

**FULLY AUTOMATED DEEP SUPERVISED AND
UNSUPERVISED LEARNING APPROACHES FOR 3D
PROTEIN CRYO-EM DENSITY MAP RECONSTRUCTION**

A Dissertation presented to
the Faculty of the Graduate School at
the University of Missouri

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in
Electrical Engineering and Computer Science

by
ADIL AL-AZZAWI
Professor Jianlin Cheng, Dissertation Supervisor
DECEMBER 2019

The undersigned, appointed by the Dean of the Graduate School, have examined the dissertation entitled:

**FULLY AUTOMATED DEEP SUPERVISED AND UNSUPERVISED
LEARNING APPROACHES FOR 3D PROTEIN CRYO-EM DENSITY MAP
RECONSTRUCTION**

presented by ADIL AL-AZZAWI, a candidate for the degree of Doctor of Philosophy and hereby certify that, in their opinion, it is worthy of acceptance.

Professor Jianlin Cheng

Professor Naz Islam

Professor Zhihai He

Professor Ye Duan

DEDICATION

I dedicate this dissertation to my gorgeous wife *Noor Ahmed Al-Hindawi*, my handsome boy *Ali Al-Azzawi*, my beautiful queen *Judy Al-Azzawi*. A special feeling of gratitude to my loving parents, *My Mother* for her ongoing love and support, *My Father*, and my big brother *Raed Al-Azzawi* who could not see this thesis completed.

I also dedicate this dissertation to my brother *Yaser Al-Azzawi*, my supportive sister *Israa Al-Azzawi*, *My Mother in Law*, *My Father in Law*, *My Sisters*, *My Sisters in Law*, and *All My Relatives*, especially my American's mom *Kathy Smith* for their continued love and support throughout.

I dedicate this work and give special thanks to *The Higher Committee of Education Development in Iraq (HCED)*, for supporting me throughout the doctorate program and to my wonderful country, *IRAQ*.

For all, thank you for always giving me the best...

ACKNOWLEDGMENTS

I would like to express the deepest appreciation to my advisor and committee chair *Professor Dr. Jianlin Cheng*. Without his guidance, suggestions, and support this dissertation would not have been possible. His mentoring has been instrumental to my research productivity and efficiency, and his view about problem solving has influenced me to always have a relentless positive attitude in all situations. I am glad to have had the opportunity to work in his lab.

I would like to thank my committee members *Dr. Ye Duan, Dr. Naz Islam, and Dr. Zhihai He*, for providing scientific guidance, encouragement and advice throughout my time as a student.

My sincere thanks to my beautiful wife *Noor Ahmed* for her support, encouragement, quiet patience and unwavering love were undeniably the bedrock upon which the past ten years of my life have been built. Her tolerance of my occasional vulgar moods is a testament in itself of her unyielding devotion and love, my handsome son *Ali Al-Azzawi*, and my queen *Judy Al-Azzawi*.

Finally, I would like to thank the previous and current members of the Bioinformatics and Data Mining (BDM) lab for the supportive role they played while working with them in the lab. I also enjoyed a friendly working environment with *Hasanain Al-Sadr, Anes Quadou, Oluwatosin Oluwadare, Tianqi, Jie Hou, Max Highsmith*, and *Chen (Chris) Chen*. I consider it a privilege to have worked alongside each one of you.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	ii
LIST OF TABLES	ix
LIST OF FIGURES	xi
ABSTRACT	xxxv
CHAPTER	
1 Introduction	1
1.1 Overview.....	1
1.2 Electron Microscopy (EM) Imaging.....	4
1.3 Electron Microscopy (Cryo-EM) Preparation	6
1.4 Computational Molecular Reconstruction Methods.....	9
1.4.1 Single Particle Picking	10
1.4.2 Particles Alignment and 2D Classification.....	15
1.4.3 3D Reconstruction	18
1.4.4 Angular Reconstitution using Fourier Transform.....	19
1.5 Outlines.....	22
2 Literature Review	25
3 AutoCryoPicker: An Unsupervised Learning Approach for Fully Automated Single Particle Picking in Cryo-EM Images	29
3.1 Overview.....	29
3.2 Background	30
3.3 Methods.....	33
3.3.1 Stage 1: Pre-processing.....	33
Step 1: Cryo-EM Image Resolution Improving	36
Step 2: Global Cryo-EM Intensity Adjustment	37

	Step3: Global Cryo-EM Contrast Enhancement.....	41
	Step 4: Cryo-EM Noise Suppressing.....	41
	Step 5: Local Particles Contrast Enhancement in cryo-EM...	42
	Step 6: Particle Edges Enhancement in cryo-EM	43
	Step 7: Particle Shape Localization in cryo-EM	44
3.3.2	Stage 2: Particle Clustering.....	46
	Intensity Based Clustering (IBC) Algorithm.....	46
	K-means and FCM Clustering Algorithms.....	49
3.3.3	Stage 3: Particle Picking.....	52
	Step 1: Cryo-EM Cluster Image Cleaning and Non-Circular Object Removal.....	52
	Step 2: Top View (Circular) Particle Detection and Picking in Cryo-EM.....	55
	Step 3: Side View (Square) Particle Detection and Picking in Cryo-EM.....	63
	Step 4: Perfect Side View (Square) Particle Detection and Picking in Cryo-EM.....	68
3.4	Results and Discussion.....	74
	3.4.1 Dataset.....	75
	3.4.2 Evaluation Metrics.....	75
	3.4.3 Particle Clustering, Detection and Picking Results.....	77
	3.4.4 Comparison with Another Particle Picking Software.....	81
3.5	Conclusion.....	91
4	SuperCryoEMPicker: Super Clustering Approach for Fully Automated Single Particle Picking in Cryo-EM	92
	4.1 Introduction.....	92
	4.2 Background.....	94
	4.3 Methods.....	96
	4.3.1 Stage 1: Pre-processing Stage.....	99
	4.3.2 Stage 2: Particles Clustering.....	105
	Clustering with base clustering algorithms	105
	Super Particle Clustering	107

4.3.3	Stage 3: Particles Picking.....	121
	Binary Mask Cleaning.....	126
	Single Particle Detection and Picking.....	121
4.4	Results and Discussion	128
4.4.1	Datasets.....	128
4.4.2	Evaluation Metrics.....	128
4.4.3	Particle Clustering, Detection and Picking Results.....	132
4.4.4	Comparison With other Particle Picking Software.....	137
4.5	Conclusions	141
5	DeepCryoPicker: Fully Automated Deep Neural Network for Single Particle Picking in cryo-EM.....	142
5.1	Introduction.....	142
5.2	Background.....	143
5.3	Methods.....	144
5.3.1	An Overview of the DeepCryoPicker Framework.....	145
5.3.2	Model 1: Fully Automated Training Particles-Selection Based Unsupervised Learning Approach.....	147
5.3.3	Stage 1: Fully Automated Training Particles-Selection.....	147
	Binary Mask Cleaning.....	157
	Perfect “good” Top-View Training Particles-Selection.....	157
	Perfect “good” Side-View Training Particles-Selection.....	161
	Perfect “good” Irregular and Complex Training Particles-Selection	162
5.3.4	Model 2: Fully Automated Single Particle Picking Based on Deep Classification Network.....	163
5.4	Experiments Results.....	167
5.4.1	Micrographs Data Collection.....	167
5.4.2	Performance Evaluation Metrics	168
5.4.3	Experiments on Unsupervised Learning Framework for Fully Automated Training Particles-Selection.....	169
5.4.4	Experiments on Automated Training Dataset Generation.....	170
5.4.5	Experiments on Training Deep Classification Model	170

5.4.6	Experiments on Testing Deep Classification Model.....	172
5.4.7	Experiments of Deep Learning Framework for Fully Single Particle Picking on cryo-EM Datasets.....	175
5.4.8	Experiments on External Testing Micrographs.....	178
5.5	Discussion.....	180
6	DeepCryoMap: Fully Automated cryo-EM Particles Alignment Approach for 3D Density Maps Reconstruction Based Deep Supervised and Unsupervised Learning Approaches	184
6.1	Background	184
6.2	Methods	189
6.2.1	Component 1: Micrographs Particles Pre-processing	193
6.2.2	Component 2: Fully Automated Single Particle Picking.....	193
6.2.3	Component 3: Fully Automated Perfect 2D Particles-Selection..	195
	Stage 1: Fully Automated Training Particles-Selection.....	195
	Stage 2: Fully Automated 2D Particle Mask Generation based Unsupervised Learning Approach.....	197
	Perfect “good” Side-view 2D Particles Images Selection	197
	Perfect “good” Top-view 2D Particle Selection.....	201
6.2.4	Component 4: Fully Automated Particles-Alignment.....	205
	Stage 1: Fully Automated Side-View Particle Alignment.....	206
	Stage 2: Fully Automated Top-View Particle Image Alignment..	215
6.2.5	Component 5: 3D Density Map Reconstruction.....	219
	Step 1: Prefect 2D Particle Alignment.....	220
	Step 2: Extract and Match Set of Sparse Points.....	222
	Step 3: 3D Fundamental Matrix Estimation.....	223
	Step 4: Reconstruct the 3D Matched Points Locations.....	226
	Step 5: Metric Reconstruction and 3D Density Map Visualization.....	227
6.3	Results and Discussion	231
6.3.1	Datasets.....	231
6.3.2	Evaluation Metrics.....	232

6.3.3	Experiments on Fully Automated Single Particle Picking	233
6.3.4	Experiments on Fully Automated Perfect 2D Single Particles Selection.....	234
6.3.5	Experiments of the Single 2D Particle Images Alignment	235
6.3.6	Experiments on Fully Automated 3D Density Map Reconstruction.....	240
6.4	Conclusions	244
7	Tools for Fully Automated Single Particle Picking and 3D Density Map Reconstruction	245
7.1	Basic Dependencies	245
7.2	AutoCryoPicker	245
7.2.1	Installation	245
7.2.2	Usage	246
	Pre-processing Stage.....	247
	Single Particle Picking without GUI	247
	Single Particle Picking with GUI	247
7.3	SuperCryoPicker	251
7.3.1	Installation	251
7.3.2	Usage	252
	Pre-processing Stage.....	252
	Single Particle Picking without GUI	252
	Single Particle Picking with GUI	253
7.4	DeepCryoPicker	256
7.4.1	Installation	256
7.4.2	Usage	257
	Component 1: Fully Automated Training Particle Selection...	257
	Component 2: Fully Automated Single Particle Picking.....	259
7.5	DeepCryoMap	260
7.5.1	Installation	260
7.5.2	Usage	261

8	Conclusion and Future Works	262
8.1	Conclusions	262
8.2	Future Works	263
	BIBLIOGRAPHY	265
	VITA	284

LIST OF TABLES

Table	Page
3.1 The results of AutoCryoPicker using the three clustering methods on the first dataset (Apoferitin). The table reports the average of the sensitivity or recall, specificity, precision, F1 score, accuracy, DICE score, and the particle clustering time (seconds).....	77
3.2 The results of AutoCryoPicker using the three clustering methods on the second dataset (KLH). The table reports the average of the sensitivity or recall, specificity, precision, F1 score, accuracy, DICE score, and the particle clustering time consuming (seconds).....	78
3.3 Statistical evaluation AutoCryoPicker and EMAN2 performance using the Apoferitin and KLH images. The table reports TP: True Positive picking results where the correct particles are picked, FN: False Negative picking results where some good particles are missed, FP: False Positive picking results where the incorrect particles (other objects such as background or artificial objects) are picked as particles.....	90
3.4 Evaluation of particle picking on Apoferitin images.....	90
3.5 Evaluation particle picking on the second KLH image.....	91
4.1 The average peak signal-to-noise ratio (PSNR), signal-to-noise ratio (SNR), and mean squared error (MSE) of the cryo-EM images without or with EMAN2 intensity adjustment according to different scaling factors.....	131
4.2 The results of the particle picking using the base clustering algorithms (k-means, FCM, and IBC)	137
4.3 The results of the super clustering methods (SP-K-means, SP-FCM, and SP-IBC).....	137
5.1 Training Particles-Selection Image Size using Good Training Particles-Selection form Apoferitin KLH, Ribosome datasets.....	164
5.2 Total Number of the Training Particles-selection using Fully Automated Good Training Particles-selection for Apoferitin KLH, Ribosome Datasets.....	169
5.3 The Whole Automated Training Particles-selection Dataset.....	170

5.4	Performance Results of the Deep Classification Network Testing using Different Parameters and Different Datasets.....	175
6.1	Total number of single particles picking using automated deep neural network for single protein particle and different datasets Apoferritin [183], and KLH [184] datasets.....	234
6.2	Performance results of fully automated perfect 2D single particles selection using the deep classification network using different parameters and datasets, learning patch size illustrates the number of subsection of an input image to the CNN which describe how many chunk of an image the been processed by the kernel at the time to estimate the good regularization property “small number of parameters” to be good across many regions of each image.....	135
6.3	The average similarity metric scores (SSIM) for the fully automated single particle alignment.....	239

LIST OF FIGURES

Figure	Page
1.1 The Overall Single-Particle Cryo-EM Workflow for Protein Structure Determination.....	3
1.2 The Overall Framework of the X-ray Crystallography.....	5
1.3 An example of a zoomed-in part of the cryo-EM where the particles that appear in a less disparity and they are hardly to pick-up. (a) the original size of the negative stain cryo-EM. (b) the zoomed-in of the cryo-EM image of (a). (c) the original size of the real world of the cryo-EM image. (d) the zoomed-in of some particles in the cryo-EM in (d).....	7
1.4 Different samples of different cryo-EM images from different datasets.....	9
1.5 Overall of the Computational Protein Structure Determination Steps.....	11
1.6 The Overview of the Micrograph Image Component Using Micrograph from the KLH dataset.....	12
1.7 The Overview of the Micrograph Image Component Using Micrograph from the Beta-galoczeta dataset.....	13
1.8 The Particle Picking Examples using Different Micrographs Images and Different Fully Automated Particle Picking Approaches (a) (b) (c).....	14
1.9 Particle Images Alignment Example by Shifting, Rotating or By Both in 2D Space.....	16
1.10 Particle Image Classification and Averaging Example.....	17
1.11 An Example 2D particle image classification and averaging using 200 Particle Images from the Ribosome dataset.....	17
1.12 The Back-projection Approach to produce the 3D Density Map.....	18
1.13 The General Sinograms for Angular Line Detection in the 3D Model using Fourier Transform.....	20
1.14 3D Model Reconstruction using Angular Refinement based Fourier Transform.....	21
1.15 3D Model Reconstruction using Projection Matching Method.....	22
3.1 The general framework of AutoCryoPicker: Fully Automated Single Particle Picking. The dashed boxes represent three stages of the approach: pre-	34

processing, particle clustering, and particle detection and picking. A solid box denotes an analysis step.....	
3.2 Cryo-EM image averaging and normalization result using EMAN2. (a) The original cryo-EM image (stack of 50 frame) in the MRC format before the averaging and normalization processing. (b) The cryo-EM image in PNG file format (single frame) after the averaging and normalization processing using EMAN2.	37
3.3 Contrast transfer correction and adjustment process. (a) Illustration of the cryo-EM image histogram after the averaging and normalization step using EMAN2 and the a two-element vector that consists of the low and the upper intensity limits by default. The values in low_high specify the bottom 2% and the top 2% of all pixel values. (b) Illustration of the cryo-EM histogram (Histogram shrinking) after automatically detecting and specifying the low and high intensity range (e.g. [0.2-0.8])	38
3.4 Cryo-EM Contrast Transfer Correction (CTC) process. (a) The original cryo-EM image after the applying the averaging and normalization process through the EMAN2 software. (b) Histogram of the original cryo-EM image. (c) The cryo-EM image after applying the mid-range stretching based on the low-high intensity range. (d) Histogram of the image in (c). (e) The cryo-EM image after applying the contrast enhancement correction (CEC) and image adjustment. (f) The histogram of the cryo-EM image after applying the contrast enhancement correction (CEC).....	40
3.5 Illustration of effects of the cryo-EM image analysis on a zoom-in selected particle region using two different examples from two datasets. (a) An original zoom-in selected particle region in the micrograph image in Apoferritin dataset. (b) The normalized single particle image region. (c) The single particle region after applying the contrast enhancement correction (CEC). (d) The single particle region after applying the histogram equalization. (e) The single particle region after applying image resonation with Wiener filtering. (f) The single particle region after applying the contrast-limited adaptive histogram equalization. (g) The single particle region after applying image guided filtering. (h) The single particle region after applying morphological image operation. (i) An original zoom-in selected particle region in a micrograph image in the KLH dataset before the preprocessing	46

steps. (j) The selected particle region in a micrograph image in the KLH dataset after normalization. (k) The selected particle region in a micrograph image in the KLH dataset after applying the contrast enhancement correction (CEC). (l) The selected particle region in a micrograph image in the KLH dataset after applying the histogram equalization. (m) The selected particle region in a micrograph image in the KLH dataset after applying image resonation with Wiener filtering. (n) The selected particle region in a micrograph image in the KLH dataset applying the contrast-limited adaptive histogram equalization. (o) The selected particle region in a micrograph image in the KLH dataset after applying image guided filtering. (p) The selected particle region in a micrograph image in the KLH dataset after applying morphological image operation.....

3.6 Different cryo-EM image clustering results using an Intensity-Based Clustering Algorithm (ICB). (a) Two sets of cryo-EM image clustering results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) on the Apoferritin dataset. Most real particles were always assigned to Cluster 1. (b) Two sets of cryo-EM image clustering results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) on the KLH dataset. Most real particles were always assigned to Cluster 1..... 49

3.7 Different cryo-EM image clustering results using the k-means clustering algorithm. (a) The two sets of cryo-EM images clusters results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) on the Apoferritin dataset. Most real particles were assigned to Cluster 2 and Cluster 3, respectively. (b) The two sets of cryo-EM image clustering results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) on the KLH dataset. Most real particles were assigned Cluster 1 and Cluster 2, respectively..... 50

3.8 Different cryo-EM image clustering results using the FCM clustering algorithm. (a) Two sets of cryo-EM images clustering results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) on Apoferritin dataset. Most real particles were assigned to Cluster 1 and Cluster 3, respectively. (b) Two sets of cryo-EM image clustering results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) on the KLH dataset. Most real particles were assigned to Cluster 2 and Cluster 3, respectively..... 51

3.9	Cryo-EM Particle Clustering Results after Binary Image Cleaning and Non-Circular Object Removal. (a) The particle clustering image before binary image cleaning and non-circular object removal on the results of ICB clustering of a cryo-EM image from Apoferritin dataset. (b) The particle clustering image after binary image cleaning and non-circular object removal on the results of ICB clustering of a cryo-EM image from Apoferritin dataset. (c) The particle clustering image before binary image cleaning and non-circular object removal on the results of ICB clustering of a cryo-EM image from KLH dataset. (d) The particle clustering image after binary image cleaning and non-circular object removal on the results of ICB clustering of a cryo-EM image from KLH dataset. (e) The particle clustering image before binary image cleaning and non-circular object removal on the results of k-means clustering of a cryo-EM image from Apoferritin dataset. (f) The particle clustering image after binary image cleaning and non-circular object removal on the results of k-means clustering of a cryo-EM image from Apoferritin dataset. (g) The particle clustering image before binary image cleaning and non-circular object removal on the results of k-means clustering of a cryo-EM image from KLH dataset. (h) The particle clustering image after binary image cleaning and non-circular object removal on the results of k-means clustering of a cryo-EM image from KLH dataset. (i) The particles clustering image before binary image cleaning and non-circular object removal on the results of FCM clustering of a cryo-EM image from Apoferritin dataset. (j) The particle clustering image after binary image cleaning and non-circular object removal on the results of FCM clustering of a cryo-EM image from Apoferritin dataset. (k) The particle clustering image before binary image cleaning and non-circular object removal on the results of FCM clustering of a cryo-EM image from KLH dataset. (l) The particle clustering image after binary image cleaning and non-circular object removal on the results of FCM clustering of a cryo-EM image from KLH dataset.....	55
3.10	(a) (d) Original cryo-EM image from the Apoferritin and KLH. (b) Edge detection result that will be used later for CHT to detect the center of each circular object in the binary cryo-EM image from the Apoferritin dataset based on using canny edge detection. (c) Edge detection results that will be used later for CHT to detect the center of each circular object in the binary cryo-EM	58

image from the Apoferritin dataset based on using the modified CHT based IBC clustering and boundary pixels list extraction (outline's boundary pixel). (e) Edge detection result that will be used later for CHT to detect the center of each circular object in the binary cryo-EM image from the KLH dataset based on using canny edge detection. (f) Edge detection results that will be used later for CHT to detect the center of each circular object in the binary cryo-EM image from the KLH dataset based on using the modified CHT based IBC clustering and boundary pixels list extraction (outline's boundary pixel).....

3.11 Top View (Circular) Particles Detection and Picking Results using Modified Circular Hough Transform (CHT). (a) The Ground truth (particles manually labelled) for the cryo-EM image from the Apoferritin dataset. (b) ICB clustering results after the binary image cleaning and non-circular objects removal (Apoferritin dataset). (c) The center of each particle illustrated by the '+' sign and the radius of each particle by the blue circle around each particle (ICB and Apoferritin dataset). (d) The bounding box for each particle object in the original cryo-EM image (ICB and Apoferritin dataset). (e) K-means clustering results after the binary image cleaning and non-circular objects removal (Apoferritin dataset). (f) The center of each particle illustrated by using the '+' sign and the radius of each particle by the blue circle around each particle (k-means results on Apoferritin dataset). (g) The bounding box for each particle (k-means results and Apoferritin dataset). (h) FCM clustering results after the binary image cleaning and non-circular objects removal (Apoferritin dataset). (i) The center of each particle illustrated by the '+' sign and the radius of each particle by the blue circle around each particle (FCM and Apoferritin dataset). (j) The bounding box for each particle in the original cryo-EM image (FCM results and Apoferritin dataset). (k) The ground truth (particles manually labeled) for the cryo-EM image from the KLH dataset. (l) ICB clustering results after the binary image cleaning and non-circular objects removal (KLH dataset). (m) The center of each particle illustrated by the '+' sign and the radius of each particle by the blue circle (ICB and KLH dataset). (n) The bounding box for each particle in the original cryo-EM image (ICB and KLH dataset). (o) K-means clustering results after the binary image cleaning and non-circular objects removal (KLH dataset). (p) Shows the center of each particle illustrated

	by the ‘+’ sign and the radius of each particle by the blue circle (k-means and KLH dataset). (q) The bounding box for each particle in the original cryo-EM image (k-means and KLH dataset). (r) FCM clustering results after the binary image cleaning and non-circular objects removal (KLH dataset). (s) The center of each particle illustrated by the ‘+’ sign and the radius of each particle by the blue circle (FCM and KLH dataset). (t) The bounding box for each particle in the original cryo-EM image (FCM and KLH dataset)	
3.12	Cryo-EM clean clustered images after the circular and non-square object removal. (a) The cryo-EM clustered images after image cleaning and small objects removal. (b) The same cryo-EM clustered images after the circular and non-square object removal.....	66
3.13	Side view (square) particles detection and picking results. (a) The original cryo-EM image (KLH dataset). (b) The result after circular and non-square object removal based on the ICB clustering algorithm. (c) Side view (square) particle detection results.....	68
3.14	Perfect square (side view) particle shape detection using the Feret object diameter using (KLH dataset). (a) Square particle image after shapes smoothing and blurring. (b) Boundary boxes (each particle) based on Feret object diameter measurement. (c) Perfect square particle shapes that are generated based on the new boundary box dimension using Feret object diameter measurement. (d) Square particle image after the outlier objects are eliminated. (e) Square particle detection results (side view) based on the new Feret boundary box dimension. (f) The final results of two different particle shape detection and picking (top and side view) based on ICB clustering and modified CHT; and perfect square (side view) particle shapes detection using Feret object diameter.	72
3.15	Automated particle picking results for both cases (top and side view) on KLH dataset The original cryo-EM images form the KLH dataset. (a) Target detection and picking results (top and side particles view) using the ICB clustering algorithm. (b) Target detection and picking results (top and side particles view) using the k-means clustering algorithm. (c) Target detection and picking results (top and side particles view) using the FCM clustering algorithm	74
3.16	Automated particle picking results on the two datasets. (a) A cryo-EM image with a high identical particle density and a lack low-frequency from the Apoferritin dataset. (e) A low SNR cryo-EM image from the Apoferritin	81

dataset. (i) A micrograph image from the KLH dataset that includes excessively overlapped particles due to confounding artifacts such as ice contamination, degraded particles, and particle aggregates. (m) A micrograph image from the KLH dataset that has a very low spatial density and different intensity levels. (b) and (f) Particle picking results using Intensity Based Clustering Algorithm (ICB) (Apoferritin dataset). (c) and (j) Particle picking results using k-means (Apoferritin dataset). (d) and (h) Particle picking results using FCM (Apoferritin dataset). (j) and (n) Particle picking results using Intensity Based Clustering Algorithm (ICB) (KLH dataset). (k) and (o) Particle picking results using k-means (KLH dataset). (l) and (p) Particle picking results using FCM (KLH dataset)

3.17 Particle picking using EMAN2 and AutoCryoPicker. (a) The manually selected reference particles of the Apoferritin dataset that were used for automated particle picking with EMAN2. (b) Zoomed-in view of the reference particles for the Apoferritin dataset. (c) EMAN2 automatic picking result based on threshold value=0.0 using the first tested image of the Apoferritin dataset. (d) EMAN2 automatic picking result based on threshold value=0.5 using the first tested image of the Apoferritin dataset. (e) EMAN2 automatic picking result based the threshold value=2.3 using the first tested image of the Apoferritin dataset. Red dots mark missed particles). (f) Ground truth of first tested image of the Apoferritin dataset. Yellow dots mark valid particles. (g) EMAN2 automatic picking result based the threshold value=2.3 using the second tested image of the Apoferritin dataset. Red dots mark missed particles). (h) Ground truth of second tested image of the Apoferritin dataset. Yellow dots mark valid particles. (i) The manually selected reference particles of the KLH dataset that were used for automated picking of top-view (circular) particles with EMAN2. (j) EMAN2 automatic picking result based the threshold value=0.5 using the first tested image of the KLH dataset. Red squares mark the false positives and the yellow dots the missing particles. (k) Zoomed-in view of the automatically picked particles (threshold value=0.5) for first tested image of the KLH dataset. (l) EMAN2 automatic picking result based the threshold value=0.5 using the second tested image of the KLH dataset. Red squares mark the false positives, and the yellow dots mark the missing particles (top-view). (m) Particle picking result from AutoCryoPicker using the first tested image of the Apoferritin

dataset. Red '+' mark the center of each particle and blue circles the top-view detected particles in the cryo-EM image. (n) Particle picking result from AutoCryoPicker using the second tested image of the Apoferritin dataset. Red '+' mark the center of each particle and blue circles the top-view detected particles in the cryo-EM image. (o) Particle picking result from AutoCryoPicker using the first tested image of the KLH dataset. Red '+' marks the center of each particle, blue circles the top-view detected particles in the cryo-EM image, and the yellow squares the side-view detected particles in the cryo-EM image. (p) Particle picking result from AutoCryoPicker using the second tested image from the KLH dataset. Red '+' marks the center of each particle, blue circles the top-view detected particles in the cryo-EM image, and the yellow squares the side-view detected particles in the cryo-EM image.....

3.18 Evaluation of particle picking using EMAN2 and AutoCryoPicker. (a) Apoferritin cryo-EM image with top-view particle shapes only. (b) The ground truth (manually particle picking labels) of the first Apoferritin cryo-EM image where each particle is marked by a yellow circle on top of each particle. (c) The particle picking results of the first Apoferritin image using EMAN2. The particles are labeled as follows: Green, True Positive (TP); red, False Negative (FN). The particle picking results of the first Apoferritin cryo-EM image using AutoCryoPicker. The particles are labeled as follows: Green, True Positive (TP); red, False Negative (FN); orange, False Positive (FP). (d) The second original Apoferritin cryo-EM image with top-view particle shapes only. (e) The ground truth (manually particle picking labels) of the second Apoferritin cryo-EM image where each particle is marked by a yellow circle on top of each particle. (f) The particle picking results of the second Apoferritin cryo-EM image using EMAN2. The particles are labeled as follows: Green, True Positive (TP); red, False Negative (FN); orange, False Positive (FP). (g) The particle picking results of the second Apoferritin cryo-EM image using AutoCryoPicker. The particles are labeled as follows: Green, True Positive (TP); red, False Negative (FN); orange, False Positive (FP). (h) The first original KLH cryo-EM image. (i) The ground truth (manually particle picking labels) of the first KLH cryo-EM image where each particle is marked by a yellow circle on top of each particle. (j) The particle picking results of the first KLH image using EMAN2. The particles are labeled as follows: Green, True

Positive (TP); red, False Negative (FN). (k) The particle picking results of the first KLH cryo-EM image using AutoCryoPicker. The particles are labeled as follows: Green, True Positive (TP); red, False Negative (FN). (l) The second original KLH cryo-EM image which has top-view particle shapes only. (m) The ground truth (manually particle picking labels) of the second KLH cryo-EM image where each particle is marked by a yellow circle on top of each particle. (n) The particle picking results of the second KLH cryo-EM image using EMAN2. (o) The particles are labeled as follows: Green, True Positive (TP); red, False Negative (FN). (p) The particle picking results of the second KLH cryo-EM image using AutoCryoPicker. The particles are labeled as follows: Green, True Positive (TP); red, False Negative (FN).....

- 4.1 Example of particles of complex and irregular shapes. (a) Ribosome Electron Microscopy Density Map. (b) a cryo-EM image that has the Ribosome particles of irregular particle shapes [27]. (c) Beta-galactosidase Electron Microscopy Density Map. (d) a cryo-EM image that contains Beta-galactosidase particles of complex shape. Both (a) and (c) are created based on using Chimera [27] and density maps from Protein Data Bank in Europe EMBL-EBI [28] 97
- 4.2 The general framework of the SuperCryoEMPicker. The dashed boxes represent three stages of the approach: pre-processing, super clustering, and particle picking. A solid box denotes an analysis step..... 99
- 4.3 : One zoom-in particle image from the ribosome dataset during the first stage of the pre-processing “intensity adjustment” using different scaling factors in the EMAN2. (a) the original zoom-in particle (manually selected and cropped from the original cryo-EM). (b) the original histogram of the cryo-EM. (c) particle image after the intensity adjustment using scale factor 5. (d) the histogram of the pre-processed image adjusted in (c). (e) particle image after the intensity adjustment with scale factor 4. (f) the histogram of the pre-processed image in (e). (g) particle image after the intensity adjustment with scale factor 3. (h) the histogram of the pre-processed image in (g). (i) particle after the intensity adjustment (scale factor 1). (j) the histogram of the pre-processed image in (i). (k) particle image after the intensity adjustment with scale factor 0.1. (l) the histogram of the pre-processed image in (k). (m) particle image after the intensity adjustment with scale factor 0.25. (n) the histogram of

102

- the pre-processed image in (m). (o) particle image after the intensity adjustment with scale factor 0.5. (p) the histogram of the pre-processed image in (o).....
- 4.4 Illustration of effects of the preprocessing procedures on Ribosome and Beta-galactosidase images. (a) the original particle image of Ribosome (one full image and one zoom-in particle). (b) the original image of Beta-galactosidase. (c) the image of Ribosome after the image resolution improvement. (d) the image of Beta-galactosidase after image resolution improvement. (e) the image of Ribosome after global intensity adjustment. (f) the image of Beta-galactosidase after global intensity adjustment. (g) the image of Ribosome after global contrast enhancement-based histogram equalization. (h) the image of Beta-galactosidase after the global contrast enhancement-based histogram equalization. (i) the image of Ribosome after the noise suppressing using Wiener filter. (j) the image of Beta-galactosidase after the noise suppressing using Wiener filter. (k) the image of Ribosome after the local particle contrast enhancement with the adaptive histogram equalization. (l) the image of Beta-galactosidase after the local particle contrast enhancement using adaptive histogram equalization. (m) the image of Ribosome after edges enhancement using guided image filtering. (n) the image of Beta-galactosidase after edges enhancement using guided image filtering. (o) the image of Ribosome after the particle shape localization using morphological image operation. (p) the image of Beta-galactosidase after the particle shape localization using morphological image operation..... 104
- 4.5 Image clustering results using k-means, FCM, and IBC. (a) an original cryo-EM image of Ribosome. (b) an original cryo-EM image of Beta-galactosidase. (c) K-means clustering results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) for Ribosome image. Most real particles were assigned to Cluster #3. (d) K-means clustering results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) for the Beta-galactosidase image. Most real particles were assigned to Cluster #2. (e) FCM clustering results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) for the Ribosome image. Most real particles were assigned to Cluster #2. (f) FCM clustering results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) for the Beta-galactosidase image. Most real particles were assigned to Cluster #1. (g) IBC clustering results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) for the Ribosome image. Most real particles were assigned to Cluster #1. 107

	(h) IBC clustering results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) for the Beta-galactosidase image. Most real particles were assigned to Cluster #1.....	
4.6	Different cryo-EM intermedia micrograph maps generation using simple linear iterative clustering (SLIC). (a) an original cryo-EM image of Ribosome. (b) the histogram of (a).(c) intermedia micrograph maps generated using simple linear iterative clustering (SLIC) based on the pre-processed image. (d) histogram of (c). (e) an intermedia micrograph map generated by SLIC from the original image. (f) histogram of (e). (g) an original cryo-EM image of Beta-galactosidase. (h) histogram of (g). (i) an intermedia micrograph map generated by SLIC from the pre-processed image. (j) histogram of (i). (k) an intermedia micrograph map generated by SLIC from the original image. (l)histogram of (k).....	111
4.7	SP-K-means evaluation and automated cluster section based on extract the total number of the particles in each cluster and select the cluster that has the minimum total number of the particles. (a) the total number of objects (particles) in the cluster index number (#1). (b) the total number of objects (particles) in the cluster index number (#2). (c) the total number of objects (particles) in the cluster index number (#3). (d) the total number of objects (particles) in the cluster index number (#4)	113
4.8	Cryo-EM super clustering results of SP-K-means, SP-FCM, and SP-IBC in comparison with the base algorithms. (a) an original cryo-EM image of Ribosome. (b) the intermedia micrograph map generated by SLIC from the pre-processed image of Ribosome. (c) an original cryo-EM image of Beta-galactosidase. (d) the intermedia micrograph map generated by SLIC for Beta-galactosidase. (e) k-means clustering results of the Ribosome. (f) SP-K-means clustering results of Ribosome. (g) k-means clustering results of the Beta-galactosidase cryo-EM image. (h) SP-K-means clustering results of the Beta-galactosidase cryo-EM image. (i) FCM clustering results of the Ribosome cryo-EM image. (j) SP-FCM clustering results of the Ribosome cryo-EM image. (k) FCM clustering results of the Beta-galactosidase cryo-EM image. (l) SP-FCM clustering results of the Beta-galactosidase cryo-EM image. (m) IBC clustering results of the Ribosome cryo-EM image. (n) SP-IBC clustering results of the Ribosome cryo-EM image. 9o) IBC clustering results of the Beta-	120

	galactosidase cryo-EM image. (p) SP-IBC clustering results of the Beta-galactosidase cryo-EM image.....	
4.9	A zoomed-in selected particle image before and after binary image cleaning and non-connected objects removal on the image masks generated by the three base clustering methods. (a) the original zoomed-in particle image. (b) the particle clustering image before binary image cleaning by k-means for Ribosome. (c) the particle clustering image after binary image cleaning by k-means for Ribosome. (d) the original zoomed-in particle image. (e) the particle clustering image before binary image cleaning by FCM clustering for Ribosome. (f) the particle clustering image after binary image cleaning by FCM clustering for Ribosome. (g) the original zoomed-in particle image. (h) the particle clustering image before binary image cleaning by IBC clustering for Ribosome. (i) the particle clustering image after binary image cleaning by IBC clustering for Ribosome.....	123
4.10	The whole cryo-EM particle clustering results before and after binary image cleaning for the three base clustering methods. (a) the Ribosome clustered image of k-means. (b) the image mask of (a) after image cleaning. (c) the Beta-galactosidase clustered image of k-means. (d) the binary image mask of (c) after image cleaning. (e) the Ribosome clustered image of FCM. (f) the binary image mask of (e) after image cleaning. (g) the Beta-galactosidase clustered image of FCM. (h) the binary mask of (g) after image cleaning. (i) the Ribosome clustered image of IBC. (j) the binary image mask of (i) after image cleaning. (k) the Beta-galactosidase clustered image of IBC. (l) the binary image mask of (k) after image cleaning.....	125
4.11	The results of detecting particles of irregular and complex shapes on Ribosome and Beta-galactosidase images. (a) particle detection and picking by k-means on the Ribosome image. (b) particles detection and picking by FCM) on the Ribosome image. (c) particles detection and picking by IBC on the Ribosome image. (d) particle detection and picking by k-means on the Beta-galactosidase image. (e) particle detection and picking by FCM on the Beta-galactosidase image. (f) particle detection and picking by IBC on the Beta-galactosidase image. (g) particle detection and picking by SP-K-means on the Ribosome image. (h) particle detection and picking by SP-FCM on the Ribosome image. (i) particle detection and picking by SP-IBC on the Ribosome image. (j) particle	127

	detection and picking by SP-K-means on the Beta-galactosidase image. (k)	
	particle detection and picking by SP-FCM on the Beta-galactosidase image. (l)	
	particles detection and picking by SP-IBC on the Beta-galactosidase image....	
4.12	The quality of Cryo-EM images before and after the pre-processing Stage. (a)	
	the average PSNR and SNR values of the cryo-EM images before and after the	
	pre-processing steps. (b) average MSE values of the cryo-EM images before	
	and after the pre-processing steps.....	132
4.13	The results of the fully automated single particle picking in cryo-EM images	
	by the base and super clustering methods. Red dots denote the missing particles	
	not detected (false negatives) and yellow dots the false positives. (a) the particle	
	picking of IBC. (b) particle picking of IBC on extremely low-SNR cryo-EM	
	image. (c) the particle picking of SP-IBC. (d) the single particle picking of SP-	
	IBC on extremely low-SNR cryo-EM image. (e) particle picking of k-means.	
	(f) the particle picking of k-means clustering algorithm on extremely low-SNR	
	cryo-EM image. (g) the particle picking of SP-K-means. (h) the particle picking	
	of SP-K-means on extremely low-SNR cryo-EM image. (i) the particle picking	
	of FCM. (j) the particle picking of FCM on extremely low-SNR cryo-EM	
	image. (k) the particle picking of SP-FCM. (l) the particle picking of SP-FCM	
	on extremely low-SNR cryo-EM image.....	133
4.14	Particle picking using EMAN2, Scipion and SuperCryoEMPicker. (a) a	
	manually selected reference particle of the Beta-galactosidase image for	
	Scipion. (b) the zoom-in view of some manually selected reference particles	
	for the Beta-galactosidase image for Scipion. (c) the final reference particles of	
	Beta-galactosidase manually selected for Scipion. (d) all the particle picking	
	results of Scipion trained on 40 manually reference particles on the image of	
	the Beta-galactosidase. (e) EMAN2 autopicking result based on different	
	manually training samples selection in the first tested image of the Beta-	
	galactosidase dataset. (f) the manually selected reference particles of Beta-	
	galactosidase for EMAN2. (g) the particle picking results of	
	SuperCryoEMPicker based on the SP-IBC clustering. (h) the particle picking	
	results of SuperCryoEMPicker based on the SP-K-means clustering. (i) the	
	particle picking results of SuperCryoEMPicker based on the SP-FCM	
	clustering.....	136

4.15	Particle picking using EMAN2, Scipion and SuperCryoEMPicker. (a) a manually selected reference particle of the Beta-galactosidase image for Scipion. (b) the zoom-in view of some manually selected reference particles for the Beta-galactosidase image for Scipion. (c) the final reference particles of Beta-galactosidase manually selected for Scipion. (d) all the particle picking results of Scipion trained on 40 manually reference particles on the image of the Beta-galactosidase. (e) EMAN2 autopicking result based on different manually training samples selection in the first tested image of the Beta-galactosidase dataset. (f) the manually selected reference particles of Beta-galactosidase for EMAN2. (g) the particle picking results of SuperCryoEMPicker based on the SP-IBC clustering. (h) the particle picking results of SuperCryoEMPicker based on the SP-K-means clustering. (i) the particle picking results of SuperCryoEMPicker based on the SP-FCM clustering.....	139
4.16	Evaluation of particle picking using EMAN2 and Super Cluster approach for Fully Automated single Particle Picking. (c) The particle picking results of Beta-galactosidase image [30] using EMAN2 [31]. (b) The particle picking results of Beta-galactosidase image [30] using super Clustering Approach for Fully Automated Single Particle Picking, the particles are labeled as follows: yellow, True Positive (TP); red, False Negative (FN), blue, False.....	140
5.1	The general workflow of the training particle-selection based unsupervised scheme and single particle picking based on deep learning scheme. The gray part of the workflow shows the micrographs data collection. The blue part of the workflow shows the fully automated training particles-selection using clustering algorithms. The red part of the workflow shows the general flow of the single particle picking using deep classification network. The yellow part of t6he workflow shows the external testing part of the DeepCryoPicker.....	146
5.2	DeepCryoPicker architecture. The orange rectangle marks the first part of the fully automated approach “fully training particles-section and dataset generation”. The blue rectangle marks the second part “fully automated single particles picking”. The green and gray rectangles mark the first and second stage of the preprocessing step respectively.....	148
5.3	Illustration of effects of the cryo-EM image analysis on a zoom-in selected particle region using two different examples from two datasets. (a1), (b1), (c1),	150

(d1), and (e1) original zoom-in particle regions (different shapes) are selected from different micrograph Apoferritin (top-view particle), KLH (top-view), KLH (side-view) Ribosome (irregular shape), and Beta-galactosidase (complex shape) respectively. (a2), (b2), (b2), and (e2) normalized single particle image region. (a3), (b3), (c3), (d3), and (e3) single particle region after applying the contrast enhancement correction (CEC). (a4), (b4), (c4), (d4), and (e4) single particle region after applying the histogram equalization. (a5), (b5), (c5), (d5), and (e5) single particle region after applying image resonation with Wiener filtering. (a6), (b6), (c6), (d6), and (e6) single particle region after applying the contrast-limited adaptive histogram equalization. (a7), (b7), (c7), (d7) and (e7) single particle region after applying image guided filtering. (a8), (b8), (c8), (d8) and (e8) single particle region after applying morphological image operation.....

- 5.4 Top-view particle clustering using different cryo-EM image clustering results and the Intensity-Based Clustering Algorithm (ICB), (a) the Apoferritin micrograph clustering image (binary mask), (b) the Apoferritin micrograph binary mask image cleaning and small object removal. (c) the perfect circular particle shape generation on the Apoferritin micrograph binary mask image. (d) the KLH micrograph clustering image (binary mask), (e) the KLH micrograph binary mask image cleaning and small object removal. (f) the perfect circular particle shape generation on the KLH micrograph binary mask image..... 151
- 5.5 Top View (Circular) Particles Detection and Picking Results using Modified Circular Hough Transform (CHT). (a) The Ground truth (particles manually labelled) for the cryo-EM image from the Apoferritin dataset. (b) The center of each particle illustrated by the '+' sign and the radius of each particle by the blue circle around each particle (ICB and Apoferritin dataset). (c) The bounding box for each particle object in the original cryo-EM image (ICB and Apoferritin dataset). (d) the ground truth (particles manually labeled) for the cryo-EM image from the KLH dataset. (e) the center of each particle illustrated by the '+' sign and the radius of each particle by the blue circle (ICB and KLH dataset). (f) the bounding box for each particle in the original cryo-EM image (ICB and KLH dataset)..... 153

5.6	Top-view particles-selection results, (a) using cryo-EM micrographs form the Apoferritin [14] dataset, (b) using cryo-EM micrographs form the KLH [13] datasets, (c) Apoferritin good particle example , (d) Apoferritin good binary mask example, (e) KLH good particle example , (f) KLH good binary mask example, (g) Apoferritin bad particle example, (d) Apoferritin bad binary mask example, (i) KLH bad particle example , (d) KLH bad binary mask example.....	155
5.7	Good and bad top-view training particles-selection results. (a) and (e) individual top-view particle binary mask form the Apoferritin [14] and KLH [13] datasets. (b) and (f) CHT perfect circle on top of the particle’s binary masks. (c) and (g) the replaced artificial perfect circle binary after the CHT for the Apoferritin [14] and KLH [13] particle’s binary mask respectively. (d) and (h) the good Apoferritin [14] and KLH [13] top-view training particles selection. (i), (l), (m), and (o) other examples of the top-view particle’s binary masks that the modified CHT has failed to draw perfect circles on top of them. (j), (l), (n), and (p) bad top-view training particle examples.....	157
5.8	Side-view particles clustering using different cryo-EM image and Intensity-Based Clustering Algorithm (ICB), (a) and (g) different KLH micrograph clustering images (binary masks), (b) and (h) KLH micrograph binary mask images after image cleaning and small object removal. (c) and (i) particle objects smoothing. (d) and (j) Feret diameter measures for the particle objects. (e) and (k) perfect side-view (square) particle shapes generation on the top of the binary image of the KLH micrograph. (i) and (l) show the overlapped particles removal and perfect side-view particles-selection results.....	159
5.9	Good and bad top-view training particles-selection results. (a) and (e) individual top-view particle binary mask form the Apoferritin [14] and KLH [13] datasets. (b) and (f) CHT perfect circle on top of the particle’s binary masks. (c) and (g) the replaced artificial perfect circle binary after the CHT for the Apoferritin [14] and KLH [13] particle’s binary mask respectively. (d) and (h) the good Apoferritin [14] and KLH [13] top-view training particles selection. (i), (l), (m), and (o) other examples of the top-view particle’s binary masks that the modified CHT has failed to draw perfect circles on top of them. (j), (l), (n), and (p) bad top-view training particle examples.....	160

5.10	Side-View (square and circular) particles-selection. (a) and (b) The Ground truth (particles manually labelled) for the different cryo-EM images from the KLH dataset [13]. (b) and (e) side-view particles-selection results using IBC clustering and perfect side-view (square) particles-selection algorithm. (c) and (f) top-view particles-selection results using modified CHT algorithm (the red '+' sign is the center of each particle, and blue circles around each particle are the radius of each particle by the blue circle around each particle.....	161
5.11	Irregular (complex) particles-selection results, (a) particle picking results using cryo-EM micrographs from the Ribosome dataset. (b) and (d) good particle binary mask examples, (c) (e) good training particle examples, (f) and (h) bad binary mask examples, (g) and (i) bad particle examples.....	162
5.12	The architecture of the deep neural network used in DeepCryoPicker. The convolutional layer and the subsampling layer are abbreviated as C and S, respectively. C3:11x11x96 means that in the third convolutional layer (C3) is comprised of 96 feature maps, each of which has a size of 11×11 , also. C3:@27x27 means that output feature maps dimensions are 27x27 pixels.....	164
5.13	(b) simulated top-view (circular) of the Apoferritin molecule shapes, (c) Apoferritin real-world top-view (circular) protein shape, (d) simulated top-view (circular) of the KLH molecule shapes, (e) KLH real-world top-view (circular) protein shape, (f) simulated side-view (square) KLH molecule shape, (g) KLH real-world side-view (circular) protein shape, (h) simulated irregular (complex) Ribosome molecule shape, (i) Ribosome irregular (complex) protein shape, (j) simulated complex beta-galactosidase molecule shape, (k) beta-galactosidase complex protein shape.....	168
5.14	Impact of the number of the training classes on the precision-recall curve, (a), (c), and (c) red, blue, and black curves are obtained with the deep classification model training datasets and represents the precision-recall curves plotted for different single particle shapes picking using different micrographs datasets (KLH, Apoferritin, and Ribosome) including 5 classes, 4 classes (with negative detection), and 4 classes (with background) respectively.....	171
5.15	Different examples of the deep classification network results. (a) A typical testing image example showing high-density top-view particle's predicted label and prediction score of the Apoferritin micrograph dataset, (b) A typical testing image example showing high-density side-view particle's predicted label and	174

	prediction score of the KLH micrograph dataset, (c) A typical testing image example showing high-density background predicted label and prediction score, (d) A typical testing image example showing high-density irregular particle's predicted label and prediction score, (e) A typical testing image example showing high-density top-view particle's predicted label and prediction score of the KLH micrograph dataset, (f) A typical testing image example showing high-density background predicted label and prediction score	
5.16	The DeepCryoPicker" fully automated single particle picking in cryo-EM images results for different micrographs datasets (a) typical micrograph showing the KLH Top and Side-View particles picking, (b) typical micrograph showing the Apoferritin Top-View particles picking, (c) typical micrograph showing the Ribosome irregular (complex) particles picking.....	177
5.17	Precision-recall cures of the fully automated different single particle shapes picking result using deep classification network and different micrographs datasets, (a) precision-recall cure of the top-view particle shapes picking, (b) precision-recall cure of the side-view particle shapes picking, (c) precision-recall cure of the irregular and complex particle shape picking.....	178
5.18	The External DeepCryoPicker testing results using different micrographs from different external testing datasets (a) typical external micrograph from the bacteriophage MS2 (EMPIAR-10075) [18] showing the Top-View particles picking, (b) typical external micrograph from the T. acidophilum 20 (EMPIAR-10186) [19] showing the Top and Side-View particles picking, (c) typical external micrograph from the beta-galactosidase 2.2 Å (EMPIAR- 10061) [19] showing the irregular (complex) particles picking.....	180
5.19	Quantity analysis on real datasets using a precision-recall curve of different single particle picking tools. The green, yellow, black, blue, and red curves represent the precision-recall curves for RELION, DeepPicker, DeepEM, PIXER, and DeepCryoPicker respectively.....	183
6.1	Different Levels of protein structures. (a) the protein primary structure (sequence of amino acids connected to form a long chain). (b) protein secondary structure. (c) alpha helical. (d) beta strand. (e) protein tertiary structure. (f) protein quaternary structure (Figure is retrieved from [163] [166]	185

6.2	3D Density Map Reconstruction using cryo-EM technique. (a) single-particle-reconstruction procedure (framework), (b) Identify the individual complex 2D image (particle picking) [163] [175].....	187
6.3	Different Complex cryo-EM density maps Visualization, the maps were captured from the EMDB (EMDB:5001 and 1042 respectively) [181] [182] and images are captured from [163]. (a) and (b) low- and high-density map resolution visualized 3D density map using so-surface of low- and high-density map resolution respectively, and slice (right) representations are shown for maps at two resolutions. Higher resolution density maps (lower number) have a greater amount of detail, while lower resolution (higher number) are smother. cryo-EM density map of visualized with iso-surfaces at 4 different density thresholds. The surfaces shown are drawn at decreasing threshold values, with the surface on the left having the highest threshold. At higher threshold, the inner and denser parts of the complex are seen, while at lower thresholds a larger outer envelope of the complex can be seen.....	189
6.4	Structural cryo-EM 3D Density Maps examples (top and side view) (b) 3D Cryo-EM map of apoferritin, (c) picked particle from an apoferritin micrograph, (d) 3D Cryo-EM map of KLH viewed from the top, (e) picked particle from a KLH micrograph showing the top view (circular particle), (f) 3D Cryo-EM map of KLH viewed from the side, (g) picked particle from a KLH micrograph showing the side-view (square particle)	190
6.5	The general workflow of the DeepCryoMap based single particle picking based on deep learning scheme and unsupervised scheme. The green part of the workflow shows the micrographs data preprocessing. The red part of the workflow shows the fully automated single particles picking using DeepCryoPicker [25]. The blue part of the workflow shows the general flow of the good (perfect) 2D single particle-selection using unsupervised learning and deep classification network. The yellow part of the workflow shows the fully automated particle alignment, and gray part shows the 3D density map reconstruction using single-particle-reconstruction.....	191
6.6	Fully automated particle picking experimental results (second component of the DeepCryoMap results). (a) and (b) the DeepCryoPicker [187] results using two datasets Apoferritin [183] and KLH dataset [184], (c) the original KLH particle picking results, (e) KLH preprocessing particle picking results, (d) the	195

	original Apoferritin particle picking results, (f) the Apoferritin preprocessing particle picking results.....	
6.7	The architecture of the deep neural network used in DeepCryoPicker [25]. The convolutional layer and the subsampling layer are abbreviated as C and S, respectively. C3:11x11x96 means that in the third convolutional layer (C3) is comprised of 96 feature maps, each of which has a size of 11×11 , also. C3:@27x27 means that output feature maps dimensions are 27x27 pixels.....	197
6.8	Perfect “good” side-view 2D particles images selection based perfect 2D mask generation. (a) original side-view particle image that is fully automated picked using the DeepCryoPicker [187] using KLH dataset [185], (b) preprocessed version of the original side-view particle, (c) initial binary mask of the (b) using IBC clustering algorithm in AutoCryoPicker [185], (d) Feret dimeter detection of (c), (e) Initial binary mask generation of (a), (f) Postprocessing version of (c), (g) Feret dimeter detection using (f), (h) Perfect binary mask generation.....	200
6.9	Perfect “good” top-view 2D particles images selection based perfect 2D mask generation. (a) and (i) two original top-view particle images that is fully automated picked using the DeepCryoPicker [25] using Apoferritin [21] and KLH dataset [23], (b) and (j) the preprocessed version of the original top-view particle images of (a) and (i) respectively, (c) and (k) the initial binary mask of the (b) and (j) using IBC clustering algorithm in AutoCryoPicker [23], (d) and (l) the cleaned circular clustered images of (c) and l) respectively, (e) and (f) the inner and outer circular mask extraction of the (d). (m) and (n) the inner and outer circular mask of (l). (k) and (o) are the filled circular binary masks of (f) and (n) respectively, (h) and (p) perfect top-view binary mask generation of (g) and (o) respectively.....	203
6.10	Fully automated side-view particle alignment using KLH dataset [184], (a) Artificial frontal view reference image generation based average binary particle object sizes (b) Original side view particle image (moving), (c) Perfect generated binary mask of (b). (d) Unalignment images projection (references (a) and moving (c)), (e) Default image alignment (initial registration). (f) Optimizer adjustment and metric configuration-based image registration. (g) Image registration based increasing the maximum iteration number. (h) Image registration-based optimization and rigid [189] transformation. (i) Image remigration using affine transform, (i) Original binary mask particle’s	212

	orientation, (k) Aligned binary mask particle's orientation, (l) Final particle alignment result.....	
6.11	Localized 2D side-view aligned particle image generation. (a) Original aligned particle image, (b) Aligned binary mask particle of the original image (a), (c) Perfect particle binary mask image and original particle image projection, (d) localized 2D side-view aligned particle image.....	213
6.12	Fully automated side-view particle image alignment using intensity-based image registration.....	214
6.13	Localized 2D top-view aligned particle image generation. (a) Original aligned particle image, (b) Prefect 2D binary mask particle of the original image (a), (c) Perfect particle binary mask and original particle image projection, (d) localized 2D top-view aligned particle image.....	215
6.14	Fully automated top-view particle image alignment. (a) and (b) original top-view particle images from Apoferritin dataset [21], (b) and (f) Center point extraction using the modified CHT algorithm [23] using the generated prefect binary masks for (a) and (b) respectively, (c) and (g) Centralized top-view particle binary mask alignment result, (h) Centralized top-view of the localized particle alignment image result.....	216
6.15	Localized 2D particle images before and after the centralized based particle image alignment, (a) localized 2D top-view particle image from Apoferritin dataset [21] before the centralized particle image alignment, (b) localized 2D top-view particle binary mask of (a) before the centralized particle image alignment, (c) localized 2D top-view particle image after centralized based particle image alignment, (d) localized 2D top-view particle mask image after centralized based particle image alignment.....	217
6.16	Fully automated localized 2D top-view particle image alignment using centralized image alignment based perfect binary 2D image.....	218
6.17	Localized 3D density map reconstruction framework using structural based motion information.....	220
6.18	Some example from the localized 2D side-view particle image showing (a) and (b) two localized particle images are not perfectly aligned.....	221
6.19	Fully automated perfect side-view particle alignment using KLH dataset [184], (a) and (b) two localized aligned particle images that are not perfectly aligned, (c) two particle image projection (a) and (b), (d) default image alignment	222

	(initial registration), (e) optimizer adjustment and metric configuration-based image registration, (f) image registration based increasing the maximum iteration number, (g) image registration-based optimization and rigid [197] transformation. (h) image remigration using affine transform.....	
6.20	Sparse points matching and extraction, (a) first tested particle image, (b) features (corners) extraction using minimum eigenvalue algorithm [204], (c) second tested particle image, (d) correlation points detection and tracking using Kanade-Lucas-Tomasi (KLT), feature-tracking algorithm [205] [206] [207] [208].....	223
6.21	Localized 3D density map reconstruction and visualization, (a) localized 3D density map in a side view, (b) same localized 3D density map in a frontal view.....	228
6.22	Reference image generation-based image fusion, (a) first original localized aligned particle image, (b) second localized perfect aligned particle image, (c) blended overlay fused particle image, by scaling the intensities of the reference image, (d) visualized the fused blended (overlay) image using red channel for the reference particle image, green channel for the aligned moving image, and yellow channel for the areas of similar intensity between the two images.....	230
6.23	Different examples of the deep classification network results. (a) A typical testing image example showing high-density top-view particle's predicted label and prediction score of the apoferritin micrograph dataset [183], (b) A typical testing image example showing high-density top-view particle's predicted label and prediction score of the KLH micrograph dataset [184], (c) and (d) typical testing image examples showing high-density side-view particle's predicted label and prediction score of the KLH micrograph dataset [184], (e) and (f) typical testing image examples showing high-density background predicted label and prediction score.....	234
6.24	Perfect binary mask generation stage for "good" side-view 2D particles images selection. (a)-(d) are the original side-view particle image from KLH dataset [184], (e)-(h) are the perfect binary mask generation for the good 2D particle sample selection.....	236
6.25	Different examples of the deep classification network results. (a) A typical testing image example showing high-density top-view particle's predicted label and prediction score of the apoferritin micrograph dataset, (b) A typical testing	237

	image example showing high-density side-view particle's predicted label and prediction score of the KLH micrograph dataset [184].....	
6.26	Different examples of the side-view fully alignment results-based intensity-image registration and perfect generated particle masks using KLH dataset [184].....	238
6.27	Different examples some examples of the localized particle image generation using the fully automated side-view particle alignment and perfect generated binary mask using KLH dataset [184].....	238
6.28	Different examples of the perfect localized 2D top-view particle alignment using KLH dataset [184].....	239
6.29	Fully automated 3D density map reconstruction for the KLH side-view protein molecule, (a) Final average 3D density map reconstruction, (b) localized alignment particle images from the KLH dataset [185].....	240
6.30	Fully automated 3D density map reconstruction for the KLH side-view protein molecule, (a) Final average 3D density map reconstruction, (b) localized aligned of the preprocessed particle images from the KLH dataset [185].....	241
6.31	Fully automated 3D density map reconstruction for the KLH top-view protein molecule, (a) Final average 3D density map reconstruction, (b) original localized alignment particle images the KLH dataset [184].....	242
6.32	Fully automated 3D density map reconstruction for the KLH top-view protein molecule, (a) Final average 3D density map reconstruction, (b) preprocessed localized alignment particle images using KLH dataset [184].....	242
6.33	Fully automated 3D density map reconstruction for the Apoferritin top-view protein molecule, (a) Final average 3D density map reconstruction, (b) original localized alignment particle images using Apoferritin dataset [183].....	243
6.34	Different Levels of protein structures. (a) the protein primary structure (sequence of amino acids connected to form a long chain). (b) protein secondary structure. (c) alpha helical. (d) beta strand. (e) protein tertiary structure. (f) protein quaternary structure (Figure is retrieved from [163] [166]..	244
7.1	AutoCryoPicker GUI demo, showing the main GUI version that has five commands: load cryo-EM, preprocessing cryo-EM, cryo-EM clustering, particles detection and picking, and performance results.....	248
7.2	An example of preprocessed and particle detection using one cryo-EM image using Apoferritin cryo-EM dataset [21].....	248

7.3	An example of preprocessed and particle detection using one cryo-EM image using KLH cryo-EM dataset [22].....	249
7.4	An example of particle detection and picking using one cryo-EM image using Apoferritin cryo-EM dataset [21].....	250
7.5	An example of particle detection and picking using one cryo-EM image using KLH cryo-EM dataset [22].....	250
7.6	SuperCryoEMPicker GUI demo, showing the main GUI version that has five commands: load cryo-EM, preprocessing cryo-EM, cryo-EM clustering, particles detection and picking, and performance results.....	251
7.7	An example of preprocessed and particle detection using one cryo-EM image using Ribosome cryo-EM dataset.....	253
7.8	An example of preprocessed and particle detection using one cryo-EM image using Beta-galactosidase cryo-EM dataset.....	254
7.9	An example of particle clustering detection and using one cryo-EM image using Ribosome cryo-EM dataset.....	254
7.10	An example of particle clustering detection and using one cryo-EM image using galactosidase cryo-EM dataset.....	255
7.11	An example of particle detection and picking using one cryo-EM image using Ribosome cryo-EM dataset.....	255
7.12	An example of particle detection and picking using one cryo-EM image using Ribosome cryo-EM dataset.....	256

ABSTRACT

One of the most important components of the human body is the protein. Protein uses for building and repairing tissues, making enzymes and hormones. It is the essential building block of bones, muscles, cartilages, skin and blood. Therefore, a large quantity of protein always needed. Proteins are stored in the form of sequence of nucleotides that can be easily converted into a sequence of amino acids, which is known as a protein primary structure. For protein to perform its job, it needs to be in its three-dimensional structure, which also known as the protein tertiary structure. Several methods were developed for this reason. The most important one among them are X-ray crystallography, nuclear magnetic resonance (NMR) spectroscopy, and recently Electron Microscopy (EM). These methods required complicated procedures that are hard to implement, very time consuming, labor intensive, required well-trained specialists. Therefore, an alternative approach that is less time and cost consuming is required. Molecular structure prediction and understanding leads to major breakthroughs in medicine to design and produce better drugs, which will increase its efficiency and reduce its side effect. Whereas for biotechnology new and more efficient enzymes can be designed which impact many areas of our daily life such as detergents, Textiles, Food and Beverages, Leather, and Bioethanol. In terms of gaining the popularity in structural biology using the Electron Microscopy (EM) technology, a hundred of thousands of single particle images are required to be extracted from two-dimensional (2D) cryo-electron microscopy (cryo-EM) to build a reliable high-resolution (3D) model. In order to reduce the radiation damage to the biomolecules of interest during the imaging process, a limited electron dose is used as the high-energy electrons can greatly damage the specimen during imaging and results in extremely noisy

micrographs. Hence, single particle images picking still present significant challenges due to that much single particle in the original (2D) micrographs arises from different sources such as the very low single-to-noise-ratio (SNR), low contrast, heavy background noise, ice contamination, particle overlap, and amorphous carbon. Many different computational methods have been proposed for the automated semi-automated single particle picking over the past decades. Most of these methods are based on different techniques such as template-based matching, edge detection, feature extraction, and convolutional computational vision. These methods for particle picking often need a large training dataset, which requires extensive manual labor. Other reference-dependent methods rely on low-resolution templates for particle detection, matching and picking, and therefore are not fully automated. To address this challenge, we develop different models such as AutoCryoPicker – a fully automated particle picking approach based on image preprocessing, unsupervised clustering and shape detection. SuperCryoEMPicker - a fully automated super particle clustering method for picking particles of complex and irregular shape in cryo-EM images. DeepCryoPicker – a fully automated deep neural network for single particle picking in cryo-EM. Our approach solves the fully automated single particle in diversity cryo-EM images. We combined two different fully automated particle picking approaches (AutoCryoPicker and SuperCryoEMPicker) to do the fully automated single particle picking. Also, we generated fully automated approach for training dataset expanding and training particle images increasing. The fully automated training particle-selection can automatically distinguish between the “good” and “bad” training examples and isolated the selected particles to positive and negative detection examples. Later, a deep neural network is designed and trained using the generated training dataset. Finally, for each testing

micrograph, we used the developed preprocessing stage to improve the quality of the low-SNR micrographs. Then, we use the trained deep neural network model and sliding windows to test every single sub-image based on using the NMS. The results indicated that DeepCryoPicker performed accurately as good as the RELION which is “semi-automated particle picking method”, and DeepEM. Another essential process for fully understanding and determining the protein structure is a 3D density map reconstruction. 3D density map of a single protein molecule gives a significant indication to understand the protein functions and structural dynamics relationship. Individual cryo-EM particles provide an opportunity to build/reconstruct a 3D density map using single protein particles. However, always using low-dose images causes radiation of the particle damage (very low particle image contrast and highly noise particle images). That makes some limitations and more challenges for the particle’s alignment during the 3D reconstruction at intermediate resolution (1–3 nm). To overcome this issue, we design a DeepCryoMap a fully automated cryo-EM particles alignment for 3D Density Maps Reconstruction Based Deep Supervised and Unsupervised Learning Approaches. At the beginning in the first two steps, we used our previous model DeepCryoPicker to fully automated pick the particle from the micrographs. The set of the picked particles are fully automated classified and labeled based on their view (top or side-view) using the deep classification network. Then, a perfect 2D particle mask is generated for every single particle and the original particle is aligned based on the binary mask. Finally, we used a 3D computer vision algorithm to reconstruct a localized 3D density map between every two single particle image that has the most corresponding features (information). Then, we average the localized 3D density maps localized to reconstruct the final 3D cryo-EM protein density map.

Chapter 1

Introduction

1.1 Overview

For decades, X-ray crystallography has been the dominant technique for obtaining high-resolution structures of macromolecules. Single-particle cryo-electron microscopy (cryo-EM) was traditionally used to provide low resolution structural information on large protein complexes that resisted crystallization (e.g., highly symmetric particles of viruses). Though the basic workflow of cryo-EM has not changed considerably over the years, recent technological advances in sample preparation, computation and especially instrumentation have revolutionized the field of structural biology [1] [2] [3], allowing it to solve large protein structures at better than 3 Å resolution [4] [5] [6] [7].

Cryoelectronic microscopy or cryo-electron microscopy (also known as a cryo-EM) is the method originally used to “take photographs” of viruses and other macromolecular complexes. It becomes increasingly popular to study the structures of protein complexes. Cryo-EM micrographs contains two-dimensional projections of the

particles in different orientations. Generally, cryo-EM images have low contrast, due to the similarity of the electron density of the protein to that of the surrounding solution, as well as the limited electron dose used in data collection. In addition, the micrographs may contain sections of ice, deformed particles, protein aggregates, etc., which can complicate particle picking. Because a large number of single-particle images must be extracted from cryo-EM micrographs to form a reliable 3D reconstruction of the underlying structure, particle recognition, represents a significant bottleneck in cryo-EM structure determination.

Molecular structure determination and understanding requires the three-dimensional (3D) structures of macromolecular and protein [8]. A hundred of thousands of single particle images are extracted from two-dimensional (2D) cryo-electron microscopy (cryo-EM) to build a reliable high-resolution (3D) reconstruction in terms of gaining the popularity in structural biology [9]. Cryo-EM micrographs contains two-dimensional projections of the particle in different orientations. Generally, cryo-EM images have low contrast, due to the similarity of the electron density of the protein to that of the surrounding solution, as well as the limited electron dose used in data collection. In addition, the micrographs may contain sections of ice, deformed particles, protein aggregates, etc., which can complicate particle picking. Because a large number of single-particle images, extracted from cryo-EM micrographs are required for a reliable 3D reconstruction of the underlying structure, particle recognition thus, represents a significant bottleneck in cryo-EM structure determination.

An overall single-particle cryo-EM workflow is shown in Figure 1.1. The first step is sample preparation. Then, the sample is placed on a grid that is rapidly plunged into a

cryogen such as liquid ethane (-180°C for liquid nitrogen stages, -268°C for He) for flash-freezing and particle trapping in a thin film of vitreous ice. In addition to capturing the protein structure at the moment of freezing, this process protects the sample to some degree from radiation damage and prevents evaporation of buffer in the high-vacuum conditions of a transmission electron microscope.

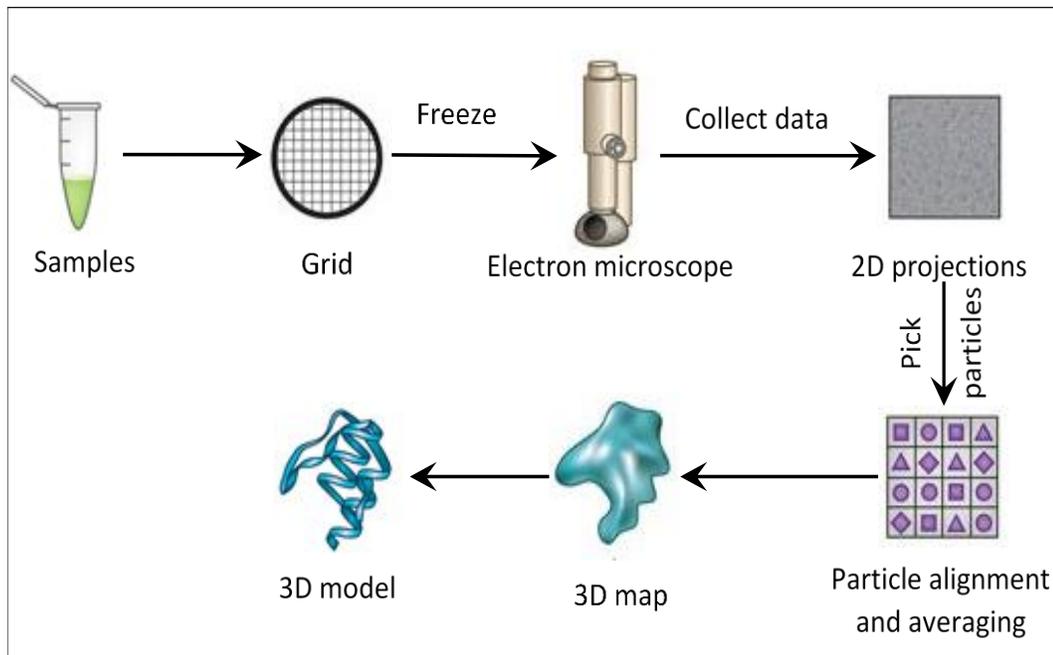


Figure 1.1: The Overall Single-Particle cryo-EM Workflow for Protein Structure Determination [10]

In order to record or acquire the 2D images of proteins by cryo-EM, a sample grid is created by freezing a solution containing proteins on a carbon film. A beam of electrons passes through a thin frozen sample of proteins to create 2D image projections [11]. Then, the 3D shape (density map) of the protein is reconstructed from the 2D images. The 2D cryo-EM images of the protein particles are taken by electron microscope, which contain randomly arranged particles along with non-particles—bits of frost, deformed particles, protein aggregates and so on. These images suffer from heavy background noise and low contrast, due to a limited electron dose used in imaging. A large number of single-particle

images, extracted from cryo-EM micrographs, are required to perform a reliable 3D reconstruction of the underlying structure. Particle recognition thus represents the first bottleneck in the practice of cryo-EM structure determination. In order to reduce the radiation damage to the biomolecules of interest during the imaging process, a limited electron dose is used as the high-energy electrons can greatly damage the specimen during imaging and results in extremely noisy micrographs [12] [13].

1.2 Electron Microscopy (EM) Imaging

The basic idea of the single particle representation in electron microscopy (EM) is that each single particle is structured in terms to illustrate a molecule of the protein or a well-defined the complex composed of many molecules such as a ribosome. In this case, the identical particles are separated out of a film, then image them (particles) in (2D image) based on using the electron microscope which is called cryo-EM images [14] [15].

One technique enjoyed a near monopoly in elucidating protein structures to this level of detail: X-ray crystallography, in which scientists persuade proteins to form into crystals, then blast X-rays at them and decipher the protein's structure from patterns that the X-rays make when they bounce off. As an example, Figure 1.2 (a) shows the X-ray crystallographic technology to image the biomolecular structures [16] [17].

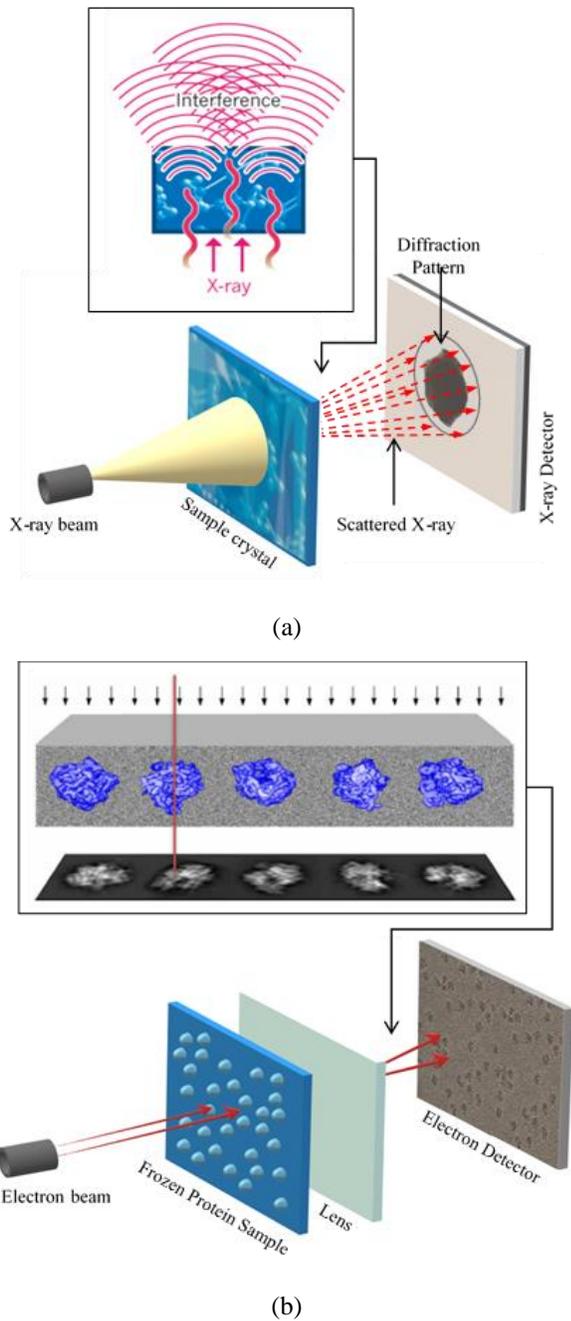


Figure 1.2: The Overall Framework of the X-ray Crystallography [16] [17]

X-rays scatter as they pass through a crystallized protein. The resulting waves interfere a diffraction pattern from which the position of atoms is deduced []. X-ray crystallography has long been dominant technology for deducing the high-resolution protein structure. Later, labs are racing a new technology to adopt the cryo-EM to image

the biomolecular structures called the cryo-electron microscopy (cryo-EM). The new technology takes the protein pictures in which be more easily to be formatted in large crystal. In the cryo-electron microscopy as shown in Figure 1.2 (b) a beam of electron is fired at a frozen protein solution. The emerging scatter electrons pass through a lens to create a magnetified image on the detector, from which their structure can be worked out [18].

1.3 Electron Microscopy (Cryo-EM) Preparation

During the sample preparation process, the particles are usually prepared in either one of two ways negative staining or cryo-EM (Vitrification). Native staining, where the particles are coated with a heavy metal salt crystal to increase the protein contrast. In this case, the particles become easy to pick up from the background as shown in Figure 1.3 (a). In case of showing the contraindication deference between the particles and the background, Figure 1.3 (b) shows a zoomed part from the negative staining. The negative stain sample preparation limits the resolution to $\sim 20\text{\AA}$ in which can introduce artificially []. In the other preparation way, the particles are prepared in different way (not crystallized) where the particles are embedded in frozen liquid (flash frozen, vitreous ice). This technique produces a less contrast image and the particles become hard to pick out as shown in Figure 1.3 (c) [10] [19].

In contrast, the vitrification technique enables much higher resolution than the negative staining by using below a 4\AA . The higher resolution of the single particle relies on the vitrification which refers to a cryo-electron microscopy (cryo-EM image) [19]. As an example, Figure 1.3 (d) shows a zoomed part from the cryo-EM where the particles appear in a less disparity and they are hardly to pick up.

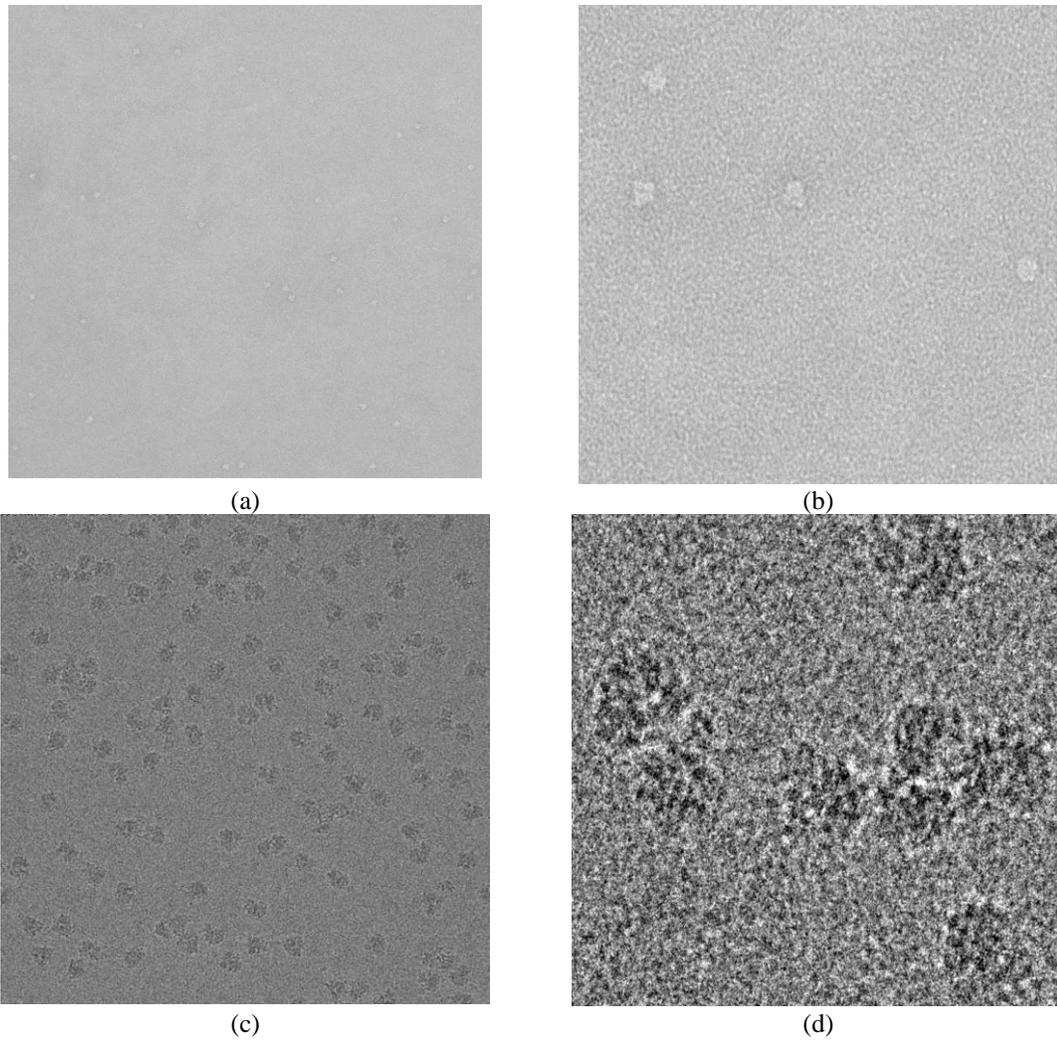
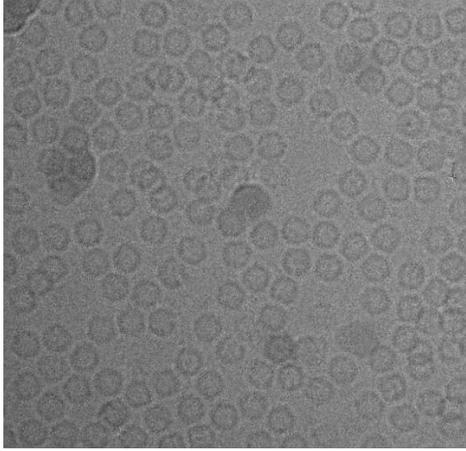
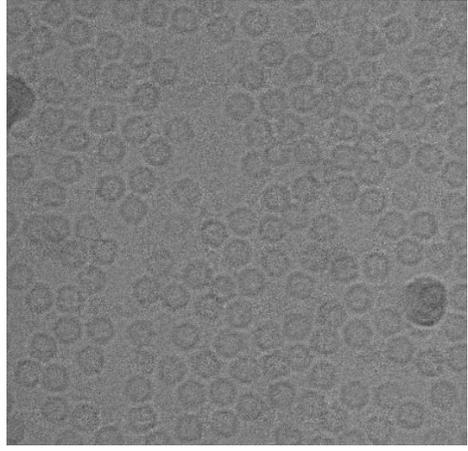


Figure 1.3: An example of a zoomed-in part of the cryo-EM where the particles that appear in a less disparity and they are hardly to pick-up. (a) the original size of the negative stain cryo-EM. (b) the zoomed-in of the cryo-EM image of (a). (c) the original size of the real world of the cryo-EM image. (d) the zoomed-in of some particles in the cryo-EM in (d)

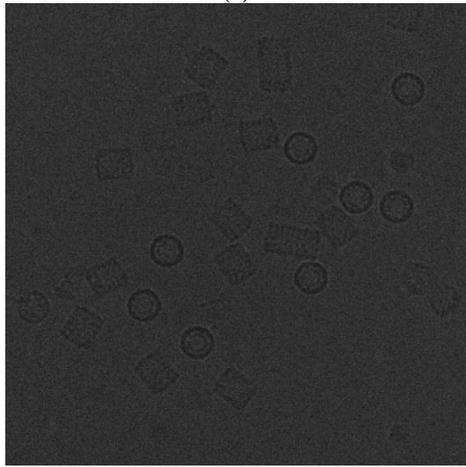
Different samples of different cryo-EM images from different datasets are shown in Figure 1.4.



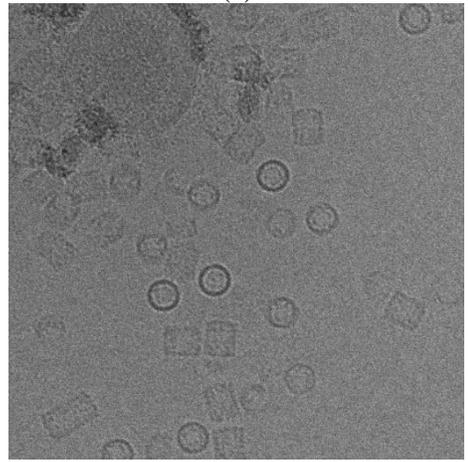
(a)



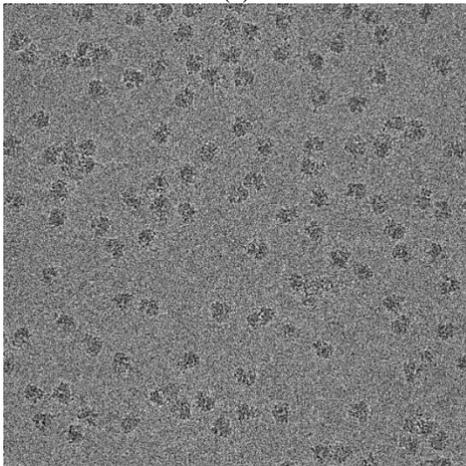
(b)



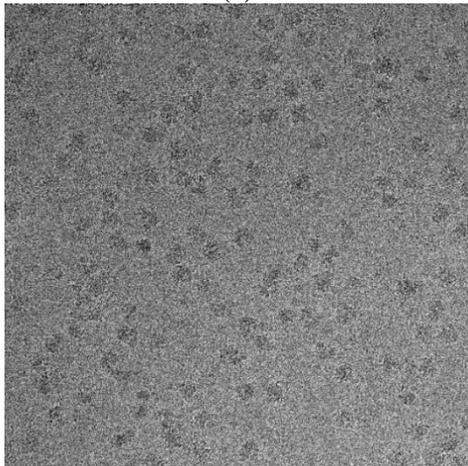
(c)



(d)



(e)



(f)

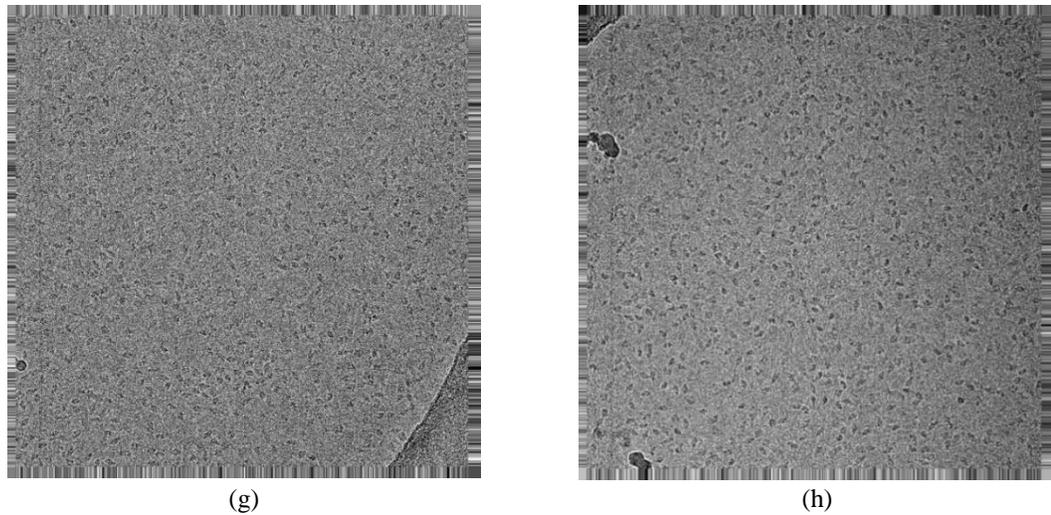


Figure 1.4: Different samples of different cryo-EM images from different datasets

1.4 Computational Molecular Reconstruction Methods

In general, there is two main steps for the molecular (protein) structure determination. The first step of the computational molecular reconstruction is the 2D cryo-EM image analysis and the second step is the 3D protein map reconstruction. In the first step, basically the raw 2D cryo-EM image (protein) is used to project a 2D high-resolution map (2D protein map) and construct a 2D projected density map. To do that, there is some steps that are preformed once to enhance the original 2D cryo-EM images (raw images) using some image pre-processing tools. Other steps are using to pick significant number of good particle examples based on some particle picking techniques using different approaches such as unsupervised learning (image clustering) or supervised learning (deep learning) for particle picking. Finally, the 3D shape (density) of the protein samples is created by projecting each recorded 2D image. In another word, the pixel value of the 2D image is presented as a sum of the values among the through line in the direction of the electron beam. Unfortunately, micrographs images have a low signal-to-noise ratio (SNR) due to damage radiation that causes only a small number of electrons can be used to create the micrograph images [],

Also, the similarity of the electron density of the protein to that of the surrounding solution produces low-contrast cryo-EM images [20].

The second step of the computational molecular reconstruction is the 3D reconstruction. In this step the high-resolution projection map (density map) is used to build a 3D model. Some essential steps are required such as background reconstruction based known view angles determination, structure refinement based unknown view angles, initial structural calculation and determination, and finally, atomic-resolution model fitting for low-resolution EM structural model construction. The overview of the main computational protein reconstruction methods is illustrated in Figure 1.5 [21] [22].

1.4.1 Single Particle Picking

The first essential step of the molecular (protein) structure determination process is the single particle picking. In general, the particle picking (single particle picking) from the micrograph images is very difficult and still a big challenge for the researcher according to the low contrast and noisy micrographs. Micrographs may also have two-dimensional projections of the particles in different orientations. Generally, cryo-EM (micrographs) images have low contrast, due to the similarity of the electron density of the protein to that of the surrounding solution, as well as the limited electron dose used in data collection [23].

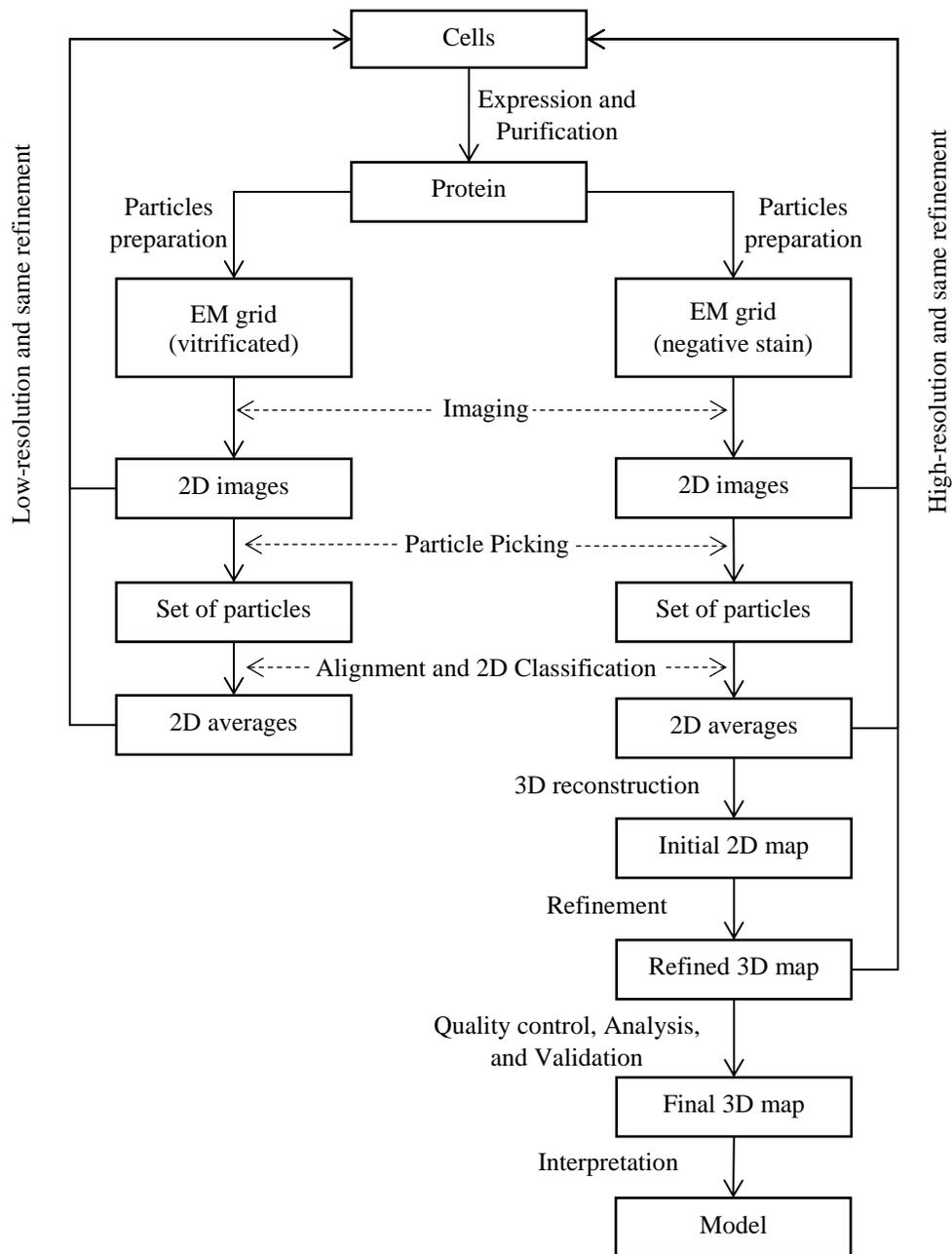


Figure 1.5: Overall of the Computational Protein Structure Determination Steps [23]

In addition, the micrographs may contain sections of ice, deformed particles, protein aggregates, etc., which can complicate particle picking as is shown in Figure 1.6. Because a large number of single-particle images must be extracted from cryo-EM

micrographs to form a reliable 3D reconstruction of the underlying structure, particle recognition, represents a significant bottleneck in cryo-EM structure determination [22].

The protein particle shapes in the most cryo-EM datasets are either common shape – circle (top view) or square (side view) as is shown in Figure 1.6 [23]. In addition to the noise, contaminants, and ice object particle shapes (side-view particle) are even overlapped or some additional objects are attached to the original particles. In contrast, another common protein particle shape in very low SNR cryo-EM images is either complex or irregular shapes [24] as is shown in Figure 1.7. In this case, detect and pick the irregular or complex particle shapes in the very low SNR cryo-EM facing two main problems. First, particles in the cryo-EM appear in non-structural object shapes which makes template matching algorithms unable to distinguish between the objects and the background as well as the particles in the very low SNR cryo-EM images have almost the same intensity level of the background.

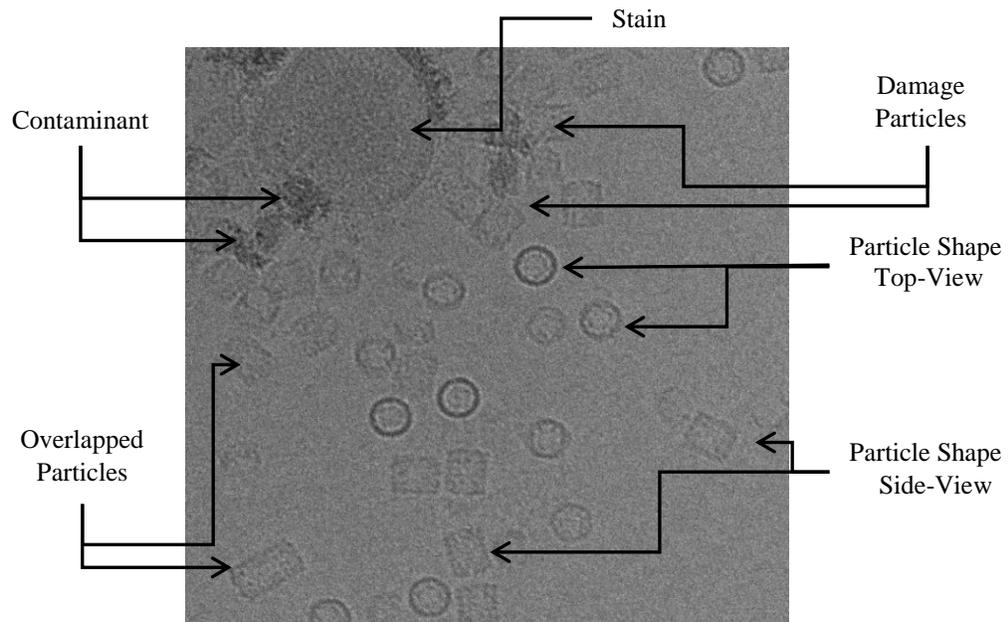


Figure 1.6: The Overview of the Micrograph Image Component Using Micrograph from the KLH dataset

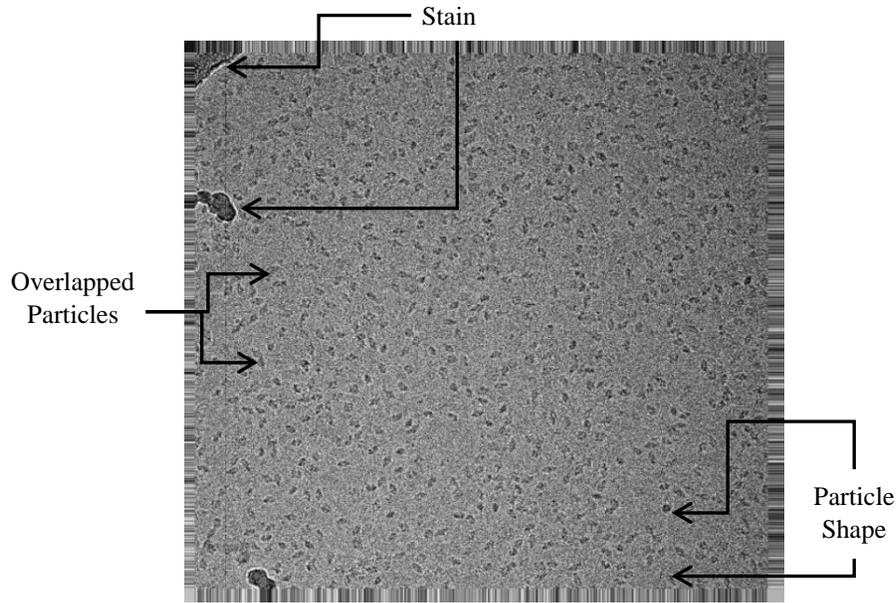
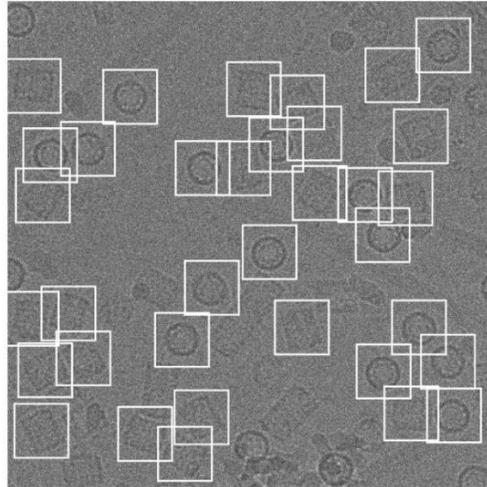
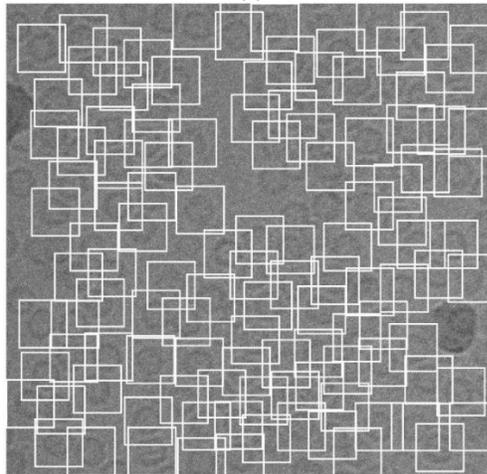


Figure 1.7: The Overview of the Micrograph Image Component Using Micrograph from the Beta-galoczeta dataset

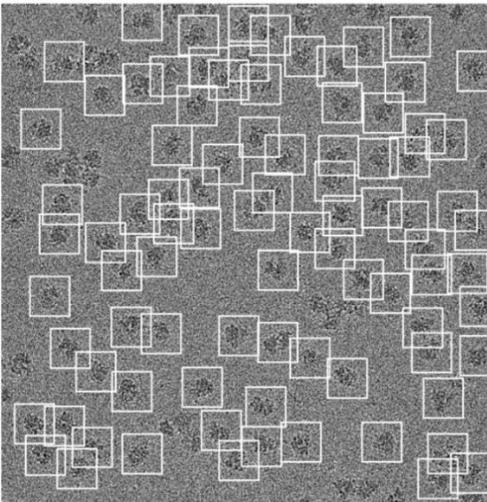
Single particle picking step is a labor-intensive step in the computational molecular reconstruction procedure and is a major obstacle for automated cryo-EM pipeline. In the past, particles from cryo-EM micrographs are often selected manually. Such a manual picking process is usually a laborious, tedious and time-consuming task which inevitably requires a considerable amount of human effort to obtain a sufficient number of good-quality particles to ensure high-resolution 3D reconstruction. In addition, manual particle selection is normally subjective and can easily introduce bias and inconsistency due to change in human judgement over time [26]. Figure 1.8 illustrate some examples of the automated single particle picking that has been developed by our models [27] [28] [29].



(a)



(b)



(c)

Figure 1.8: The Particle Picking Examples using Different Micrographs Images and Different Fully Automated Particle Picking Approaches (a) (b) (c)

1.4.2 Particles Alignment and 2D Classification

In single particle reconstruction stage, images of each complex particle are isolated from the micrograph by using the specification boundary box dimension from the single particle picking stage as is shown in Figure 1.8. In this case, the selected particle from the single particle picking stage look different. The reason that because that selected particle are similar copies of different view, different conformation, and either some of them are partially damaged. For this reason, to unify them, class average can be used here to generate multiclass that is one single class image uses to represent a group of particle images that all particles look the same [30] [31].

Single particle images are very noisy, so image averaging is the process that is used to reduce the improve the quality of the image quality based on improve the single-to-noise-ratio [32]. In terms of doing the single particle image averaging, the particle images are classified and grouped in different class based on the view and other particle properties, then the particle images being averaged to improve the image quality [33] [34].

In general, the classification methods are divided in to two approaches: supervised and unsupervised learning approaches. The supervised learning approach divides the particle images to different category according to the based template similarity or the reference similarity (training dataset). Unsupervised learning approach divides the particle images according to the intrinsic properties such as find the classes that are matched or projected presenting in the same view (clustering) [32]. During the 2D particle image classification, particle images are aligned based on the references by shifting each image or rotating in the 2D space. For instance, Figure 1.9 shows an example of some particle images that aligned by shifting and rotating each one based on the particle reference image.

Figure 1.9 (a) shows the original reference of the particle image while Figure 1.9 (b) shows the aligned particle image after the shifting, Figure 1.9 (c) shows the particle image after the rotation, finally Figure 1.9 (d) shows the particle image after both rotation and shifting [31].

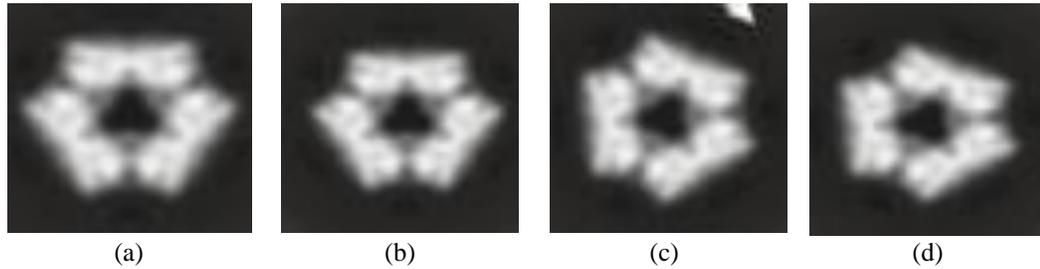
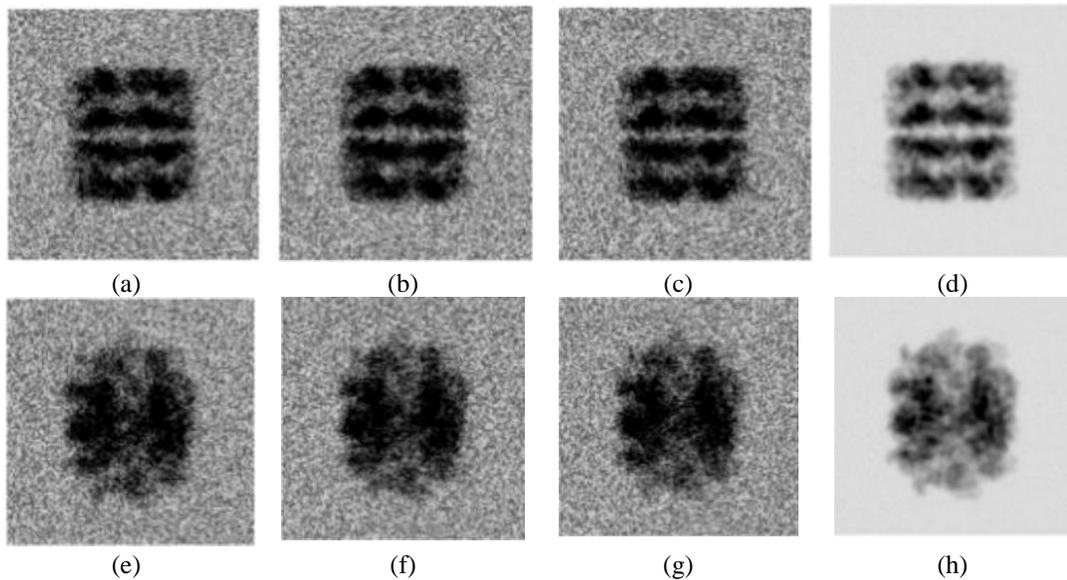


Figure 1.9: Particle Images Alignment Example by Shifting, Rotating or By Both in 2D Space [31]

Particle images during the single-particle reconstruction process are separated into different classes as is shown in Figure 1.9 (a-c), (e-g), and (i-k). Then, particle images are averaged on the same particle view to construct the 3D view. Basically, the particle images are averaged on the same particle view to construct the 3D view. Basically, the particle images are back projected to produce a 3D density map as is shown in Figure 1.10 and the results of back-projected of each different clustered class are shown in Figure 1.9 (d), (h), and (l). The 3D density map captures the electron density of the complex macromolecular [35].



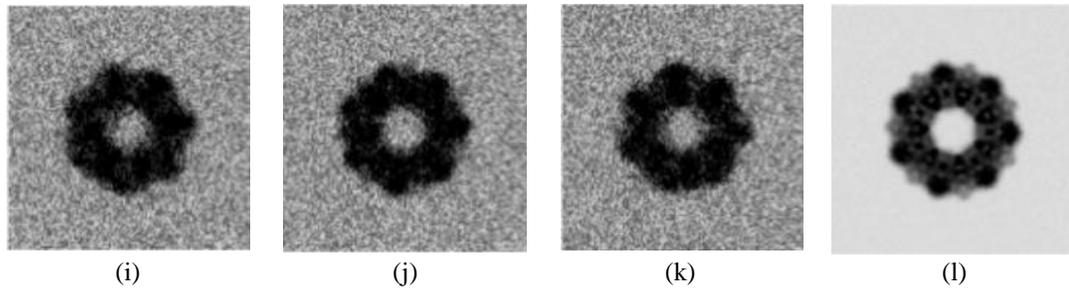


Figure 1.10: Particle Image Classification and Averaging Example [35]

The main advantages of the 2D classification during the image averaging for the single particle images are the cryo-EM (micrograph) are such a noisy image with very high low-contrast and low-signal-to-noise ratio which is probability (>10%). Particle images averaging is similar to the image enhancement technique that looks like the image contrast in enhanced and increased which the signal-to-noise-ratio (SNR) is increased too.

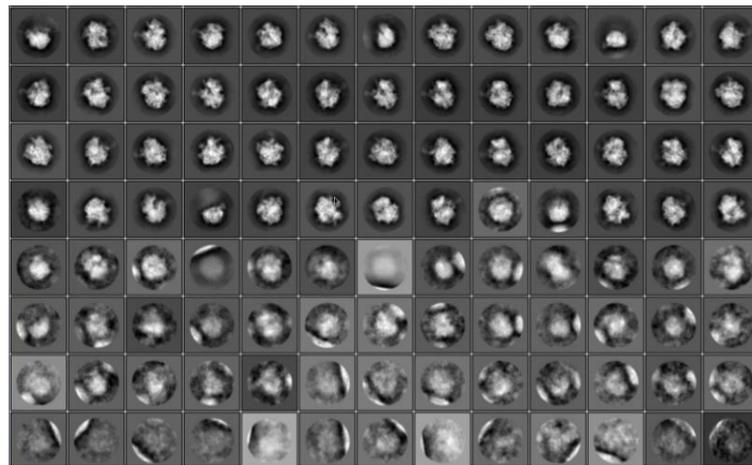


Figure 1.11: An Example 2D particle image classification and averaging using 200 Particle Images from the Ribosome dataset [31]

In another word, the class averaging produces a class averages particle image version that is easy to identify the particle image features and it helps in alignment during the 3D image reconstruction. Also, the 2D particle images classification helps to identify the bad or the unwanted particles that are improved technically by increase the quality of the data. Finally, this process (2D class averaging) is unbiased the whole process by relying

on using fewer manual references examples. An example of the 2D particle image classification and averaging using the Ribosome dataset (200 particle images) is shown in Figure 1.11 [31] [35].

1.4.3 3D Reconstruction

The Transmission Electron Microscopy (TEM) is the process to record and project the 2D object (particle) of a 3D dimension. The 3D object of the particle images is constructed basically based on using the 2D classification images (3D orientations) of each of those averaging images. Then, using those orientations by back-propagate them to produce the 3D density map which captures the electron density throughout the macromolecular complex as is shown in Figure 1.12 below [31] [35].

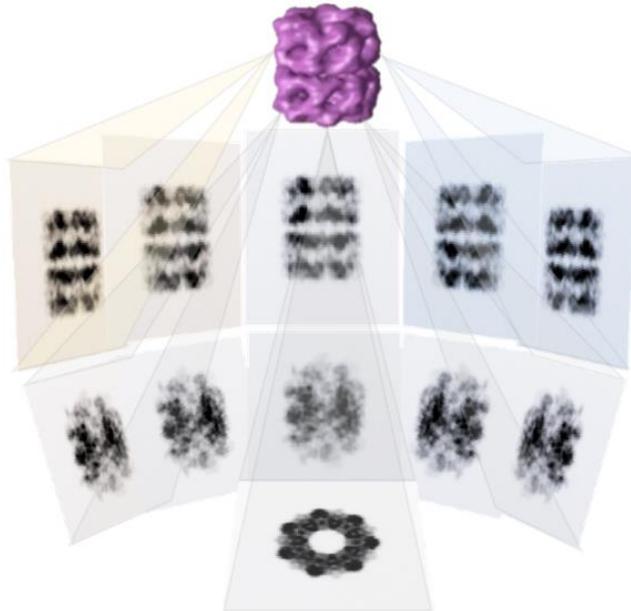


Figure 1.12: The Back-projection Approach to produce the 3D Density Map [31]

According to the nature of the biological macromolecules data, the micrographs datasets are heterogeneous. Heterogeneity means the impurities that is presented in the data

sample. 3D classification is an important step in the single particle model construction. Basically, the method proposes to sort the data samples into homogeneous subsets. In another word, the 3D classification is a process that the particles are selected and sorted into such a distinct homogenous population from different heterogeneous mixture of particles [32] [35].

Basically, generating a 3D dimensional reconstruction model requires recording different number of 2D projection of the same particles and then aligning them perfectly and properly by using the backpropagation technique. However, to do that we need to know the electron tomography angle projection [36]. In order to align and project the particle image back properly, we need to know the exact projection angle that is related to the original objects or the relation between them [32]. Fourier transform of the 2D projection is one of the most suitable methods that is used to transform the 2D projection equals the center section through its 3D. Fourier Transform (FT) basically perpendicular the projection of the angle definition. There are two main methods to identify the angle. The first one is the angular reconstitution method (common line method). The second one is the projection matching method (reference based) [31] [32].

1.4.4 Angular Reconstitution using Fourier Transform

Angular reconstitution is based on the idea that two different projections of a 3D object has always a common 1D (one dimensional) line in the projection space. In another word, the amplitude and phase are similar into two images [32]. The common line can be found in 2 3D object (particle) which can be extracted from the 3D by aligning the 2D projection along common lines. FT transform converts the 2D class averaging (particles) to another domain (space) that the common line can be extracted based on relying on the same

computed amplitudes phases [31] [32]. Figure 1.13 illustrate the main framework that is used based on compute the FT to identify the best line matching for each individual 2D class averaging.

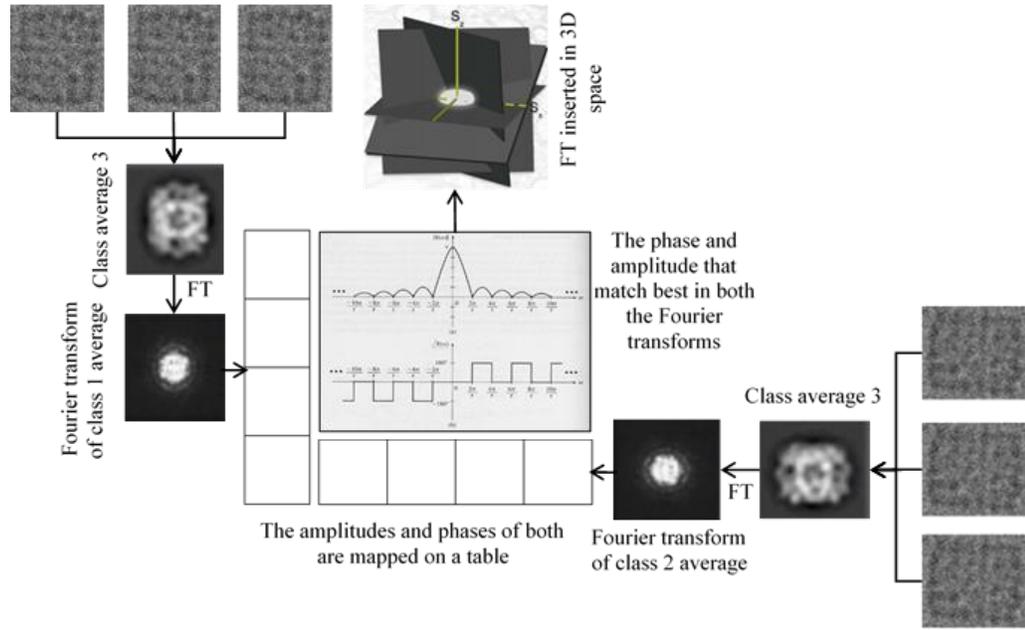


Figure 1.13: The General Sinograms for Angular Line Detection in the 3D Model using Fourier Transform [32]

Then, one all the 2D classes are interested from the 2D space in a 3D space, r =the Fourier Transform is inverted of the particle 3D object (particle) and produces a 3D structural object. The 3D object (particle) is re-projected among all the directions to perform the 2D projections as is shown in Figure 1.14. The 3D refinement iterative process is required especially when the low-resolution initial model is used. In this case, the high-resolution 3D model is obtained to produce a higher resolution 3D structure protein model. The refinement process usually starts with such an initial 3D model based on the identify set of angular [32].

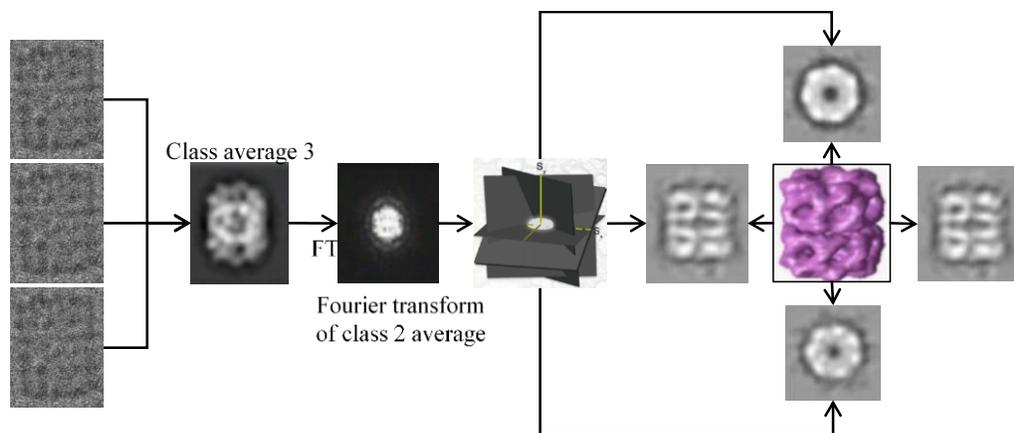


Figure 1.14: 3D Model Reconstruction using Angular Refinement based Fourier Transform [32]

1.4.5 Projection Matching

Projection matching is another method that is mainly used for 3D structure construction. It is known as a similarity or semi-similarity structure. In this method, the main process is re-project the reference model (similar) among all possible directions. Then, the re-projected directions are matched using the 2D images in the 2D classes and back projected to generate a new model. Figure 1.15 illustrate the whole framework of the 3D model reconstruction using the projection matching method [31] [32] [35] [36].

The main advantage of the projection matching technique basically does not require an initial model at the first beginning step. This is a big challenge for the 3D model construction since a similar structure is available in some cases. In contrast, the main disadvantages of the projection matching method are that it cannot be used to determine a new structure since it bases on the reference model. However, the projection matching method can wrongly lead to the references that biasedly build a wrong structural model [31] [32] [36].

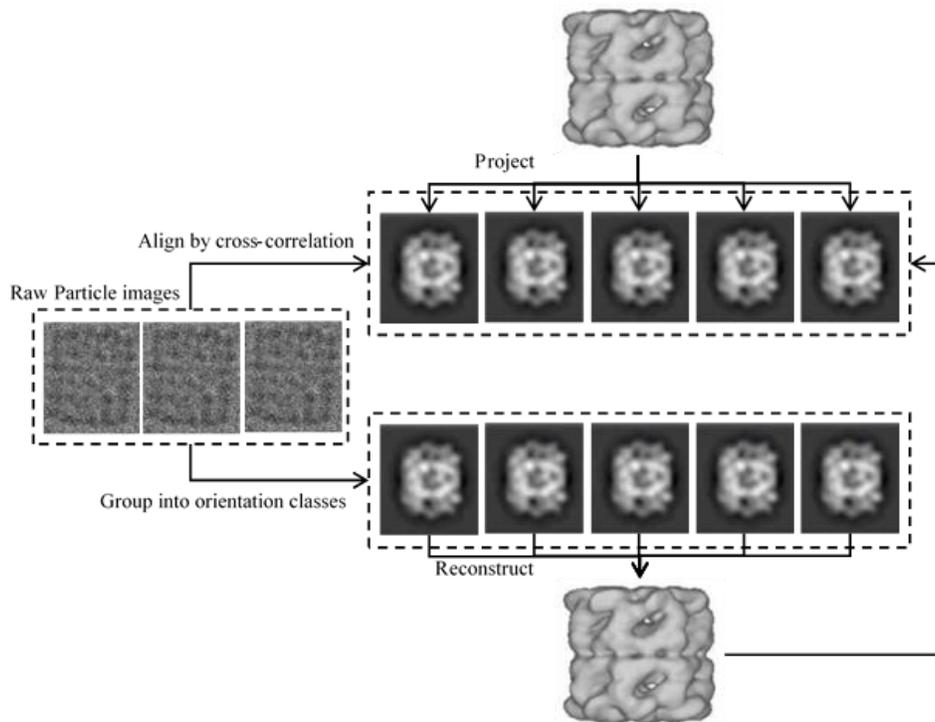


Figure 1.15: 3D Model Reconstruction using Projection Matching Method [32]

1.5 Outline

The content of each chapter in this dissertation is described as follow. Chapter 1, the introduction, gives the general background about single-particle cryo-EM workflow for protein structure determination, the overall framework of the x-ray crystallography, overall of the computational protein structure determination steps. The main content of this chapter is from the following publication:

Al-Azzawi, A., Ouadou, A., Tanner, J.J. et al. AutoCryoPicker: an unsupervised learning approach for fully automated single particle picking in Cryo-EM images. BMC Bioinformatics 20, 326 (2019) doi:10.1186/s12859-019-2926-y. [184].

Chapter 2 illustrates some recent related works for the semi-automated single particle picking in cryo-EM data.

Chapter 3 describes the AutoCryoPicker: an unsupervised learning approach for fully automated single particle picking in Cryo-EM images. The main content of this chapter is from the following publication:

Al-Azzawi, A., Ouadou, A.; Tanner, J.J.; Cheng, J., AutoCryoPicker: an unsupervised learning approach for fully automated single particle picking in Cryo-EM images. BMC Bioinformatics 20, 326 (2019) doi:10.1186/s12859-019-2926-y. [184]

Chapter 4 describes the CuperCryoPicker, a super-clustering approach for fully automated single particle picking in cryo-EM. The main content of this chapter is from the unpublished manuscript:

Al-Azzawi, A.; Ouadou, A.; Tanner, J.J.; Cheng, J., A Super-Clustering Approach for Fully Automated Single Particle Picking in Cryo-EM. Genes 2019, 10, 666. [185]

Chapter 5 describes the DeepCryoPicker: fully automated deep neural network for single protein particle picking in cryo-EM. The main content of this chapter is from the unpublished manuscript:

Al-Azzawi, Adil, A Ouadou, Anes, A Max R, Highsmith, Tanner, John J., A Cheng, Jianlin, DeepCryoPicker: Fully Automated Deep Neural Network for Single Protein Particle Picking in cryo-EM, bioRxiv preprint first posted online Sep. 10, 2019; doi: <http://dx.doi.org/10.1101/763839>.

Chapter 6 describes the DeepCryoMap: fully automated cryo-EM particles alignment approach for 3D density maps reconstruction based deep supervised and unsupervised learning methods. The main content of this chapter is from the unpublished manuscript:

*Al-Azzawi, A., A, Ouadou, Anes, Ye Duan, Tanner, John J., Cheng, Jianlin,
DeepCryoPicker: Fully Automated Deep Neural Network for Single Protein Particle
Picking in cryo-EM. (To be submitted Nov/Dec. 2019)*

Chapter 7 is the conclusions and future works.

Chapter 2

Literature Review

Many different computational methods have been proposed for the automated semi-automated single particle picking over the past decades. Most of these methods are based on different techniques such as template-based matching, edge detection, feature extraction, and convolutional computational vision [12].

Single particle picking using template-based matching methods are very sensitive to noise and result a substantial fraction of false positives since the template-based matching methods rely on the local cross-correlation in which result from the false correlation peak [37] [38] [39] [40] [41]. Thus, some initial “good references” are selected in advance to ensure that those manual selected examples have less noise comparing with the other in the same (2D) micrographs. Similarly, the edge-based [7] and feature-based methods [42] [43] [44] are dramatically reduction of the performance results since they are sensitive to the lower contrast of the (2D) micrographs [7]. For this reason, also some “good

examples” are selected in advance in which avoiding the low-contrast particle examples in the (2D) micrographs. However, in different method [45] [46] which using one-layer artificial neural network or support vector machine (SVM) [47] in which the extracted features from the single particle images are insufficient for the classifier to distinguish between the “good particles” and the “bad ones”. In most cases, the ‘bad particles’ examples include the local aggregates, overlapped particles, background noise fluctuations, carbon-rich areas, and ice contamination. Thus, after initialized the classifier either the neural network or the SVM, an additional step “manual verification and selection” is required to sort out the “good examples” and isolate them from the “bad ones” [47].

Deep learning during the past few years progressively has grown in the machine learning field. It bases on extract features from big data by generating different layers from deep neural networks and outcome with robustness results against the low SNRs in many convolutional techniques in the computer vision field [48]. Furthermore, deep learning appears to be suitable approach for cryo-EM processing as the size of the micrographs (2D cryo-EM) data continually increases while the SNR of micrographs remains low [12]. Recently, three methods for single particle picking have been proposed based deep learning approach. EMAN2.21 (particle picking with convolution neural network [48], DeepEM [49], DeepPicker [50], FasetParticlePicker [51], and PIXER [12].

EMAN2.21 [48] proposed to train two CNNs. One for pick particles from the (2D) micrographs while another to distinguish between “good particles” and “bad ones”. For the good and bad references, both should be precisely selected based on two criteria. First, the good training samples “references” should be in pure good particles. Second, the bad training samples are a collection from noisy background references which are selected from

the bad noise region in the 2D micrograph in addition to some bad particle references such as large aggregation, ice contamination, or overlap particles.

DeepEM [49] To tackle the problem of the automated free-template particle picking, DeepEM proposed an automated particle recognition using binary classification approach based deep CNN learning. DeepEM requires manually select hundreds of particles (selected by humans) to create the training dataset that has both positive and negative examples of each training dataset. Then, using the sliding window to classify the sub images to particles or background.

DeepPicker [50] proposed fully automated particle picking approach using other molecules as training data to train the network based on using two CNNs modules (model training and particle picking). DeepPicker considers the absence of training data by suggesting an alternative training scheme called “semi-automated particle picking with an alternative training strategy”. This technique requires a small set of manually user’s selection training dataset (positive and negative particle samples) to train the CNN model and initialize the particle selection process. Then, the trained CNN classifier is used to select particle images from different testing (2D) micrographs that have the same protein molecule shape.

FastParticlePicker [51] proposed a fast-single particle picking in cryo-EM based on the standard approach of the object detection network using fast R-CNN and Caffe [52]. The FastParticlePicker requires to extract the coordinates (upper-left and lower-right corners) of each particle bounding box for each single particle in each individual (2D) micrograph to train the fast R-CNN on good training examples while the rest regions are

either a background or bad training examples. Then, cropping the (2D) micrographs with a sliding window and the testing performance relies on the classification network.

Chapter 3

AutoCryoPicker: An Unsupervised Learning Approach for Fully Automated Single Particle Picking in Cryo-EM Images

3.1 Introduction

An important task of macromolecular structure determination by cryo-electron microscopy (cryo-EM) is the identification of single particles in micrographs (particle picking). Due to the necessity of human involvement in the process, current particle picking techniques are time consuming and often result in many false positives and negatives. Adjusting the parameters to eliminate false positives often excludes true particles in certain orientations. The supervised machine learning (e.g. deep learning) methods for particle picking often need a large training dataset, which requires extensive manual annotation. Other reference-dependent methods rely on low-resolution templates for particle detection, matching and

picking, and therefore, are not fully automated. These issues motivate us to develop a fully automated, unbiased framework for particle picking.

We design a fully automated, unsupervised approach for single particle picking in cryo-EM micrographs. Our approach consists of three stages: image preprocessing, particle clustering, and particle picking. The image preprocessing is based on multiple techniques including: image averaging, normalization, cryo-EM image contrast enhancement correction (CEC), histogram equalization, restoration, adaptive histogram equalization, guided image filtering, and morphological operations. Image preprocessing significantly improves the quality of original cryo-EM images. Our particle clustering method is based on an intensity distribution model which is much faster and more accurate than traditional K-means and Fuzzy C-Means (FCM) algorithms for single particle clustering. Our particle picking method, based on image cleaning and shape detection with a modified Circular Hough Transform algorithm, effectively detects the shape and the center of each particle and creates a bounding box encapsulating the particles.

AutoCryoPicker can automatically and effectively recognize particle-like objects from noisy cryo-EM micrographs without the need of labeled training data or human intervention making it a useful tool for cryo-EM protein structure determination.

3.2 Background

For decades, X-ray crystallography has been the dominant technique for obtaining high-resolution structures of macromolecules. Single-particle cryo-electron microscopy (cryo-EM) was traditionally used to provide low resolution structural information on large protein complexes that resisted crystallization (e.g., highly symmetric particles of viruses). Though the basic workflow of cryo-EM has not changed considerably over the years,

recent technological advances in sample preparation, computation, and especially instrumentation, have revolutionized the field of structural biology [52] [53] [54], allowing it to solve large protein structures at better than 3 Å resolution [55] [56] [57] [58].

Cryo-EM micrographs contains two-dimensional projections of the particles in different orientations. Generally, cryo-EM images have low contrast, due to the similarity of the electron density of the protein to that of the surrounding solution, as well as the limited electron dose used in data collection. In addition, the micrographs may contain sections of ice, deformed particles, protein aggregates, etc., which can complicate particle picking. Because a large number of single-particle images must be extracted from cryo-EM micrographs to form a reliable 3D reconstruction of the underlying structure, particle recognition, represents a significant bottleneck in cryo-EM structure determination.

To address the bottleneck, numerous computational approaches have been proposed to facilitate the particle picking process [59] [60] [61] [62] [63] [64] [65]. These methods can roughly be divided into two categories: generative methods [66] [67] [68] and discriminative classification methods [69] [70] [71] (e.g. the recent deep learning methods [72] [73]). The generative methods measure the similarity of an image region to a reference to identify particle candidates from micrographs. A typical generative method employs a template-matching technique with a cross-correlation similarity measure to accomplish particle selection. The discriminative methods first train a classifier on a labeled dataset of positive and negative particle examples, then apply it to detecting particle images from micrographs images.

DeepPicker [72] is a deep learning method for semi-automated particle selection and picking. The first part of the method involved the manual creation of training data. The

second part was fully automated by learning patterns from the training data to classify particles. DeepEM [73] uses a convolutional neural network (CNN) to recognize particles. The CNN was trained on a manually curated dataset. The training dataset was augmented by adding additional particles images generated by image rotation.

The existing unsupervised approaches distinguish the particle-like objects from background noise in micrographs via an unsupervised learning manner without the need of any labeled training data [61] [62] but, they do not fully exploit the intrinsic and unique characteristics of particles to facilitate automated particle picking. Therefore, the unsupervised approaches are often combined with the reference template matching or classification-based approaches to achieve good picking results. However, in this case, the training dataset has to be manually created to train the model. Although these approaches have greatly reduced time and effort spent on single-particle data analysis, most of them are not fully automated and still require substantial human intervention to initialize the particle selection process. For instance, most methods require users to prepare an initial set of high-quality reference particles used as templates to search for similar particle candidates from micrographs, while the discriminative approaches usually demand the user to manually pick a number of positive and negative samples to train the classifier first.

In this chapter, we develop a fully automated approach for particle picking (AutoCryoPicker) that is based on advanced image preprocessing, robust clustering via the intensity distribution, and sophisticated shape detection. The experimental results demonstrate that the fully automated particle picking scheme can accurately detect a number of particles that is comparable to those picked manually. The clustering method is also more accurate than k-means and Fuzzy C-means (FCM) for particle clustering.

Therefore, our new automated picking approach can significantly reduce time and labor spent on single-particle data analysis and thus greatly relieves a bottleneck in the automated cryo-EM structure determination pipeline.

3.3 Methods

Our AutoCryoPicker framework for automated particle picking is shown in Figure 3.1. In this framework, a user is not required to manually pick any particle from the micrographs. The fully automated approach has three main stages: preprocessing, clustering, and particle picking. In the preprocessing stage, several image processing methods are applied to enhance the input cryo-EM images such as image normalization, Contrast Enhancement Correction (CEC), etc. Clustering is done using three different algorithms k-means [74], Fuzzy C-Means (FCM) [75], and a new robustness clustering algorithm, which is the intensity-Based Clustering (IBC) that addresses some typical clustering issues such as cluster destabilization due to random initialization of cluster centers. In the particle picking stage, a final set of particles is selected from clustered particle candidates.

3.3.1 Stage 1: Pre-processing

A standard cryo-EM image is stored in the Mixed Raster Content (MRC) format, which defines a three-dimensional grid (array) of voxels each with a value corresponding to electron density or electric potential. In order to apply various image preprocessing techniques to improve the quality of noisy cryo-EM images, we convert cryo-EM images in the MRC format into widely used 16-bits PNG format using EMAN2 [76].

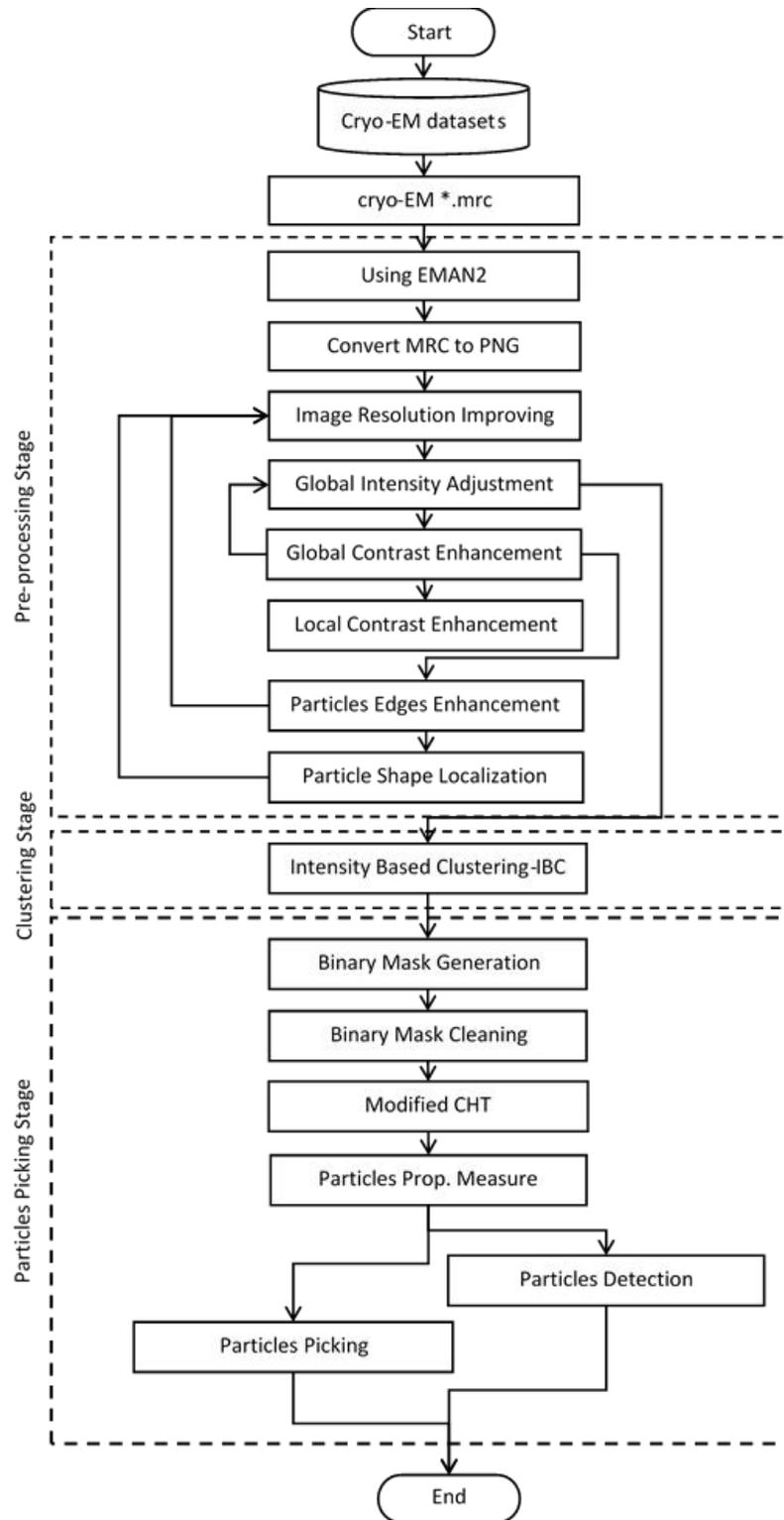


Figure 3.1: The general framework of AutoCryoPicker: Fully Automated Single Particle Picking. The dashed boxes represent three stages of the approach: pre-processing, particle clustering, and particle detection and picking. A solid box denotes an analysis step.

Since our goal is to use the unsupervised learning algorithm to cluster pixels based on the difference in intensity levels in any cryo-EM image, we select a set of advanced preprocessing tools to improve the quality of cryo-EM images. Those tools are tested on two different datasets.

There are two benefits of using the preprocessing. Firstly, those tools improve the contrast of the cryo-EM images by increasing the particle's intensity. Secondly, pre-grouping the pixels inside each particle makes them easier to be isolated by the clustering algorithm. Specifically, the preprocessing tools are selected based on three main objectives: enhancing the global contrast of the cryo-EM, enhancing the local contrast and increasing the intensity level of each particle, and enhancing the particle shapes inside the cryo-EM images.

In order to improve the entire contrast between particles and the background, image normalization is used first and then contrast enhancement and correction is applied to increase the global intensity value. To increase the global image contrast, histogram equalization is applied to enhance the pixel intensity level and then image restoration is used to recover and improve the quality of an image. To improve the local contrast and enhancing the definitions of edges in each particle, adaptive histogram equalization is employed. Moreover, guided image filtering is used to perform edge-preserving smoothing of each particle in the cryo-EM image. Finally, morphological image operation is applied to enhance the particle shape and make the particle regions similar to each other and different from the background regions. These preprocessing methods are described in detail in the following steps.

Step 1: Cryo-EM Image Resolution Improving

Cryo-EM images are affected by different factors that either corrupt the micrograph image signal by some gaussian noise or the image resolution. Different cryo-EM images have different artificial objects such as ice, which in some cases, have different thickness and similar ranges of the particle's pixel intensity value. In this case, in a single cryo-EM image, a small number of particles may not have significant difference of scatter power. Technically, the cryo-EM image resolution can be improved using computational image (signal) averaging based on blur motion elimination. This is selected as a main step of the contrast transfer function (CTF) based on the image quality evolution of the single particle cryo-EM and 3D reconstruction tool of viruses [77].

Different cryo-EM images have different intensity value ranges. In order to unify the range values, we renormalize the micrograph by setting the background mean to zero and background variance to one. In this normalization, the pixel values become the Z-score, i.e., the number of sigma's above noise level as shown in Equation (3.1) [78]:

$$x' = \frac{x - \bar{x}}{\sigma} \quad (3.1)$$

where \bar{x} is the mean of the intensity pixel values, and σ is the standard deviation. For instance, for an image consisting of 50 frames, we used the image averaging and normalization function in EMAN2 [76] to average the 50 frames, resulting in a converted cryo-EM image for further processing and analysis as shown in Figure 3.2.

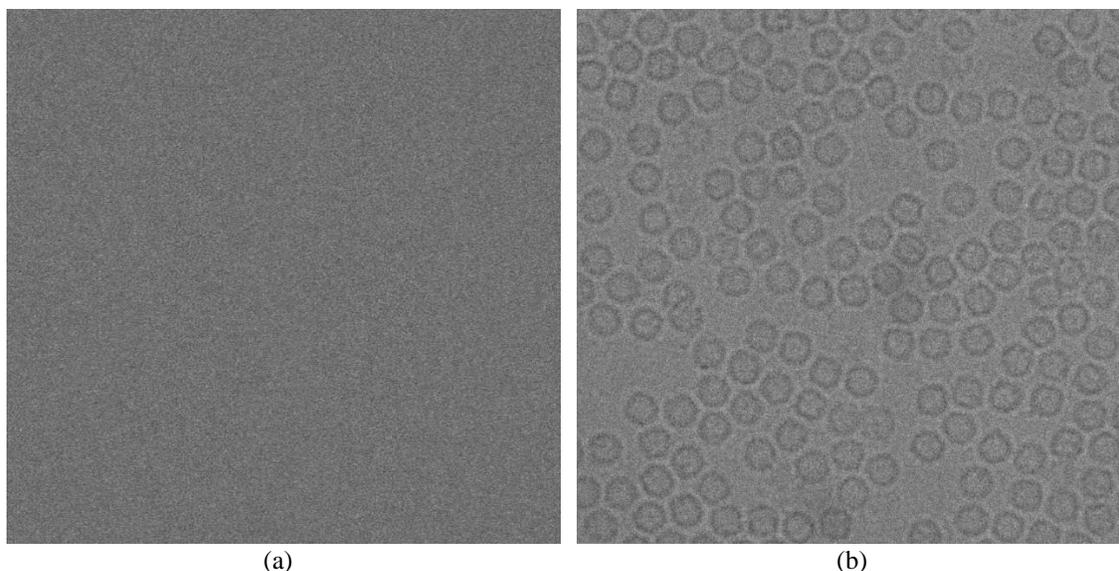
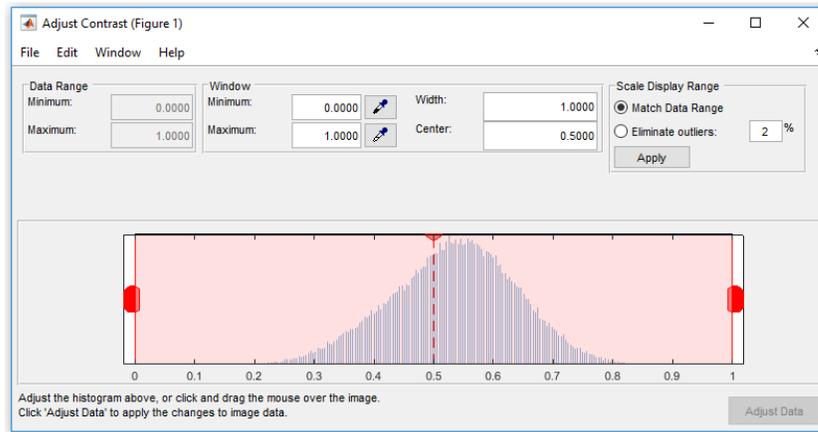


Figure 3.2: Cryo-EM image averaging and normalization result using EMAN2. (a) The original cryo-EM image (stack of 50 frame) in the MRC format before the averaging and normalization processing. (b) The cryo-EM image in PNG file format (single frame) after the averaging and normalization processing using EMAN2.

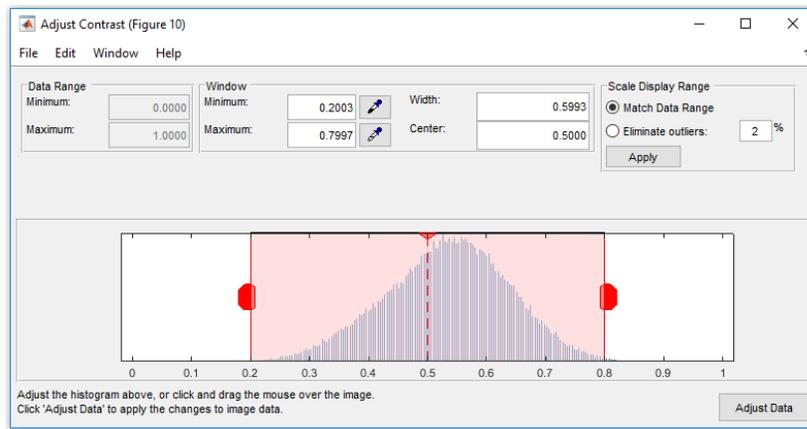
Step 2: Global Cryo-EM Intensity Adjustment

Low-dose micrograph imaging models the exposure to a very low intensity beam in a large defocus area that has both good particle distribution and thin ice. This imaging mode produces very low intensity cryo-EM images. To overcome this problem, intensity adjustment is applied to map the cryo-EM image intensity values to a new range. An Intensity Enhancement Correction (IEC) procedure is used to identify the descent image intensity and improve signal to noise ratio in cryo-EM images. In order to enhance the global intensity adjustment, we apply three different steps.

First: Find Limits to Contrast Stretch: In this step, the range of image intensity is specified by detecting the low and high values via a MATLAB function “*stretchlim*”, which returns a two-element vector that consists of the low and upper intensity limits as shown in the cryo-EM histogram in Figure 3(a).



(a)



(b)

Figure 3.3: Contrast transfer correction and adjustment process. (a) Illustration of the cryo-EM image histogram after the averaging and normalization step using EMAN2 and the a two-element vector that consists of the low and the upper intensity limits by default. The values in low_high specify the bottom 2% and the top 2% of all pixel values. (b) Illustration of the cryo-EM histogram (Histogram shrinking) after automatically detecting and specifying the low and high intensity range (e.g. [0.2-0.8]).

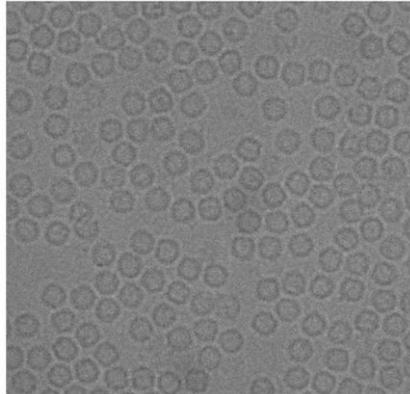
By default, values in low and high intensities specify the bottom 2% and the top 2% of pixel values. In this case, the intensity level of each cryo-EM should be unified. The gray values returned can be used by the “imadjust” function [28] to increase the contrast of an image as shown in Figure 3.3(b).

Second: Mid-Range Stretching: In this step, the cryo-EM image intensity values are stretched to improve their quality. The gray scale image pixels are mapped into the

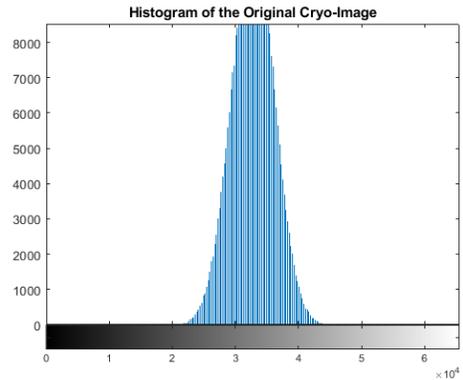
range [0 1] by dividing the intensity values of each pixel as shown in Equation (3.2).

$$x_{ij} = \frac{\text{Input Image}}{\text{High Range}} \quad (3.2)$$

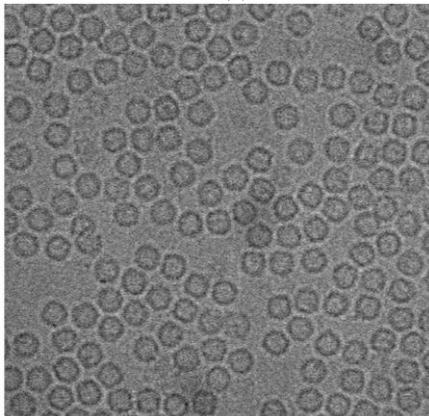
where i and j are the row and column index of cryo-EM image matrix respectively and the *High Range* is the highest intensity value in the input image. Figure 3.4(a) shows an original cryo-EM image, Figure 3.4(b) the histogram of the original image, Figure 3.4(c) a cryo-EM image after mid-range stretching and Figure 4(d) the histogram of the stretched image. The histogram in Figure 3.4(d) is more stretched than the original one in Figure 3.4(b).



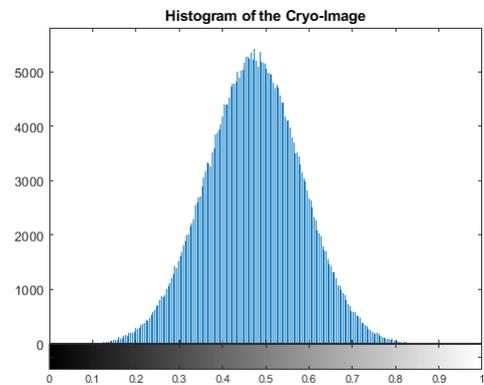
(a)



(b)



(c)



(d)

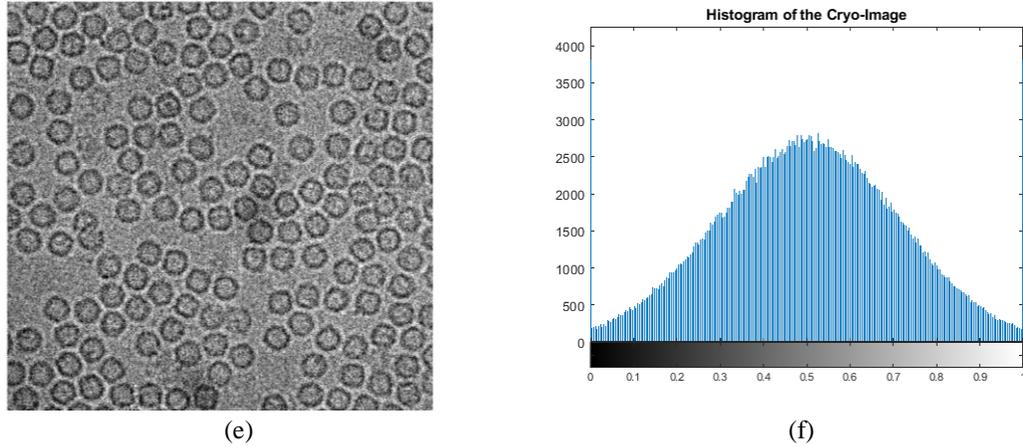


Figure 3.4: Cryo-EM Contrast Transfer Correction (CTC) process. (a) The original cryo-EM image after the applying the averaging and normalization process through the EMAN2 software. (b) Histogram of the original cryo-EM image. (c) The cryo-EM image after applying the mid-range stretching based on the low-high intensity range. (d) Histogram of the image in (c). (e) The cryo-EM image after applying the contrast enhancement correction (CEC) and image adjustment. (f) The histogram of the cryo-EM image after applying the contrast enhancement correction (CEC).

Third: Intensity Adjustment: The intensity values of the cryo-EM image are adjusted to new values in a condensed smaller range by using the MATLAB function “imadjust” [79]. Figure 3.4(e) shows an example of a cryo-EM image with contrast enhancement correction (CEC) and image adjustment, and Figure 3.4(f) shows the histogram of Figure 4(e) where the histogram looks more stretching and the contrast of the cryo-EM is enhanced compared with the original image in Figure 3.4(a).

For better demonstrating the effects of the preprocessing steps, we zoom-in one particle image from different datasets. Figure 3.5(a) and (i) show two original particle images from two different datasets. Figure 3.5(b) and (j) show the cryo-EM Image resolution being improved by image averaging and normalization. We can notice that image noise has been reduced. Figure 3.5(c) and (k) illustrates the same single particle images after the global intensity adjustment using Intensity Enhancement Correction (IEC). In comparison with the same particle region in the original micrograph after

normalization (Figure 3.5(b)), the particles in Figure 3.5(c) and (k) has more intensity contrast and are more isolated from the background than the ones in Figure 3.5(a) and (b), which will make it easier for clustering algorithms to identify them.

Step 3: Global Cryo-EM Contrast Enhancement

Due to the low-dose micrograph imaging mod on a large defocuses particles area, cryo-EM images have low contrast areas where the particles are difficult to detect. Histogram equalization [80] based on a uniform distribution is used to increase and enhance the intensity value of the image pixels. It increases and improves the global image contrast by mapping the original image histogram to a uniform histogram. Figure 3.5(d) and (l) show an example of a selected particle region in the micrograph after global contrast enhancement-based histogram equalization. Compared with the previous step (e.g. Figure 3.5(c) and (k)), the particle object regions have more contrast with the background.

Step 4: Cryo-EM Noise Suppressing

Due to the small electron doses and low contrast between protein and solvent, cryo-EM images tend to be rather noisy [81]. Image restoration is applied to denoise single particle cryo-EM images [82]. Based on the prior knowledge of the degradation process, the image restoration recovers and improves the quality of an image by identifying the type of noise and then removing it. Since the cryo-EM images are often corrupted by typically gaussian noise, the Weiner filter is chosen to model the noise. The Wiener filter is applied to remove additive noise and invert the blurring in cryo-EM images [83]. It minimizes the overall mean square error in the process of inverse filtering and noise smoothing. The Wiener filter in the Fourier domain can be expressed as in Equation (3.3).

$$W(f_1, f_2) = \frac{H * (f_1, f_2)S_{xx}(f_1, f_2)}{|H(f_1, f_2)|^2S_{xx}(f_1, f_2) + S_{\eta\eta}(f_1, f_2)} \quad (3.3)$$

where $S_{xx}(f_1, f_2) + S_{\eta\eta}(f_1, f_2)$ are respectively the power spectra of the original image and the additive noise, and $H(f_1, f_2)$ is the blurring filter. Figure 5(e) and (m) show two different zoom-in particles after applying noise suppressing based image restoration using Wiener filtering. We notice that, in both cases, some background noise is removed, and the structure of the particle object appears more distinctly than the particle object in the previous step (Figure 3.5(a)-(d)).

Step 5: Local Particles Contrast Enhancement in cryo-EM

In general, the particle picking process depends on the quality of the particles in the cryo-EM. Since there are too many low-quality particle shapes in the cryo-EM images, the local features of the particles such as the contrast, intensity level, and edges, need to be improved and enhanced [77]. Using adaptive histogram equalization (AHE) [83] the particle edges are locally enhanced in the cryo-EM. This is done by improving the local contrast between the particles and background. It provides a sophisticated technique for contrast dynamic range modification (CDRM) based on the intensity histogram shape description. It is applied to small regions of cryo-EM images, called tiles. It enhances the contrast of each tile so that the histogram of the output region approximately matches a specified histogram. The Adaptive Histogram Equalization combines neighboring tiles using bilinear interpolation to eliminate artificially induced boundaries. It is based on a probability model to enhance the contrast condition of each small region (sub-rejoin) using Equation (3.4) [83]:

$$p_{rx}(i) = \frac{1}{4} + \left(1 - \frac{1}{4}\right) \Phi[(x - \mu_{ij})\sigma_i^{-1}] \quad (3.4)$$

where $p_{rx}(i)$ is the image contrast-limited adaptive histogram equalization function of pixel value and Φ denotes the cumulative gaussian distribution function for each region, which has a separate location parameter estimate for each region. 1/4 is a constant for the 4-choice task [83]. Figure 3.5(f) and (n) show two different zoom-in particles after applying local particles contrast enhancement based on contrast-limited adaptive histogram equalization. The particle object intensity (contrast) is significantly improved and enhanced. In both examples, particles look darker and have a higher contrast than the previous particle images (Figure 3.5(e) and (m)).

Step 6: Particle Edges Enhancement in cryo-EM

In order to localize each particle object in the cryo-EM image, particle edges enhancement is proposed to isolate the particle shapes in the cryo-EM image. Edge-preserving smoothing technique is used to locally smooth and enhance the particle edges in order to localize different particles in any cryo-EM. Guided image filtering [84] is employed to perform edge-preserving and smoothing using the content of a second image, called a guidance image, to influence the filtering. The guided filter generates the filtered output by considering the content of a guidance image, which can be the input image itself or a different image. It has a theoretical connection with the matting Laplacian matrix [84] and can better utilize the structures in the guidance image. Let us assume that I is a guidance image filter, p is an input cryo-EM image, and q is an output image. Both I and p are given beforehand and can be identical. The filtered output at a pixel i is expressed as a weighted average as shown in Equation (3.5) [84]:

$$W_{ij} = \frac{1}{|W|^2} \sum_{k:i \in W_k} \left(1 + \frac{(I_i + \mu_k)(I_j + \mu_k)}{\sigma_k^2 + \epsilon} \right) \quad (3.5)$$

where i and j are pixel indices. The filter kernel W_{ij} is a function of the guidance cryo-EM image I and independent of p as in Equation (3.6) [84]:

$$q_i = W_{ij}(I)p_j \quad (3.6)$$

where q_i is the output image after the image guidance filtering and p_j is the input image after the image guidance filtering. A function “imguidedfilter” is used to implement the guided filtering. It performs the edge-preserving smoothing of the cryo-EM image in order to reduce the noise while keeping the particle edges. Figure 3.5(g) and (o) show two different zoom-in particles after applying particle edges enhancement using image guided filtering. The overall contrast of the particle in the cryo-EM image is improved. Compared to the same particle in the previous step (Figure 3.5(f) and (n)), particle edges appear more smoothly and some dark spots around the particle object become smoother and brighter while particle object edges become darker. In addition, the particle edges are more connected and have higher contrast than the background.

Step 7: Particle Shape Localization in cryo-EM

The last step of the pre-processing stage is the particle object localization and isolation step. In this step, we use morphological image processing [80], which is a collection of non-linear operations related to the shape or morphology of features in an image. Logical operations are applied to make particle regions like each other and different from the background regions. We apply an opening dilation operation followed by erosion with the same structuring element as shown in Equation (7) [80]:

$$A \bullet B = (A \oplus B) \ominus B \quad (3.7)$$

where A is the original cryo-EM image and B is the structure element. Figure 5 (h) and (p) show two different zoom-in particles after applying shape localization using morphological image operation (image closing with a structural element 5×5). The particle object is significantly improved and more isolated from the background. Also, the particle object structure is fully connected and has a higher contrast. The particle background is smoother, (see Figure 3.5(g) and (o)).

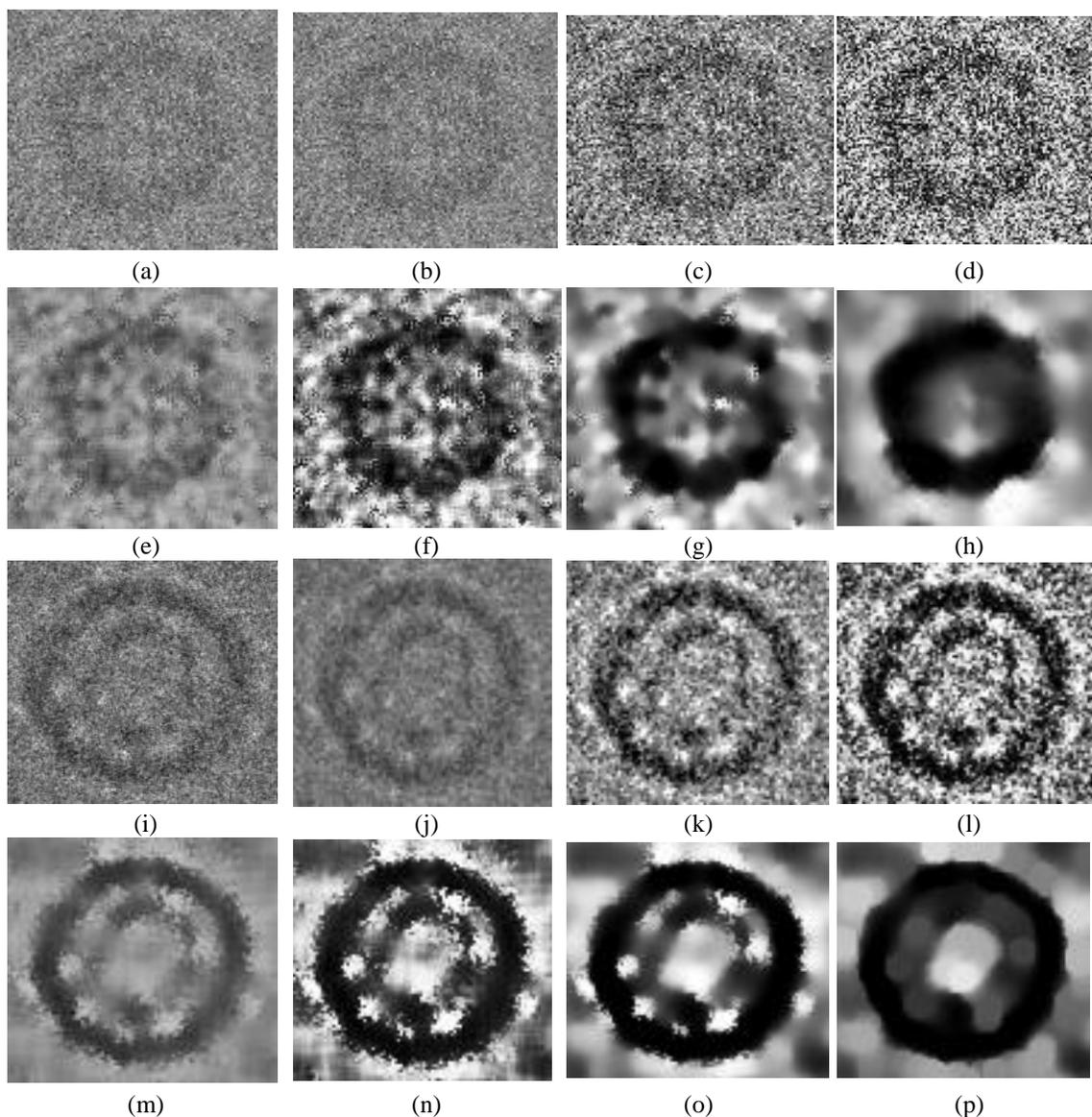


Figure 3.5: Illustration of effects of the cryo-EM image analysis on a zoom-in selected particle region using two different examples from two datasets. (a) An original zoom-in selected particle region in the micrograph image in Apoferritin dataset. (b) The normalized single particle image region. (c) The single particle region after applying the contrast enhancement correction (CEC). (d) The single particle region after applying the histogram equalization. (e) The single particle region after applying image resonance with Wiener filtering. (f) The single particle region after applying the contrast-limited adaptive histogram equalization. (g) The single particle region after applying image guided filtering. (h) The single particle region after applying morphological image operation. (i) An original zoom-in selected particle region in a micrograph image in the KLH dataset before the preprocessing steps. (j) The selected particle region in a micrograph image in the KLH dataset after normalization. (k) The selected particle region in a micrograph image in the KLH dataset after applying the contrast enhancement correction (CEC). (l) The selected particle region in a micrograph image in the KLH dataset after applying the histogram equalization. (m) The selected particle region in a micrograph image in the KLH dataset after applying image resonance with Wiener filtering. (n) The selected particle region in a micrograph image in the KLH dataset applying the contrast-limited adaptive histogram equalization. (o) The selected particle region in a micrograph image in the KLH dataset after applying image guided filtering. (p) The selected particle region in a micrograph image in the KLH dataset after applying morphological image operation.

3.3.2 Stage 2: Particle Clustering

In this stage, a binary mask is constructed using unsupervised learning clustering methods for particle isolation. Two standard clustering algorithms K-means [74] and FCM) [24] as well as a new intensity-based clustering (IBC) algorithm are applied. This clustering algorithm is based on an intensity distribution model, $P(i; d)$, which relates the intensity difference value d to the signed difference intensity values, i .

Intensity Based Clustering (IBC) Algorithm

Let $\{I_1, I_2, \dots, I_n\}$ be a set of images of the same modality containing the same anatomical structure of various subjects (i.e. particles in the cryo-EM images) and let $\{x^{(1)}, x^{(2)}, \dots, x^{(L)}\}$ be the set of all pixels in an image. Each pixel (x^l) will be grouped into several consistent “clusters” where the number of clusters is determined according to

a specific intensity interval size. To determine the initial number of clusters in the ICB algorithm, for example $K = 4$, if the adjusted intensity range is $[0.2, 0.8]$ as shown in Figure 3(b) and the interval size is 0.15, there are 4 initial cluster levels: the intensity level $[0.2-0.35]$ will be assigned to Cluster 1, $[0.35-0.5]$ to Cluster 2, $[0.5-0.65]$ to Cluster 3, and $[0.65-0.8]$ to Cluster 4. Here, $x^{(i)}$ is a real intensity value in a specific range, $1 \leq i \leq L$. Let $\{\theta_1, \theta_2, \dots, \theta_K\}$ be the set of the average intensity values of K clusters. The centers (θ_i, θ_j) of K clusters are initialized as evenly distributed intervals in the intensity range at the equal step size according to Equation (3.8):

$$S_{size} = \frac{I_{Range}}{K \times 0.15} \times 0.15 \quad (3.8)$$

where the I_{Range} is the difference between the maximum and the minimum intensity level in each image and 0.15 is the selected interval step size value. Let U_j be the index of the cluster whose center (θ_j) is closest to $x^{(i)}$. We denote the cluster assignment of all pixels after the desired centroid and cluster label (C) is predicted as $\{\langle x^{(1)}, U_1 \rangle, \langle x^{(2)}, U_2 \rangle, \dots, \langle x^{(n)}, U_j \rangle\}$. Each pixel $x^{(i)}$ is assigned to a specific cluster $k^{(j)}$ whose cluster center (θ_k) is closest to the $x^{(i)}$ according to the absolute intensity difference value using Equation (3.9):

$$C^{(i)} = |x^{(i)} - \theta_k| \quad (3.9)$$

The final clusters centroids are updated iteratively according to the average intensity (θ_j) of clusters according to Equation (3.10):

$$\theta_j = \frac{\sum_{i=1}^m \{x^{(i)}\}}{\sum_{i=1}^m \{L^{(i)} = j\}} \quad (3.10)$$

The procedure of the clustering algorithm for cryo-EM image clustering is shown below.

Algorithm 3.1 Intensity Based Clustering (IBC)

```
1: input: pre-processed cryo-EM image  $I_p$ 
2: return: clustered image  $I_c$ 
3: Initialize the minimum and maximum intensity mapping threshold values.
4: Identify the cluster number  $K$  to be 4 by mapping the input image intensity level
   to 4 levels based on the intensity adjusted min and max threshold values.
5: Convert the 2-D image  $I_p$  into 1-D  $I_v$  which has the intensity values of all the
   pixels.
6:    $L \leftarrow \text{height} \times \text{width}$  where  $L$  is the total number of pixels in the  $I_p$ 
7:    $V_{Max} \leftarrow \text{Max}[I_v]$  /*maximum values of intensity in the image*/
8:    $V_{Min} \leftarrow \text{Min}[I_v]$  /*minimum values of intensity in the image*/
9: for  $i = 1$  to  $K$  do
10:   $\theta_k \leftarrow \text{Int}_s[i]$  /*Initialize the cluster centroids  $\theta_1, \theta_2, \dots, \theta_K \in R^n$  by
    computing the interval step size using Equation (8) */
11: end for
12: repeat
13:  for  $i=1$  to every intensity pixel  $x^{(i)}$  do
14:    for  $j = 1$  to each cluster  $K$  do
15:      Assign  $x^{(i)}$  the cluster  $k$  whose center ( $\theta_k$ ) is closest to  $x^{(i)}$  according to
        the absolute intensity difference using Equation (9).
16:    end for
17:  end for
18:  for  $n = 1$  to each cluster  $K$  do
19:    Recompute the centroid of each cluster according to the average intensity
        ( $\theta_j$ ) of each cluster using Equation (3).
20:  end for
21: until convergence, i.e. there is no change in cluster centers.
```

Figure 3.6(a) and (b) show an example of different cryo-EM clustering results by using the intensity-based clustering method (ICB) with two cryo-EM datasets (Apoferritin [85] and KLH datasets [86]). It is noticed that the particles are most stably grouped in Cluster 1. Generally, the particles of the different images of the same protein can be best identified in the same specific cluster by the ICB method according to our experiments. However, the particles are not most stably grouped in the same cluster by k-

means and FCM algorithms due to their random initialization of cluster centers.

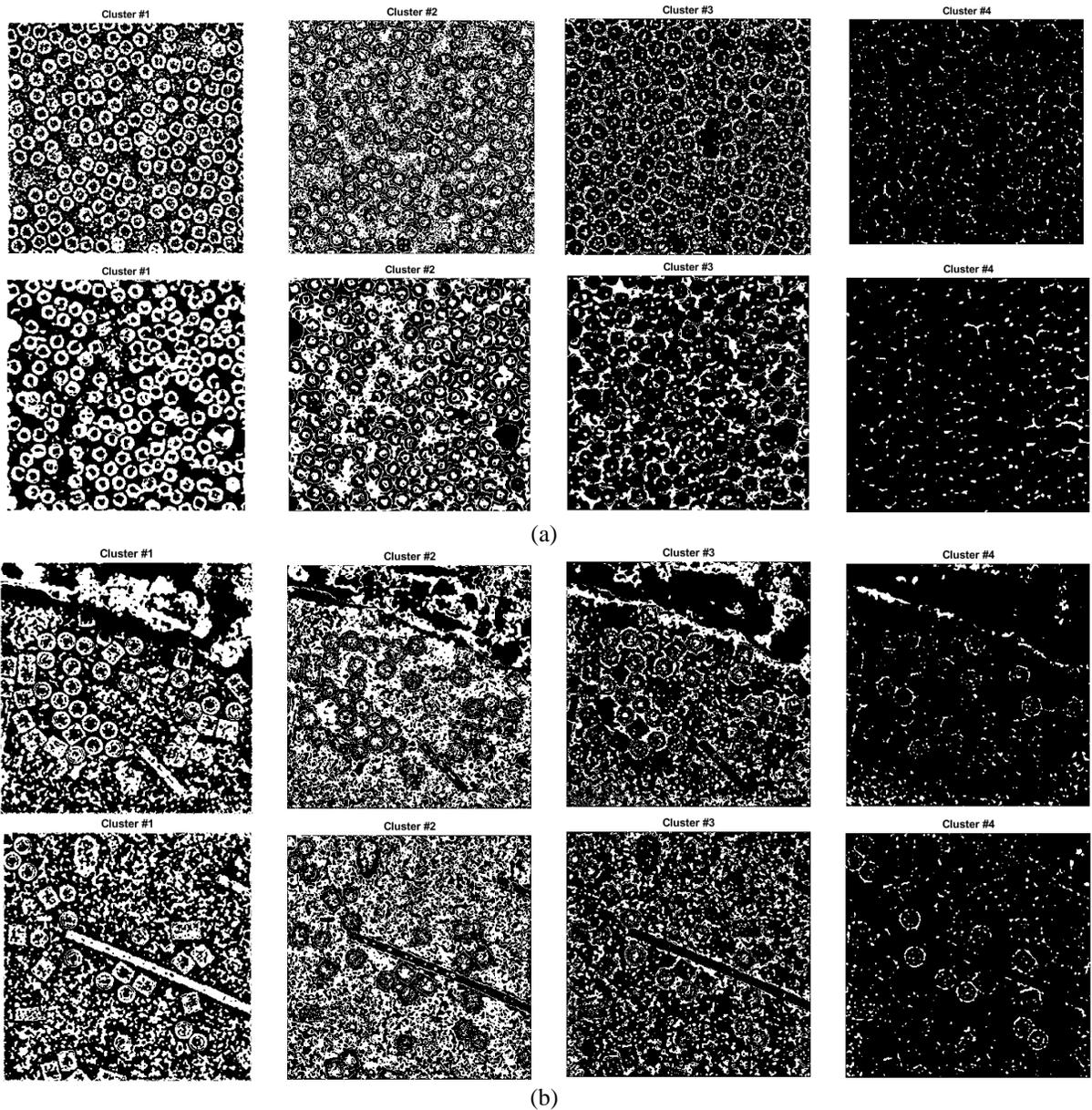
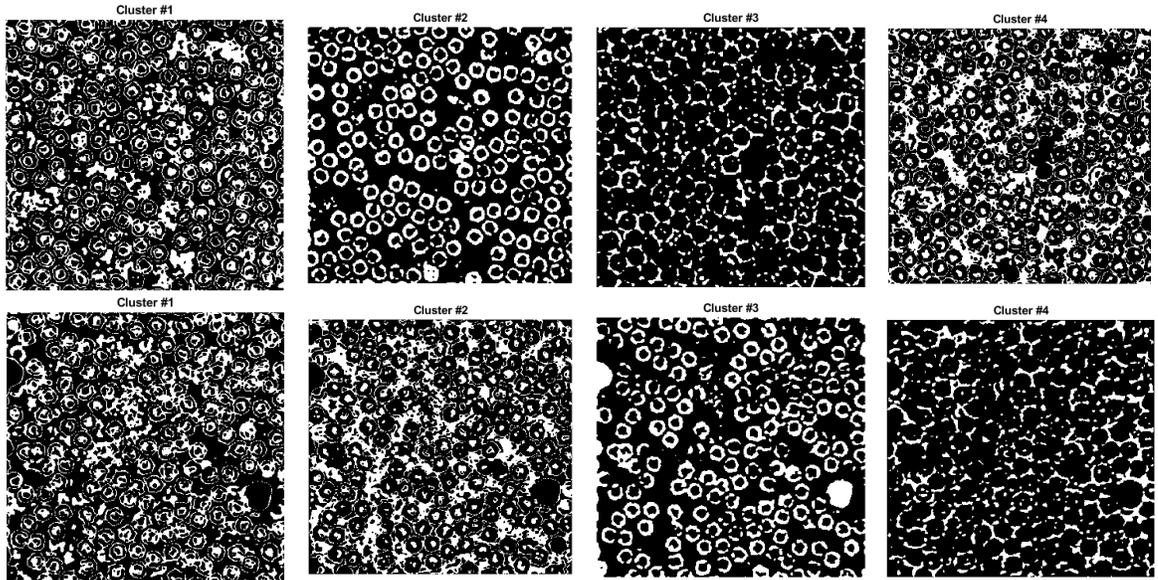


Figure 3.6: Different cryo-EM image clustering results using an Intensity-Based Clustering Algorithm (ICB). (a) Two sets of cryo-EM image clustering results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) on the Apoferritin dataset. Most real particles were always assigned to Cluster 1. (b) Two sets of cryo-EM image clustering results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) on the KLH dataset. Most real particles were always assigned to Cluster 1.

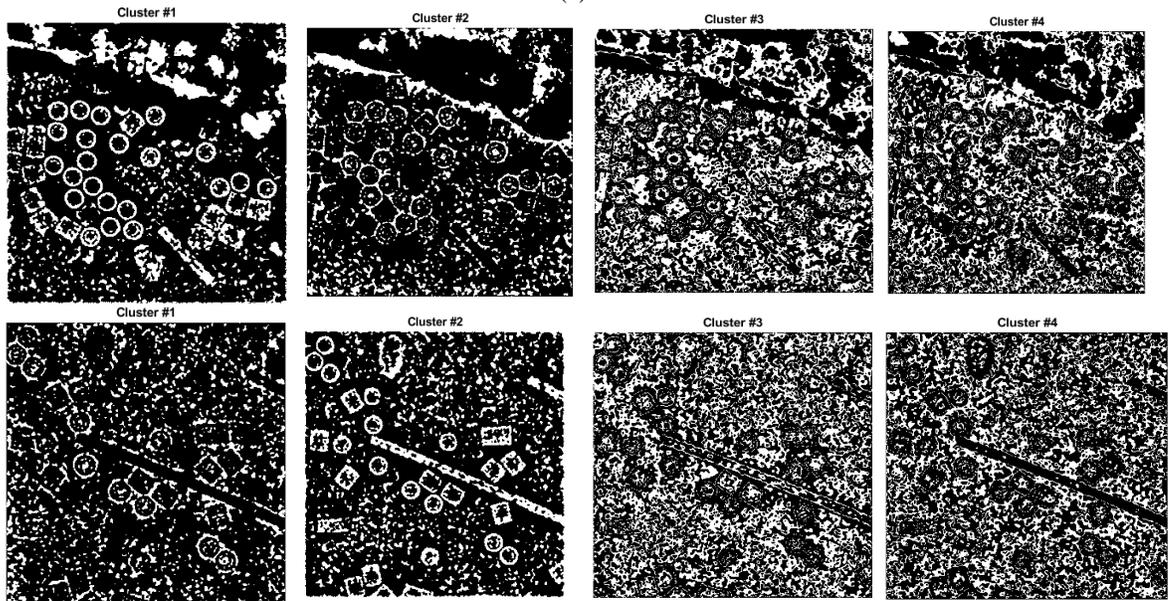
K-means and FCM Clustering Algorithms

Figure 3.7 and 3.8 show the clustering results of the same cryo-EM images using k-means

and FCM respectively. Note that the particles are located in different clusters. For instance, the particles clustering for two cryo-EM images in the first dataset (Apoferritin) using k-means is shown in Figure 3.7(a). The particles are grouped in two different clusters (Cluster 2 and 3, respectively). Figure 3.7(b) shows the same issue for the k-means on the second dataset (KLH). The same problem happens to FCM (Figure 3.8).



(a)



(b)

Figure 3.7: Different cryo-EM image clustering results using the k-means clustering algorithm. (a) The two sets of cryo-EM images clusters results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) on the Apoferritin dataset. Most real particles were assigned to Cluster 2 and Cluster 3, respectively. (b) The two sets of cryo-EM image clustering results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) on the KLH dataset. Most real particles were assigned Cluster 1 and Cluster 2, respectively.

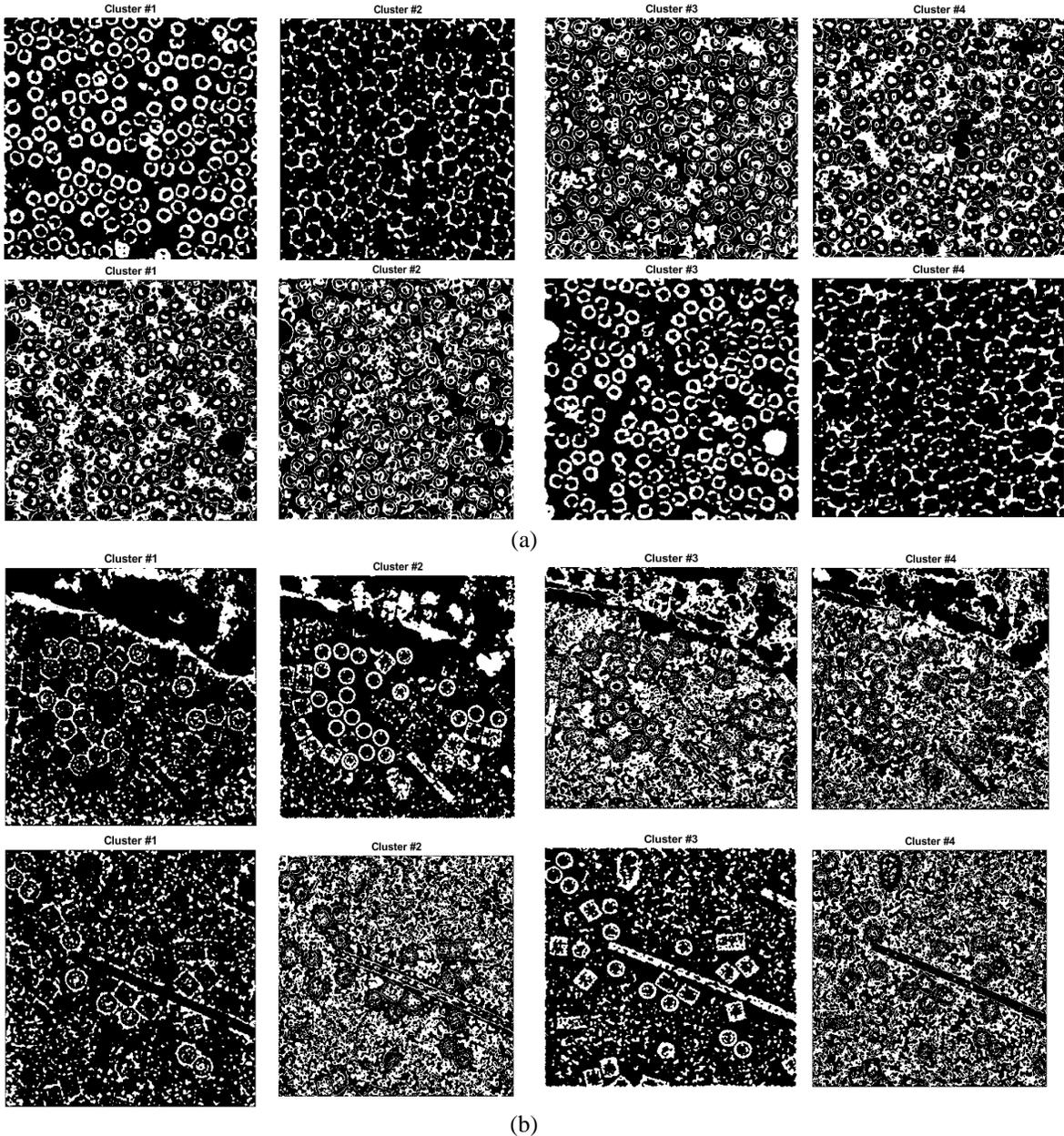


Figure 3.8: Different cryo-EM image clustering results using the FCM clustering algorithm. (a) Two sets of cryo-EM images clustering results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) on Apoferritin dataset. Most real particles were assigned to Cluster 1 and Cluster 3, respectively. (b) Two sets of cryo-EM image

clustering results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) on the KLH dataset. Most real particles were assigned to Cluster 2 and Cluster 3, respectively.

3.3.3 Stage 3: Particle Picking

The last stage of the AutoCryoPicker framework has two main steps. The first step is binary mask image cleaning and the second step is particle object detection and picking. In the first step, some post-processing operations (e.g. binary image region and hole filling, morphological image operation using image opening, and small object removal from the binary image) are performed to clean the binary mask produced in the clustering stage. In the second step, a modified Circular Hough Transform algorithm (CHT) [87] is applied to detect particles in the cleaned binary mask.

Step 1: Cryo-EM Cluster Image Cleaning and Non-Circular Object Removal

A binary mask of each cryo-EM cluster image is cleaned based on removal of the small and non-circular objects via size filtering and roundness filtering. An intermediate binary image $I_{clustered}$ is generated first to enlarge each object in the original binary clustered image by applying the morphological image operation on the original clustered image (binary) C_{Image} using image opening according to Equation (3.11)

$$I_{clustered} = (C_{Image} \ominus S_{sub_image}) \oplus S_{sub_image} \quad (3.11)$$

Where $I_{clustered}$ is the original clustered image, S_{sub_image} is the structural sub image using circular structure 5×5 , \ominus and \oplus denote erosion and dilation respectively. Then, the small objects and non-circular ones are removed from the intermediate images based on the object roundness class which is determine by computing the area and perimeters using the connected component pixel index list and the circularity based on the

Equation (3.12):

$$Circularities = \frac{allPerimeters^2}{4 \times \pi \times allAreas} \quad (3.12)$$

The image cleaning and small object removal algorithm is shown below.

Algorithm 3.2 Image Cleaning and Non-Circular Object Removal

- 1: **input:** I_c /*cluster cryo-EM image */
 - 2: **return:** I_{cc} /*cleaned cluster image */
 - 3: $I_{c1} \leftarrow imopen(I_c)$ /* Generate an intermediate clustered image by enlarge the small object using the image opening according to Equation (4) */.
 - 4: $L \leftarrow bwlabel(I_{c1})$ /* Label each object in the cluster image using MATLAB function (*bwlabel*) */.
 - 5: **for** $i=1$ to L **do** /* for each object in the intermediate clustered image*/
 - 6: $I_{object} \leftarrow state(L(k))$ /* determine the connected components (objects) in the image, including a list of indexing pixel locations for each one using MATLAB function (*regionprops*) */.
 - 7: $I_{object} \leftarrow bwareaopen(state(L(k)))$ /*remove the object that has not a fully connected edge using MATLAB function (*bwareaopen*)*/.
 - 8: **end for**
 - 9: $obj_{number} \leftarrow is\ member(I_{object})$ /*extract the number of object (particles)*/
 - 10: $L \leftarrow bwlabel$ /*label each object (particle)*/
 - 11: **for** $i=1$ to L **do** /* for each object (particles) */
 - 12: Do size filtering and roundness filtering
 - 13: $Areas \leftarrow [props.Area]$ /* Determine the region area of each connected component (object) using MATLAB function (*region props('Area')*) */
 - 14: $Perimeters \leftarrow [props.Perimeter]$ /* Determine the region perimeters of each connected component (object) using MATLAB function (*region props (Perimeter)*) */
 - 15: $Circularities \leftarrow allPerimeters^2 / ((4 \times \pi \times allAreas))$ /* Determine the region circularities of each connected component (object) using Equation (5).
 - 16: $Threshold_{area} \leftarrow 50000$ /*determine the average objects "roundness" circularities value. */
 - 17: $keeperObjects \leftarrow circularities < 3 \ \& \ Areas < threshold_{area}$ /* Keep objects that less than or equal to the average object's "roundness" circularities value using MATLAB function (*bwareaopen*) */.
 - 18: Get actual index numbers instead of a logical vector
 - 19: $I_{c2} \leftarrow$ produce new binary image with only the small, round objects in it
-

```

20:  $I_{cc} \leftarrow bwareaopen(I_c)$  /*remove the object that has not a fully connected
    edge*/
21: end for
22: Construct the output image containing only the object circular “roundness” object
    classes in each image.

```

Figure 3.9 shows the cryo-EM image cluster cleaning results (particles clustering) before and after image cleaning step. Figure 3.9(b), (f), and (j) show the particles clustering and cleaning results for the cryo-EM images from the Apoferritin dataset using ICB, k-means, and FCM respectively. Figure 3.9(d), (h), and (l) show the particles clustering and cleaning results for the cryo-EM images from the KLH dataset using ICB, k-means, and FCM respectively. It noticed that the proposed algorithm (ICB) produces significantly cleaner clustering images than the other two standard clustering algorithms.

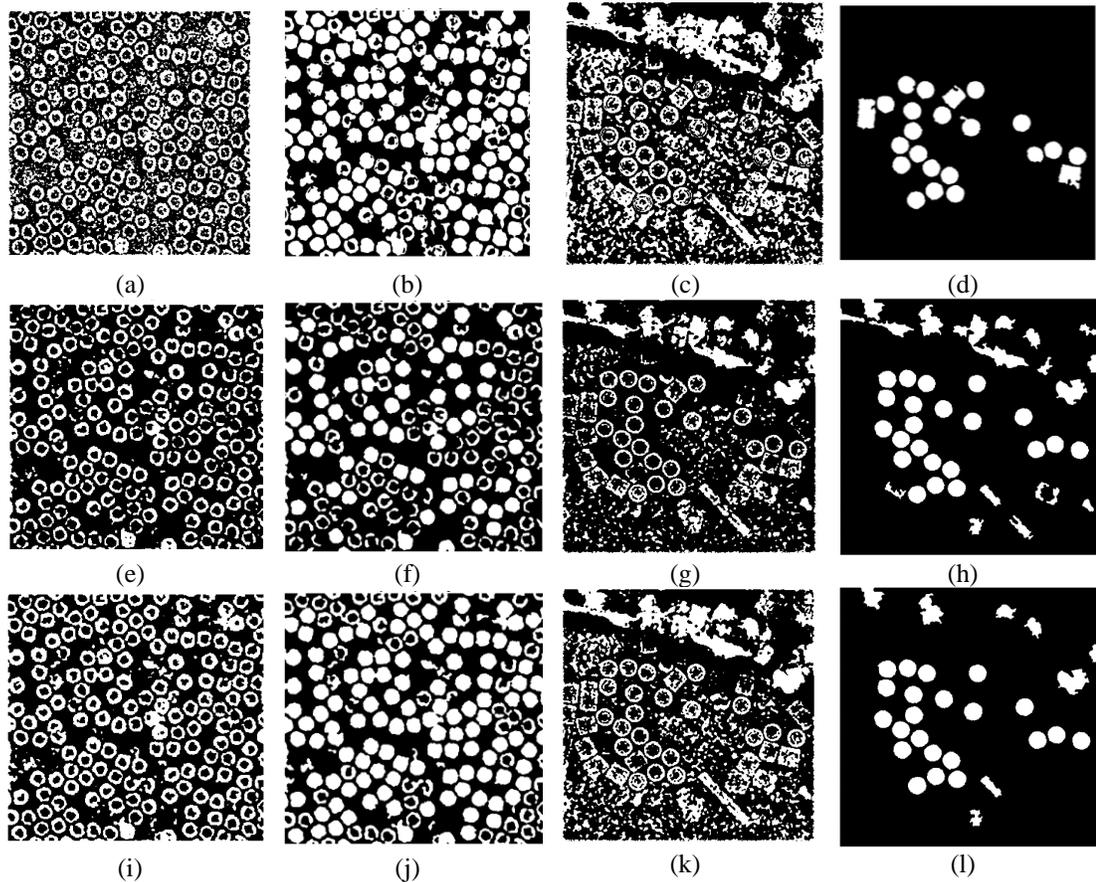


Figure 3.9: Cryo-EM Particle Clustering Results after Binary Image Cleaning and Non-Circular Object Removal. (a) The particle clustering image before binary image cleaning and non-circular object removal on the results of ICB clustering of a cryo-EM image from Apoferritin dataset. (b) The particle clustering image after binary image cleaning and non-circular object removal on the results of ICB clustering of a cryo-EM image from Apoferritin dataset. (c) The particle clustering image before binary image cleaning and non-circular object removal on the results of ICB clustering of a cryo-EM image from KLH dataset. (d) The particle clustering image after binary image cleaning and non-circular object removal on the results of ICB clustering of a cryo-EM image from KLH dataset. (e) The particle clustering image before binary image cleaning and non-circular object removal on the results of k-means clustering of a cryo-EM image from Apoferritin dataset. (f) The particle clustering image after binary image cleaning and non-circular object removal on the results of k-means clustering of a cryo-EM image from Apoferritin dataset. (g) The particle clustering image before binary image cleaning and non-circular object removal on the results of k-means clustering of a cryo-EM image from KLH dataset. (h) The particle clustering image after binary image cleaning and non-circular object removal on the results of k-means clustering of a cryo-EM image from KLH dataset. (i) The particles clustering image before binary image cleaning and non-circular object removal on the results of FCM clustering of a cryo-EM image from Apoferritin dataset. (j) The particle clustering image after binary image cleaning and non-circular object removal on the results of FCM clustering of a cryo-EM image from Apoferritin dataset. (k) The particle clustering image before binary image cleaning and non-circular object removal on the results of FCM clustering of a cryo-EM image from KLH dataset. (l) The particle clustering image after binary image cleaning and non-circular object removal on the results of FCM clustering of a cryo-EM image from KLH dataset.

Step 2: Top View (Circular) Particle Detection and Picking in Cryo-EM

Since the regular shape of the protein particle in the test cryo-EM dataset is a common shape – circle (top view), a Circular Hough Transform (CHT) [80] is used to detect particles in cluster images. For another common particle shape in cryo-EM images - square, a square shape detector would be needed. The CHT first generates a binary image based on each object’s edges. It then calculates the center and radius of each detected circular object. Each circular object in the binary image is defined by three parameters: the coordinates of the center (a, b) and the radius R as Equations (3.13) and (3.14) show [29]:

$$x = a + R \times \text{Cos}\theta \tag{3.13}$$

$$y = b + R \times \text{Sin}\theta \quad (3.14)$$

where θ ranges from 0 to 360° , and R is the radius. In this case, CHT is looking for each circular object of a particular radius (R) based on every boundary point (p) in the clustered image (binary image) using the original coordinated system (xy) space as shown in Equations (6) and (7). Then, every point in the (xy) space is equivalent to a circle in the (ab) space by rearranging Equations (3.13) and (3.14) to form the following Equations (3.15) and (3.16).

$$a = x_1 - R \times \text{Cos}\theta \quad (3.15)$$

$$b = y_1 - R \times \text{Sin}\theta \quad (3.16)$$

Next, all (R), indexed by theta, are retrieved from Hough space. For each of these $R(\theta)$, a vote is placed in Hough space at $p + R(\theta)$. Finally, the cells that receive the greatest number of votes are selected as the centers of the circular objects. In most applications of CHT it is common to use Canny edge detection [80] for the construction of the binary map. In this application Canny edge detection fails to identify sufficient points for CHT to detect the circular object center (top-view particles) as is shown in Figure 10(b) and (e). To overcome this issue, the canny edge detection step is replaced by our IBC algorithm. The pixels which makeup the outline of each circular object is extracted to form a boundary pixel list. This is done by removing interior pixels and then treating non-zero pixels as belonging to the object and considering zero valued pixels as the background. In our method a vector P , which contains two elements, is extracted by specifying the row and column coordinates of each point on the object boundary by tracing the 4-connected neighbors (setting them to 1). Each direction of the object boundary is traced to specify the

direction of the object boundary. The result is a Q-by 2 matrix called the B factor, where Q is the number of boundary pixels for the region. The modified Circular Hough Transform algorithm (CHT) is shown below.

Algorithm 3.3 Circular Hough Transformation (CHT)

- 1: **input:** I_{cc} /*cleaned cluster image */
 - 2: **return:** number of circular object (particles) in the cryo-EM
 - 3: center (x, y) of each circular particle,
 - 4: radius r of each one.
 - 5: $P \leftarrow Trace[object]$ /* Determine and extract the boundary pixels list by specifying the row and column coordinates of each point on the object boundary */
 - 6: Construct the output binary image containing only the object circular boundary for each object.
 - 7: /* *Hough Transform Begin**/
 - 8: **for** $i = 1$ to each edge point **do**
 - 9: Draw a circle with centre (x, y) in the edge point with r where (x, y) is the image pixels with position x , and y , r is the circular radius.
 - 10: Increment all coordinates (x, y) that the perimeter of the circle passes through in the accumulator.
 - 11: Find one or several maxima in the accumulator
 - 12: Map the found parameters (r, a, b) corresponding to the maxima back to the original image, where a , and b is the centre of the maxima.
 - 13: **end for**
 - 14: /* *Hough Transform End**/
-

The results of replacing the canny edge detection by our IBC algorithm are shown in Figure 3.10(c) and (f) using the same images that are used in original CHT (using canny edge detection) in Figure 3.10(b) and (e).

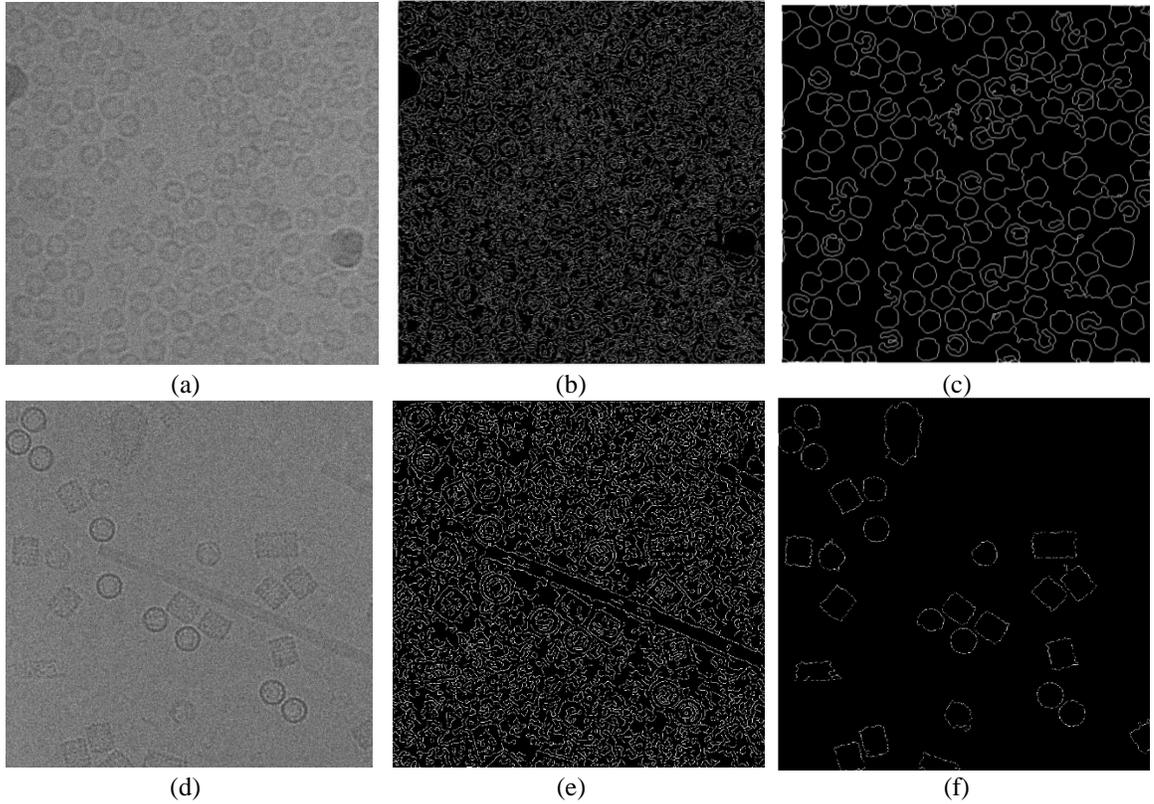
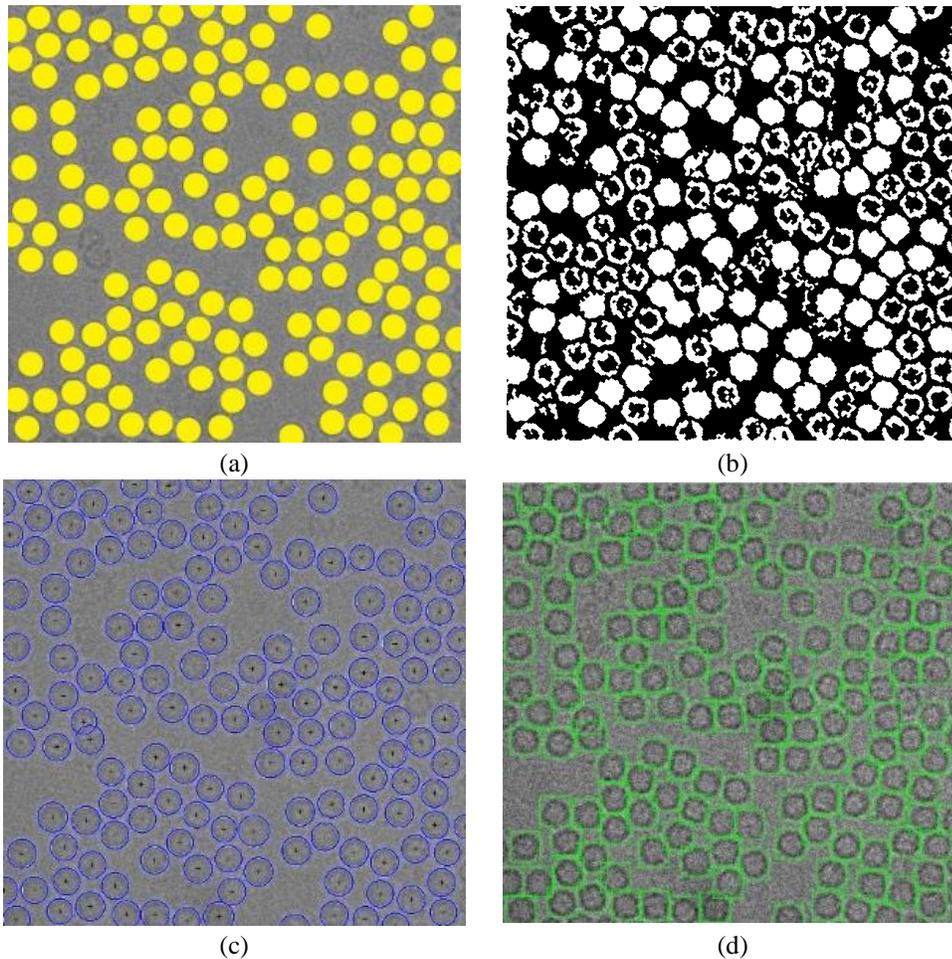
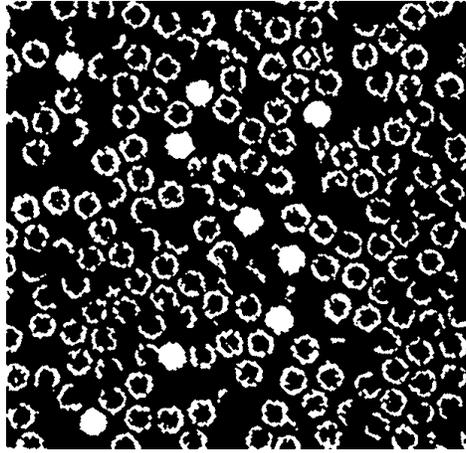


Figure 3.10: (a) (d) Original cryo-EM image from the Apoferritin and KLH. (b) Edge detection result that will be used later for CHT to detect the center of each circular object in the binary cryo-EM image from the Apoferritin dataset based on using canny edge detection. (c) Edge detection results that will be used later for CHT to detect the center of each circular object in the binary cryo-EM image from the Apoferritin dataset based on using the modified CHT based IBC clustering and boundary pixels list extraction (outline's boundary pixel). (e) Edge detection result that will be used later for CHT to detect the center of each circular object in the binary cryo-EM image from the KLH dataset based on using canny edge detection. (f) Edge detection results that will be used later for CHT to detect the center of each circular object in the binary cryo-EM image from the KLH dataset based on using the modified CHT based IBC clustering and boundary pixels list extraction (outline's boundary pixel).

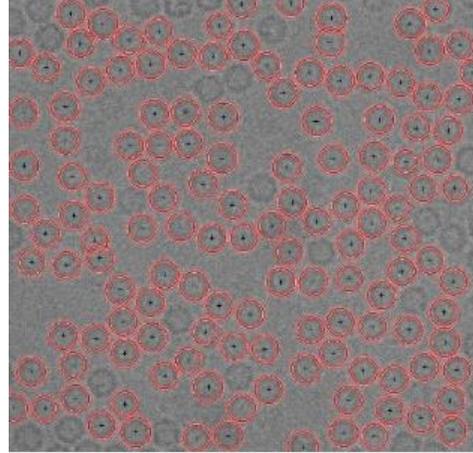
The detection algorithm returns the center and radius of each particle as is shown in Figure 3.11(c), (f), (i), (m), (p), and (s) based on the clustering results of the different clustering algorithms (ICB, k-means, and FCM) respectively for Apoferritin and KLH datasets. For instance, Figure 11(c) shows the center and radius of each particle illustrated by a '+' sign and a blue circle. A bounding box is drawn around each particle object in the cryo-EM image (Figure 3.11(d)). Figure 3.11(c) and (d) show the results of the particle

object detection and picking based on the ICB clustering and the Circular Hough Transform (CHT) on the first dataset (Apoferritin). Figure 3.10(m) and (n) show the same results on the second dataset (KLH dataset). Figure 3.11(f) and (g) show the results of the particle object detection and picking based on k-means clustering and the Circular Hough Transform (CHT) on the first dataset. Figure 3.11(p) and (q) show the same results on the second dataset. Finally, Figure 3.11(h) and (j) show the results of the particle object detection and picking based on the FCM clustering and the Circular Hough Transform (CHT) on the first dataset. Figure 3.11(s) and (t) show the same results on the second dataset.

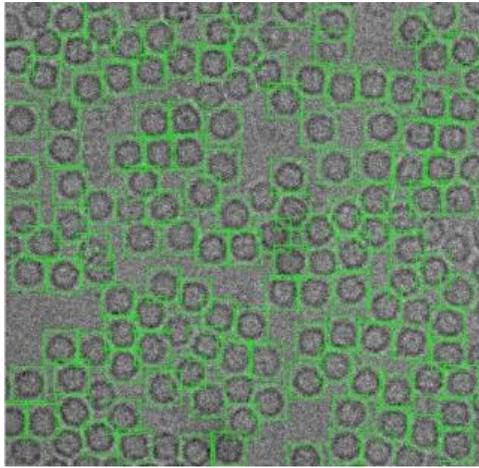




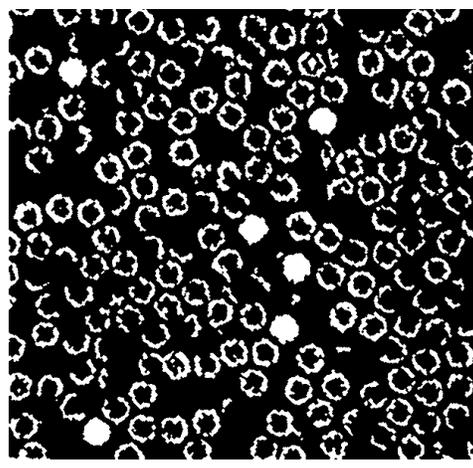
(e)



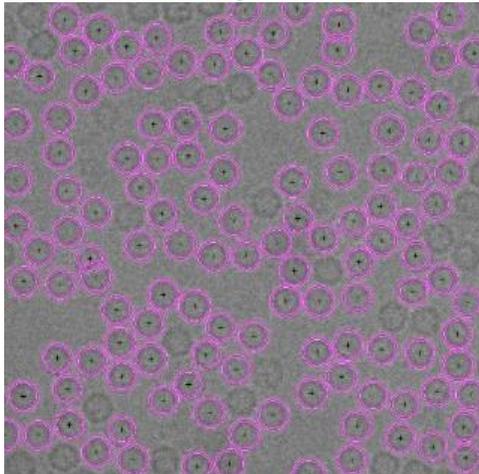
(f)



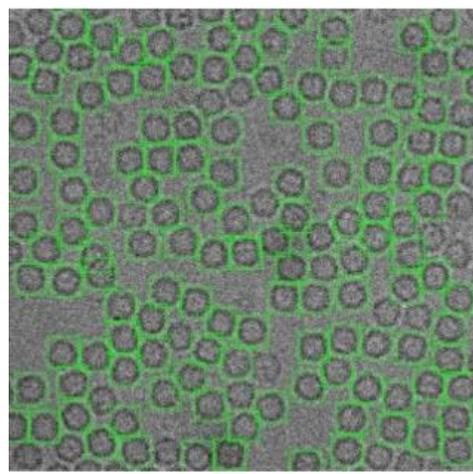
(g)



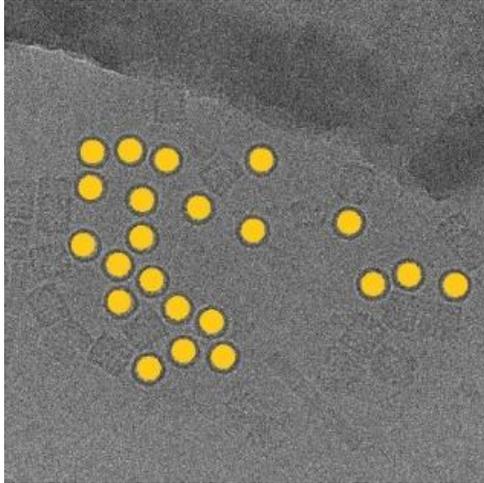
(h)



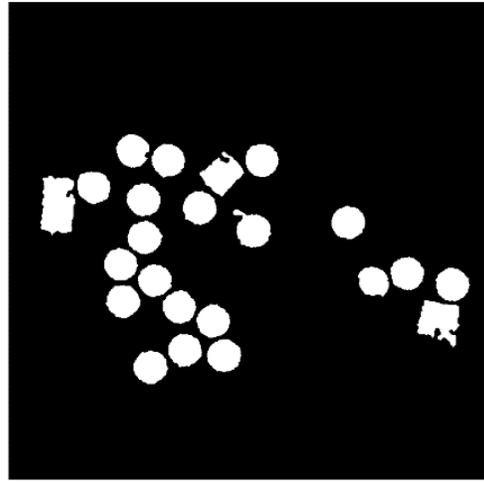
(i)



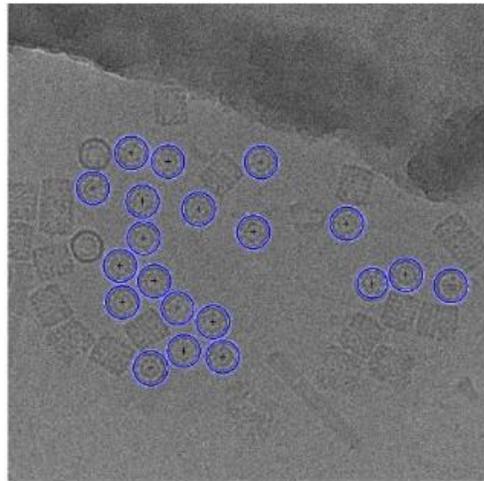
(j)



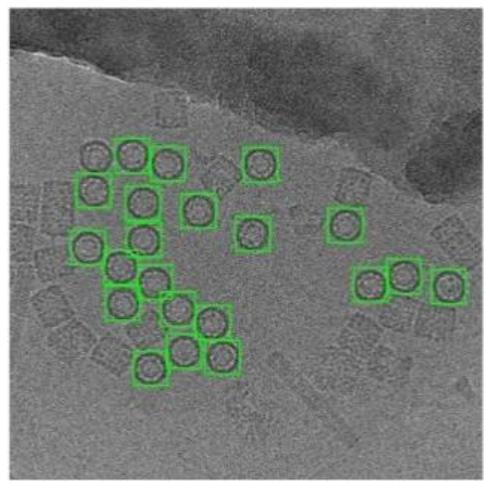
(k)



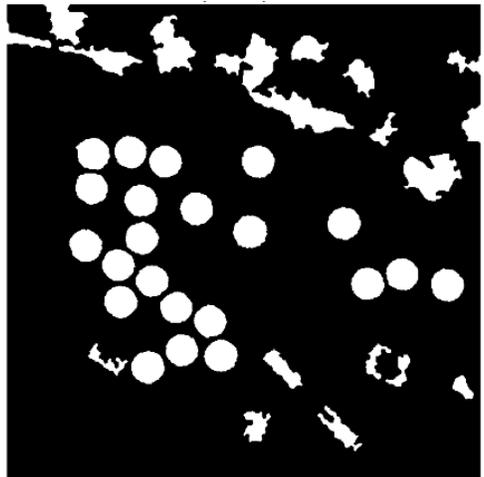
(l)



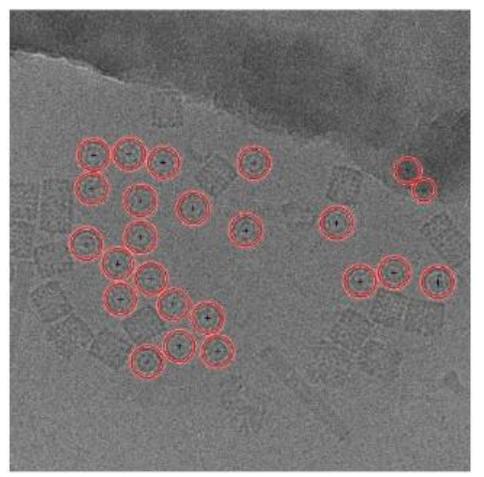
(m)



(n)



(o)



(p)

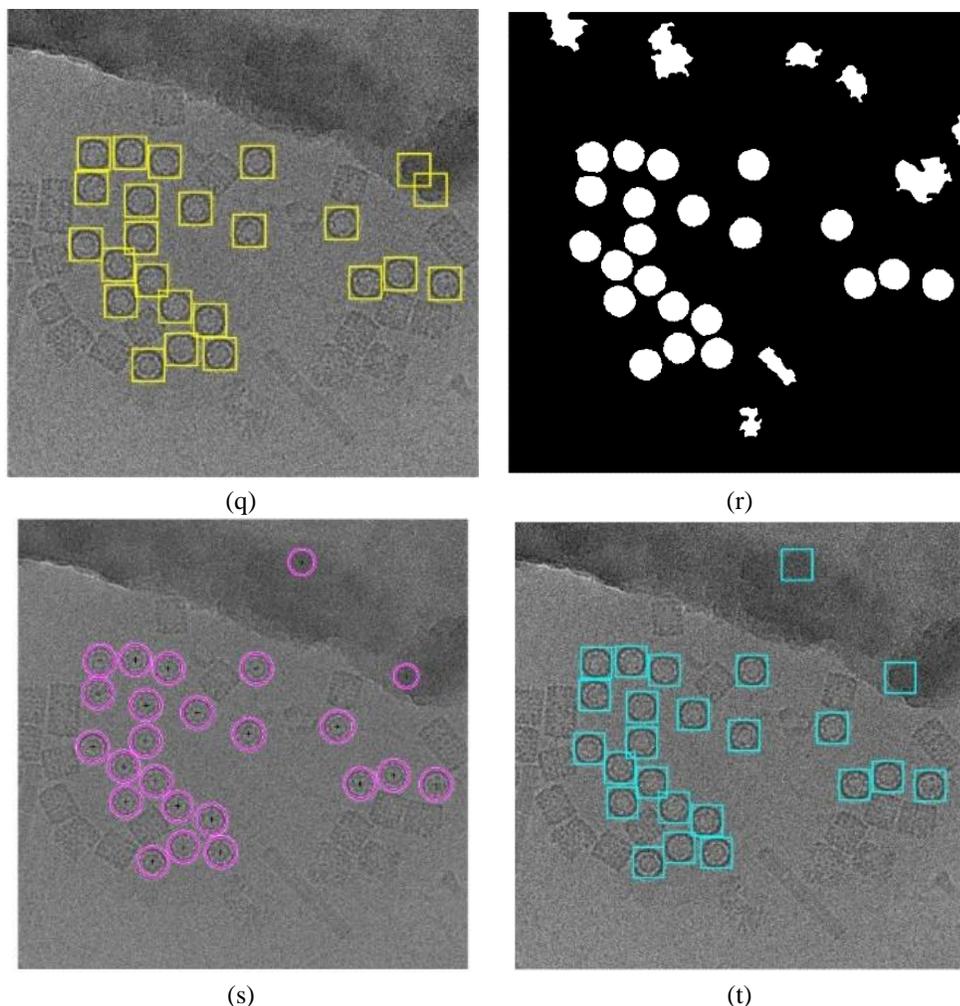


Figure 3.11: Top View (Circular) Particles Detection and Picking Results using Modified Circular Hough Transform (CHT). (a) The Ground truth (particles manually labelled) for the cryo-EM image from the Apoferritin dataset. (b) ICB clustering results after the binary image cleaning and non-circular objects removal (Apoferritin dataset). (c) The center of each particle illustrated by the '+' sign and the radius of each particle by the blue circle around each particle (ICB and Apoferritin dataset). (d) The bounding box for each particle object in the original cryo-EM image (ICB and Apoferritin dataset). (e) K-means clustering results after the binary image cleaning and non-circular objects removal (Apoferritin dataset). (f) The center of each particle illustrated by using the '+' sign and the radius of each particle by the blue circle around each particle (k-means results on Apoferritin dataset). (g) The bounding box for each particle (k-means results and Apoferritin dataset). (h) FCM clustering results after the binary image cleaning and non-circular objects removal (Apoferritin dataset). (i) The center of each particle illustrated by the '+' sign and the radius of each particle by the blue circle around each particle (FCM and Apoferritin dataset). (j) The bounding box for each particle in the original cryo-EM image (FCM results and Apoferritin dataset). (k) The ground truth (particles manually labeled) for the cryo-EM image from the KLH dataset. (l) ICB clustering results after the binary image cleaning and non-circular objects removal (KLH dataset). (m) The center of each particle illustrated

by the ‘+’ sign and the radius of each particle by the blue circle (ICB and KLH dataset). (n) The bounding box for each particle in the original cryo-EM image (ICB and KLH dataset). (o) K-means clustering results after the binary image cleaning and non-circular objects removal (KLH dataset). (p) Shows the center of each particle illustrated by the ‘+’ sign and the radius of each particle by the blue circle (k-means and KLH dataset). (q) The bounding box for each particle in the original cryo-EM image (k-means and KLH dataset). (r) FCM clustering results after the binary image cleaning and non-circular objects removal (KLH dataset). (s) The center of each particle illustrated by the ‘+’ sign and the radius of each particle by the blue circle (FCM and KLH dataset). (t) The bounding box for each particle in the original cryo-EM image (FCM and KLH dataset).

Step 3: Side View (Square) Particle Detection and Picking in Cryo-EM

Another common particle shape in the cryo-EM images is a square (side view). In this case, we add another step called circular and non-square object removal from the ICB clustering image after the cleaning and small object removal step in case of keeping the side view particle shapes (square). The main idea of the circular and non-square object removal is illustrated by two steps. First, each object (particle) in the cryo-EM (clustered image) is smoothed using a gaussian filter based on Equation (3.17) [80]:

$$g(m, n) = G_{\sigma}(m, n) \times f(m, n) \quad (3.17)$$

where $g(m, n)$ is the output smoothed image, $f(m, n)$ is the original input image (clustered), and G_{σ} is the gaussian kernel (mask) which is constructed based on using Equation (3.18) [80]:

$$G_{\sigma} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{m^2+n^2}{2\sigma^2}\right)} \quad (3.18)$$

where σ is the sigma (which represents the signal width), m , and n is the image dimension. Next, a parallel of the image cleaning and non-circular object removal algorithm is constructed called the circular and non-square object (particles) removal. In this case, first we remove any other objects such as ice artifacts in the binary image (clustered) by removing any object that does not have a square shape. The square object

shapes are determined by extracting the square object's edges using the MATLAB function (bwareaopen). Then, instead of keeping all circular objects found in each region by Algorithm 3.2, objects with circularity value below the computed average are removed. Finally, we implement the reserve objective function of the Algorithm 2 to eliminate any other object that has region's area size larger than the computed threshold value (average region size). This keeps the squares and removes the circular objects from the final mask (clustered image) of the cryo-EM. The circular and non-square object (particles) removal algorithm is shown below.

Algorithm 3.4 Circular and Non-Square Object (Particles) Removal

- 1: **input:** I_{cc} /*cleaned cluster image */
 - 2: **return:** I_{cs} /*cleaned cluster image with square shapes only*/
 - 3: $L \leftarrow \text{bxlabel}(I_{c1})$ /* Label each connected components (objects) in the cleaned image, including a list of pixel locations for each one using MATLAB function (bxlabel)*/
 - 4: Generate a gaussian kernel (mask) using Equation (11).
 - 5: **for** each object in the cleaned clustered image **do**
 - 6: Smooth each object shape using gaussian filter according Equation (10) with specific kernel size=5x5.
 - 7: **end for**
 - 8: **for** $i=1$ to L **do** /* **for** each object in the cleaned clustered image */
 - 9: Remove the object that has not a fully connected edge using MATLAB function (bwareaopen)
 - 10: $\text{allAreas} \leftarrow [\text{Area}(L(i))]$ /* Determine the region area of each connected component (object) using MATLAB function (regionprops('Area')) */
 - 11: $\text{allPerimeter} \leftarrow [\text{Perimeter}(L(i))]$ /* Determine the region perimeters of each connected component (object) using MATLAB function (regionprops('Perimeter')) */
 - 12: $\text{circularities} \leftarrow (\text{object})$ /* Determine the region circularities of each connected component (object) using Equation (5) */
 - 13: $\text{keeperObjects} \leftarrow \text{circularities} < 3 \ \& \ \text{Areas}$ /* Remove all the connected components that are bigger or equal to the average pixels */
 - 14: **end for**
 - 15: Construct an intermediate image containing only the non-circular object "roundness" in each image.
-

```

16: for  $i=1$  to  $L$  do /* for each object in the intermediate clustered image */
17:     Determine the region area of each connected component (object) using
        MATLAB function (regionprops('Area'))
18:      $Max\_Allowable\_Area \leftarrow Max(Area(i))$  /* find the max area for all
        objects*/
19: end for
20: for each object in the intermediate clustered image do
21:     Determine the region area of each connected component (object) using
        MATLAB function (regionprops('Area'))
22:     Determine the region perimeters of each connected component (object)
        using MATLAB function (regionprops('Perimeter'))
23:     Determine the region circularities of each connected component (object)
        using Equation (5).
24:     If  $circularities < Max\_Allowabl\_Area$  then
25:          $RoundObjects \leftarrow circularities < 3 \ \& \ Max\_Allowable\_Area$  /*
            Keep objects that less than the maximum allowable area using
            MATLAB function (bwareaopen)*/
26:          $obj\_position \leftarrow is\ member(I_{object})$  /*Extract each object position
            using MATLAB function (ismember)*/
27:     end if
28: end for
29:  $I_{cs} \leftarrow ismember(I_{cc}, RoundObjects) > 0$  /* Construct the cleaned output
        image containing only the squarest object classes occurring in each image */

```

Figure 3.12 shows an example of the cryo-EM clustered images after the circular and non-square object removal. For instance, Figure 3.12(a) shows cryo-EM clustered images after image cleaning and small objects removal although, Figure 3.12(b) shows the same cryo-EM clustered images after the circular and non-square object removal. After this step, the cleaned image has only the square particle shapes (side view) in the whole cryo-EM images. We can notice that not all the particles (side view) are cleaned after the second post-processing step, but some of them are according to the similarity between the $Max_Allowable_Area$ value and the circularities of each square particle object. If the circularity values between each particle shapes (side view-square and top view-circle) are very close, they are eliminated from the cleaned image.

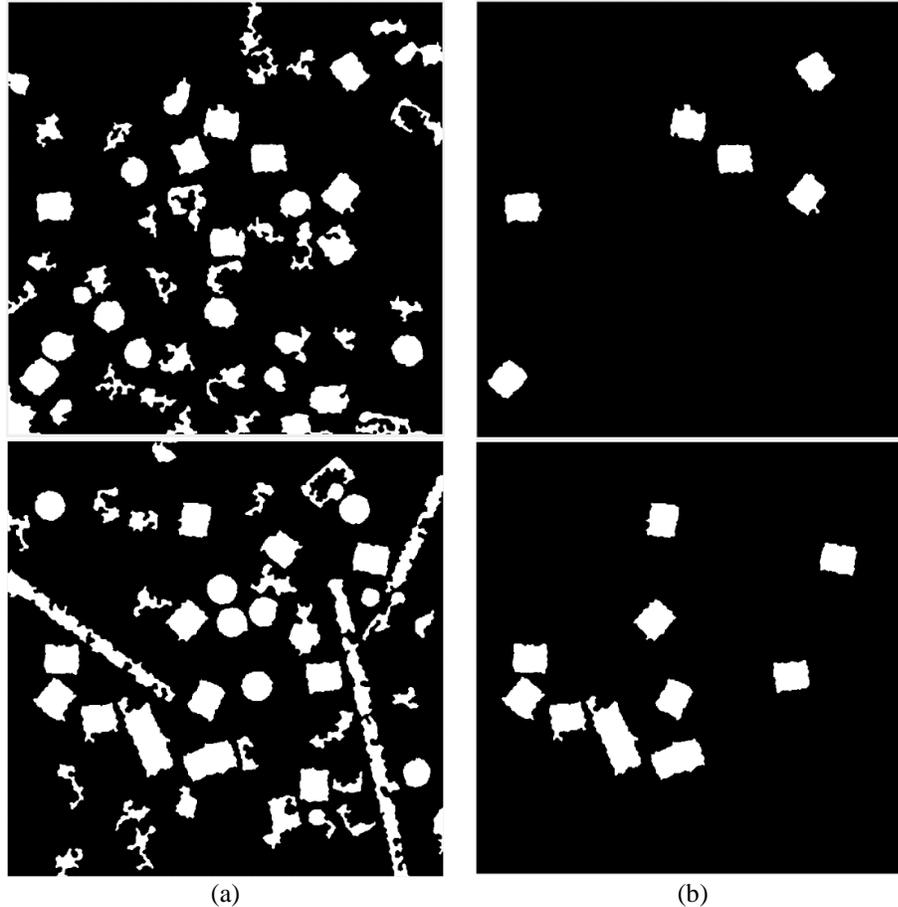


Figure 3.12: Cryo-EM clean clustered images after the circular and non-square object removal. (a) The cryo-EM clustered images after image cleaning and small objects removal. (b) The same cryo-EM clustered images after the circular and non-square object removal.

After the circular objects and artifacts have been being removed, the cryo-EM cleaned mask becomes significantly clear for detecting and selecting each square particle. The cleaned binary image has almost only the square objects (particles side view), in this case, we apply the square (side-view) particle detection and picking. The side-view particle detection and picking algorithm based on first determining the connected components (for each objects) in the cleaned cryo-EM image, including a list of pixel area and locations for each one. Second, since some of the artifact objects have a fully connected component and almost the same size the side view particles, we determine and extract the smallest

rectangle region area. Third, we determine the region area of each connected component (object) and keep the objects that are less than the smallest rectangle region area. Finally, bounding boxes are drawn around each discontinuous region (rectangle region area) after determining the connected components (objects) in the image. A list of pixel locations for each object is returned along with the extracted centroid, defined by the horizontal and vertical coordinates (x, y) . The square (side-view) particle detection and picking is shown below.

Algorithm 3.5 Square (Side View) Particle Detection and Picking

```

1: input:  $I_{CS}$  /*cleaned cluster image with square shapes only*/
2: return:  $I_{CPS}$  /*cleaned cluster image with perfect square shapes*/
3:  $L \leftarrow \text{bwlabel}(I_{C1})$  /* Label each connected components (objects) in the
   cleaned image and extract the total number of objects including a list of pixel
   locations for each one using MATLAB function (bwlabel)*/
4: for  $i=1$  to  $L$  do /* for each object in the cleaned clustered image*/
5:    $Stats \leftarrow \text{regionprops}(I_{CS})$  /* Determine the class measure properties of
   each connected component using MATLAB function (regionprops). */
6:    $Areas \leftarrow [props.Area]$  /* Determine the region area of each connected
   component (object) using MATLAB function ( $\text{regionprops}('Area')$ ) */
7:    $Min_{area} \leftarrow \min[Areas(i)]$  /* Determine and extract the smallest rectangle
   region area */.
8:    $keeperObjects \leftarrow Areas < Min_{area}$  /*keep objects that are less than the
   smallest rectangle region area. */
9: end for
10: for  $i=1$  to  $\text{size}(keeperObjects)$  do /* for each rectangle region area. in the
   cleaned clustered image */
11:    $[x, y] \leftarrow \text{centroid}(keeperObjects)$  /*Determine the connected
   components (objects) in the image, including a list of pixel locations for each
   one and extract the centroid is the horizontal coordinate (or x-coordinate) and
   vertical coordinate (or y-coordinate) using MATLAB function
   ( $\text{regionprops}('centroid')$ ) */.
12:   Draw all bounding box for each discontinuous region (rectangle region area).
13: end for

```

The results of the side-view particle shapes detection (square particles) are shown

in Figure 3.13(c) using different cryo-EM image samples from the KLH dataset.

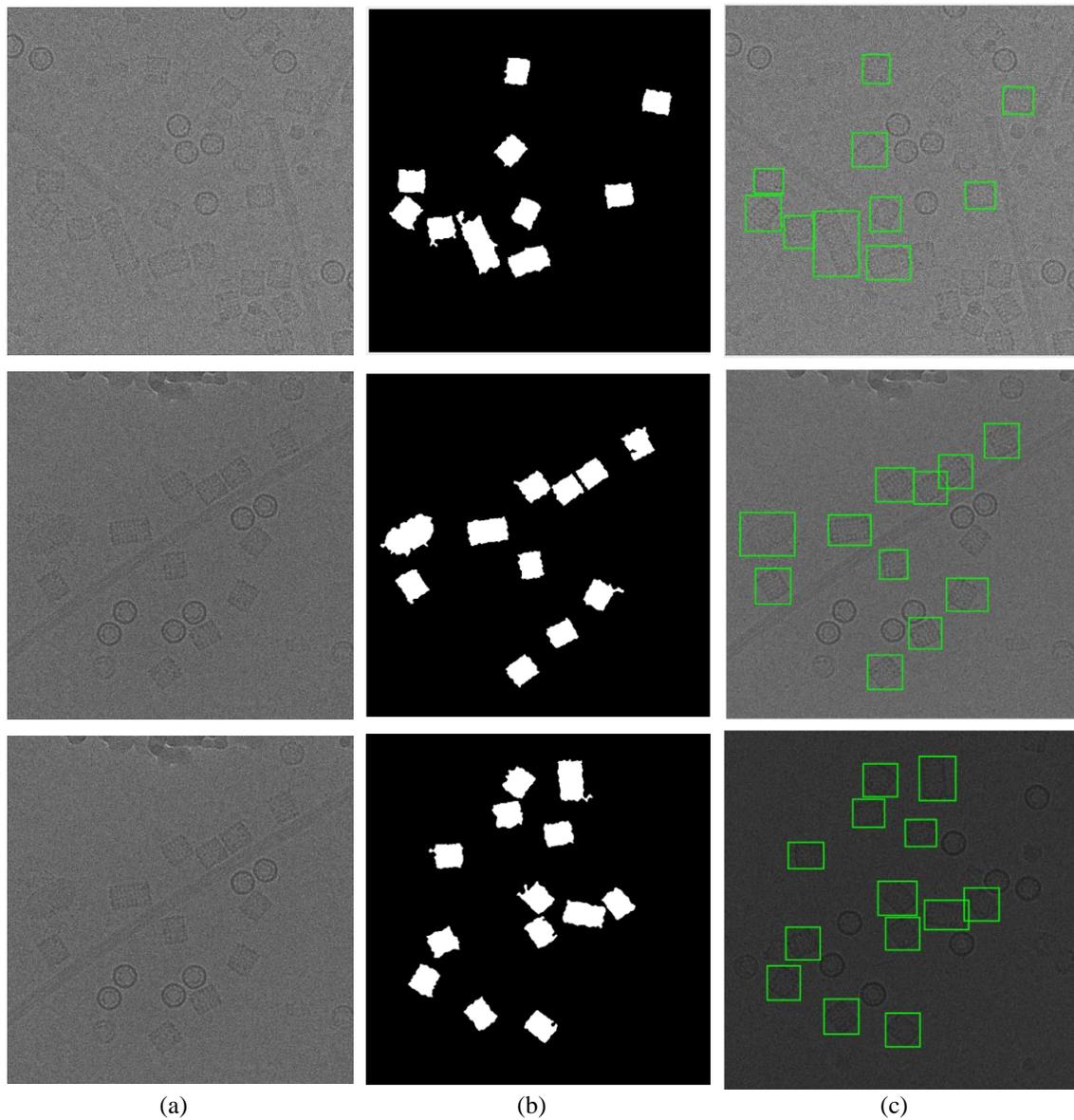


Figure 3.13: Side view (square) particles detection and picking results. (a) The original cryo-EM image (KLH dataset). (b) The result after circular and non-square object removal based on the ICB clustering algorithm. (c) Side view (square) particle detection results.

Step 4: Perfect Side View (Square) Particle Detection and Picking in Cryo-EM

Side-view particle detection (square) and picking is not very accurate. We can notice that some additional objects are attached to the original square particles in addition to some

overlapped particles, which are also selected as shown in the final detected results in Figure 13(b). To overcome this problem, we design another post processing algorithm called perfect square particles shape detection and picking.

The perfect square particles shape detection and picking has three main steps. The first step removes the small attached objects by smoothing each particle. Each particle is convolved with a 50x50 averaging filter kernel. The main particles smoothing (averaging) can be defined in Equation (3.19) [80]:

$$G_{\sigma} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{m^2+n^2}{2\sigma^2}\right)} \quad (3.19)$$

where $C(i, j)$ is defined as each particle sub image, B is the smoothing (averaging) kernel, p and q are the particle sub image dimensions. In the second step, after the small attached object are removed for each particle, we use the Feret diameter measures approach [88] to measure and correct the particle object dimensions. New perfect particle shapes are generated based on the maximum and the minimum Feret diameter. The maximum and minimum dimensions (width) of the particle object are used to identify the antipodal vertex pairs from the convex hull vertices set. Based on the new boundary box dimension, new perfect shapes are generated and inserted above each particle in the clean clustered cryo-EM image. The last step eliminates the outliers object (overlapped particles) by defining the average particles size and eliminating the outliers that have particle size larger than the average size. Then, the new boundary box is drawn based on the dimension of new particle object shapes. The perfect square particles shape detection and picking is shown below.

Algorithm 3.6 Perfect Square Particles Shape Detection and picking

1. **input:** I_{cps} /*cleaned cluster image with perfect square shapes*/
 2. $Kernek_{size} \leftarrow 5 \times 5$
 3. **return:** number of perfect square particles
-

-
4. I_{cps} /*cleaned cluster image with perfect square shapes*/
 5. $L \leftarrow bxlabel(I_{c1})$ /* Label all connected components (objects) in the cleaned image and extract the total number of objects including a list of pixel locations for each one using MATLAB function (*bxlabel*)*/.
 6. **for** each object in the cleaned clustered image **do**
 7. $I_{smoothed} \leftarrow Smooth(I_{object}, Kernek_{size})$ /* remove the small object by smoothing each object shape using Equation (10) with specific kernel size=5x5 */.
 8. **end for**
 9. **for** $i=1$ to each object (particle object) L **do** /* each object in the cleaned clustered image*/
 10. $I_{object} \leftarrow state(L(k))$ /* get each particle where k is the total number of objects in the cluster cryo-EM*/
 11. $L \leftarrow bxlabel(I_{smoothed})$ /*measure set of properties specified by properties for each 8-connected component in the binary image using MATLAB function (*regionprops*) */.
 12. Calculate the ferret properties
 13. **for** $i=1$ to each pixel in the cleaned binary mask **do**
 14. $P_{list} \leftarrow PixelList(objects)$ /*convert each object pixel to coordinates as an x-y order including a list of pixel locations for each one using MATLAB function (*PixelList*)*/
 15. $P_{hull} \leftarrow PixelHull(P_{list})$ /*extract the pixel hull diamond shapes using MATLAB function (*PixelHull*) */
 16. $P_{pairs} \leftarrow VerticPair(P_{hull})$ /*Determine the maximum Feret diameter and its orientation (maximum diameter) */
 17. $Feter_{diameter} \leftarrow Min(P_{pairs})$ /*computes the minimum ferret diameter*/
 18. $Area_{bounding} \leftarrow Min(Feter_{dim})$ /*extract the minimum bounding box area*/
 19. **end for**
 20. $P_{list} \leftarrow PixelList(objects)$ /*Convert each object pixel to coordinates as an x-y order including a list of pixel locations for each one using MATLAB function (*PixelList*)*/
 21. **for** $i=1$ to size(objects) **do** /* each object (particle) in the cleaned binary mask*/
 22. Extract the bounding box dimension
 23. Extract the 2D convex hull of the points (X, Y) for each object (particle) /* X and Y are column-vectors which presents a vector of point indices arranged in a counter-clockwise cycle around the hull */
 24. **end for**
 25. $I_s \leftarrow Insert(Area_{bounding}(x, y))$ /*Generate the final image with perfect squares shape generation*/
 26. $L \leftarrow bxlabel()$ /*find the connected components of all objects (particles) in binary image*/
 27. $Stats \leftarrow regionprops(L)$ /* measure properties of particle region*/
 28. determine and eliminate the outliers object (particles)
-

```

29. for  $i=1$  to  $L$  do /* each object in the binary image*/
30.    $Average\_area \leftarrow average(area(L(i)))$  /* find the average object area in
      binary image*/
31. end for
32. for  $i=1$  to  $L$  do /*for each object in the binary image*/
33.   if each  $particle\_area \leq average\_area$  then
34.     keep this object by getting the actual index numbers instead of a logical
      vector.
35.      $I_{object} \leftarrow particle\_area \leq average\_area$  /* keep objects that are
      less that the average rectangle region area */
36.      $obj\_postion \leftarrow is\ member(I_{object}$  /* Extract each object position using
      MATLAB function ( $ismember$ )*/.
37.   end if
38. end for
39. Determine the connected components (objects) in the image, including a list
      of pixel locations for each one and extract the centroid is the horizontal
      coordinate (or x-coordinate) and vertical coordinate (or y-coordinate) using
      MATLAB function ( $regionprops('centroid')$ ).
40. Draw all bounding boxes for each discontinuous region (rectangle region
      area).

```

Figure 3.14 shows an example of the perfect square particle shapes detection using Feret object diameter. Figure 3.14(a) shows the square particle shapes in the image after the shapes are smoothed and blurred. Figure 3.14(b) shows the new boundary box of each particle based on the Feret diameter measures. Figure 3.14(c) shows the perfect square particle shapes based on the Feret object diameter measurement. Figure 3.14(d) shows the square particles image after eliminating the outlier objects (overlapped particles). Figure 14(e) shows the square particle detection results (side view) based on the new Feret boundary box. Finally, Figure 3.14(f) shows the final results of different particle shape detection and picking (top and side view) based ICB clustering, modified CHT, and perfect square (side view) particle shapes detection using Feret object diameter.

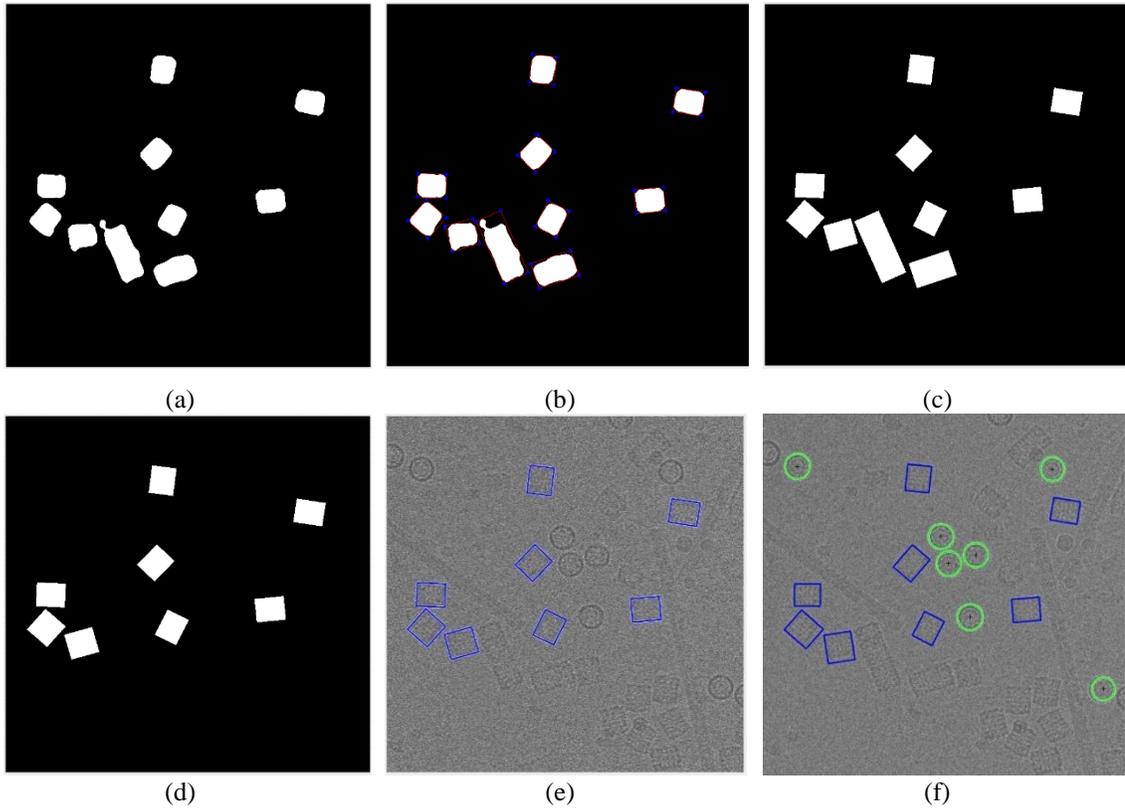
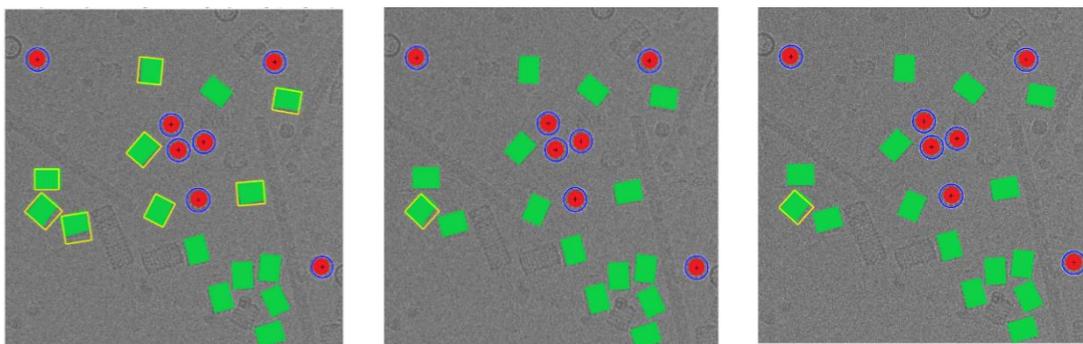


Figure 3.14: Perfect square (side view) particle shape detection using the Feret object diameter using (KLH dataset). (a) Square particle image after shapes smoothing and blurring. (b) Boundary boxes (each particle) based on Feret object diameter measurement. (c) Perfect square particle shapes that are generated based on the new boundary box dimension using Feret object diameter measurement. (d) Square particle image after the outlier objects are eliminated. (e) Square particle detection results (side view) based on the new Feret boundary box dimension. (f) The final results of two different particle shape detection and picking (top and side view) based on ICB clustering and modified CHT; and perfect square (side view) particle shapes detection using Feret object diameter.

It is noticed that there is almost no true positive (top view particles-circle) missing. In contrast, there are some true positive example of square particles (side view) missing. Figure 3.15 shows some extra cases of the particle detection and picking results for both cases (top and side view) using three different algorithms (ICB, k-means, and FCM). Figure 3.15(a) shows the original cryo-EM image, while Figure 3.15(b), (c), and (d) shows the target detection and picking image using ICB, k-means, and FCM respectively. Those examples have been manually labeled in the case of showing the detection and picking

performance. The red dots illustrate hand labeling of the circular particles (top view) while the green squares illustrate hand labeling the squares particles (side view), although, the blue circles showing the particle AutoCryoPicker detection and picking results for the top view particles, and the yellow squares showing the side view particles detection and picking results.

Figure 3.15 illustrates some cases in which AutoCryoPicker failed to detect and pick in both top and side views on the KLH dataset. In the third test example (Figure 3.15(b), (c), and (d)), there is one top view circular particle not detected by ICB, k-means, and FCM respectively. Figure 3.15(b) also shows some side view square particles not recognized by ICB clustering. In both cases (top and side view particles), there are almost no false positive detections by ICB clustering, indicating that AutoCryoPicker rarely picked objects from either the background area or icy area. Figure 3.15(c) and (d) show some side view particles not detected by k-means and FCM respectively. k-means and FCM missed more particles than ICB clustering. They had some false positives (Figure 3.15(c) and (d)). In one case, a side view was mistakenly detected as a top view, and in another case a background area was detected as a top view.



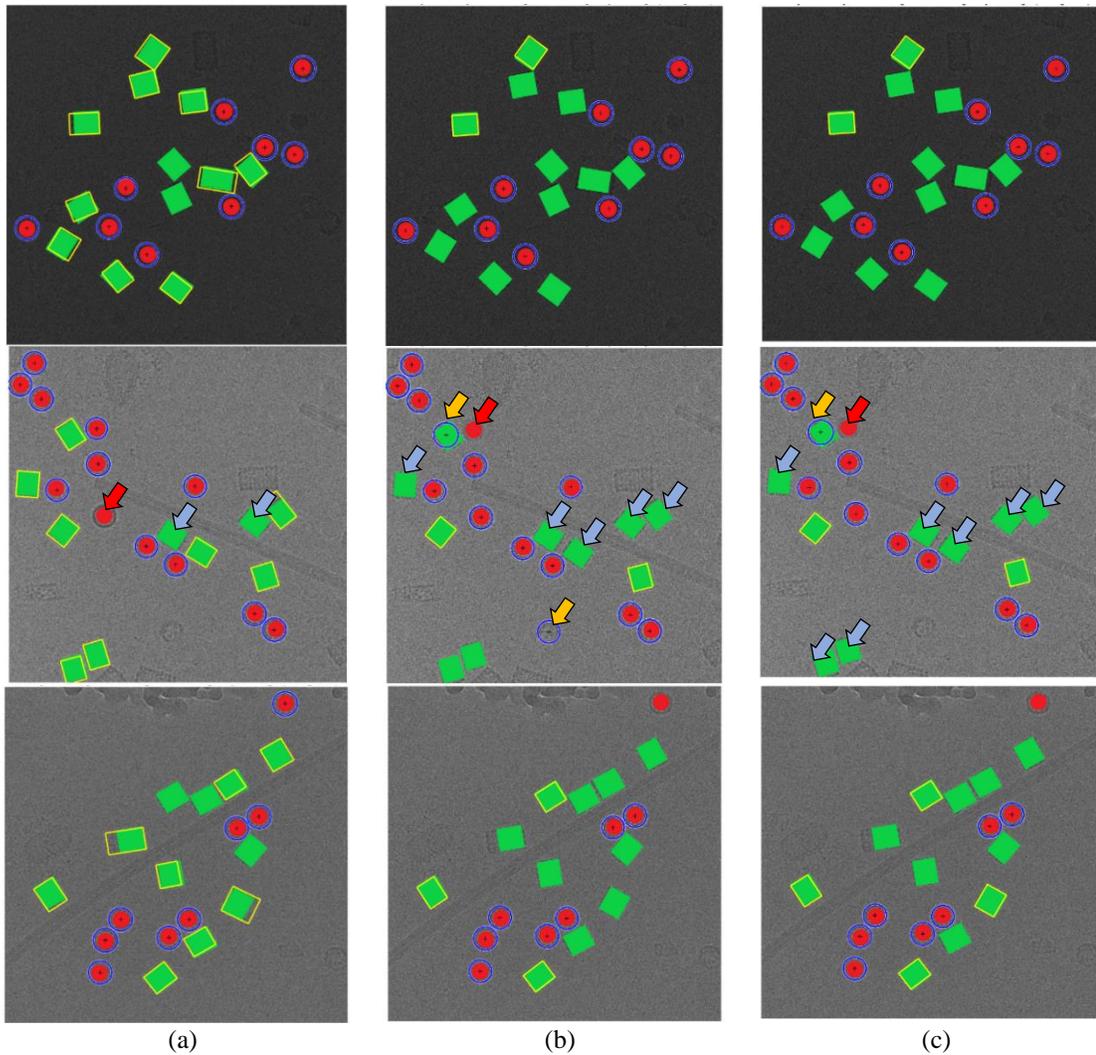


Figure 3.15: Automated particle picking results for both cases (top and side view) on KLH dataset. The original cryo-EM images form the KLH dataset. (a) Target detection and picking results (top and side particles view) using the ICB clustering algorithm. (b) Target detection and picking results (top and side particles view) using the k-means clustering algorithm. (c) Target detection and picking results (top and side particles view) using the FCM clustering algorithm.

3.4 Results and Discussion

We evaluate the performance of AutoCryoPicker in the three stages according to multiple metrics such as clustering accuracy, particle misclassification (or particles detection) rate, Dice, and time complexity.

3.4.1 Dataset

Images from two datasets (Apoferritin dataset and Keyhole Limpet Hemocyanin (KLH) dataset) are used to evaluate AutoCryoPicker. The particles in the two datasets are regular shapes, which are ideal for testing AutoCryoPicker because it is designed to detect and pick regular (e.g. circular) particle shapes. Two common shapes of protein particles in cryo-EM images are circles and rectangles. Apoferritin dataset [34] uses a multi-frame MRC image format (32 Bit Float). The size of each micrograph is 1240 by 1200 pixels. It consists of 20 micrographs each having 50 frames at 2 electrons/ $\text{\AA}^2/\text{frame}$, where the beam energy is 300 kV. The particle shape in this dataset is circular. The Keyhole Limpet Hemocyanin (KLH) dataset from US National Resource for Automated Molecular Microscopy [35] uses a single frame image format in a JPG file format. The size of each micrograph is 2048 by 2048 pixels. It consists of 82 micrographs at 2.2 electrons/ $\text{\AA}^2/\text{pixel}$, where the beam energy is 300 kV. There are two main types of projection views in this dataset: the top view (circular particle shape) and the side view (square particle shape). The KLH dataset [33] is a standard test dataset for particle picking. The KLH dataset is a challenging dataset because of different specimens (different particles) and confounding artifacts (ice contamination, degraded particles, particle aggregates, etc.).

3.4.2 Evaluation Metrics

In addition to the proposed clustering algorithm (ICB), we select two popular cluster algorithms (k-means and FCM). We compare them based on three factors. The first one is the running time. K-means and FCM based pairwise distance comparison is more time consuming. The second one is the effectiveness, which includes the clustering accuracy, misclassification rate, dice criteria, precision, recall, and the f1 measure. The third factor

is the clustering destabilization. Because K-means and FCM use random selection for cluster initialization, they may group the same points into different clusters in different runs. This requires an extra manual step to select the most appropriate cluster representing particles, which is not fully automated. In contrast, the ICB clustering algorithm is based on computing the interval size to determine the range of the intensity of cluster centers. Therefore, the particles that have the similar intensity values will be grouped into the same cluster.

For the particles clustering stage, we use clustering accuracy and misclassification rate which are defined by Equations (20) and (21), respectively. Each evaluation metric is calculated according to the numbers in a confusion matrix such as the True Positive (TP) which refers to the number of correct detections of positive cases, true Negative (TN) the number of correct detections of negative cases, False Positive (FP) the number of incorrect detections of positive cases and False Negative (FN) the number of incorrect detections of negative cases as Equations (3.20), (3.21) and (3.22) show respectively [89].

$$Accuracy = \frac{TP}{TP + TN} * 100 \quad (3.20)$$

$$Misclassification Rate = \frac{FP + FN}{Total} * 100 \quad (3.21)$$

Moreover, Dice Criteria (DIC) is also used for the similarity measure between a cluster image and the Ground Truth (GT). DC is defined by Equation (10) [90]:

$$Dice = \frac{2(A \cap B)}{A + B} * 100 \quad (3.22)$$

where, A is the cluster image and B is the ground truth image of A . Finally, we use the precision, recall, and F1 measure scores [89] to evaluate the particle picking results in the particle picking stage. The precision, recall, and F measure are defined by Equations

(3.23), (3.24) and (3.25), respectively [88]:

$$Precision = \frac{TP}{TP + FP} * 100 \quad (3.23)$$

$$Recall = \frac{TP}{TP + FN} * 100 \quad (3.24)$$

$$F1 \text{ measure} = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (3.25)$$

3.4.3 Particle Clustering, Detection and Picking Results

In order to evaluate the performance of automated particle clustering and picking, we generated a true reference by manually picking the particles on the images. Figure 3.11(a), and (k) show two different cryo-EM images from the two datasets (Apoferritin and KLH), respectively. The results on one image from the Apoferritin dataset are shown in Figure 3.10(d), (g), and (j) while the results for KLH dataset are shown in Figure 3.11(n), (q), and (t). It was demonstrated that most of the particles were correctly picked by AutoCryoPicker. Table 3.1 reports the recall, precision, accuracy, F1 score, and the running time of AutoCryoPicker based on three clustering algorithms: K-means, FCM, and ICB. On the Apoferritin dataset the AutoCryoPicker based on ICB clustering achieves a higher accuracy of 95.36% than 84.59% and 78.46% of k-means and FCM respectively. Also, ICB ran significantly faster in particles clustering (average time 1.71 seconds versus 10.29 seconds and 30.98 seconds of k-means and FCM, respectively).

Table 3.1: The results of AutoCryoPicker using the three clustering methods on the first dataset (Apoferritin). The table reports the average of the sensitivity or recall, specificity, precision, F1 score, accuracy, DICE score, and the particle clustering time (seconds).

Measures	ICB	k-means	FCM
Sensitivity/Recall (%)	98.11	87.90	83.60
Specificity (%)	97.76	87.97	85.85
Precision (%)	97.11	88.81	87.99

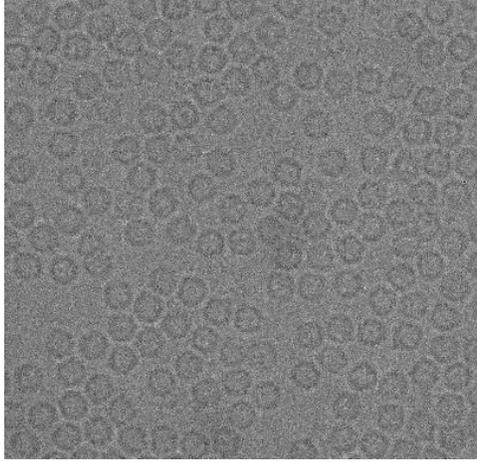
Misclassification Rate (%)	7.784	7.666	15.881
F1 Score (%)	97.61	84.59	83.10
Accuracy (%)	95.36	81.64	78.46
DICE Score (%)	97.76	87.97	85.85
Time consuming (sec.)	1.71	10.29	30.98
Clustering Selection Approach	Fully Automated	Manually	Manually

Table 3.2 shows the results on the KLH dataset. AutoCryoPicker based on ICB achieves a higher accuracy 91.82% than that of k-means and FCM (i.e. 87.50% and 80.83% respectively). The average clustering time of the whole dataset using ICB was 4.7 seconds on average, faster than the k-means by 23.8 seconds and 105.8 seconds of the FCM.

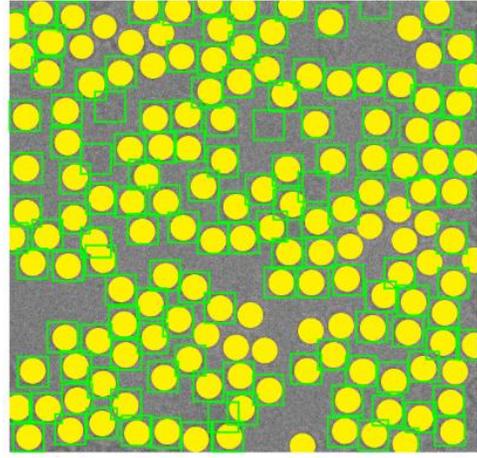
Table 3.2: The results of AutoCryoPicker using the three clustering methods on the second dataset (KLH). The table reports the average of the sensitivity or recall, specificity, precision, F1 score, accuracy, DICE score, and the particle clustering time consuming (seconds).

Measures	ICB	k-means	FCM
Sensitivity/Recall (%)	96.23	93.42	84.67
Specificity (%)	95.095	92.71	94.7925
Precision (%)	95.095	92.71	94.7925
Misclassification Rate (%)	3.77	6.58	15.33
F1 Score (%)	95.595	92.825	88.61
Accuracy (%)	91.8275	87.5025	80.835
DICE Score (%)	95.595	92.825	89.5
Time consuming (sec.)	4.714643	23.8332305	105.676302

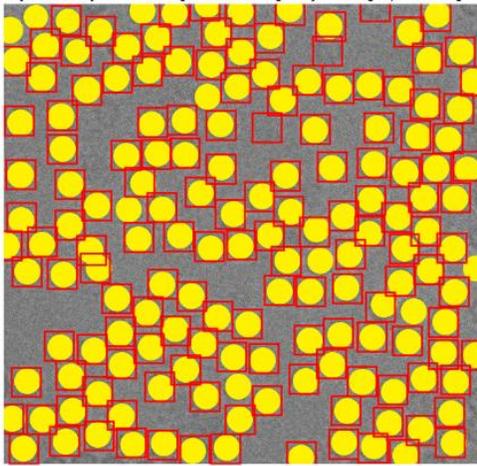
Two different cases from each of the two datasets are illustrated in Figure 3.16. Figure 3.16(a) shows cryo-EM images of a high particle density from the Apoferritin dataset with a low-frequency and Figure 3.16(b) a cryo-EM image of low SNR. Figure 3.16(c) and (d) shows two different micrograph cases from the KLH dataset that consist of excessively overlapped particles and some confounding artifacts such as ice contamination, degraded particles, and particle aggregates. AutoCryoPicker still performed very well on these cases. Figure 3.16(e)-(p) show the particle picking results using ICB, k-means, and FCM methods on the two datasets, respectively.



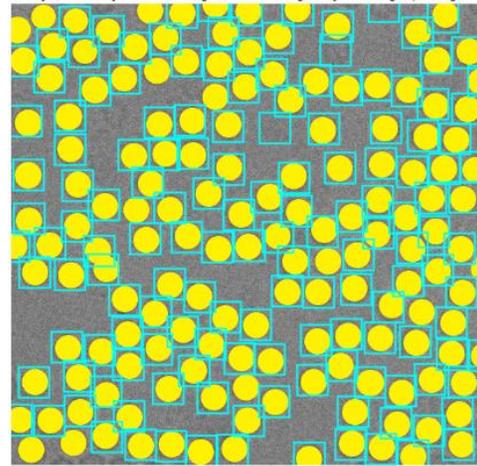
(a)



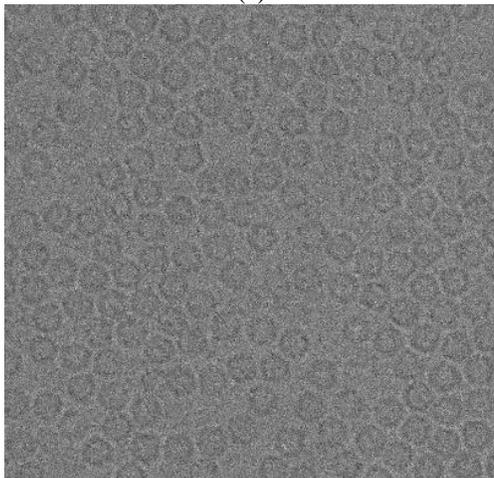
(b)



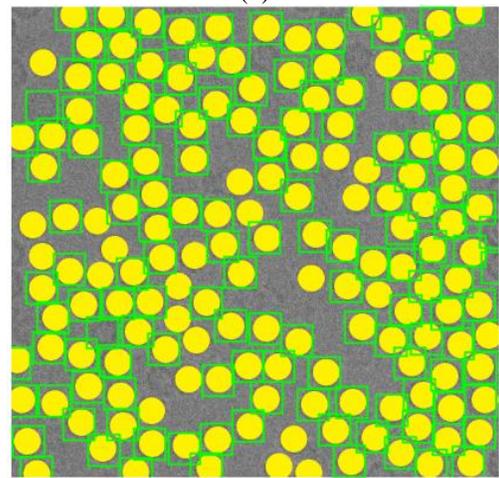
(c)



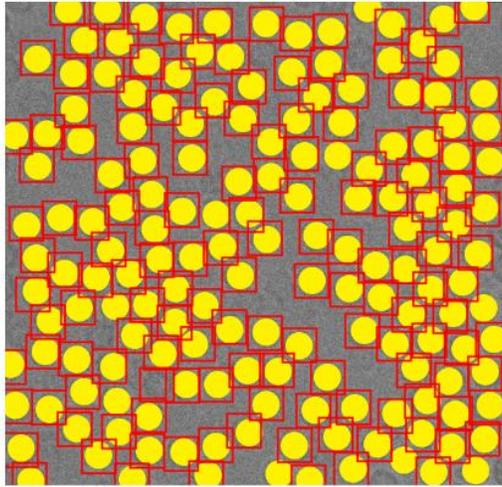
(d)



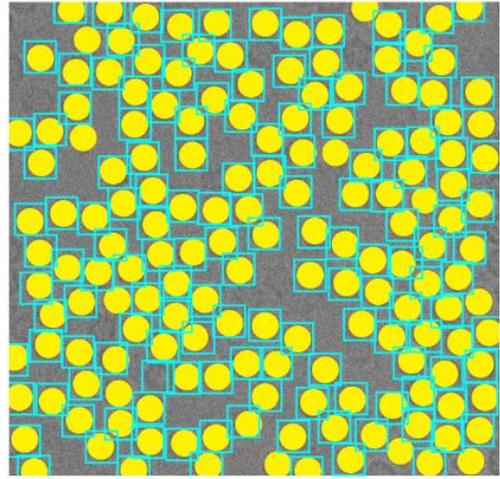
(e)



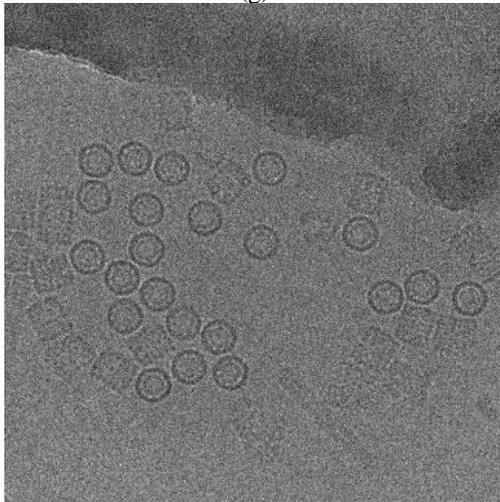
(f)



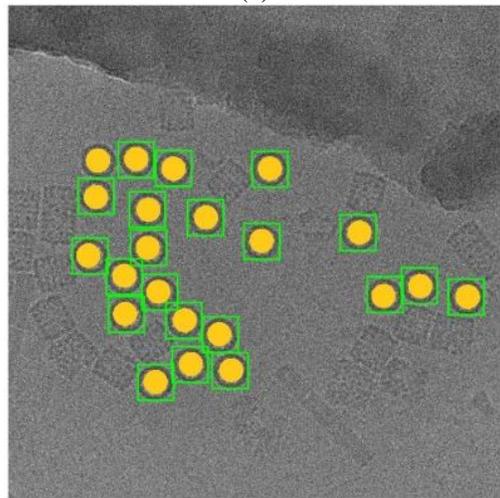
(g)



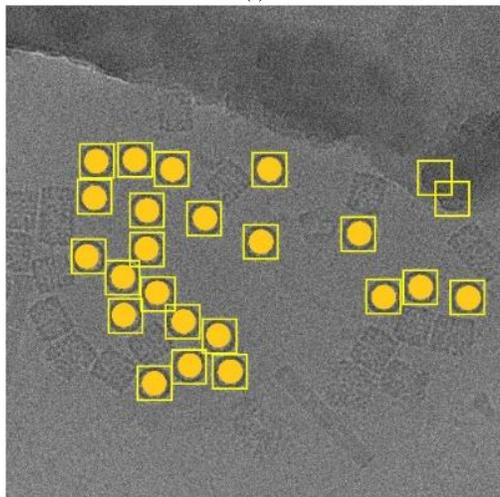
(h)



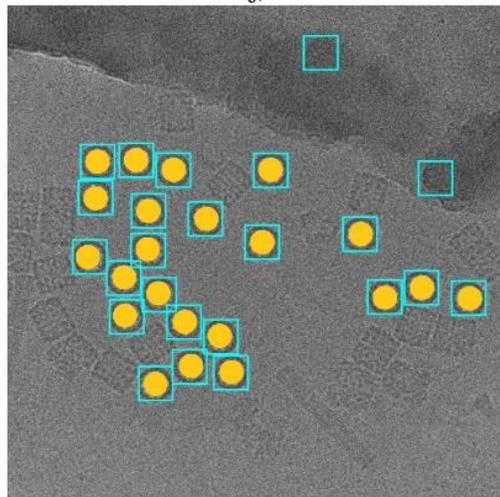
(i)



(j)



(k)



(l)

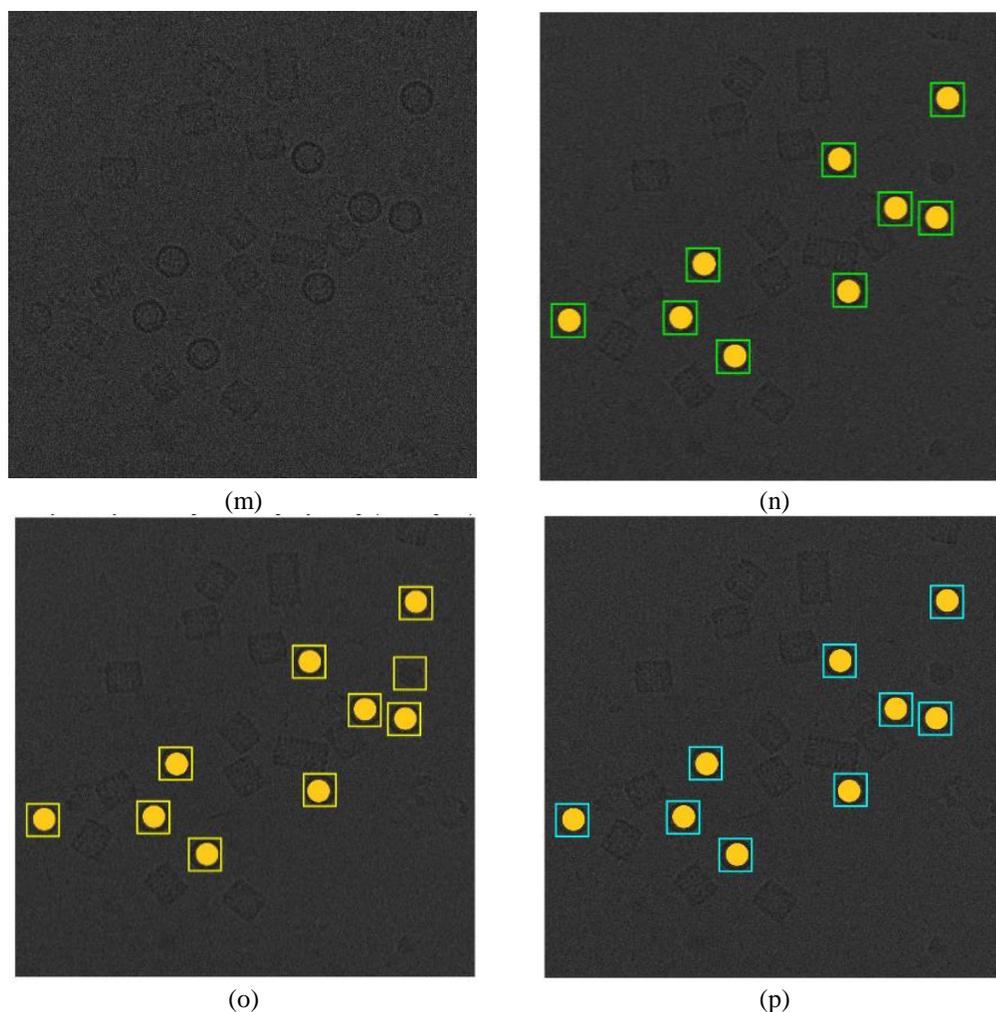


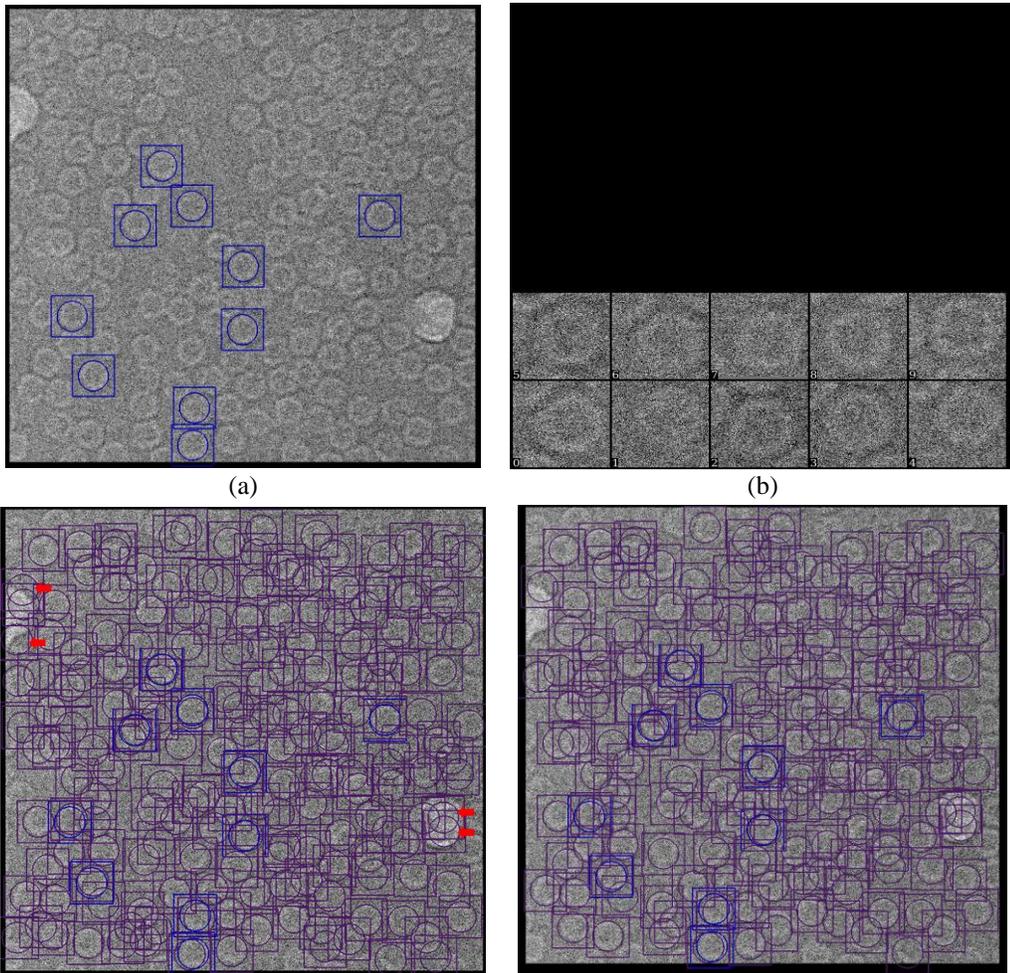
Figure 3.16: Automated particle picking results on the two datasets. (a) A cryo-EM image with a high identical particle density and a lack low-frequency from the Apoferritin dataset. (e) A low SNR cryo-EM image from the Apoferritin dataset. (i) A micrograph image from the KLH dataset that includes excessively overlapped particles due to confounding artifacts such as ice contamination, degraded particles, and particle aggregates. (m) A micrograph image from the KLH dataset that has a very low spatial density and different intensity levels. (b) and (f) Particle picking results using Intensity Based Clustering Algorithm (ICB) (Apoferritin dataset). (c) and (j) Particle picking results using k-means (Apoferritin dataset). (d) and (h) Particle picking results using FCM (Apoferritin dataset). (j) and (n) Particle picking results using Intensity Based Clustering Algorithm (ICB) (KLH dataset). (k) and (o) Particle picking results using k-means (KLH dataset). (l) and (p) Particle picking results using FCM (KLH dataset).

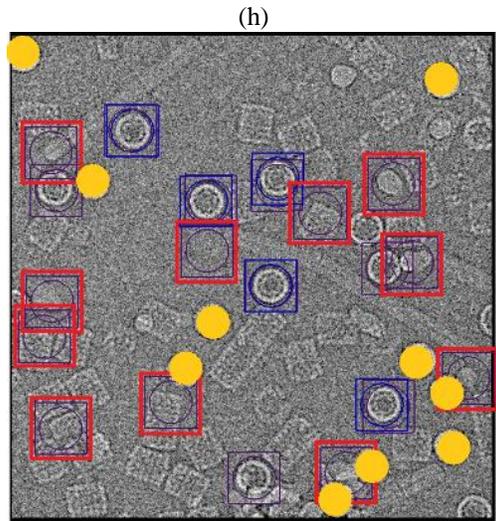
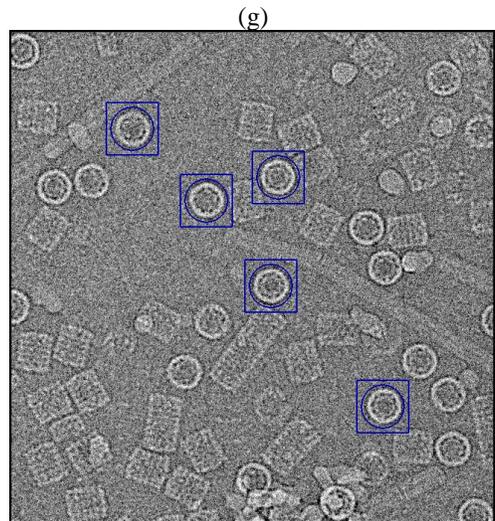
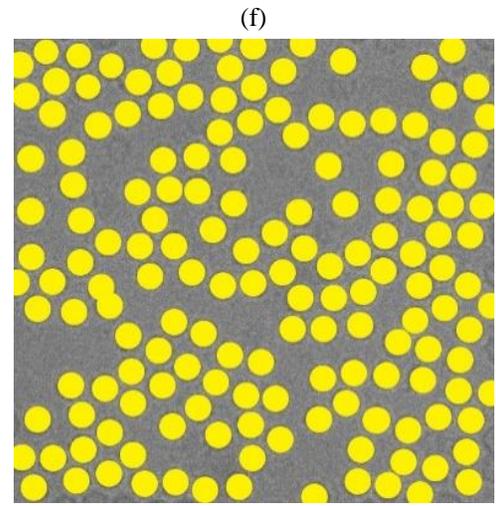
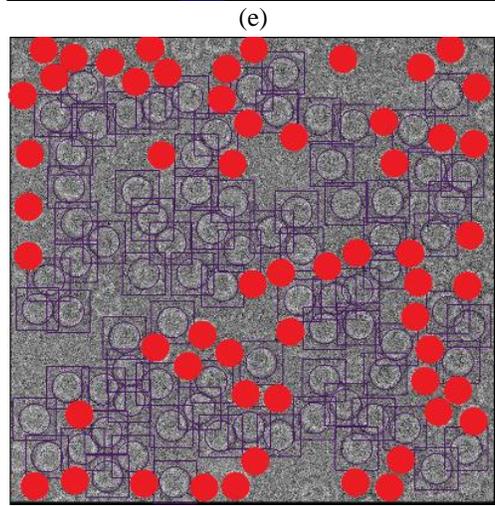
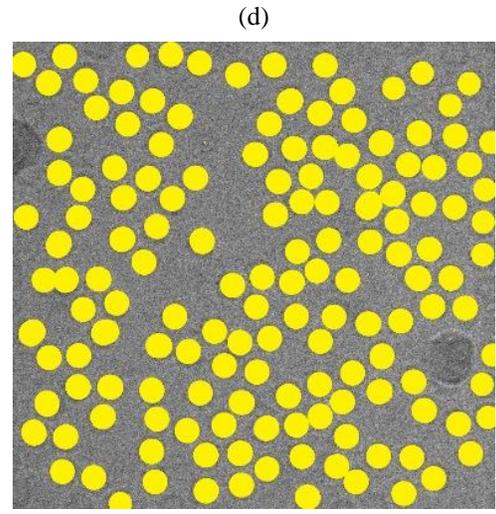
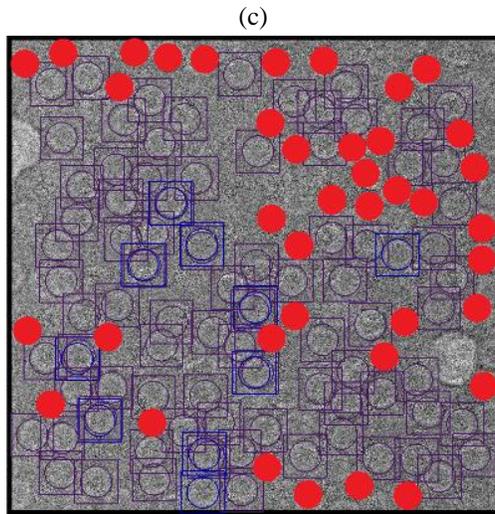
3.4.4 Comparison with Another Particle Picking Software

EMAN2 was selected as an example of particle picking software for cryo-EM images [76].

The “e2boxer.py” program of EMAN2 was applied to the same images input to AutoCryoPicker. For the Apoferritin images, a reference set of 10 particles was selected

manually (Figure 3.17(a), 3.17(b)) and then automated picking was performed with different threshold values (lower threshold results in more particles picked). For example, use of arbitrarily low threshold values of 0.0 and 0.5 results in most of the valid particles being selected; however, false positives likely corresponding to thick ice were also selected (Figures 3.17(c), 3.17(d)). Increasing the threshold to a more reasonable value of 2.3 resulted in no false positives at the expense of leaving several good particles unpicked (Figures 3.17(e), 3.17(g)). The lack of particle set completeness is evident by comparison to the ground truth result (Figures 3.17(f), 3.17(h)). In comparison, AutoCryoPicker successfully captured all the valid particles on the images without any false positives (Figures 3.17(m), 3.17(n)).





(i)

(i)

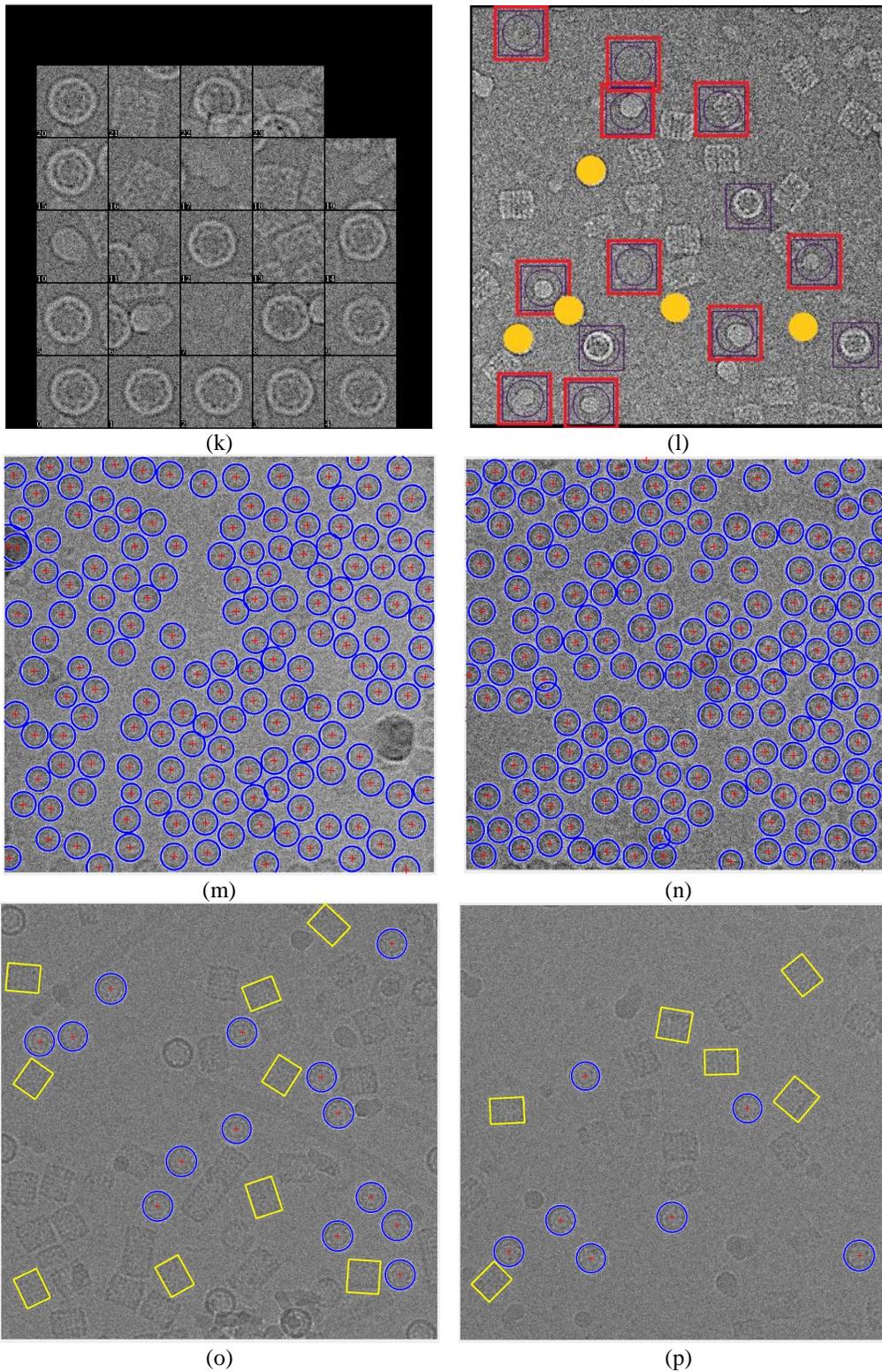


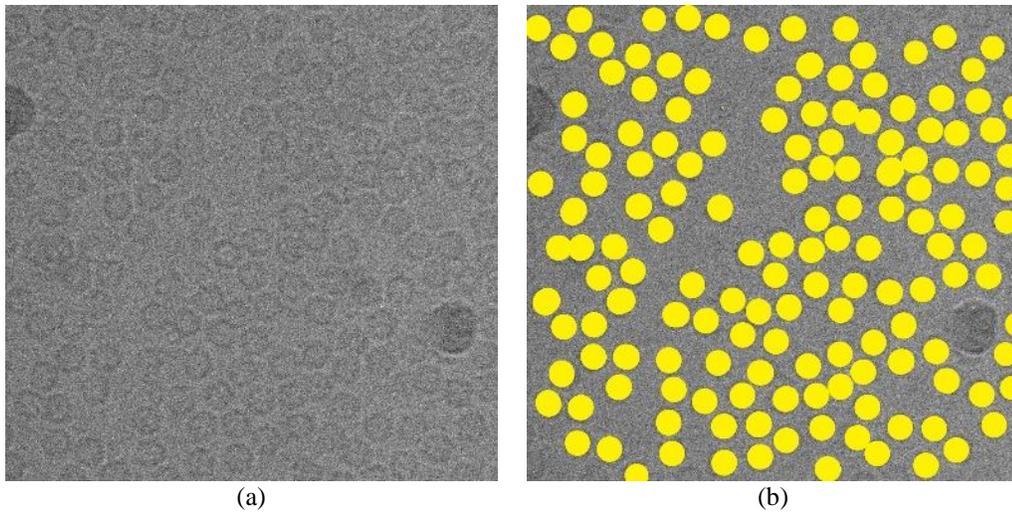
Figure 3.17: Particle picking using EMAN2 and AutoCryoPicker. (a) The manually selected reference particles of the Apoferritin dataset that were used for automated particle picking with EMAN2. (b) Zoomed-in view of the reference particles for the Apoferritin dataset. (c) EMAN2 automatic picking result based on

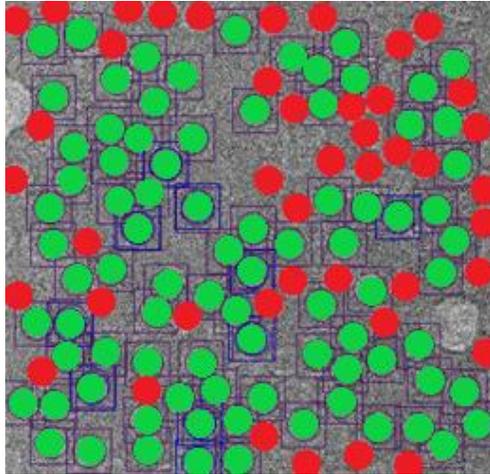
threshold value=0.0 using the first tested image of the Apoferritin dataset. (d) EMAN2 automatic picking result based on threshold value=0.5 using the first tested image of the Apoferritin dataset. (e) EMAN2 automatic picking result based the threshold value=2.3 using the first tested image of the Apoferritin dataset. Red dots mark missed particles). (f) Ground truth of first tested image of the Apoferritin dataset. Yellow dots mark valid particles. (g) EMAN2 automatic picking result based the threshold value=2.3 using the second tested image of the Apoferritin dataset. Red dots mark missed particles). (h) Ground truth of second tested image of the Apoferritin dataset. Yellow dots mark valid particles. (i) The manually selected reference particles of the KLH dataset that were used for automated picking of top-view (circular) particles with EMAN2. (j) EMAN2 automatic picking result based the threshold value=0.5 using the first tested image of the KLH dataset. Red squares mark the false positives and the yellow dots the missing particles. (k) Zoomed-in view of the automatically picked particles (threshold value=0.5) for first tested image of the KLH dataset. (l) EMAN2 automatic picking result based the threshold value=0.5 using the second tested image of the KLH dataset. Red squares mark the false positives, and the yellow dots mark the missing particles (top-view). (m) Particle picking result from AutoCryoPicker using the first tested image of the Apoferritin dataset. Red '+' mark the center of each particle and blue circles the top-view detected particles in the cryo-EM image. (n) Particle picking result from AutoCryoPicker using the second tested image of the Apoferritin dataset. Red '+' mark the center of each particle and blue circles the top-view detected particles in the cryo-EM image. (o) Particle picking result from AutoCryoPicker using the first tested image of the KLH dataset. Red '+' marks the center of each particle, blue circles the top-view detected particles in the cryo-EM image, and the yellow squares the side-view detected particles in the cryo-EM image. (p) Particle picking result from AutoCryoPicker using the second tested image from the KLH dataset. Red '+' marks the center of each particle, blue circles the top-view detected particles in the cryo-EM image, and the yellow squares the side-view detected particles in the cryo-EM image.

Similarly, results of using EMAN2 autopicking with the circular particles in the KLH images yielded incomplete recording of the valid particles and several false positives (Figures 3.17(j), 17(l)). In contrast, AutoCryoPicker was able to identify almost all of the true particles (both the circular and rectangular projections) in the KLH images, without the generation of false positives (Figures 3.17(o), 3.17(p)).

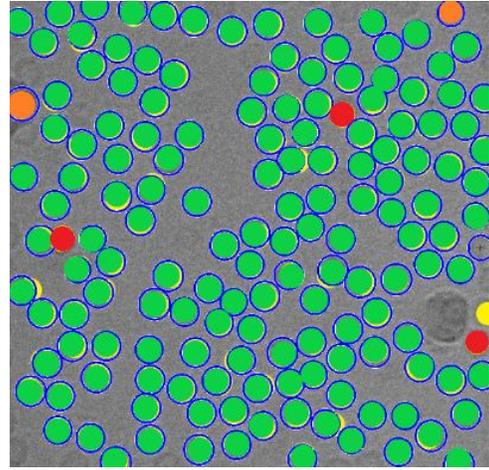
Quantitative assessment of the comparison is shown in Figure 3.18 and Tables 3.3-3.5. Figure 3.18(a) and (e) show two original images from the Apoferritin dataset where the images have top-view particle shapes only. Figure 3.18(b) and (f) show the manually particle picking labels (Ground Truth) where each particle is marked by a yellow circle on

top of each particle in the original images. Figure 3.18(c) and (g) show the particle picking performance results using EMAN2. In terms of evaluating each particle's picking tool in addition to the AutoCryoPicker, three criteria are selected to label and evaluate the particles picking performance results. True Positive (TP) picking where the correct particles are marked by the green circles. False Negative (FN) picking where the missed particles are marked by red circles. False Positive (FP) picking where the incorrectly picked particles are marked by orange circles. Figure 3.18 (d) and (h) show the same criteria of the particle picking results using AutoCryoPicker. Similarly, two images from the KLH dataset are shown in Figure 3.18(i) and (m). Figure 18(j) and (n) show the particles ground truth (hand picking and labeling). Figure 3.18(k) and (o) illustrate the performance results of the particle picking using EMAN2. Figure 3.18(l) and (p) show the same performance results using AutoCryoPicker.

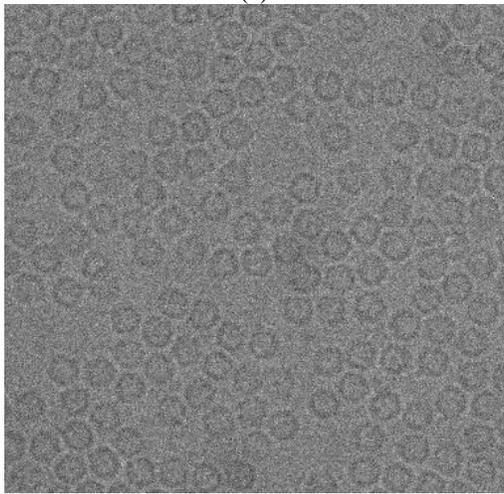




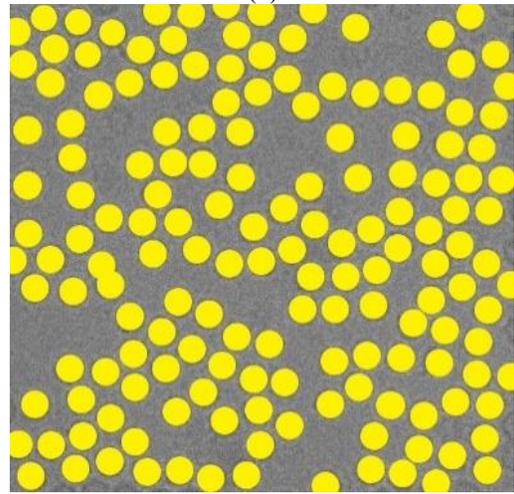
(c)



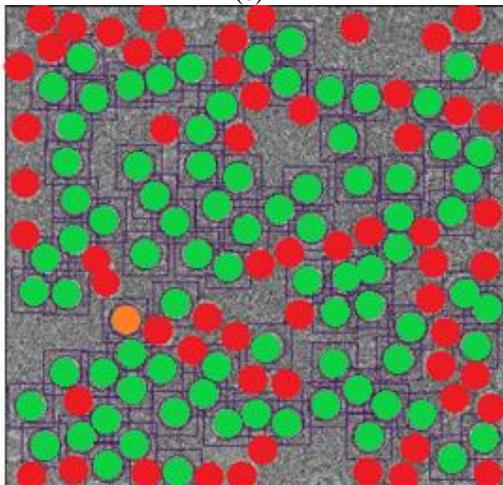
(d)



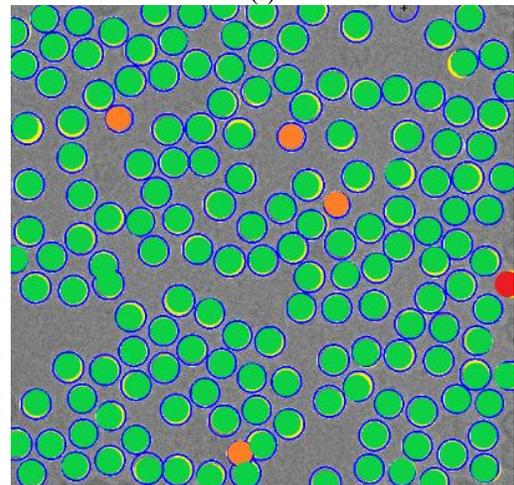
(e)



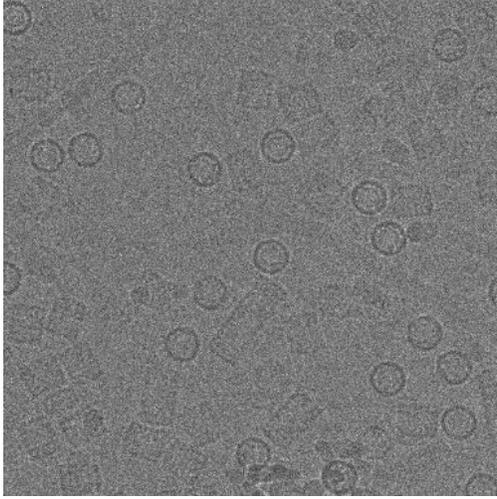
(f)



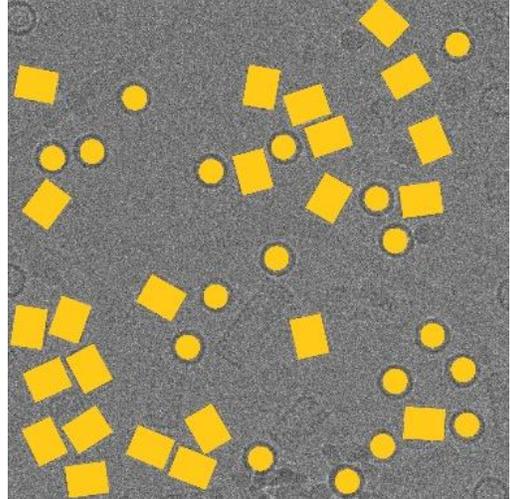
(g)



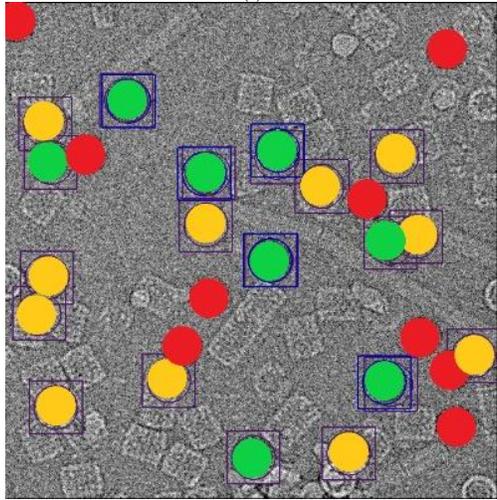
(h)



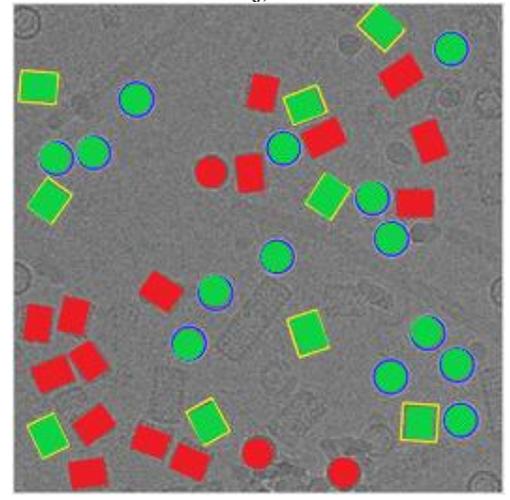
(i)



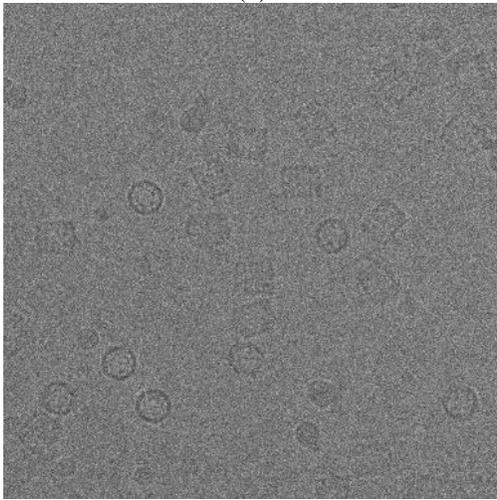
(j)



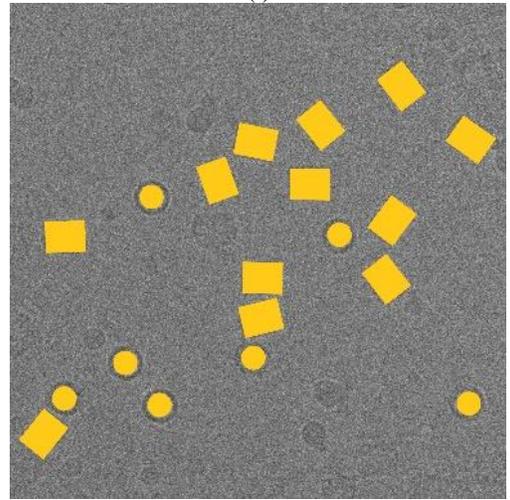
(k)



(l)



(m)



(n)

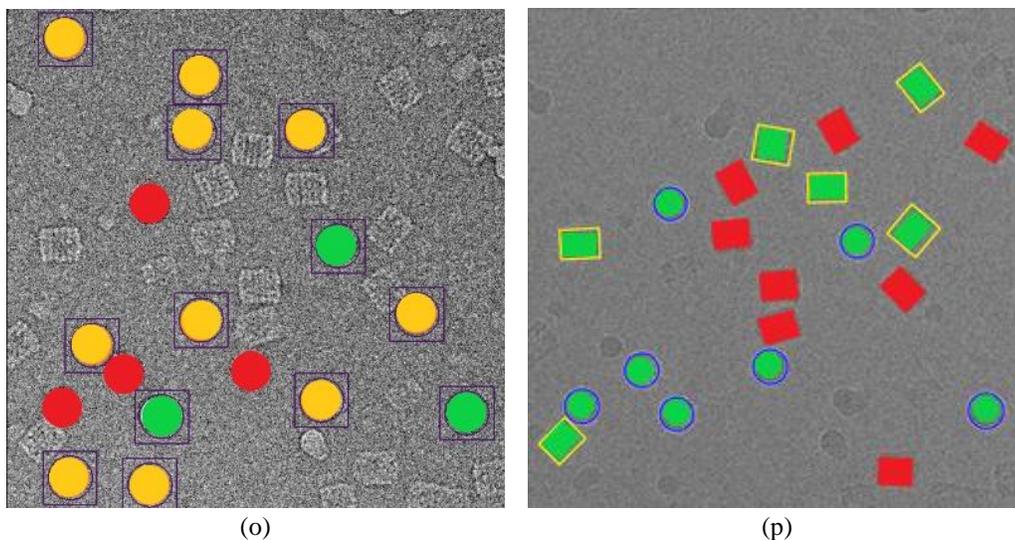


Figure 3.18: Evaluation of particle picking using EMAN2 and AutoCryoPicker. (a) Apoferritin cryo-EM image with top-view particle shapes only. (b) The ground truth (manually particle picking labels) of the first Apoferritin cryo-EM image where each particle is marked by a yellow circle on top of each particle. (c) The particle picking results of the first Apoferritin image using EMAN2. The particles are labeled as follows: Green, True Positive (TP); red, False Negative (FN). The particle picking results of the first Apoferritin cryo-EM image using AutoCryoPicker. The particles are labeled as follows: Green, True Positive (TP); red, False Negative (FN); orange, False Positive (FP). (d) The second original Apoferritin cryo-EM image with top-view particle shapes only. (e) The ground truth (manually particle picking labels) of the second Apoferritin cryo-EM image where each particle is marked by a yellow circle on top of each particle. (f) The particle picking results of the second Apoferritin cryo-EM image using EMAN2. The particles are labeled as follows: Green, True Positive (TP); red, False Negative (FN); orange, False Positive (FP). (g) The particle picking results of the second Apoferritin cryo-EM image using AutoCryoPicker. The particles are labeled as follows: Green, True Positive (TP); red, False Negative (FN); orange, False Positive (FP). (h) The first original KLH cryo-EM image. (i) The ground truth (manually particle picking labels) of the first KLH cryo-EM image where each particle is marked by a yellow circle on top of each particle. (j) The particle picking results of the first KLH image using EMAN2. The particles are labeled as follows: Green, True Positive (TP); red, False Negative (FN). (k) The particle picking results of the first KLH cryo-EM image using AutoCryoPicker. The particles are labeled as follows: Green, True Positive (TP); red, False Negative (FN). (l) The second original KLH cryo-EM image which has top-view particle shapes only. (m) The ground truth (manually particle picking labels) of the second KLH cryo-EM image where each particle is marked by a yellow circle on top of each particle. (n) The particle picking results of the second KLH cryo-EM image using EMAN2. (o) The particles are labeled as follows: Green, True Positive (TP); red, False Negative (FN). (p) The particle picking results of the second KLH cryo-EM image using AutoCryoPicker. The particles are labeled as follows: Green, True Positive (TP); red, False Negative (FN).

Table 3 illustrates the statistical evaluation of the performance results based on the TP, FN, FP for each single particle picking algorithm, as well as the particle shape class and total number of the particles (ground truth) in each image. Note that AutoCryoPicker performed better in detecting two different particle shapes on same images (Table 3.3).

Table 3.3: Statistical evaluation AutoCryoPicker and EMAN2 performance using the Apoferritin and KLH images. The table reports TP: True Positive picking results where the correct particles are picked, FN: False Negative picking results where some good particles are missed, FP: False Positive picking results where the incorrect particles (other objects such as background or artificial objects) are picked as particles.

Cryo-EM images	Particle Shape	Total Particles Number	AutoCryoPicker			EMAN2		
			TP	FN	FP	TP	FN	FP
Apoferritin 1	Top-View	151	148	3	2	84	67	0
Apoferritin 2	Top-View	160	159	1	5	83	76	1
KLH image 1	Top-View	17	14	3	0	8	9	11
KLH image 2	Top-View	7	7	0	0	3	4	10
KLH image 1	Side-View	24	8	15	0	N/A	N/A	N/A
KLH image 2	Side-View	14	6	8	0	N/A	N/A	N/A

Table 3.4 illustrates the evaluation of different single particle picking methods by reporting the average performance results using images from the Apoferritin dataset. AutoCryoPicker achieves a higher recall (98.70) and accuracy (96.55) compared to EMAN2 (53.92 and 53.76, respectively). Also, AutoCryoPicker achieved a higher f1 score (98.25) and dice score (98.24), as well as a low false negative rate (1.31).

Table 3. 4: Evaluation of particle picking on Apoferritin images.

Measures	AutoCryoPicker	EMAN2
Sensitivity/Recall (%)	98.70	53.92
Precision (%)	97.81	99.41
Misclassification Rate (%)	1.31	46.09
F1 Score (%)	98.25	69.90
Accuracy (%)	96.55	53.76
DICE Score (%)	98.24	69.90

Finally, Table 3.5 shows the performance results of different particle picking

methods using KLH images. The performance results in Table 5 have been calculated based on the circular particle detection only (top-view particles) since EMAN2 was challenged in detecting two different particle shape in the same image at the same time as shown in Table 3. In this case, AutoCryoPicker achieves higher recall (90.87), precision (98.48), F1 score (94.24), accuracy (89.36), dice score (94.24) and low miss classification rate (9.14).

Table 3. 5: Evaluation particle picking on the second KLH image.

Measures	AutoCryoPicker	EMAN2
Sensitivity/Recall (%)	90.87	59.44
Precision (%)	98.48	70.46
Misclassification Rate (%)	9.14	40.57
F1 Score (%)	94.24	59.96
Accuracy (%)	89.36	43.33
DICE Score (%)	94.24	37.22

3.5 Conclusion

Accurate particle picking in cryo-EM images still requires substantial human intervention and, therefore, can be labor-intensive and time-consuming. To address this challenge, we develop AutoCryoPicker – a fully automated particle picking approach based on image preprocessing, unsupervised clustering and shape detection. Our experiments show that the approach can significantly improve signal to noise ratio in cryo-EM images and pick particles rather accurately. Therefore, the automated method can relieve scientists from the laborious work of picking cryo-EM particles and help improve the efficiency and effectiveness of cryo-EM based protein structure determination. We conclude that AutoCryoPicker has the potential for being incorporated into the particle picking pipelines of other cryo-EM image processing software.

Chapter 4

SuperCryoEMPicker: Super Clustering Approach for Fully Automated Single Particle Picking in Cryo-EM

4.1 Introduction

Structure determining of complex proteins and macromolecular in the Cryo-EM (Cryo-electron microscopy) still a big challenge which requires substantial human intervention, labor-intensive and time-consuming. For the preparation stage, the researcher must indicate, detect, and select hundreds of thousands of good input single-particle examples for cryo-EM reconstruction. The performance of the existing tools still does not meet the requirements of the researcher in this field according to the variety of particles shapes and the quality of micrographs. Some cryo-EM images have very complex (irregular) protein shape and extremely low signal-to-noise ratio (SNR) which some existing automated particle-selection methods still required a large number of manually high-quality particles

to identify and detect them. To address this issue, we propose a fully automated single particle picking method (SuperCryoEMPicker) based on the idea of the super clustering using unsupervised learning.

We design a fully automated, unsupervised approach for single particle picking in cryo-EM micrographs that focus on identify, detect, and pick the complex and irregular protein shapes in the extremely low signal-to-noise micrographs. To adjust the low SNR micrographs, our model has two preprocessing stages. First, the original three-dimensional grid of voxels cryo-EM (MRC file format) is converted to another graphic file and the global intensity level is adjusted using a suite scientific cryo-EM image processing tools EMAN2. Second, the new micrographs format (PNG file) allows us to apply some advanced image processing tools to in case of improving the quality of the cryo-EM images. In this case, we design a fully automated, unsupervised approach for single particle picking in cryo-EM micrographs. Particle clustering is the second stage where the binary mask is generating from the original cryo-EM. Two different clustering approaches have been used. The first one is the regular clustering using k-means, fuzzy c-means (FCM), and the intensity-based clustering (ICB). The second approach is the super clustering approach. Super clustering approach is mainly based on generate an intermedia micrograph map using the simple linear iterative clustering (SLIC) and the original clustering algorithms. Experimental results show that the super particle clustering and picking from the intermedia micrograph map using super clustering SP-k-means, SP-FCM, and SP-ICB have a more robust detection and selection than those directly selected from the original noisy micrographs.

SuperCryoEMPicker can automatically and effectively identify and recognizes very

complex particle-like objects from an extremely low-SNR micrographs condition. As a fully automated particle detection and selection method, the proposed method, can help the researchers from the laborious work for manually particles identification election work, also without the need of labeled training data and human intervention, therefore is a useful tool for cryo-EM protein structure determination.

4.2 Background

For decades, X-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy have been the principal technologies of high-resolution structural biology, accounting for over 95% of the current holdings of the Protein Data Bank. However, in the past few years, major technological advances have fueled a “resolution revolution” in cryo-electron microscopy (cryo-EM) [91-93], and cryo-EM has emerged as a leading structural biology technology capable of determining protein structures to resolutions rivaling X-ray crystallography [94-99].

Identification of particles in micrographs (particle picking) is a critical step in structure determination by cryo-EM. The micrographs result from passing an electron beam through a thin vitrified sample to create 2D image projections of the particle under study [10]. Ultimately, the 3D shape (density map) of the protein is reconstructed from the 2D images. The 2D cryo-EM images contain randomly arranged particles along with non-particles—bits of frost, deformed particles, protein aggregates and so on. These images have high background noise and low contrast, due to a limited electron dose used in imaging. A large number of single-particle images need to be picked from cryo-EM micrographs to perform a reliable 3D reconstruction of the underlying protein structure. Particle picking thus represents an early bottleneck in the practice of cryo-EM structure

determination.

Particle picking methods can be basically divided into three categories, generative methods [101-102], discriminative classification [103-106] including the recent deep learning approaches [107] [18], and unsupervised learning (clustering) methods [109]. Typically, the generative method employs a template-matching technique which measures the similarity to a reference to identify particle candidates from micrographs. This technique requires initial high-quality particle templates manually selected by an expert. The discriminative classification technique requires preparing an initial set of manually labeled reference particles as the training dataset to train a classifier (e.g. a deep convolutional neural network) to detect particles. Therefore, generative and discriminative methods are not fully automated.

A typical generative method employs a template-matching technique with a cross-correlation similarity measure to accomplish particle selection. Template-based matching methods are very sensitive to noise and result a substitutional fraction of false positives since the template-based matching methods rely on the local cross-correlation in which result from the false correlation peak [110]. Thus, some initial “good references” are selected in advance to ensure that those manual selected examples have less noise comparing with the other in the same (2D) micrographs. The discriminative methods first train a classifier based on a labeled dataset of positive and negative examples, and then apply this trained classifier to detect and recognize particle images from micrographs. Also, some “good examples” are selected in advance in which avoiding the low-contrast particle examples from the micrographs. In most cases, the “bad particles” examples include the local aggregates, overlapped particles, background noise fluctuations, carbon-rich areas,

and ice contamination. Thus, after initialized the classifier, an additional step “manual versification and selection” is required to sort out the “good examples” and isolate them from the “bad ones” [111]. In contrast, the unsupervised approaches distinguish the images of particle-like objects from background noise in micrographs via an unsupervised learning manner (i.e., without any labeled training data). Therefore, the unsupervised approaches are often combined with the template-matching or classification-based approaches to achieve decent picking results [112].

To aid the streamlining of particle picking, we propose a super fully automated approach (SuperCryoEMPicker) for picking single particles of complex shape in cryo-EM images, leveraging the new super clustering technique. The method improves the base clustering algorithms (e.g. k-means) using the super pixel algorithm (simple linear iterative clustering-SLIC) [113]. Specifically, the super clustering algorithm applies a base clustering algorithm such as k-means [114], fuzzy c-Means (FCM) [115] and the intensity-based clustering (IBC) [116] to generate super pixels map firstly, which is then used for fully automated particle picking in cryo-EM images without human intervention. We demonstrate that our fully automated super clustering approach can accurately detect and select a sufficient number of complex particles that are comparable to those picked manually. Therefore, it can significantly reduce time and labor spent on particle picking and relieve a bottleneck in the cryo-EM structure determination pipeline.

4.3 Methods

The protein particle shapes in the most cryo-EM dataset are either common shape – circle (top view) or square (side view) which our last model AutoCryoPicker [116] has been designed to detect and pick them perfectly based on design a new clustering algorithm

(ICB: Intensity-Based Clustering) and using different sophisticated templated matching algorithms [117] [118]. In contrast, another common protein particle shape in very low SNR cryo-EM images is either complex or irregular shapes.

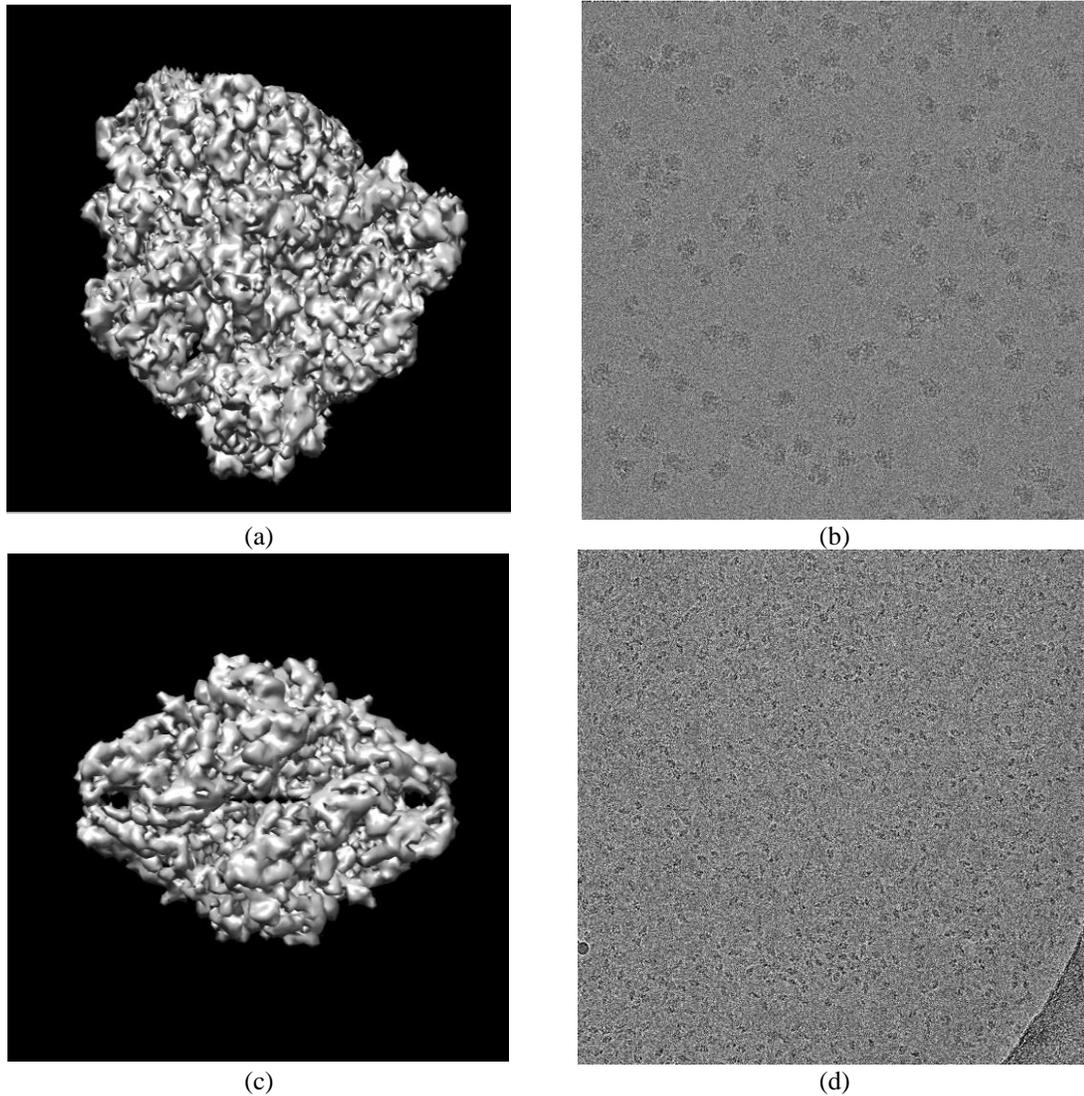


Figure 4.1: Example of particles of complex and irregular shapes. (a) Ribosome Electron Microscopy Density Map. (b) a cryo-EM image that has the Ribosome particles of irregular particle shapes [119]. (c) Beta-galactosidase Electron Microscopy Density Map. (d) a cryo-EM image that contains Beta-galactosidase particles of complex shape [119]. Both (a) and (c) are created based on using Chimera [122] and density maps from Protein Data Bank in Europe EMBL-EBI.

In this case, detect and pick the irregular or complex particle shapes in the very low SNR cryo-EM facing two main problems. First, particles in the cryo-EM appear in non-

structural object shapes which makes template matching algorithms unable to distinguish between the objects and the background as well as the particles in the very low SNR cryo-EM images have almost the same intensity level of the background. To address this problem, we proposed a super fully automated approach (SuperCryoEMPicker) for picking single particles of complex shape in cryo-EM images, leveraging the new super clustering technique. The super clustering approach is designed especially for picking particles of irregular and complex shapes as shown in Figure 3.1(a) and (b).

The framework of the super clustering approach is shown in Figure 3.2. It is divided into three main stages: preprocessing, particle clustering, and particle picking. In the first stage (preprocessing), the 3D grid (array) of voxels (MRC file) is converted to the PNG image file format using EMAN2 [121] in order to apply various image preprocessing techniques. Then, some advanced preprocessing steps are used to improve the quality of cryo-EM images. In the second stage the binary mask of the cryo-EM image is generated. Two kinds of clustering methods are implemented in this stage. The first kind is the base clustering methods including k-means [114], FCM) [115], intensity-based clustering (IBC) [116]. The second kind is the super clustering (SP) approach (superpixels based simple linear iterative clustering (SLIC) [113]), which is implemented to improve the three based clustering algorithms (k-means, FCM, and IBC), leading to three super clustering algorithms (SP-k-means, SP-FCM, and SP-IBC). In the third stage, based on the generated binary mask, a final set of particles are selected and picked from clustered particle candidates after some post-processing steps such as binary mask cleaning and particle property measurement.

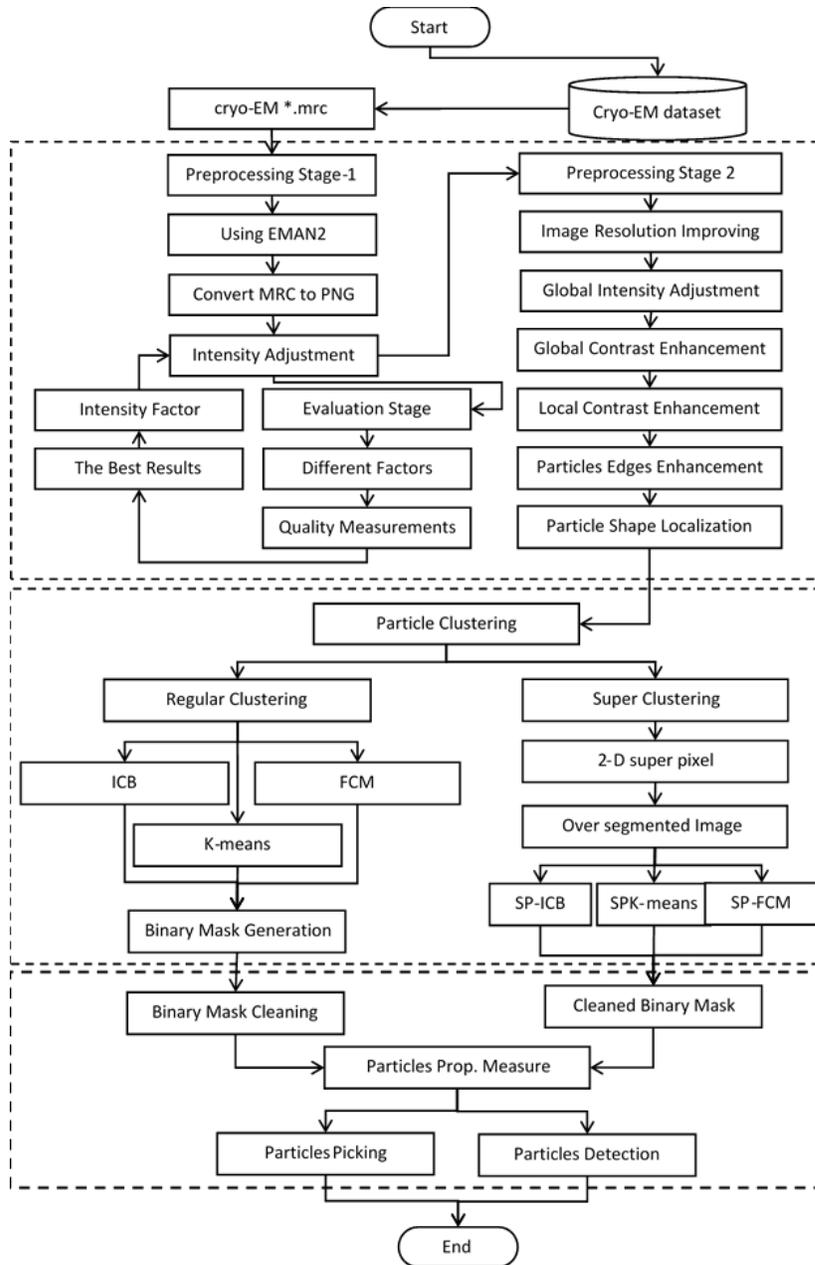
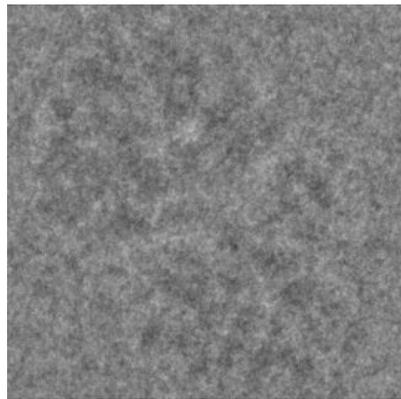


Figure 4.2: The general framework of the SuperCryoEMPicker. The dashed boxes represent three stages of the approach: pre-processing, super clustering, and particle picking. A solid box denotes an analysis step.

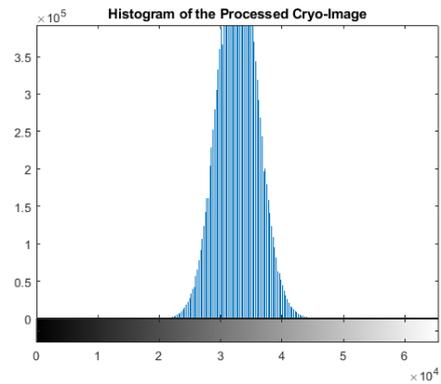
4.3.1 Stage 1: Pre-processing Stage

EM images are stored in the 3D grid in the voxels (MRC) file format. In this stage, the EMAN2 software [121] is used to adjust the global intensity of the cryo-EM and convert them from MRC file format to the PNG image format in order to apply standard imaging

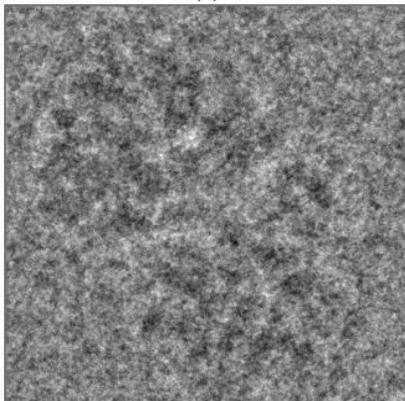
processing tools to them. Figure 3.3 shows some samples of using different scaling factors with EMAN2 [121] to adjust the intensity of the cryo-EM images. Figure 3.3(c) and (d) shows the same zoomed-in particle image after using scale factor 5 and its histogram respectively, which has better contrast than the original image Figure 3.3(a). However, the quality of images in Figure 3.3(j) and (l) adjusted with scaling factors 0.1 and 0.5, is lower than the original one.



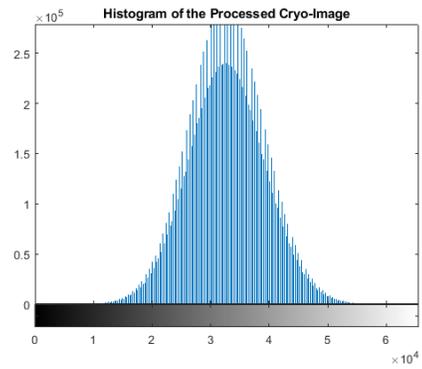
(a)



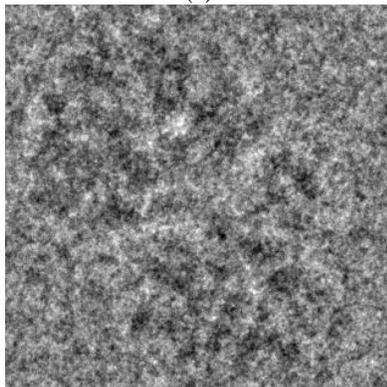
(b)



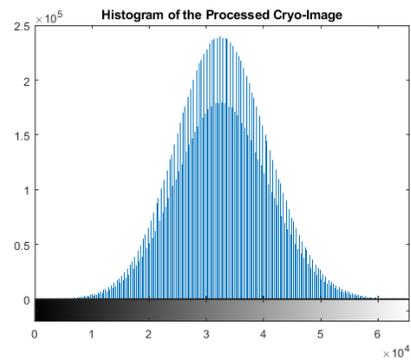
(c)



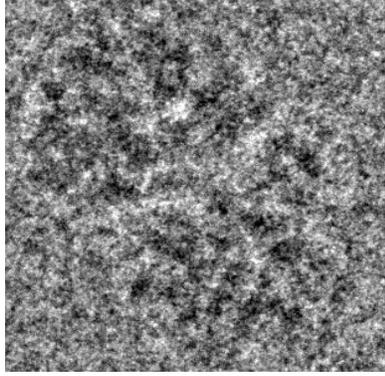
(d)



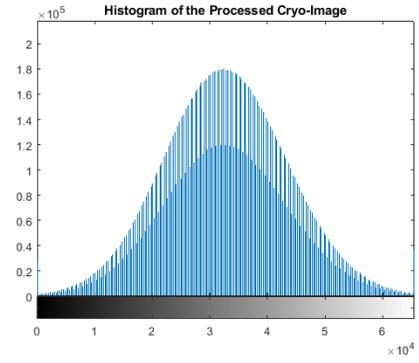
(e)



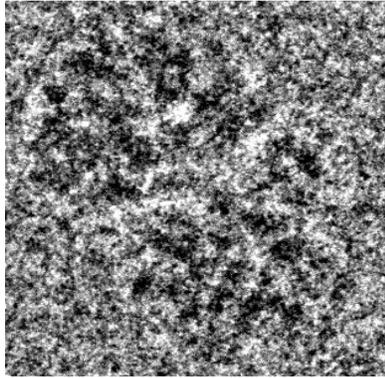
(f)



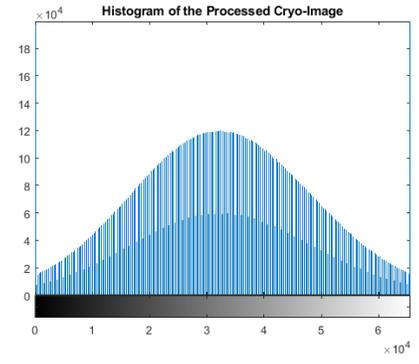
(g)



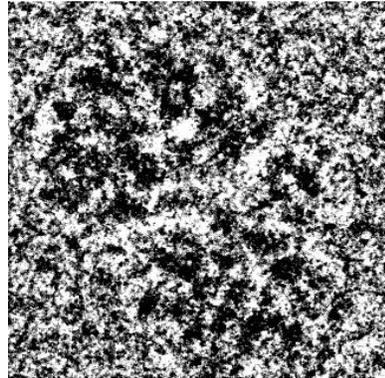
(h)



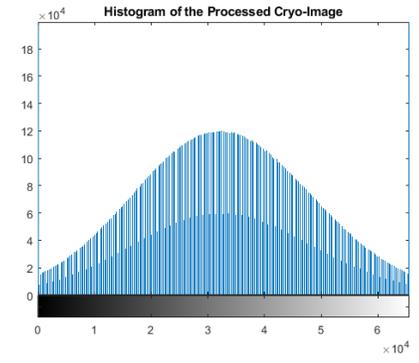
(i)



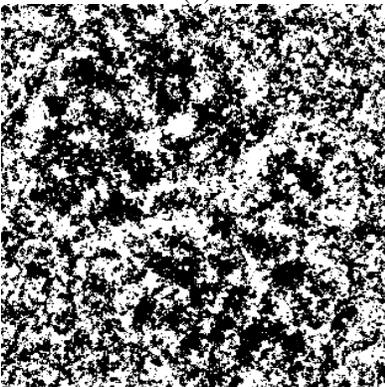
(j)



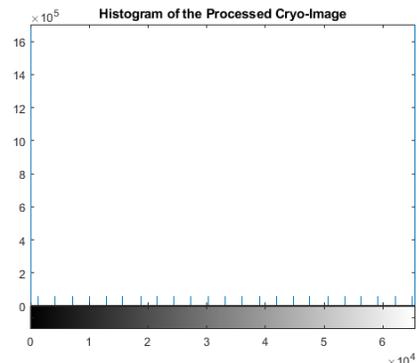
(k)



(l)



(m)



(n)

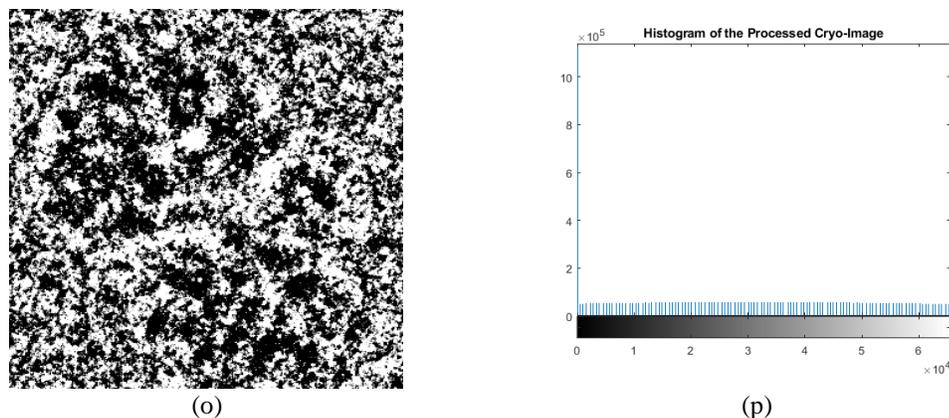
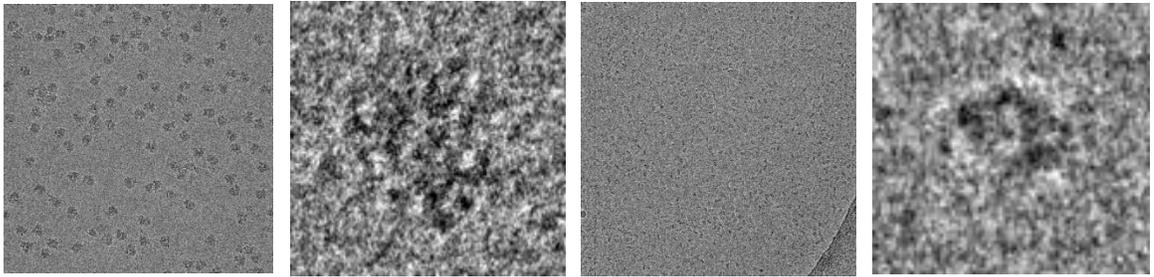


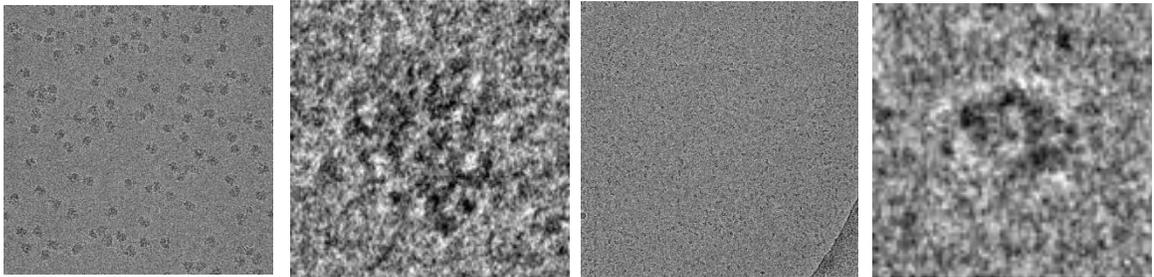
Figure 4.3: One zoom-in particle image from the ribosome dataset during the first stage of the pre-processing “intensity adjustment” using different scaling factors in the EMAN2. (a) the original zoom-in particle (manually selected and cropped from the original cryo-EM). (b) the original histogram of the cryo-EM. (c) particle image after the intensity adjustment using scale factor 5. (d) the histogram of the pre-processed image adjusted in (c). (e) particle image after the intensity adjustment with scale factor 4. (f) the histogram of the pre-processed image in (e). (g) particle image after the intensity adjustment with scale factor 3. (h) the histogram of the pre-processed image in (g). (i) particle after the intensity adjustment (scale factor 1). (j) the histogram of the pre-processed image in (i). (k) particle image after the intensity adjustment with scale factor 0.1. (l) the histogram of the pre-processed image in (k). (m) particle image after the intensity adjustment with scale factor 0.25. (n) the histogram of the pre-processed image in (m). (o) particle image after the intensity adjustment with scale factor 0.5. (p) the histogram of the pre-processed image in (o).

In the second step, different image preprocessing procedures (image resolution, global intensity adjustment, global contrast enhancement-based histogram equalization, noise suppressing using Wiener filter, local particle contrast enhancement with the adaptive histogram equalization, edges enhancement using guided image filtering) are applied to improve the quality of the cryo-EM images as in AutoCryoPicker [116]. The results of the preprocessing procedures for Ribosome [119] and Beta-galactosidase [120] images are shown in Figure 4.3.



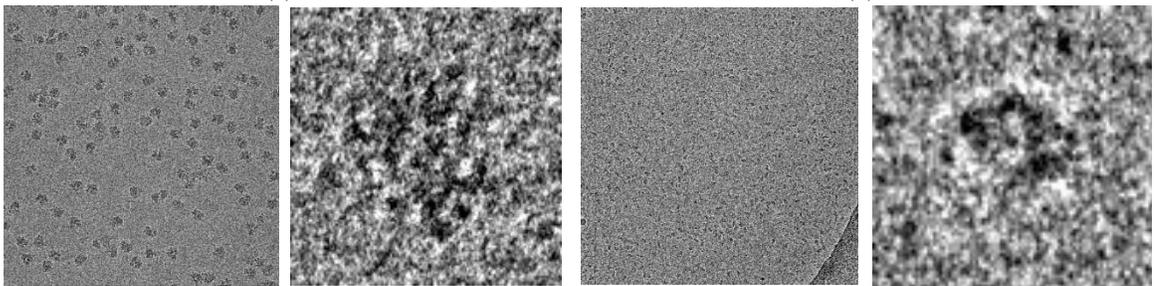
(a)

(b)



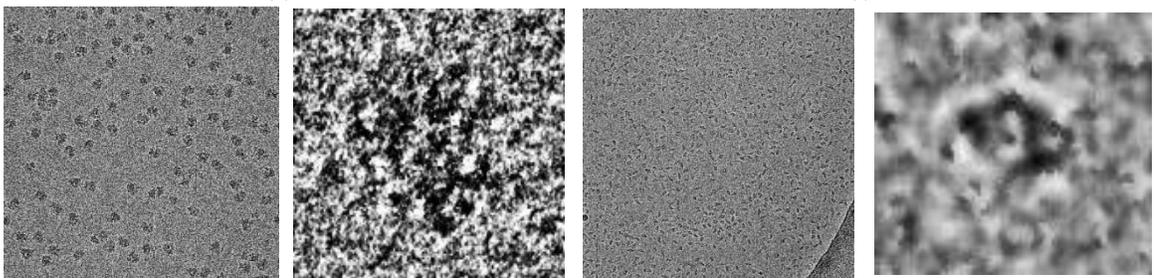
(c)

(d)



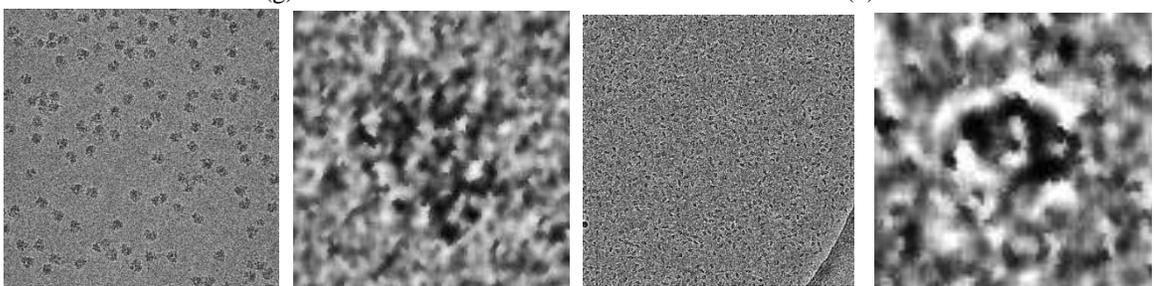
(e)

(f)



(g)

(h)



(i)

(j)

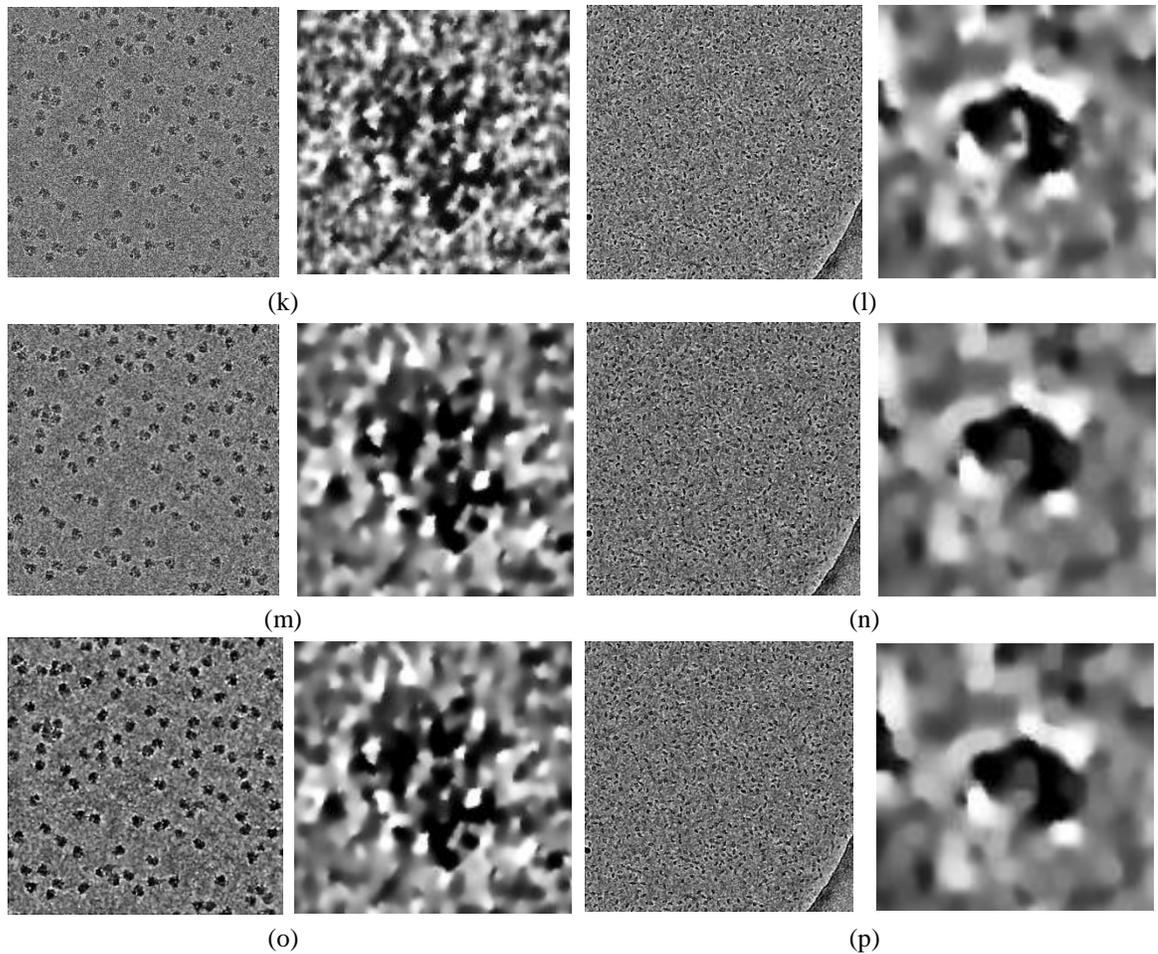


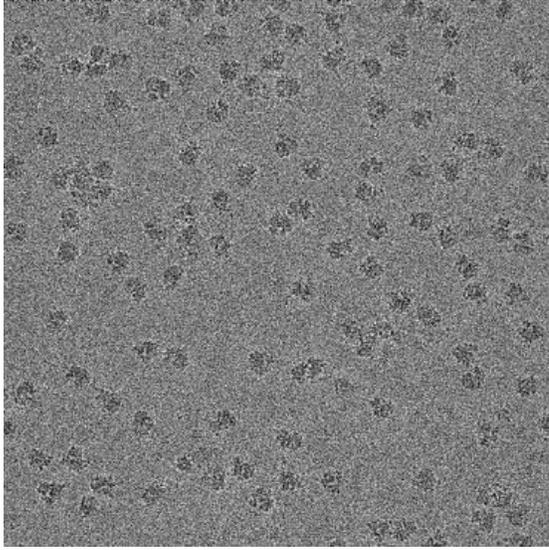
Figure 4.4: Illustration of effects of the preprocessing procedures on Ribosome and Beta-galactosidase images. (a) the original particle image of Ribosome (one full image and one zoom-in particle). (b) the original image of Beta-galactosidase. (c) the image of Ribosome after the image resolution improvement. (d) the image of Beta-galactosidase after image resolution improvement. (e) the image of Ribosome after global intensity adjustment. (f) the image of Beta-galactosidase after global intensity adjustment. (g) the image of Ribosome after global contrast enhancement-based histogram equalization. (h) the image of Beta-galactosidase after the global contrast enhancement-based histogram equalization. (i) the image of Ribosome after the noise suppressing using Wiener filter. (j) the image of Beta-galactosidase after the noise suppressing using Wiener filter. (k) the image of Ribosome after the local particle contrast enhancement with the adaptive histogram equalization. (l) the image of Beta-galactosidase after the local particle contrast enhancement using adaptive histogram equalization. (m) the image of Ribosome after edges enhancement using guided image filtering. (n) the image of Beta-galactosidase after edges enhancement using guided image filtering. (o) the image of Ribosome after the particle shape localization using morphological image operation. (p) the image of Beta-galactosidase after the particle shape localization using morphological image operation.

4.3.2 Stage 2: Particles Clustering

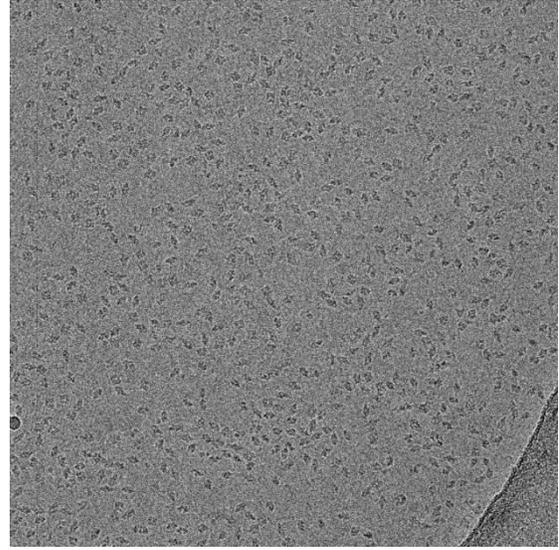
In order to pick each possible particle in the cryo-EM image, a binary mask that clusters particles are needed. Both the base clustering algorithms (k-means [114], FCM [115], IBC [116]) as well the super clustering algorithm built on top of the base algorithms (SP-K-means, SP-FCM, and SP-IBC) via pixel posterization using simple linear iterative clustering (SLIC) [113] to create the binary mask according to the following steps.

Clustering with base Clustering Algorithms

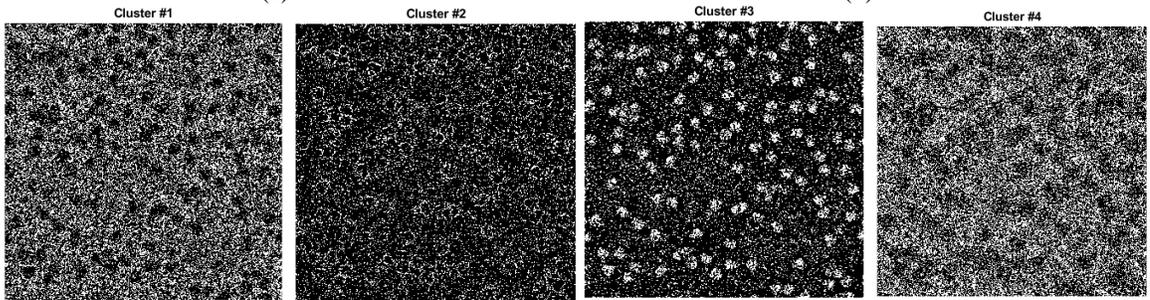
We apply three based clustering algorithms including k-means, FCM and IBC to cluster particles. The number of the cluster has been chosen based on the predefine clusters number that our ICB clustering algorithm has defined [116]. The initial number of clusters in the ICB algorithm is based on two factors, the adjusted intensity range and the interval size. The adjusted intensity range is automatically computed from the preprocessed cryo-EM image based on the lower and the upper bound of the intensity level while interval size is computed based on the ratio between the between the deference of the maximum and minimum intensity level and the adjusted intensity range. For instance, the initial number of clusters is ($K = 4$), if the adjusted intensity range is from 0.2 to 0.8 and the interval size is 0.15, there are 4 initial cluster levels: the intensity level [0.2- 315 0.35] will be assigned to Cluster 1, [0.35-0.5] to Cluster 2, [0.5-0.65] to Cluster 3, and 316 [0.65-0.8] to Cluster 4. Figure 4.5 illustrates the results of the three base clustering algorithms. More details about applying the three algorithms to particle clustering can be found in [116].



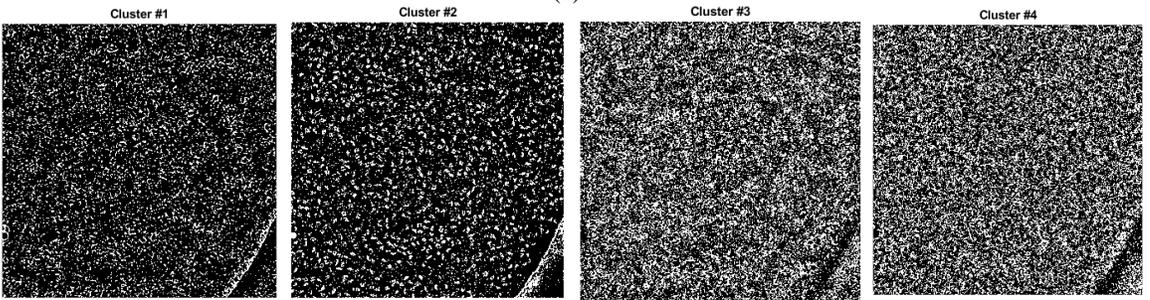
(a)



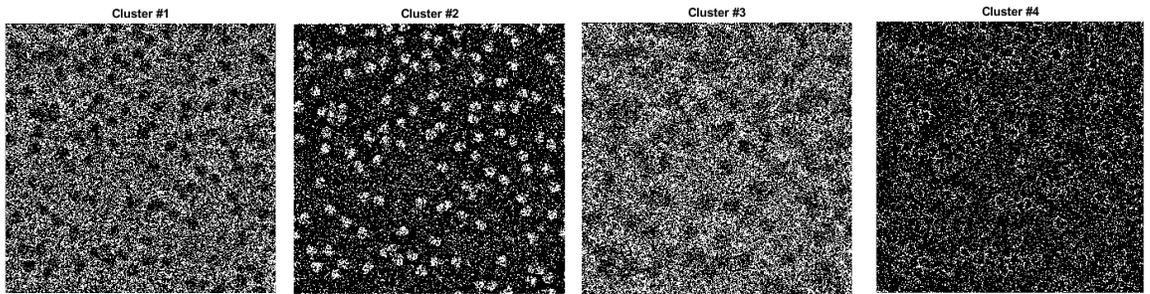
(b)



(c)



(d)



(e)

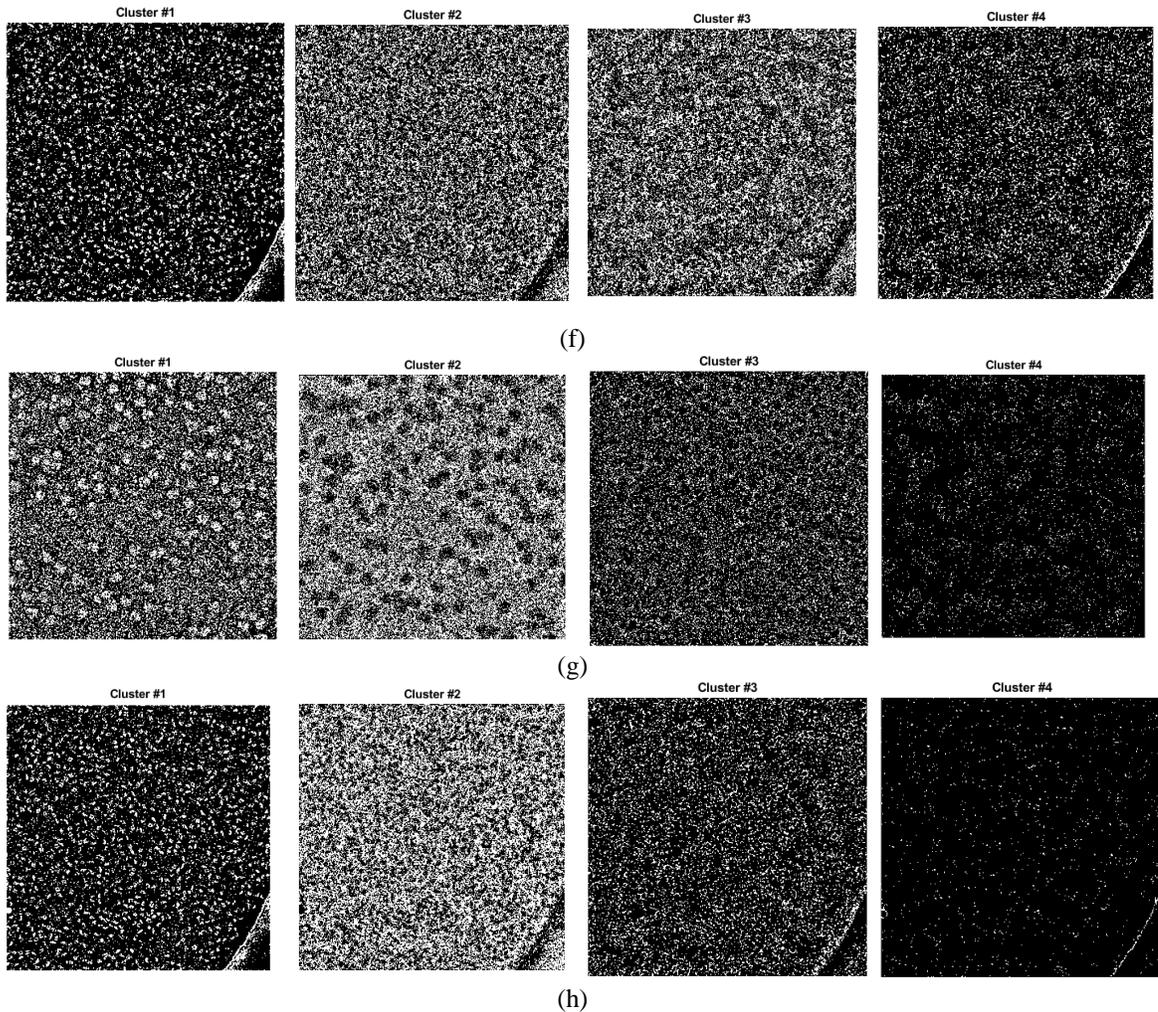


Figure 4.5: Image clustering results using k-means, FCM, and IBC. (a) an original cryo-EM image of Ribosome. (b) an original cryo-EM image of Beta-galactosidase. (c) K-means clustering results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) for Ribosome image. Most real particles were assigned to Cluster #3. (d) K-means clustering results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) for the Beta-galactosidase image. Most real particles were assigned to Cluster #2. (e) FCM clustering results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) for the Ribosome image. Most real particles were assigned to Cluster #2. (f) FCM clustering results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) for the Beta-galactosidase image. Most real particles were assigned to Cluster #1. (g) IBC clustering results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) for the Ribosome image. Most real particles were assigned to Cluster #1. (h) IBC clustering results (Cluster #1, Cluster #2, Cluster #3 and Cluster #4) for the Beta-galactosidase image. Most real particles were assigned to Cluster #1.

Super Particle Clustering

We designed a super clustering approach to further improve cryo-EM binary mask image

generation based on pixel posterization using the simple linear iterative clustering (SLIC) [113]. In this approach, an intermedia image map (super pixel over segmentation image) is generated and used as the input for the three base clustering algorithms (k-means, FCM, and IBC) to perform clustering, leading to three super clustering methods: SP-K-means, DP-FCM, and SP-IBC for fully automated single particle picking in cryo-EM.

The SLIC super pixel method [113] combines the two kinds of distances between pixels i and j (Equation (4.1) for intensity distance and Equation (4.2) for spatial distance) into a single one in Equation (4.3) [113]:

$$d_c = \sqrt{(l_j - l_i)^2} \quad (4.1)$$

$$d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \quad (4.2)$$

where l is the intensity level and x and y are the spatial pixel information.

$$D' = \sqrt{\left(\frac{d_c}{N_c}\right)^2 + \left(\frac{d_s}{N_s}\right)^2} \quad (4.3)$$

where N_c and N_s are the maximum distance within a cluster. Basically, SLIC normalizes the distances of the intensity value and the spatial information by their respective maximum values. The maximum spatial distance is calculated based on the expected spatial distance within a given cluster that corresponding to the sampling grid interval (S). To produce the roughly equaled sized super pixels, the interval grid S is calculated as is shown in Equation (4.4) [113]:

$$N_s = S = \sqrt{\frac{N}{k}} \quad (4.4)$$

Where k is the desired number of the approximated equally sized super pixels, and N is the lowest gradient position (3×3) neighborhood. In this case, to avoid the centering a

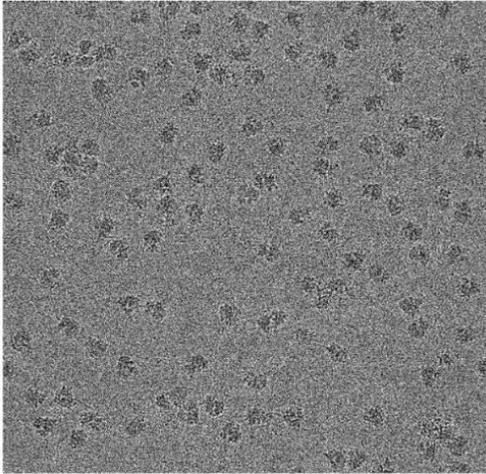
super pixel in edges the centers of the grid interval (S) are moving to the seed location corresponding to the lowest gradient position (3×3) neighborhood. Although, to reduce the distance computations, SLIC only computes the distance from each pixel to each cluster center within $2S \times 2S$ region. Since the intensity level can be significantly vary from image to image and from cluster to cluster, the calculation of the maximum intensity distance N_c is straightforward. N_c is fixed as a constant m so that weighted distance measure is calculated as Equation (4.5) [113].

$$D' = \sqrt{\left(\frac{d_c}{m}\right)^2 + \left(\frac{d_s}{S}\right)^2} \quad (4.5)$$

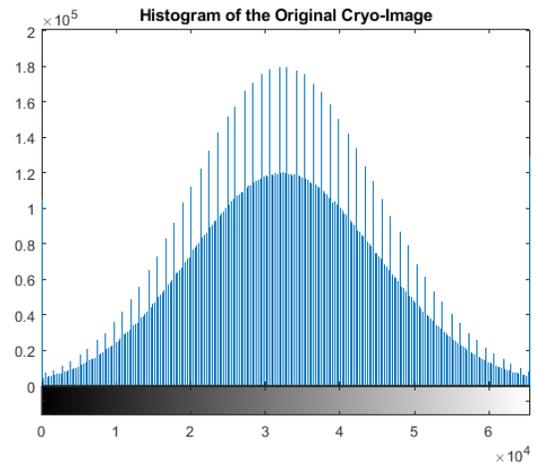
Where m is a fixed constant number, and S is the grid interval. Multiplying Equation (5.5) by m^2 leads to a simplified distance measure in Equation (4.6).

$$D' = \sqrt{d_c^2 + \left(\frac{d_s}{S}\right)^2 m^2} \quad (4.6)$$

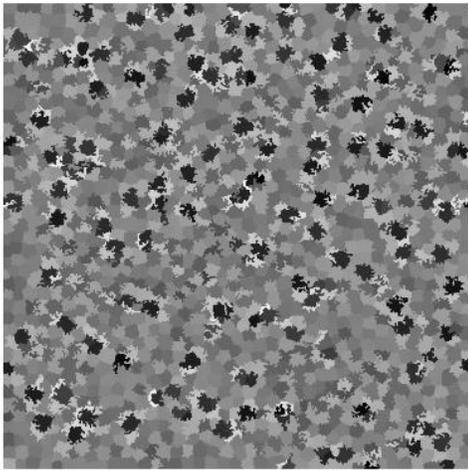
Also, m allows us to weight the relative importance of the spatial information and the intensity similarity. When m is large, there will be a small area-to-perimeter ratio (more impact), otherwise, an area close to the image boundary that is less regular. Figure 4.6 shows the examples of different intermedia cryo-EM image maps of super pixel clustering for Ribosome and Beta-galactosidase images. We compare the intermedia cryo-EM maps generated from the original cryo-EM images without preprocessing and with the ones from the pre-processed images.



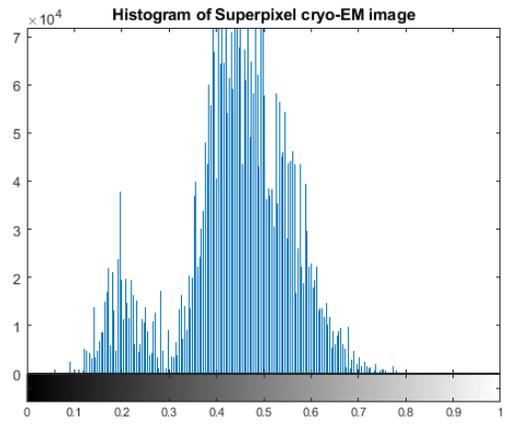
(a)



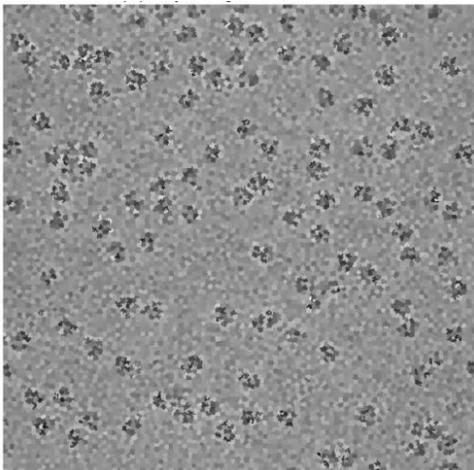
(b)



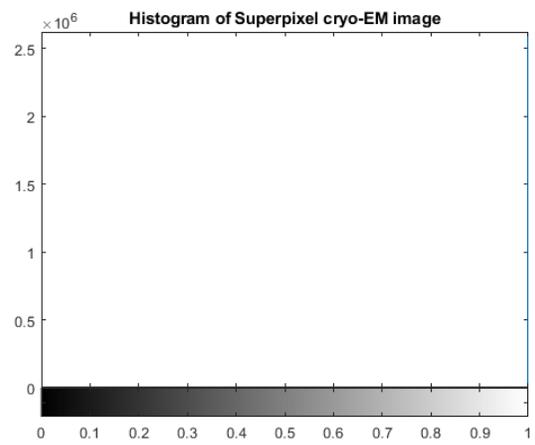
(c)



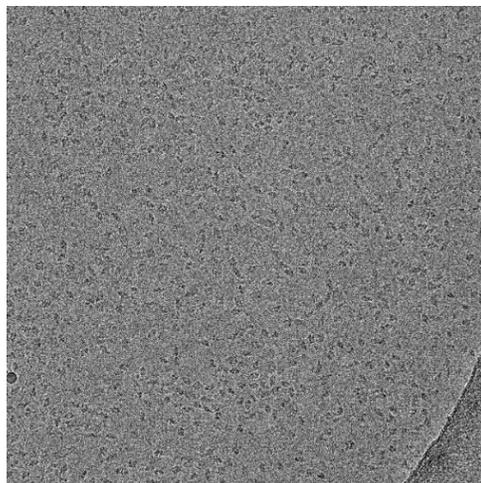
(d)



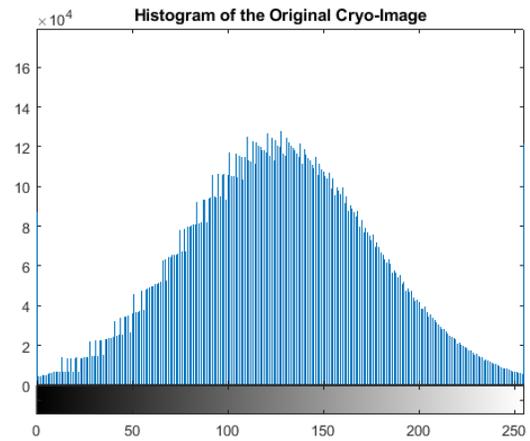
(e)



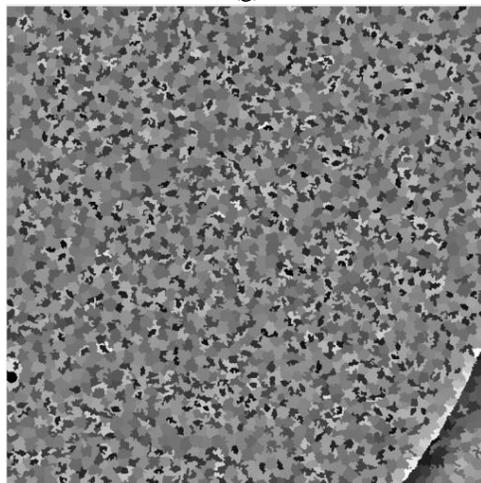
(f)



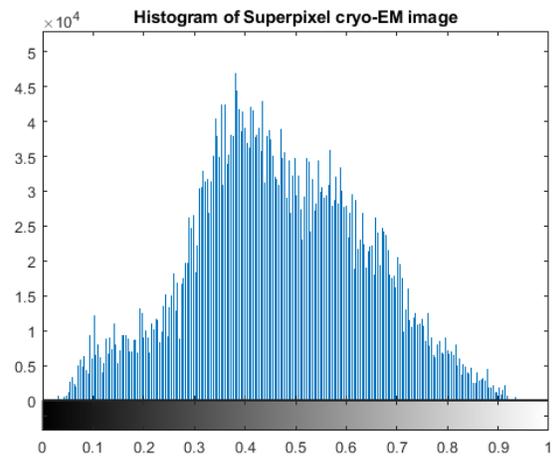
(g)



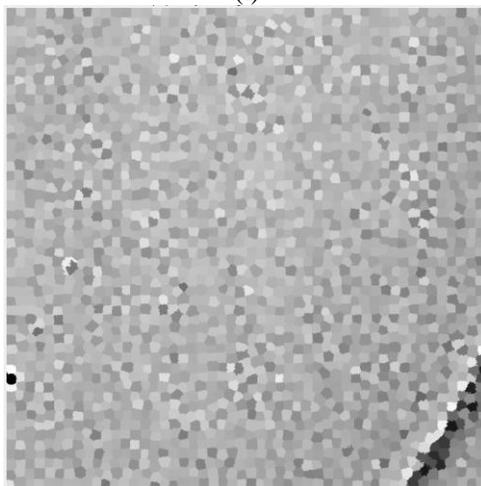
(h)



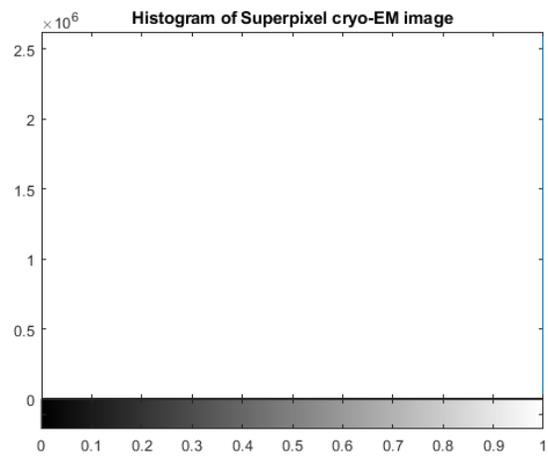
(i)



(j)



(k)

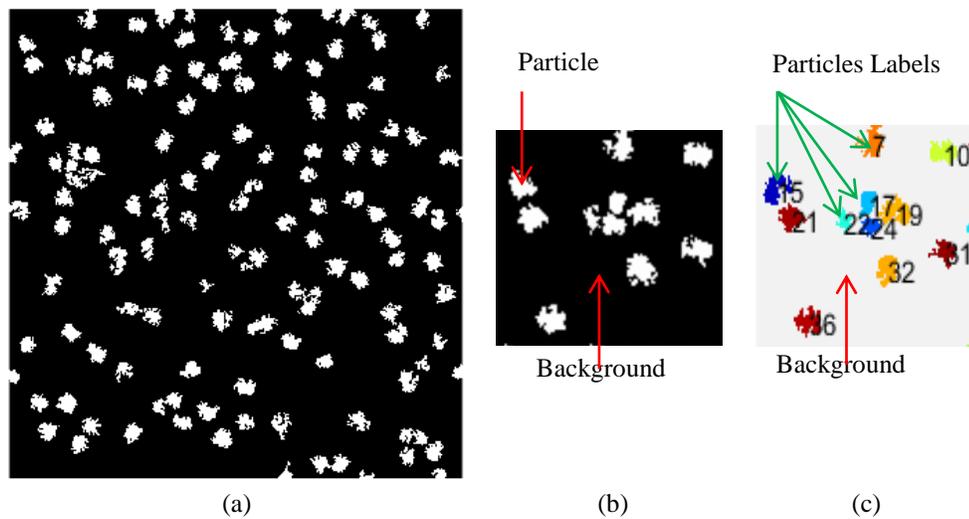


(l)

Figure 4.6: Different cryo-EM intermedia micrograph maps generation using simple linear iterative clustering (SLIC). (a) an original cryo-EM image of Ribosome. (b) the histogram of (a). (c) intermedia micrograph maps generated using simple linear iterative clustering (SLIC) based on the pre-processed image.

(d) histogram of (c). (e) an intermedia micrograph map generated by SLIC from the original image. (f) histogram of (e). (g) an original cryo-EM image of Beta-galactosidase. (h) histogram of (g). (i) an intermedia micrograph map generated by SLIC from the pre-processed image. (j) histogram of (i). (k) an intermedia micrograph map generated by SLIC from the original image. (l) histogram of (k).

The SP-K-means automatically selects the proper cluster number for particles after the non-zero element (clustered particles) is extracted from each cluster image as is showing in Figure 4.7. The SP-K-means does not require the extra human intervention to select the most appropriate cluster as the original k-means [114] does. SP-K-means is a fully automated clustering algorithm based on the assumption that each group of white pixels (non-zero elements) represents one particular single particle in different cluster image and the black pixels represent the cryo-EM background (see Figure 4.7(b)). SP-K-Means labels each group of pixels (particles) in each cluster images (binary) and number each single particle (see Figure 4.7(c)). Based on the total number of particles in each cluster image (see Figure 7(d),(e),(f), and (g)), the SP-K-Means selects the optimal cluster that has the less number of labels which represent the target cluster that has the correct particles number and position in the original image (see Figure 4.7(f)).



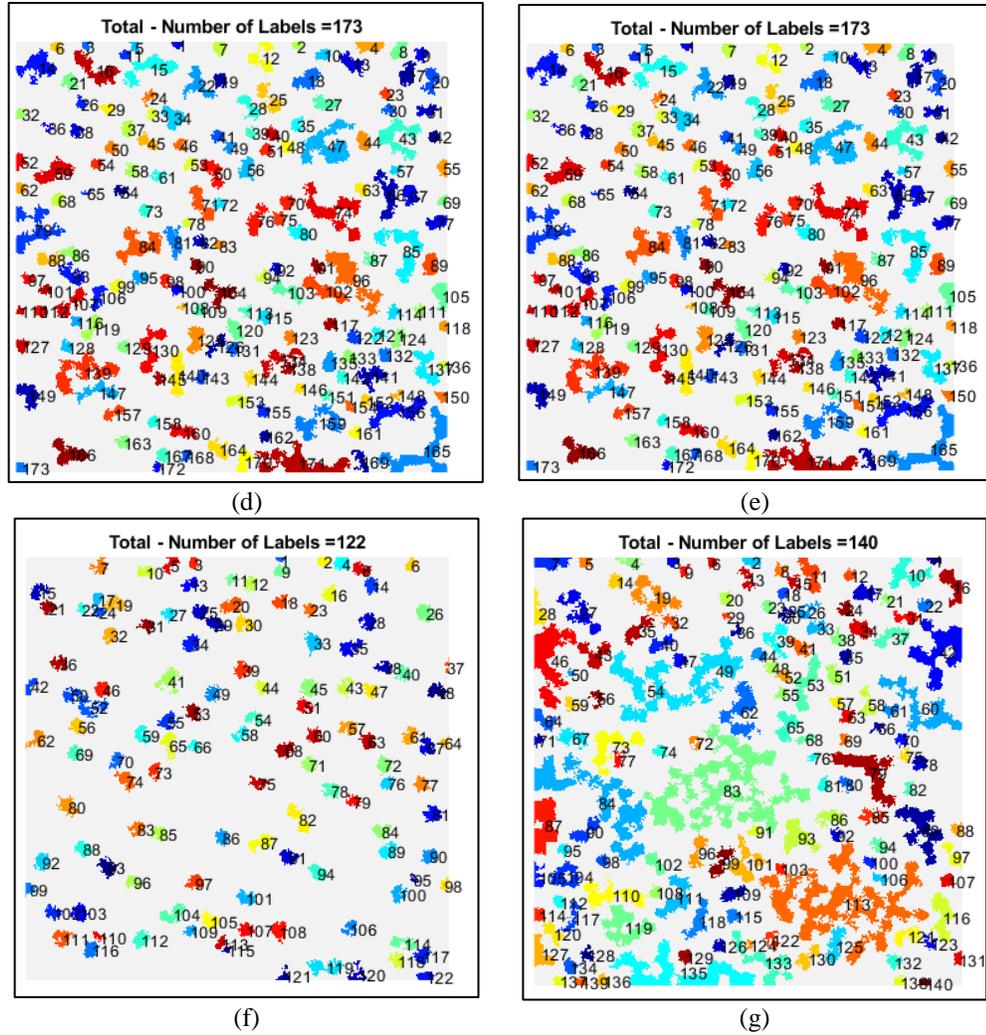


Figure 4.7: SP-K-means evaluation and automated cluster section based on extracting the total number of the particles in each cluster and selecting the cluster that has the minimum total number of particles; (a) the total number of objects (particles) in the cluster index number (#1); (b) the total number of objects (particles) in the cluster index number (#2); (c) the total number of objects (particles) in the cluster index number (#3); (d) the total number of objects (particles) in the cluster index number (#4).

The SP-K-means does not require the extra human intervention to select the most appropriate cluster as the original k-means [114] does. Based on extract the total number of the non-zero element in each cluster and select the cluster that has the minimum total number of the non-zero element (see Figure 7(c)), the SP-K-means is a fully automated clustering algorithm.

The super pixel k-means clustering (SP-K-means) is shown in Algorithm 4.1. Similarly, SP-FCM and SP-IBC are described in Algorithm 4.2 and Algorithm 4.3. The three super clustering methods are fully automated, generate cleaner image masks and run faster than their base clustering algorithms. For instance, the running time of clustering one image using k-means is (46.07 seconds) versus 117.06 seconds of SP-k-means.

Algorithm 4.1 SP-K-Means Clustering Algorithm

Input: pre-processed cryo-EM image I_p
Return super clustered image I_{sc}
Set number of clusters, K
Generate the 2-D super pixel from the super clustered image
begin /*SLIC*/
 Initialize $C_k = [l_k, x_k, y_k]^T$ /* the cluster centres*/
 Move the cluster centre cluster centres to the lowest gradient position in a 3×3 neighbourhood.
 Set label $l(i) = -1$ for each pixel i .
 Set distance $d(i) = \infty$ for each pixel i .
 repeat
 for $k = 1$ to K **do**
 for each pixel i in $2S \times 2S$ region around C_k **do**
 Compute distance D between C_k and i .
 if $D < d(i)$ **then**
 set $d(i) = D$.
 set $l(i) = k$.
 end if
 end for
 end for
 Compute new cluster centre θ_k .
 Compute residual error E .
 until $E \leq threshold$
 generate binary mask
 end /*SLIC*/
 repeat
 for $n = 1$ to N **do**
 Determine the closest representative, θ_k , for x_n
 Set label for data point n to k
 end for

for $k = 1$ to K **do**

Update cluster representative θ_k to the mean with cluster label k

$$\theta_k = \frac{\sum_{n=1}^N u_{nk} x_n}{\sum_{n=1}^N u_{nk}}$$

end for

until change in cluster centres are small

for $k = 1$ to K **do** /*for each clustered image*/

$I_{sc} \leftarrow \text{Min}(\text{Nonzero}(I_k))$ /*extract the total number of the non-zero element in each cluster and select the cluster that has the minimum total number of the non-zero element*/

end for

Algorithm 4.2 SP-FCM Clustering Algorithm

Input: pre-processed cryo-EM image I_p

Return super clustered image I_{sc}

Set number of clusters, k

Generate the 2-D super pixel over segmentation image

begin /*SLIC*/

Initialize the cluster centres $C_k = [l_k, x_k, y_k]^T$

Move the cluster centre cluster centres to the lowest gradient position in a 3×3 neighbourhood.

Set label $l(i) = -1$ for each pixel i .

Set distance $d(i) = \infty$ for each pixel i .

repeat

for $k = 1$ to K **do**

for each pixel i in $2S \times 2S$ region around C_k **do**

Compute distance D between C_k and i .

if $D < d(i)$ **then**

Set $d(i) = D$.

Set $l(i) = k$.

end if

end for

end for

Compute new cluster centre θ_k .

Compute residual error E .

until $E \leq \text{threshold}$

generate binary mask

end /*SLIC*/

```

repeat
  for n = 1 to N do
    Update membership  $u_{nk}$  by taking sum of distance ratios of cluster k and all
    clusters.

```

$$u_{nk} = \sum_{i=1}^K \left(\frac{d(x_n, \theta_k)}{d(x_n, \theta_i)} \right)^{-\frac{1}{m-1}}$$

```

  end for
until change in cluster centres are small
for k = 1 to K do /*for each clustered image*/
   $I_{sc} \leftarrow \text{Min}(\text{Nonzero}(I_k))$  /*extract the total number of the non-zero element in
  each cluster and select the minimum one as a final
  selected clustered image*/
end for

```

Algorithm 4.3 SP-IBC Clustering Algorithm

```

Input: pre-processed cryo-EM image  $I_p$ 
Return super clustered image  $I_{sc}$ 
Set number of clusters,  $k$ 
Generate the 2-D super pixel over segmentation image
begin /*SLIC*/
  Initialize the cluster centres  $C_k = [l_k, x_k, y_k]^T$ 
  Move the cluster centre cluster centres to the lowest gradient position in a  $3 \times 3$ 
  neighbourhood.
  Set label  $l(i) = -1$  for each pixel  $i$ .
  Set distance  $d(i) = \infty$  for each pixel  $i$ .
  repeat
    for  $k = 1$  to  $K$  do
      for each pixel  $i$  in  $2S \times 2S$  region around  $C_k$  do
        Compute distance D between  $C_k$  and  $i$ .
        if  $D < d(i)$  then
          Set  $d(i) = D$ .
          Set  $l(i) = k$ .
        end if
      end for
    end for
    Compute new cluster centre  $\theta_k$ .
    Compute residual error  $E$ .

```

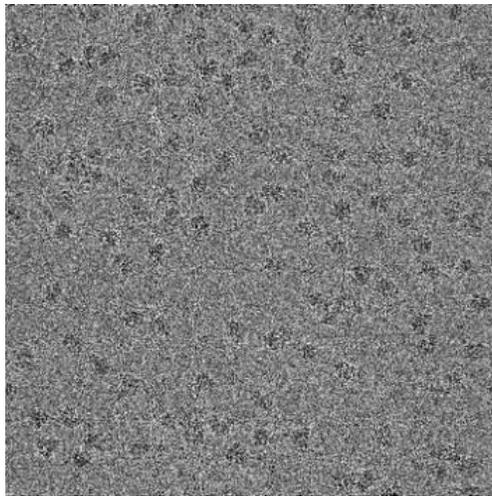
```

until  $E \leq threshold$ 
  Generate binary mask
end /*SLIC*/
Transform the intermedia 2D image map  $I_m$  into 1-D  $I_v$  which has the intensity
values of all the pixels.
 $L \leftarrow height \times width$  where L is the total number of pixels in the  $I_p$ 
 $V_{Max} \leftarrow Max[I_v]$  /*maximum values of intensity in the image*/
 $V_{Min} \leftarrow Min[I_v]$  /*minimum values of intensity in the image*/
 $K \leftarrow 4$  /*set the number of initial cluster based on the interval size=.15*/
for  $i = 1$  to  $K$  do
   $Int_s \leftarrow \left( \frac{V_{Max} - V_{Min}}{K \times .15} \right) \times .15$  /*set the interval size as the vector of intensity
range (max-min) divided by the cluster number  $K$ */
end for
for  $i = 1$  to  $K$  do
   $\theta_k \leftarrow Int_s[i]$  /*initialize the cluster centre based on the interval size*/
end for
repeat
  for  $i = 1$  to  $K$  do
    for  $j = 1$  to  $L$  do
       $Cluster[ind] \leftarrow \min(abs(\theta_k[i] - I_v[j]))$  /*assign  $x_n$  the
cluster  $k$  whose center ( $\theta_k$ ) is closest to  $x_n$  according to the
absolute intensity difference between the two*/
    end for
  end for
  for  $n = 1$  to  $K$  do
     $\theta_k[n] \leftarrow \sum_{n=1}^k \frac{Cluster[k]}{count(ones)}$  /*update the mean  $\theta_k$  of each cluster by
calculating the average intensity values of the
pixels assigned to the cluster */.
  end
until there is no change in cluster centres.

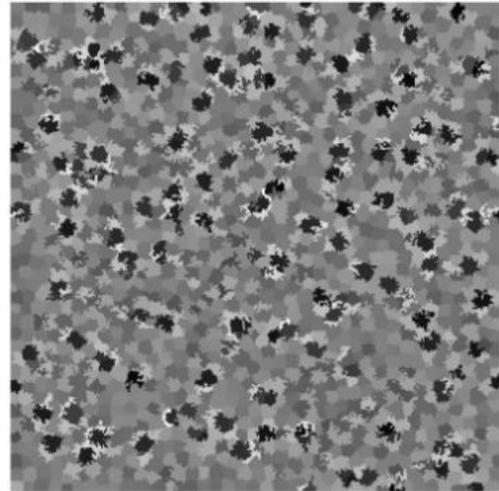
```

Figure 4.8 shows some examples of the super clustering results for Ribosome [119] and Beta-galactosidase [120]) and comparing them with the base clustering algorithms. Figure 4.8 shows that the super clustering results are better than those of the base algorithms. The three super clustering methods are fully automated, generate cleaner image masks and run faster than their base clustering algorithms. For instance, the running time

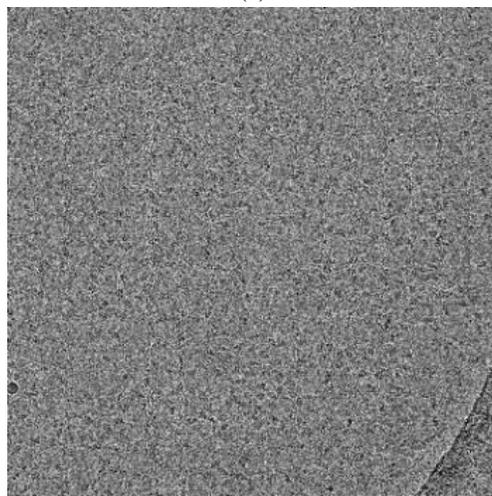
of clustering one image using k-means is (46.07 seconds) versus 117.06 seconds of SP-k-means.



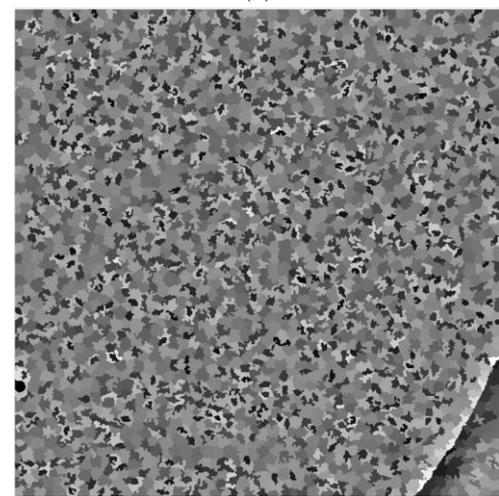
(a)



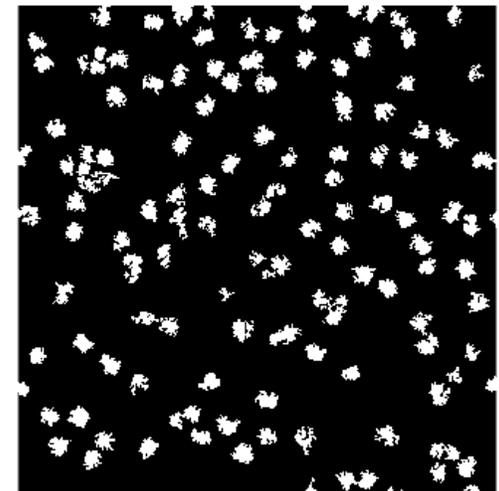
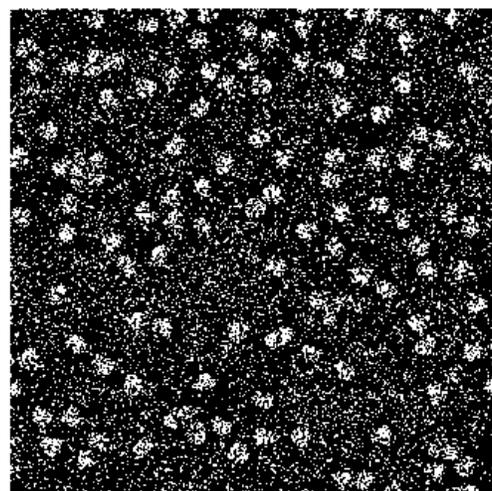
(b)



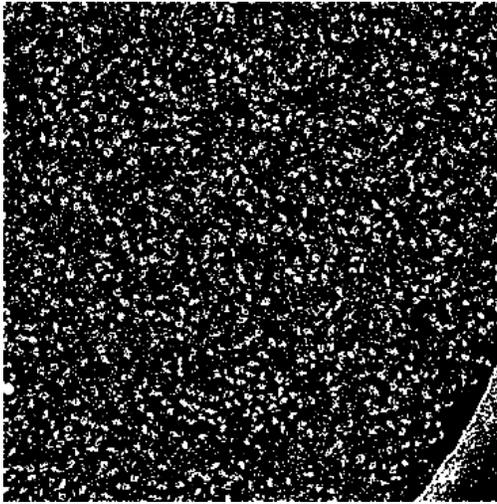
(c)



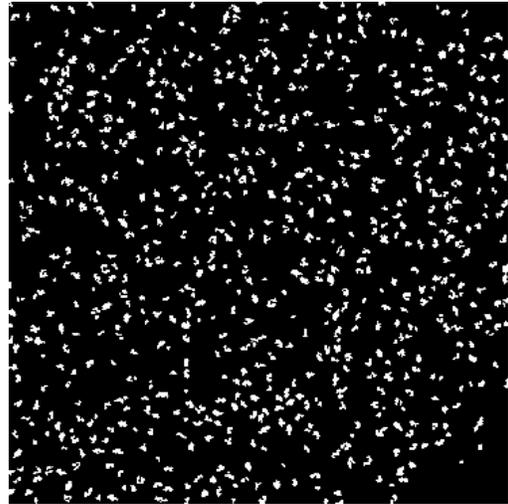
(d)



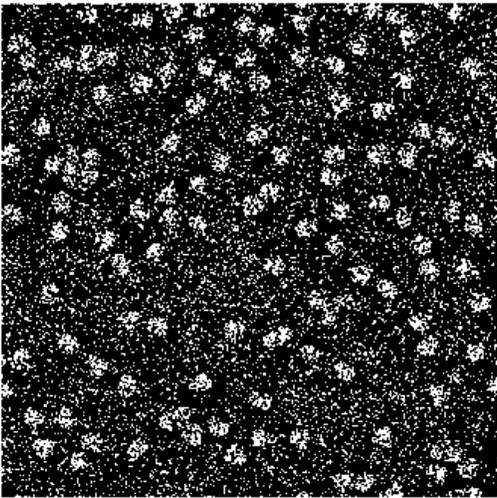
(e)



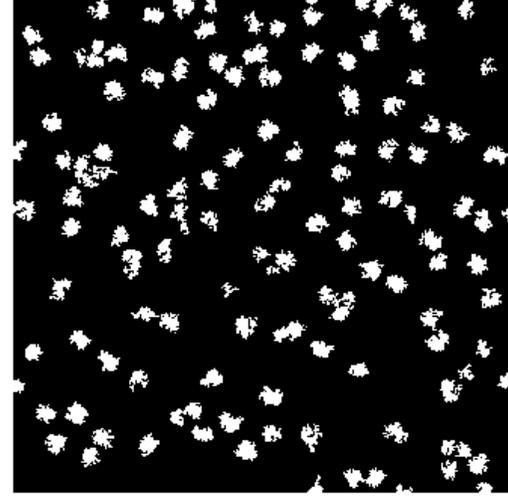
(f)



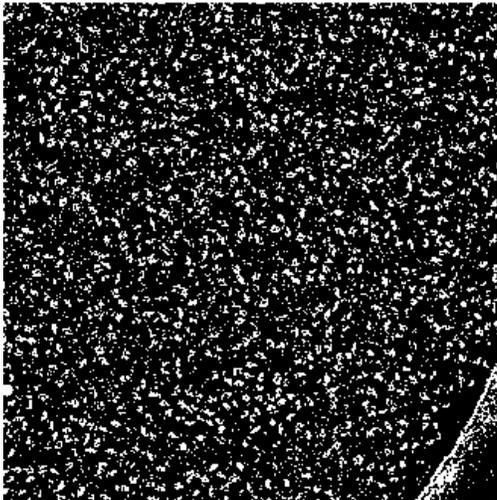
(g)



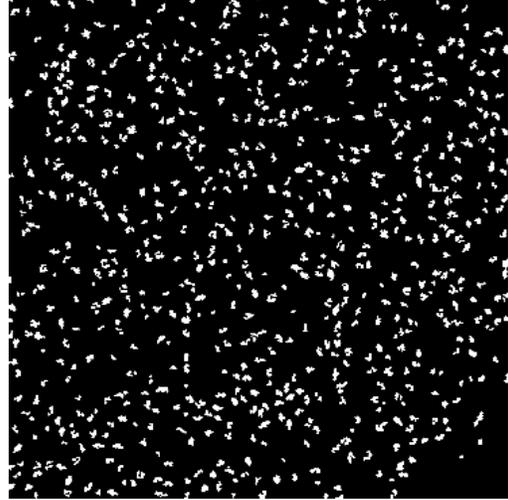
(h)



(i)



(j)



(k)

(l)

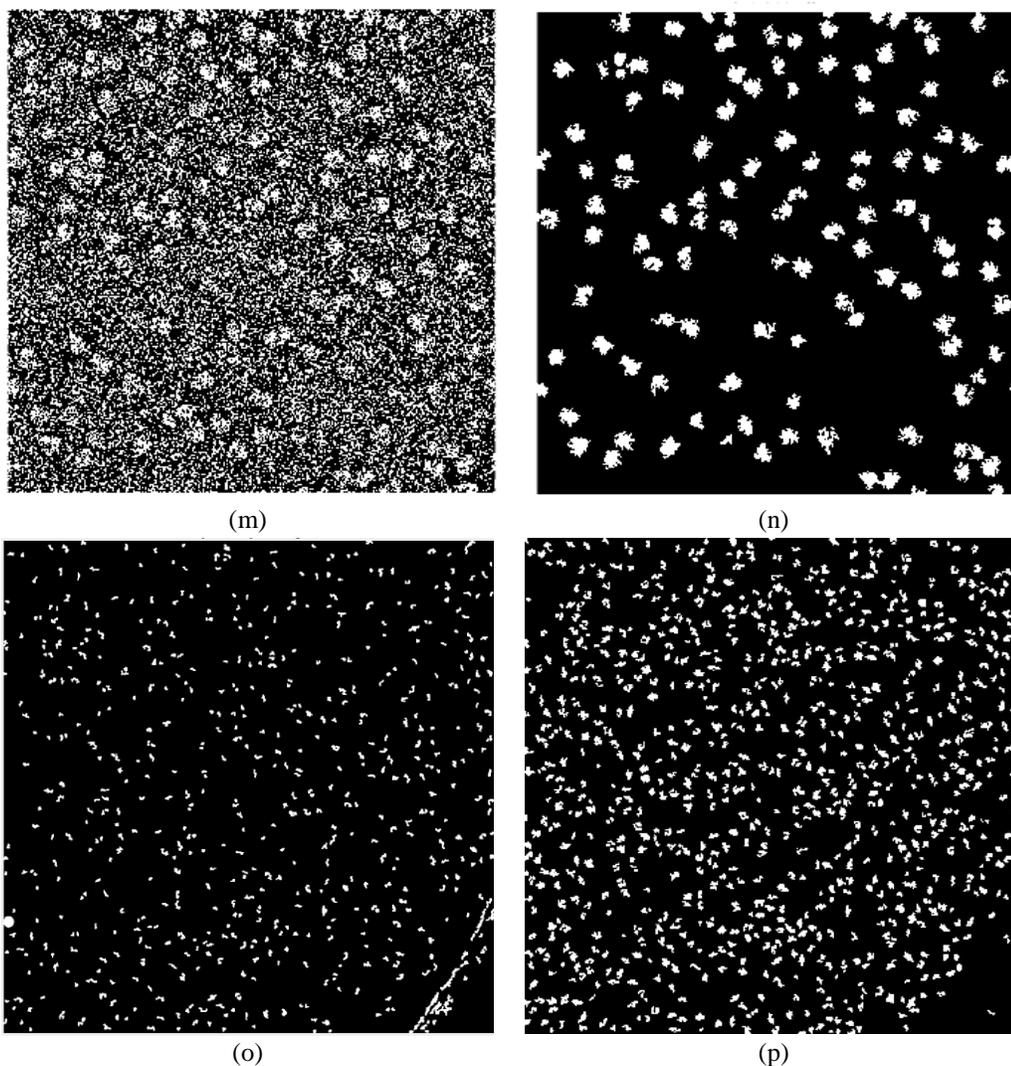


Figure 4.8: Cryo-EM super clustering results of SP-K-means, SP-FCM, and SP-IBC in comparison with the base algorithms. (a) an original cryo-EM image of Ribosome. (b) the intermedia micrograph map generated by SLIC from the pre-processed image of Ribosome. (c) an original cryo-EM image of Beta-galactosidase. (d) the intermedia micrograph map generated by SLIC for Beta-galactosidase. (e) k-means clustering results of the Ribosome. (f) SP-K-means clustering results of Ribosome. (g) k-means clustering results of the Beta-galactosidase cryo-EM image. (h) SP-K-means clustering results of the Beta-galactosidase cryo-EM image. (i) FCM clustering results of the Ribosome cryo-EM image. (j) SP-FCM clustering results of the Ribosome cryo-EM image. (k) FCM clustering results of the Beta-galactosidase cryo-EM image. (l) SP-FCM clustering results of the Beta-galactosidase cryo-EM image. (m) IBC clustering results of the Ribosome cryo-EM image. (n) SP-IBC clustering results of the Ribosome cryo-EM image. (o) IBC clustering results of the Beta-galactosidase cryo-EM image. (p) SP-IBC clustering results of the Beta-galactosidase cryo-EM image.

4.3.3 Stage 3: Particles Picking

The particle picking stage has two main steps. The first step is the binary mask image cleaning, while the second step is the particle detection and picking. The image masks generated by the super clustering methods (e.g. Figure 7(f), (j), (n), (h), (l) and (p)) are clean and do not required a post processing stage to clean them up. But the binary mask image cleaning step is required to remove some small and noisy objects from the images mask generated by the base clustering algorithms (e.g. Figure 7(e), (i), (m), (g), (k) and (o)).

4.3.4 Binary Mask Cleaning

The regular binary mask for each cryo-EM cluster image generated by a base clustering method (k-means, FCM, and IBC) is cleaned through the removal of the small and non-connected objects. The image cleaning algorithm is shown in Algorithm 4.

Algorithm 4.4 Image Cleaning

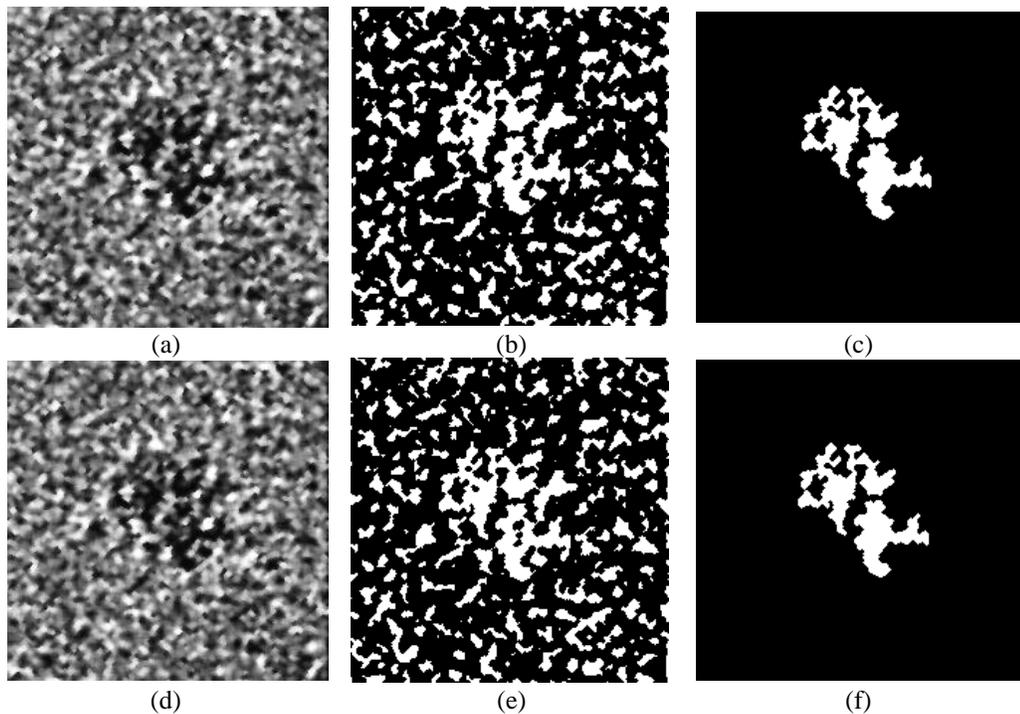
```
Input:  $I_c$  /*cluster cryo-EM image */
Return:  $I_{cc}$  /*cleaned cluster image */
 $I_{c1} \leftarrow \text{imopen}(I_c)$  /* apply image opening on the cluster image to enlarge small
blobs*/
 $L \leftarrow \text{bwlabel}(I_{c1})$  /* label each object in the cluster image which returns a label
matrix L that contains 8-connected object in the cluster */
for  $i=1$  to  $L$  do /* for each object in the clustered image*/
     $I_{\text{object}} \leftarrow \text{state}(L(k))$  /* get each particle where k is the total number of
objects in the cluster cryo-EM*/
     $I_{\text{object}} \leftarrow \text{remove}(\text{state}(L(k)))$  /*remove all the connected components that
are smaller than the p pixels*/
    centroid = stats(k).Centroid /*mark and get the actual index numbers and
the centroid of each object */
end for
 $\text{obj}_{\text{number}} \leftarrow \text{is member}(I_{\text{object}})$  /*extract the number of object (particles)*/
 $L \leftarrow \text{bwlabel}$  /*label each object (particle)*/
```

```

for  $i=1$  to  $L$  do /* for each object (particles) */
  Do size filtering and roundness filtering
   $p \leftarrow [\text{props. label}]$  /* extract the 8-connected labeled object */
   $\text{keeperObjects} \leftarrow \text{props. label} > p$  /* remove each object that less than 8-
    connected component in the binary image */
   $I_{c2} \leftarrow \text{keeperObjects}$  /* get actual index numbers instead of a logical vector */
   $I_{cc} \leftarrow \text{bwareaopen}(I_{c2})$  /* produce new binary image with only the small, or
    non-connected object */
end for
  Generate and getting a new binary image

```

Figure 4.9 shows the examples of the image cleaning and non-connected object removal on the image masks generated by the three base clustering algorithms. Figure 4.9 shows that image cleaning separates particles from the background noise in the image masks generated by the three base clustering methods well.



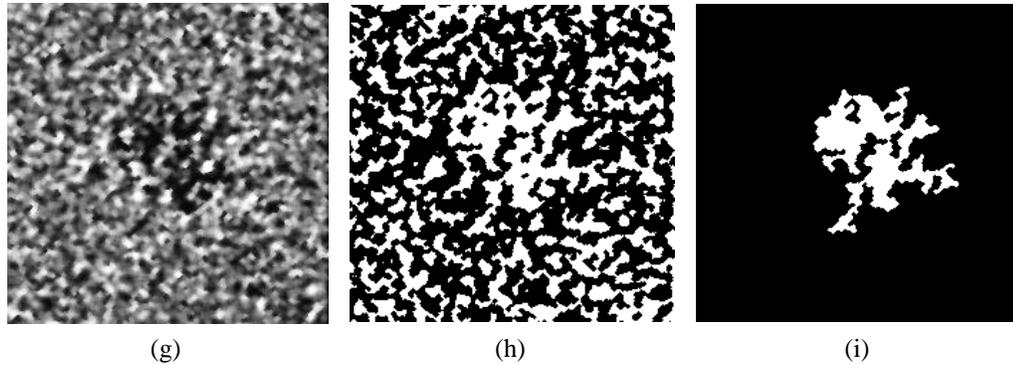
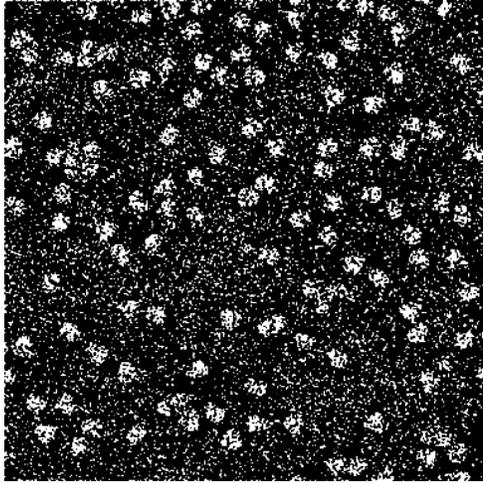
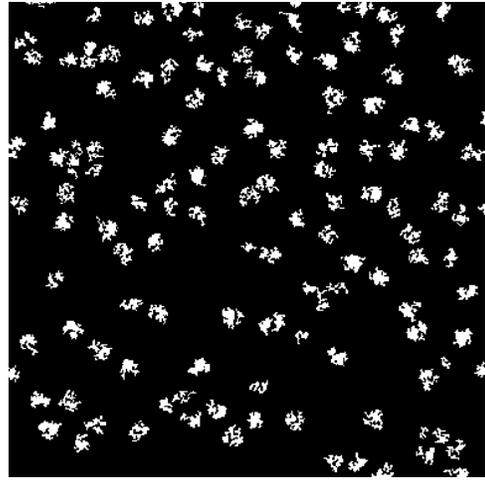


Figure 4.9: A zoomed-in selected particle image before and after binary image cleaning and non-connected objects removal on the image masks generated by the three base clustering methods. (a) the original zoomed-in particle image. (b) the particle clustering image before binary image cleaning by k-means for Ribosome. (c) the particle clustering image after binary image cleaning by k-means for Ribosome. (d) the original zoomed-in particle image. (e) the particle clustering image before binary image cleaning by FCM clustering for Ribosome. (f) the particle clustering image after binary image cleaning by FCM clustering for Ribosome. (g) the original zoomed-in particle image. (h) the particle clustering image before binary image cleaning by IBC clustering for Ribosome. (i) the particle clustering image after binary image cleaning by IBC clustering for Ribosome.

Figure 4.10 shows the whole particle images of Ribosome [119] and Beta-galactosidase [120] before and after image cleaning for k-means [114], FCM [115], and IBC [116] clustering respectively. For instance, Figure 4.10(a) and (b) show the whole particle image of Ribosome [119] cryo-EM before and after image cleaning for k-means respectively. Although, Figure 10(c) and (d) show the whole particle image of Beta-galactosidase [120] cryo-EM before and after image cleaning for k-means respectively. In both cases, we notice that in the non-particles (small objects) and noise in the clustering image after image cleaning is cleaned and are removed from the background where the objects (particles) become clearer to detect and pick.



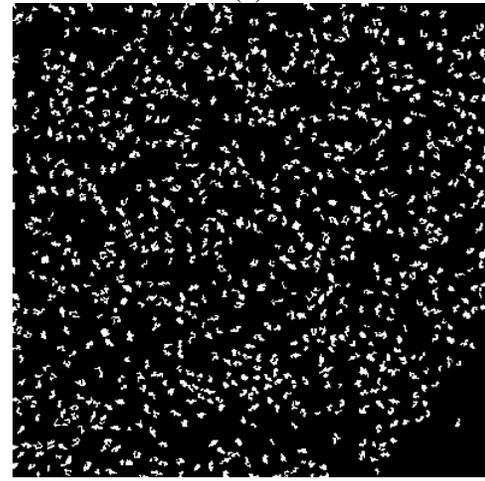
(a)



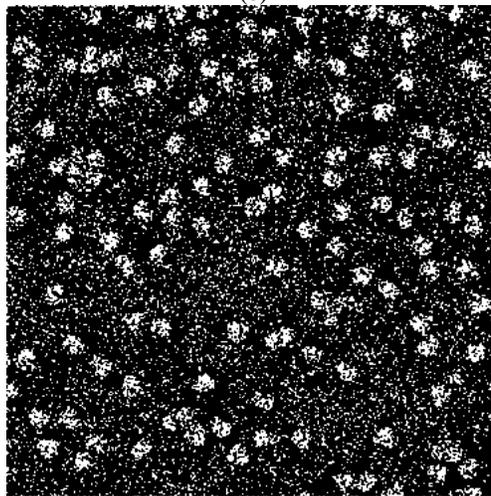
(b)



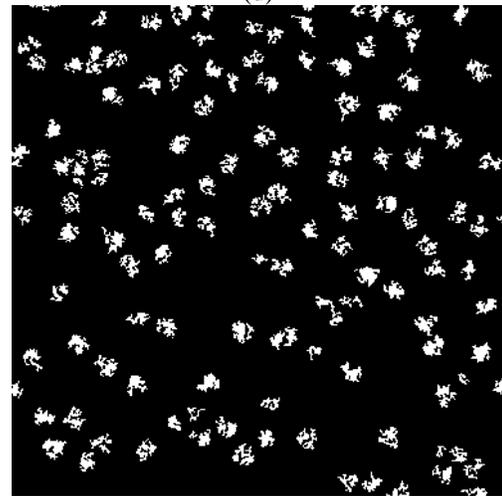
(c)



(d)



(e)



(f)

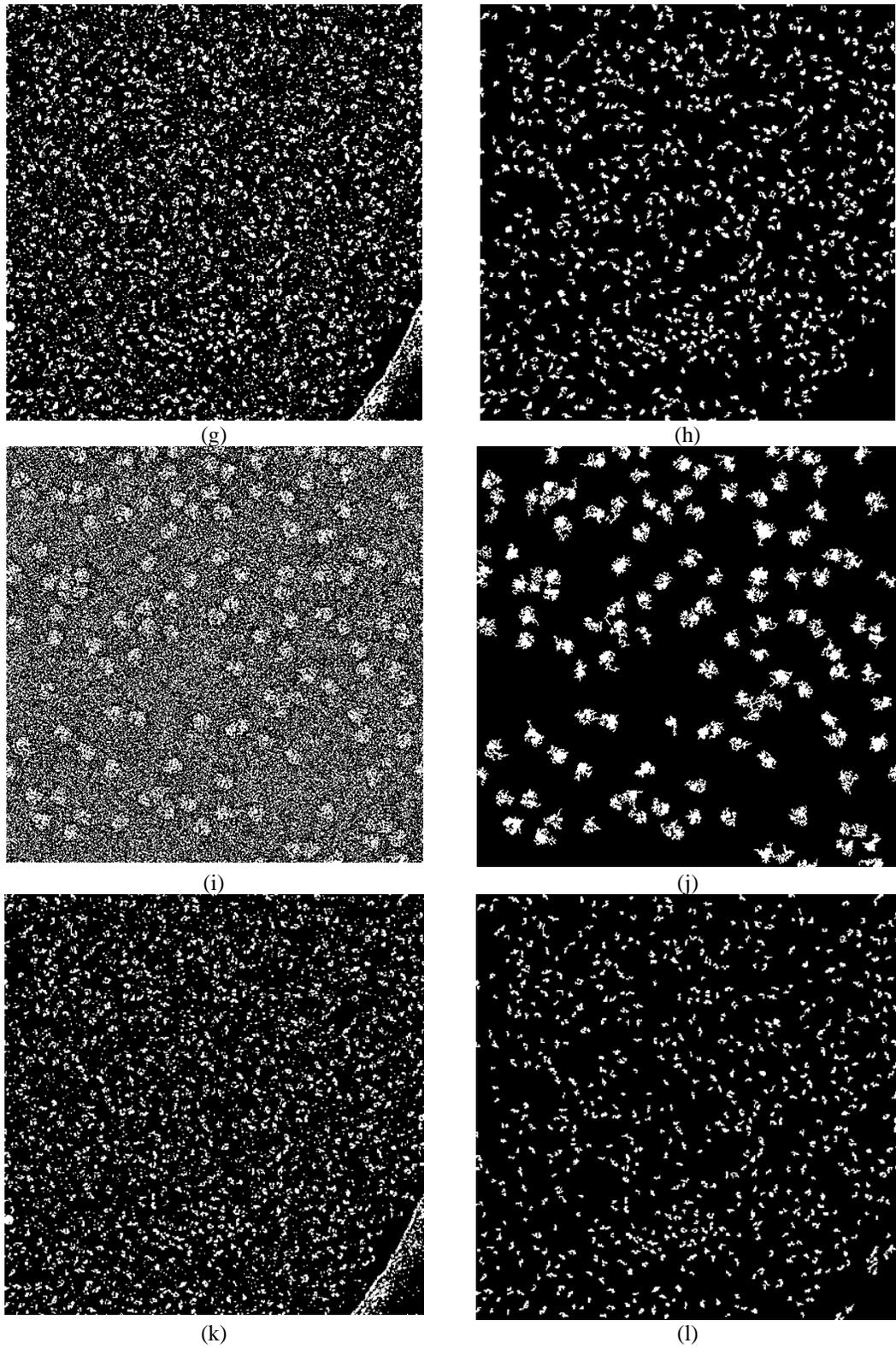


Figure 4.10: The whole cryo-EM particle clustering results before and after binary image cleaning for the three base clustering methods. (a) the Ribosome clustered image of k-means. (b) the image mask of (a) after

image cleaning. (c) the Beta-galactosidase clustered image of k-means. (d) the binary image mask of (e) after image cleaning. (e) the Ribosome clustered image of FCM. (f) the binary image mask of (e) after image cleaning. (g) the Beta-galactosidase clustered image of FCM. (h) the binary mask of (i) after image cleaning. (i) the Ribosome clustered image of IBC. (j) the binary image mask of (i) after image cleaning. (k) the Beta-galactosidase clustered image of IBC. (l) the binary image mask of (k) after image cleaning.

4.3.5 Single Particle Detection and Picking

Since the shapes of the protein particles are complex or irregular, a particle detection and picking step is applied on clean image masks to detect each single particle. The particle detection and picking algorithm is described in Algorithm 4.5.

Algorithm 4.5 Single Particle Detection and Picking

```

Input:  $I_{cs}$  /*cleaned cluster image with square shapes only*/
Return:  $I_{cps}$  /*cleaned cluster image with perfect square shapes*/
 $L \leftarrow \text{bwlabel}(I_{c1})$  /* label each object in the cluster image */
for  $i=1$  to  $L$  do /* for each object in the clustered image*/
    Stats  $\leftarrow$  regionprops( $I_{cs}$ ) /* measure properties of particle region*/
    Areas  $\leftarrow$  [props.Area /* compute all the shape measurements and the
                    pixel value measurements as well*/
end for
for  $i=1$  to  $\text{size}(\text{keeperObjects})$  do /* for each particle object*/
     $[x, y] \leftarrow \text{centroid}(\text{keeperObjects})$  /*extract the centroid is the
    horizontal coordinate (or x-coordinate) and vertical coordinate (or y-
    coordinate) */
    Draw all bounding box for a discontinuous region
end for

```

The detection algorithm returns a bounding box drawn around each particle detected in the cryo-EM. Figure 4.11 shows the detection results of two different cryo-EM images of Ribosome and Beta-galactosidase. The results of the super clustering methods are significantly better than the base clustering methods for both Ribosome [119] and Beta-galactosidase images [120]

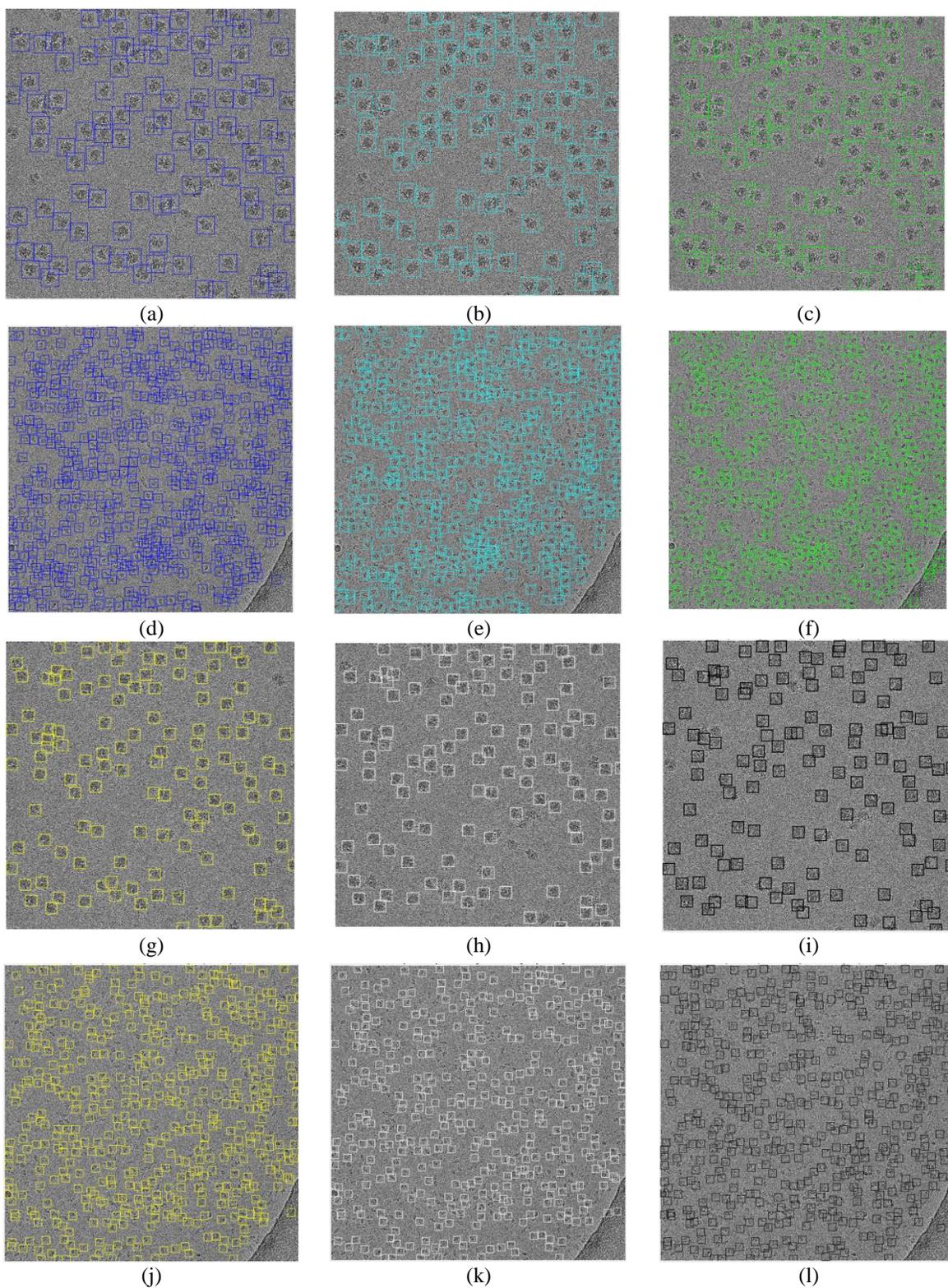


Figure 4.11: The results of detecting particles of irregular and complex shapes on Ribosome and Beta-galactosidase images. (a) particle detection and picking by k-means on the Ribosome image. (b) particles detection and picking by FCM) on the Ribosome image. (c) particles detection and picking by IBC on the

Ribosome image. (d) particle detection and picking by k-means on the Beta-galactosidase image. (e) particle detection and picking by FCM on the Beta-galactosidase image. (f) particle detection and picking by IBC on the Beta-galactosidase image. (g) particle detection and picking by SP-K-means on the Ribosome image. (h) particle detection and picking by SP-FCM on the Ribosome image. (i) particle detection and picking by SP-IBC on the Ribosome image. (j) particle detection and picking by SP-K-means on the Beta-galactosidase image. (k) particle detection and picking by SP-FCM on the Beta-galactosidase image. (l) particles detection and picking by SP-IBC on the Beta-galactosidase image.

4.4 Results and Discussion

4.4.1 Datasets

Images from two datasets (Ribosome [119] and Beta-galactosidase [120]) are used to evaluate the fully automated particle picking using the base and super clustering methods. The two datasets were download form the Electron Microscopy Public Image Archive (EMPIAR-10028) [32]. Ribosome dataset [119] is in a multi-frame MRC image format (32 Bit Float). The size of each micrograph is 4096 by 4096 pixels. It consists of 1081 micrographs each having 16 frames per image.

Beta-galactosidase dataset [120] is in the single-frame MRC image format (32 Bit Float). The size of each micrograph is 4096 by 4096 pixels. It consists of 84 micrographs. The micrographs are the average, without any realignment, of 24 raw movie frames (accumulating 24 electron per squared Angstrom in a 1.5 second exposure).

4.4.2 Evaluation Metrics

In general, the signal-to-Noise ratio is the way that the noise signal is measured in either signal or an image. In another words, a better way to assess the amount of noise in an image is to measure the ratio of pure pixels (called mean of the signal or mean of the pixel values) to the noisy pixels which is the standard deviation of the signal (called the noise standard deviation) as it given in Equation (4.7).

$$SNR = \frac{\text{Mean signal}}{\text{Noise Standard Deviation}} = \frac{\left(\frac{1}{n} \sum_{i=1}^n x_i^2\right)^{\frac{1}{2}}}{\sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x}_i)^2}} \quad (4.7)$$

where n is the total number of pixels in the micrograph. In order to reduce the radiation damage to the biomolecules of interest during the imaging process of the microscopy, a limited electron dose is used as the high-energy electrons can greatly damage the specimen during imaging and results in extremely noisy micrographs. Moreover, the micrographs contain two-dimensional projections of the particle in different orientations. Generally, cryo-EM images have low contrast, due to the similarity of the electron density of the protein to that of the surrounding solution, as well as the limited electron dose used in data collection. In addition, the micrographs may contain sections of ice, deformed particles, protein aggregates, etc., which can complicate particle picking. Although, the cryo-EM images (micrographs) of the protein particles are taken by electron microscope, which contain randomly arranged particles along with non-particles—bits of frost, deformed particles, protein aggregates and so on. These images suffer from heavy background noise and low contrast, due to a limited electron dose used in imaging. For these reasons, we called the micrographs are the extremely low signal-to-noise ratio.

Typically, for the first pre-processing stage, we use the EMAN2 software [121] to adjust the global intensity of the cryo-EM and convert them from MRC file format to the PNG image format in order to apply standard imaging processing tools to them. In terms of selecting the best results, we used different scaling factors with EMAN2 [121] to adjust the intensity of the cryo-EM images. Then, we compute different evaluation metric such as the peak signal-to-noise ratio (PSNR), signal-to-noise ratio (SNR), and mean squared error (MSE) to evaluate the improvement of the quality of cryo-EM images in the whole

dataset in each different scaling factor and compare the results with the original micrographs (i.e. without the using the intensity adjustment scaling factor).

Based on the SNR evaluation metric Equation (7) where noise signal is measured, the PSNR often measure ration between the maximum signal (pure pixels) and noise (corrupted pixels). PNSR uses a logarithmic decibel scale to measure the ration between the maximum signal and noise that have a very wide dynamic range as is given in Equation (4.8).

$$SNR = 10 \times \log_{10} \left(\frac{MAX_i^2}{MSE} \right) \quad (4.8)$$

where MAX_i is the maximum possible pixel value of the micrograph and MSE is mean squared error given in Equation (4.9):

$$SNR = \frac{1}{m \times n} \sum_{i=1}^n \sum_{j=1}^n (|I(i, j) - \hat{I}(i, j)|) \quad (4.9)$$

Where $m \times n$ is the micrograph image size, I is the original micrograph and \hat{I} is the pre-processed micrograph.

For the preprocessing stage, we use common image preprocessing criteria such as peak signal-to-noise ratio (PSNR), signal-to-noise ratio (SNR), and mean squared error (MSE) to evaluate the improvement of the quality of cryo-EM images [124]. For the particles clustering and detection stages, we use the accuracy, precision, recall and F1-score (i.e. the geometry mean of precision and recall) to evaluate the particle clustering/detection results. Preprocessing Results

Table 4.1 reports the average quality measurements of the cryo-EM images with/without EMAN2 intensity adjustment. The average quality measurements (PSNR, SNR and MSE) of the original cryo-EM images were 28.06, 6.99 dB, 26218.13,

respectively. The intensity adjustment with many scaling factors improved the quality. The best scaling factor that increased the PSNR and SNR and at the same time decreased the MSE was the “sane”. “sane” picks a good range of scaling factors automatically. The three scores of using the “sane” scaling factor were improved by 29.27, 8.10 dB) and 0.198643. Figure 4.12(a) compares the average PSNR and SNR scores of the cryo-EM images before and after all the preprocessing steps in the preprocessing stage. Figure 4.12(b) shows the MSE scores of the cryo-EM images before and after all the preprocessing steps.

Table 4.1: The average peak signal-to-noise ratio (PSNR), signal-to-noise ratio (SNR), and mean squared error (MSE) of the cryo-EM images without or with EMAN2 intensity adjustment according to different scaling factors.

Intensity Adjustment Scaling Factor	PSNR	SNR	MSE
Original Image (without Adjustment)	28.06118	6.99260 dB	0.262181
Scaling Factor=10	27.55033	20.00023 dB	0.294908
Scaling Factor=5	28.06118	13.98521 dB	0.262181
Scaling Factor=4	28.34026	12.05305 dB	0.245863
Scaling Factor=3	28.84182	9.59165 dB	0.219047
Scaling Factor=2	29.91141	6.43044 dB	0.17123
Scaling Factor=1	32.61282	2.87674 dB	0.091926
Scaling Factor=.1	44.37584	0.16536 dB	0.006125
Scaling Factor=.5	35.81839	1.26105 dB	0.043942
Scaling Factor=.25	39.23404	0.55222 dB	0.020013
Scaling Factor=sane	29.26647	8.09887 dB	0.198643

The average PSNR score is increased from 77.43 to 78.57 and the average SNR score from 3.40 to 4.05. The average MSE is reduced from 0.302 to 0.233. The range of PSNR scores has increased from [77.429-77.43] for the original cryo-EM images to [78.52-78.64] for the preprocessed ones. The range of the SNR scores increased from [3.36-3.44] for the original images to [4.04-4.052] for the preprocessed ones. The range of MSE scores is decreased from [0.3026-0.3033] to [0.23-0.237] after the preprocessing steps. According

to Student's t test, the p-values of the changes of PSNR, SNR and MSE scores caused by the preprocessing are $7.10e-35$, $2.46e-35$ and $6.06e-36$ respectively, indicating that the preprocessing steps significantly improve the quality of cryo-EM images.

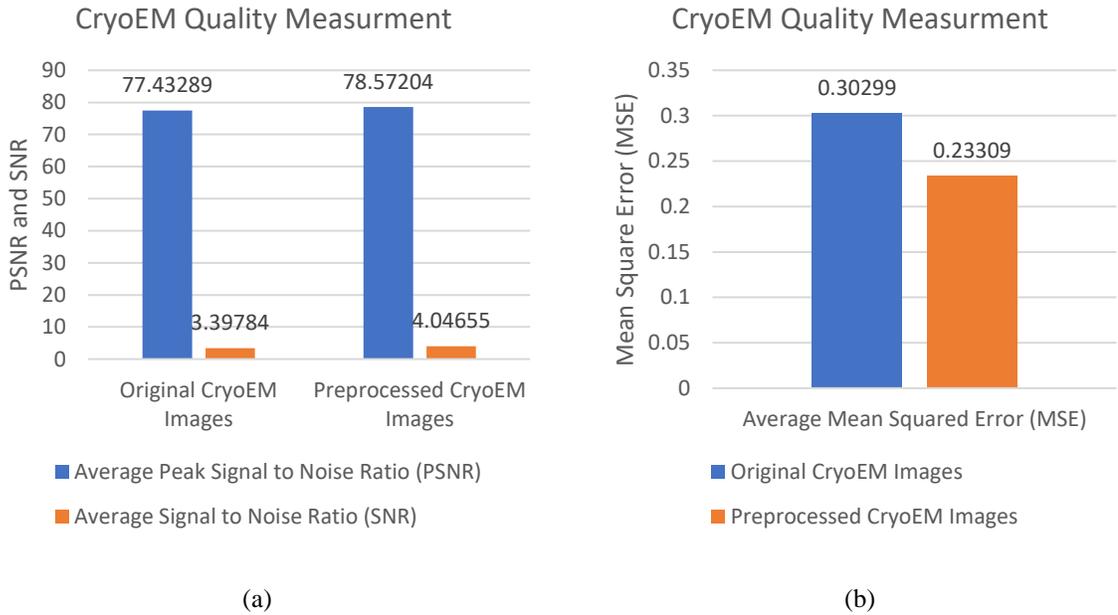


Figure 4.12: The quality of Cryo-EM images before and after the pre-processing Stage. (a) the average PSNR and SNR values of the cryo-EM images before and after the pre-processing steps. (b) average MSE values of the cryo-EM images before and after the pre-processing steps.

4.4.3 Particle Clustering, Detection and Picking Results

In order to evaluate the performance of automated particle clustering and picking, we generated a true reference by manually picking the particles on the images. Figure 4.13 illustrates the of the entire workflow of the super clustering approach for fully automated complex and irregular single particle picking in cryo-EM.

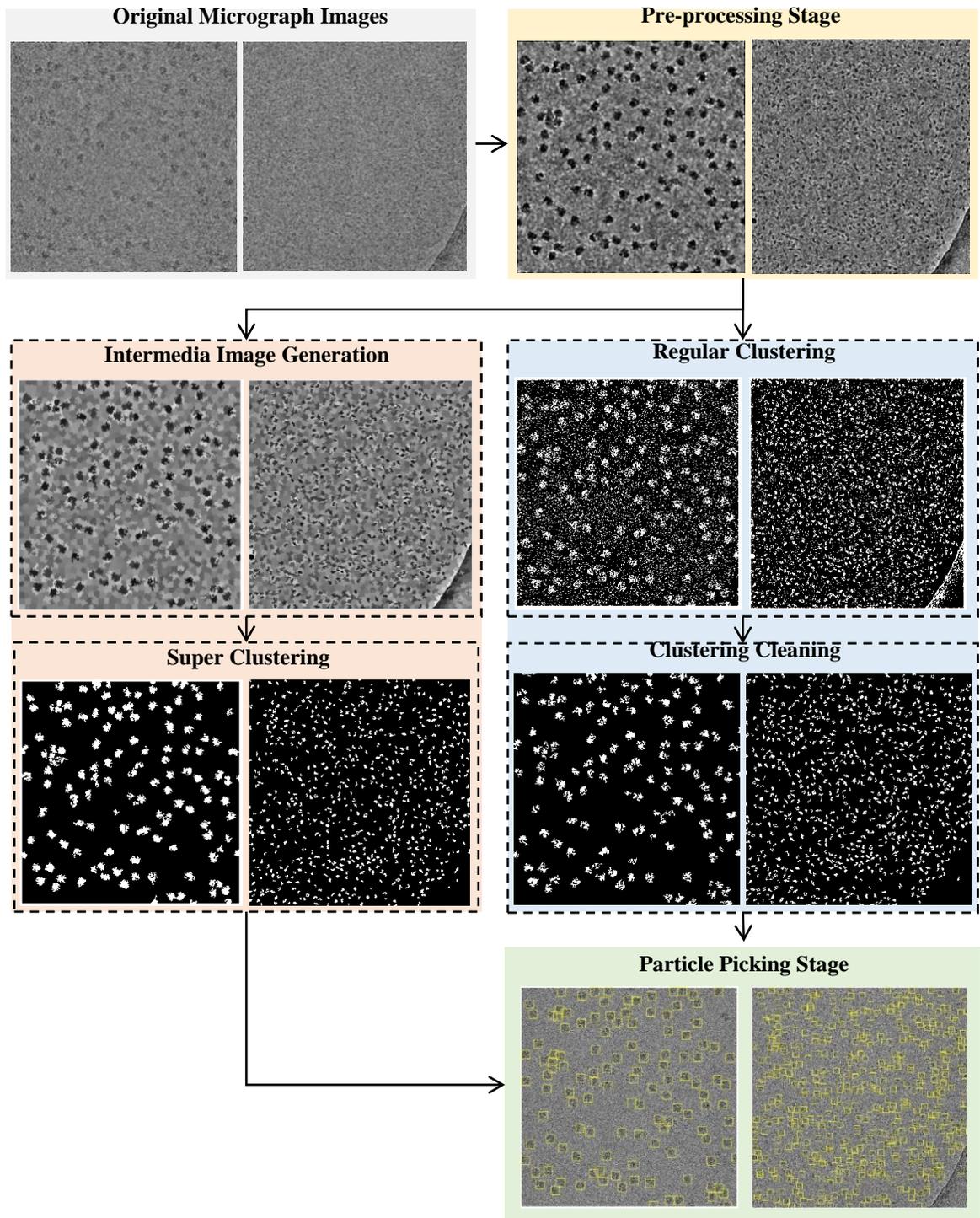
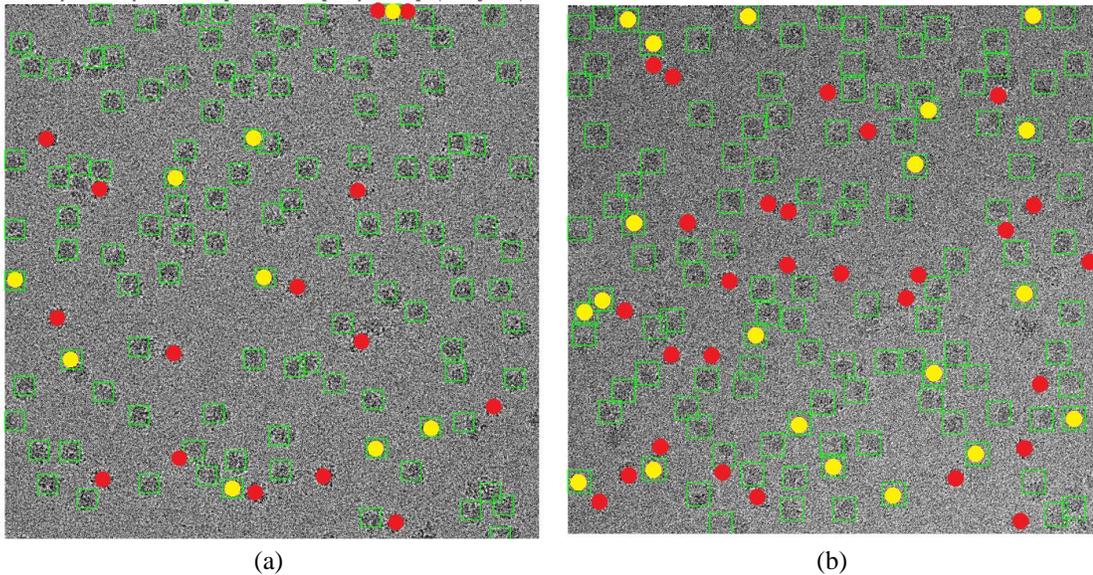
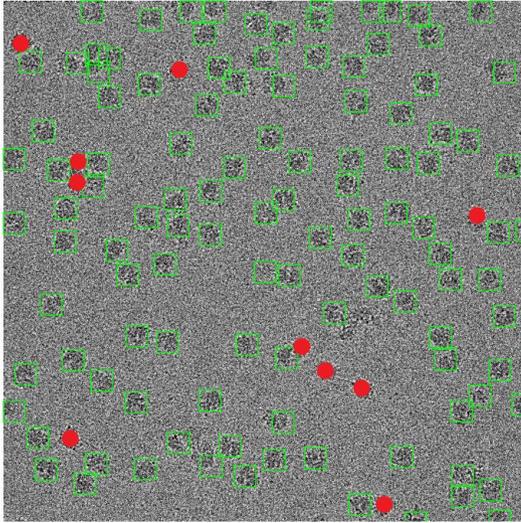


Figure 4.13: The entire workflow of the super clustering approach for fully automated complex and irregular single particle picking in cryo-EM.

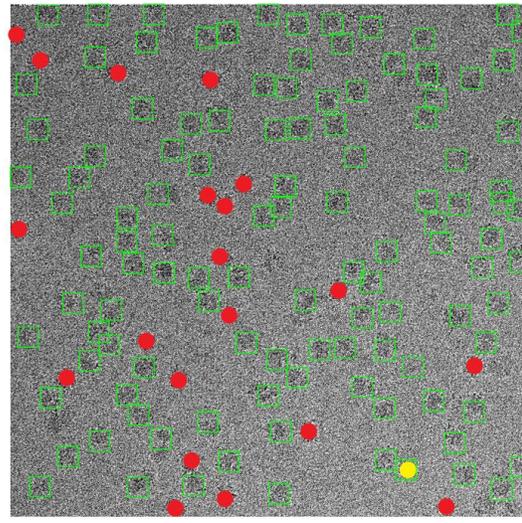
Super clustering approach is designed for fully automated single particle picking in cryo-EM. Our framework contains three stages. The first stage is the micrographs pre-processing (shown on the yellow box of Figure 4.13). The second stage is clustering stage which has two different approaches regular clustering approach (shown on the blue box of Figure 4.13) and super clustering approach (shown on the orange box of Figure 4.13). The third stage is the single particle picking (shown on the green box of Figure 4.13).

Figure 4.14 shows some examples of the fully automated complex single particle shape detection and picking using the super clustering methods and base cluster methods. The true particles failed to be detected (false negatives) are denoted by red dots. Yellow dots represent the non-particle background (e.g. icy) objects falsely detected as particles (false positives).

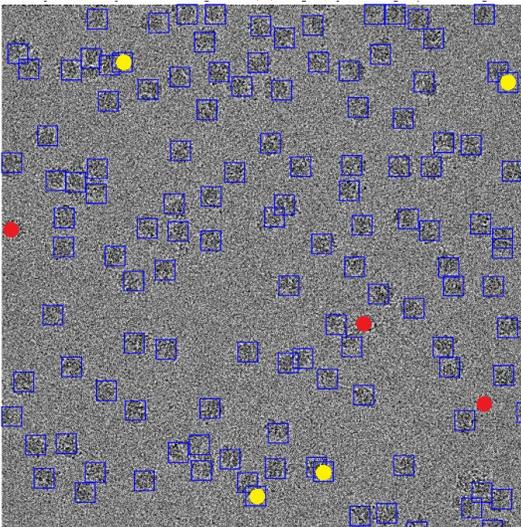




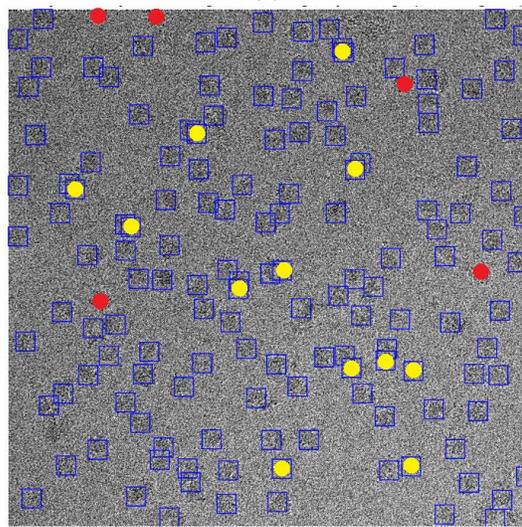
(c)



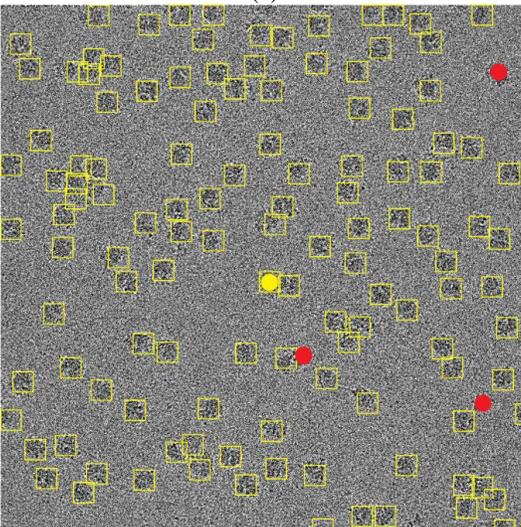
(d)



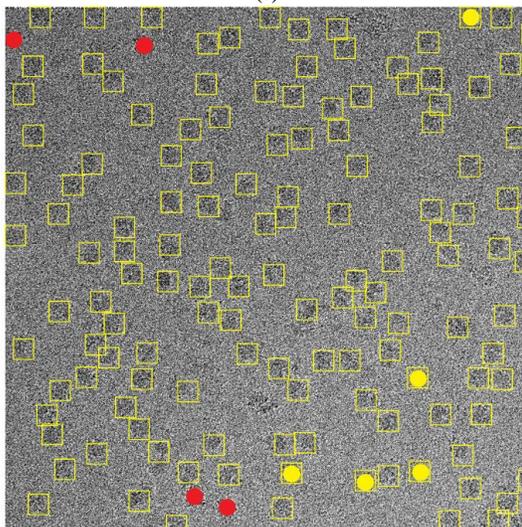
(e)



(f)



(g)



(h)

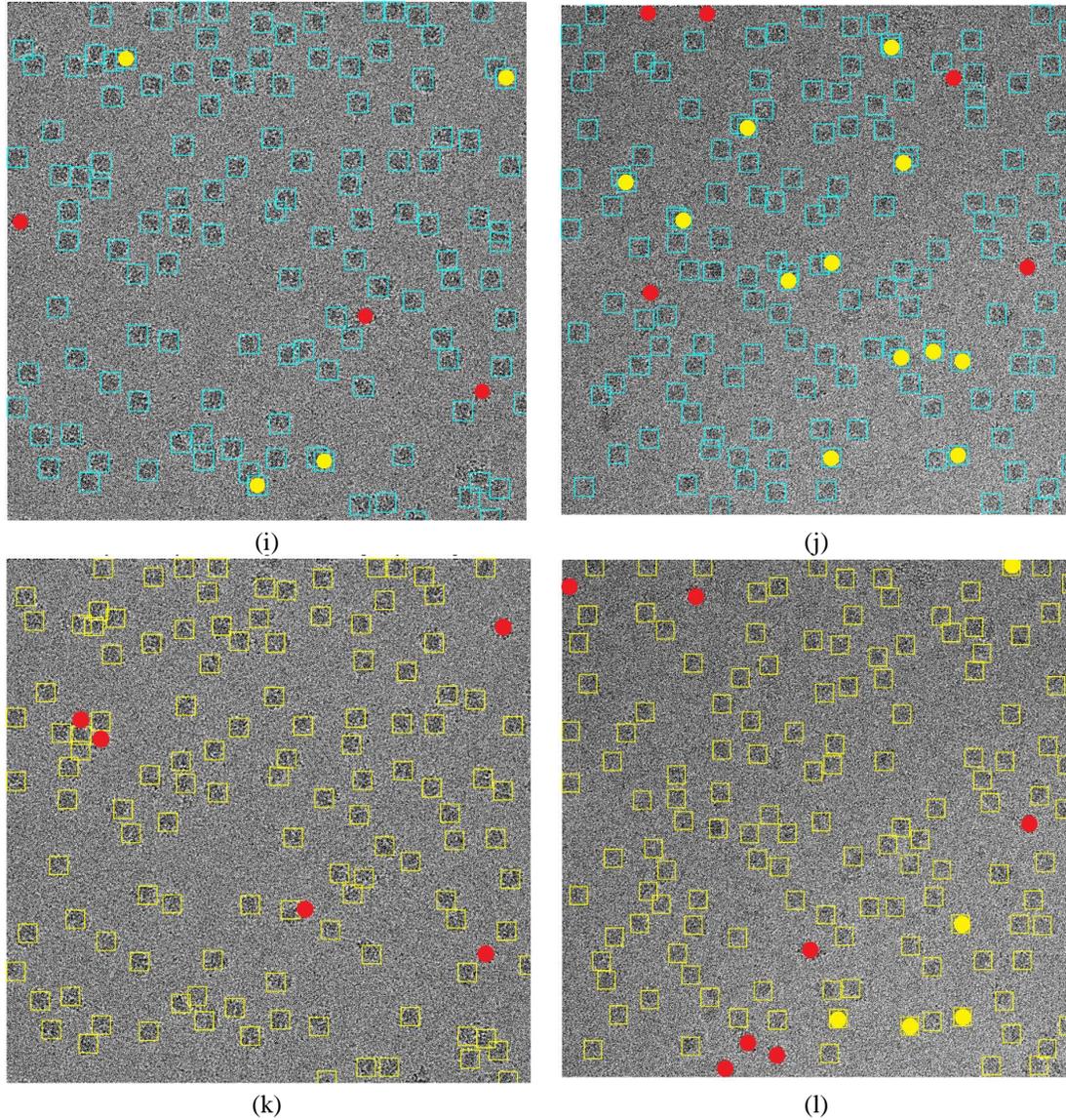


Figure 4.14: The results of the fully automated single particle picking in cryo-EM images by the base and super clustering methods. Red dots denote the missing particles not detected (false negatives) and yellow dots the false positives. (a) the particle picking of IBC. (b) particle picking of IBC on extremely low-SNR cryo-EM image. (c) the particle picking of SP-IBC. (d) the single particle picking of SP-IBC on extremely low-SNR cryo-EM image. (e) particle picking of k-means. (f) the particle picking of k-means clustering algorithm on extremely low-SNR cryo-EM image. (g) the particle picking of SP-K-means. (h) the particle picking of SP-K-means on extremely low-SNR cryo-EM image. (i) the particle picking of FCM. (j) the particle picking of FCM on extremely low-SNR cryo-EM image. (k) the particle picking of SP-FCM. (l) the particle picking of SP-FCM on extremely low-SNR cryo-EM image.

Table 4.2 reports the recall, precision, accuracy, F1 score, and the running time of

the base single particle picking methods (IBC, K-means, and FCM). Table 4.3 shows the results of the super clustering methods (SP-IBC, SP-K-means, and SP-FCM). Generally, speaking, the super clustering methods achieve the better performance than their corresponding base methods according to almost all the metrics. SP-K-means clustering achieves a higher accuracy of 95.48% than 94.08% and 88.98% of SP-FCM and SP-IBC respectively. SP-IBC runs substantially faster the other methods and all the three super clustering methods are fully automated.

Table 4.2: The results of the particle picking using the base clustering algorithms (k-means, FCM, and IBC).

Measures	IBC	k-means	FCM
Sensitivity/Recall (%)	82.42	97.14	97.14
Precision (%)	87.07	94.50	94.50
Accuracy (%)	76.95	91.98	91.98
F1 Score (%)	84.68	95.80	95.80
Time consuming (sec.)	16.77	113.55	412.86
Automation	Full Automated	Manually selection of clusters	Manually selection of clusters

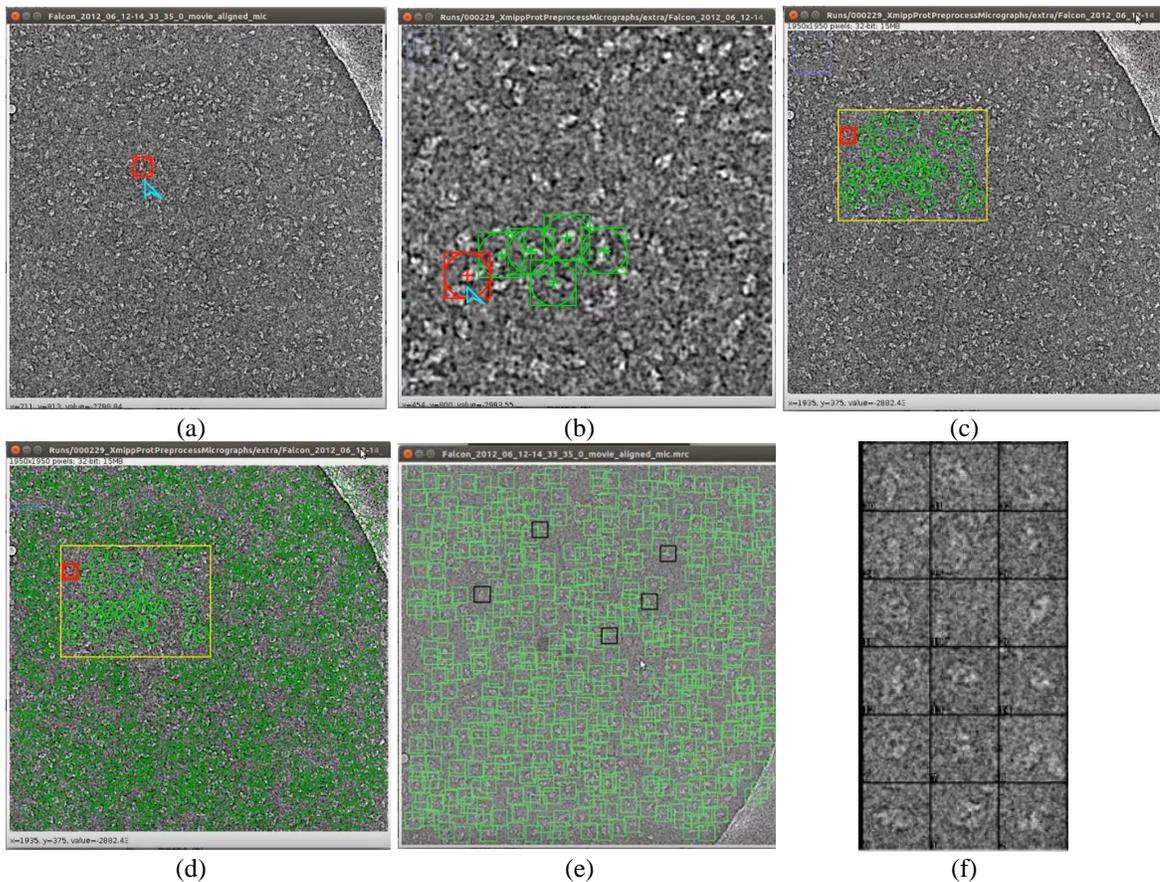
Table 4.3: The results of the super clustering methods (SP-K-means, SP-FCM, and SP-IBC)

Measures	SP-IBC	SP-k-means	SP-FCM
Sensitivity/Recall (%)	92.44	97.50	95.71
Precision (%)	96.62	97.86	98.19
Accuracy (%)	88.98	95.48	94.08
F1 Score (%)	94.48	97.68	96.93
Time consuming (sec.)	11.09	27.31	353.28
Clustering Approach	Full Automated	Full Automated	Full Automated

4.4.4 Comparison With other Particle Picking Software

We compare SuperCryoEMPicker with two external methods: Scipion [125] and EMAN2 [121] in terms of computational efficiency, detection quality and automation. Both Scipion [125] and EMAN2 [121] needed a reference set of particles to be selected manually (Figure

4.15(a for Scipion) and Figure 15(e) for EMAN2), which was used to train the methods to pick more particles (Figure 4.15(d) using Scipion and Figure 4.15(g) and (h) using EMAN2). Using of the arbitrarily manually selected particles resulted in most of the true particles being selected (Figure 4.15(c) using Scipion and (g) using EMAN2). However, some false positives likely corresponding to thick ice were also incorrectly selected (Figures 4.5(d) using Scipion and (h) using EMAN2). Increasing the number of the manually selected particles can reduce the number of false positives at the expense of increasing the number of false negatives (Figures 4.15(h) for EMAN2). In comparison, SuperCryoEMPicker successfully captured all the true particles on the images without using any manually selected samples for training (Figures 4.15(i-l)).



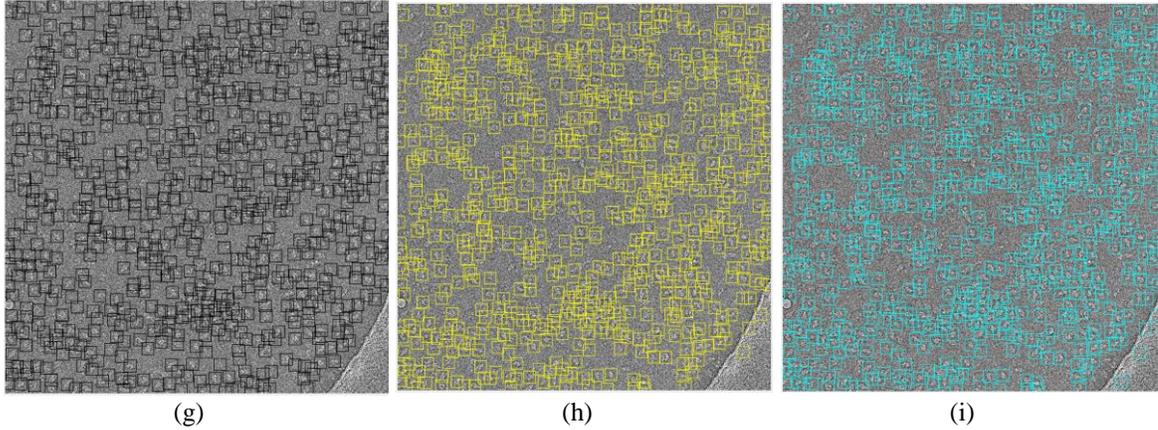


Figure 4.15: Particle picking using EMAN2, Scipion and SuperCryoEMPicker. (a) a manually selected reference particle of the Beta-galactosidase image for Scipion. (b) the zoom-in view of some manually selected reference particles for the Beta-galactosidase image for Scipion. (c) the final reference particles of Beta-galactosidase manually selected for Scipion. (d) all the particle picking results of Scipion trained on 40 manually reference particles on the image of the Beta-galactosidase. (e) EMAN2 autopicking result based on different manually training samples selection in the first tested image of the Beta-galactosidase dataset. (f) the manually selected reference particles of Beta-galactosidase for EMAN2. (g) the particle picking results of SuperCryoEMPicker based on the SP-IBC clustering. (h) the particle picking results of SuperCryoEMPicker based on the SP-K-means clustering. (i) the particle picking results of SuperCryoEMPicker based on the SP-FCM clustering.

Quantitative assessment of the comparison is shown in Figure 16 and Table 4. Figure 16 (a) and (b) show a micrograph from the Beta-galactosidase dataset [120] after the particle picking using EMAN2 [121] using 5 particle references only are selected (more references are selected the more time consuming and more accurate result will get) and our super clustering approach. Figure 16 (a) show the particle picking performance results using EMAN2 [121]. In terms of evaluating each particle's picking tool in addition to our fully automated particle picking approach, three criteria are selected to label and evaluate the particles picking performance results. True Positive (TP) picking where the correct particles are marked by the yellow circles. False Negative (FN) picking where the missed particles are marked by red circles. False Positive (FP) picking where the incorrectly picked particles are marked by blue circles. Figure 16 (b) shows the same criteria of the particle

picking results using the super clustering approach.

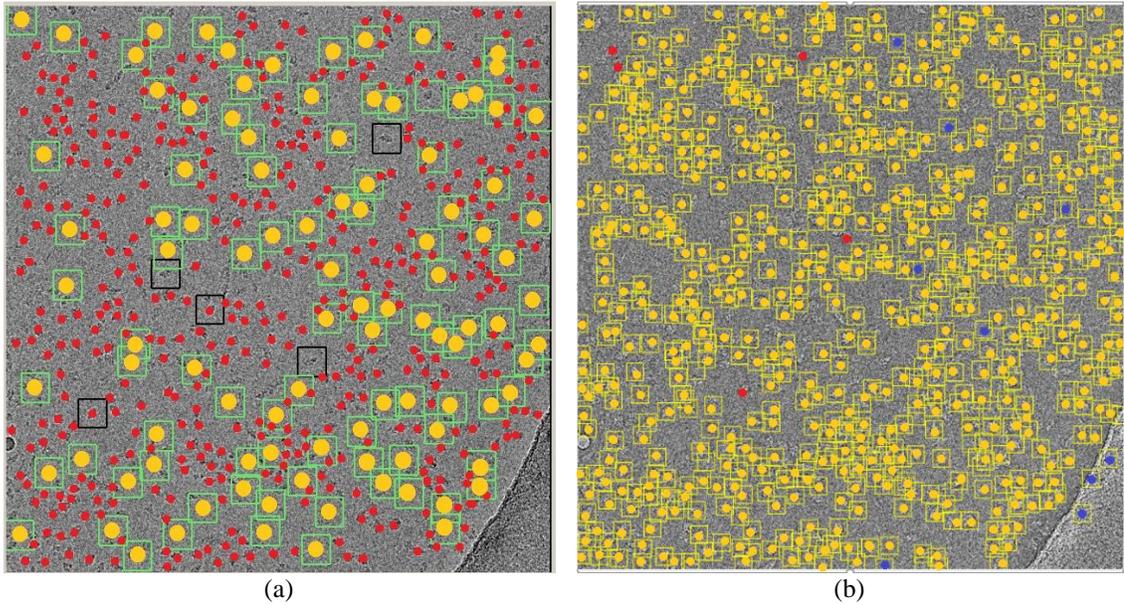


Figure 4.16: Evaluation of particle picking using EMAN2 and Super Cluster approach for Fully Automated single Particle Picking. (c) The particle picking results of Beta-galactosidase image [30] using EMAN2 [31]. (b) The particle picking results of Beta-galactosidase image [30] using super Clustering Approach for Fully Automated Single Particle Picking, the particles are labeled as follows: yellow, True Positive (TP); red, False Negative (FN), blue, False

Table 4.4 illustrates the statistical evaluation of the performance results based on the TP, FN, FP for each single particle picking algorithm, as well as the particle shape class and total number of the particles (ground truth) in each image. Note that the super Cluster Approach for Fully automated single Particle Picking performed better in detecting the shapes by achieving 99.13% sensitivity, 98.45% precision, and 97.61% accuracy.

Table 4.4: Statistical evaluation super clustering approach for Fully Automated Single Particle Picking and EMAN2 [31] performance using Beta-galactosidase image [30]. The table reports TP: True Positive picking results where the correct particles are picked, FN: False Negative picking results where some good particles are missed, FP: False Positive picking results where the incorrect particles (other objects such as background or artificial objects) are picked as particles.

Evaluation Metric	Our Approach	EMAN2
Total Particles Number	579	579

True Positive (TP)	574	101
False Negative (FN)	5	478
False Positive (FP)	9	0
Sensitivity/Recall (%)	0.99136442	0.174439
Precision (%)	0.98456261	1
Accuracy (%)	0.97619048	0.174439
F1 Score (%)	0.98795181	0.297059

4.5 Conclusions

In this chapter, we designed SuperCryoEMPicker - a fully automated super particle clustering method for picking particles of complex and irregular shape in cryo-EM images. SuperCryoEMPicker based on the super clustering methods are more accurate and run faster than the particle picking based on the base clustering methods. It is also more accurate than two external semi-automated particle picking methods that require users to manually picking some reference particles for training. Therefore, SuperCryoEMPicker is a useful and reliable tool for automated single particle picking in cryo-EM images.

Chapter 5

DeepCryoPicker: Fully Automated Deep Neural Network for Single Particle Picking in cryo-EM

5.1 Introduction

Micrographs (cryo-EM) have been widely used in the determination and understanding of the three-dimensional (3D) structures of macromolecules and proteins. Thousands of single particle images are extracted by researchers via two-dimensional (2D) cryo-electron microscopy and can be used to build reliable high-resolution (3D) reconstructions. This method has gained recent popularity in structural biology. However, because of the wide variety of different particle shapes found in micrographs, and the extremely high signal-to-noise ratio (SNR) of micrographs, single particle image picking still presents significant challenges and acquiring a sufficient quantity of high-quality particles requires excessive human labor. We propose a fully automated approach for single particle picking based on

two models. The first model is called the “fully automated training particles-selection and variety of training dataset generation based unsupervised learning approach”. The second model is the “fully automated single particle picking based on deep neural (classification) network”. The experimental results indicate that the DeepCryoPicker compares favorably with other semi-automated methods such as DeepEM, DeepPicker and RELION.

5.2 Background

In order to build a reliable high-resolution (3D) reconstruction structural biologists must extract hundreds of thousands of single particle images from two-dimensional (2D) cryo-electron microscopy [126] [127]. The use of high-energy electrons can result in radiation damage to specimens during imaging and result in extremely noisy micrographs, consequently a limited electron dose is preferred [128] [129]. The signal-to-noise-ratio (SNR) of original (2D) micrographs tends to be very low, with noise from a variety of causes including low contrast, particle overlap, ice contamination and amorphous carbon [130]. Hence, the task of single particle picking still presents major challenges.

Over the past decade many different computational methods have been proposed for the automated and semi-automated single particle picking tasks. Most of these methods are based on different techniques such as template-based matching, edge detection, feature extraction, and conversional computational vision [128]. Recently, Deep Learning has progressively grown in the field of machine learning. Many Deep Learning algorithms from the field of computer vision use convolutional techniques to extract features from big data via layers in neural networks [131]. Furthermore, deep learning appears to be a suitable approach for cryo-EM processing as the size of the micrographs (2D cryo-EM) data continually increases while the SNR of micrographs remains low [128]. Recently, six

deep learning-based approaches to particle picking have been proposed. EMAN2.21 [121] (particle picking with convolution neural network [132], DeepEM [133], DeepPicker [133], FasetParticlePicker [134], RELION [135], and PIXER [128]).

Hence, we propose a fully automated deep neural network for single particle picking based on fully automated training particle-selection using unsupervised learning algorithms. First, we design a fully automated training particle-selection based on unsupervised learning algorithms using two clustering approaches (regular clustering algorithm using the Intensity-Based Clustering IBC) [136], and super clustering algorithms using the super k-means). Both have been previously proposed in our two fully automated particle picking models (AutoCryoPicker [136] and SuperCryoPicker [137]). Second, to accommodate the low-SNR cryo-EM images, we have designed a general framework of micrograph preprocessing that has been used in both our last two models [136] [137]. We also provide a set of advanced preprocessing tools to improve the quality of the low-SNR micrographs.

Those tools are tested on different cryo-EM datasets such as KLH [138], Apoferritin [139], Ribosome [140], and Beta-galactosidase [141]. Third, to solve the considerable number of false-positive (FP) particle detection images, we use Non-Maximum Suppression (NMS) [142] during the testing phase in order to reduce the number of false-positive particle detections. In the occurrence of multiple bounding box predictions around the same particle object, NMS will retain only one box, thus reducing the number of false-positive detections in the final results.

5.3 Methods

DeepCryoPicker consists of two models: (1) Model 1: fully automated training particles-

selection based on unsupervised learning; (2) Model 2: fully automated single particle picking based on deep classification network. The orange rectangle marks the first part of the fully automated approach “fully training particles-section and dataset generation” while the blue rectangle marks the second part “fully automated single particles picking”. The rest, green and gray rectangles mark the first and second stage of the preprocessing step.

5.3.1 An Overview of the DeepCryoPicker Procedure

DeepCryoPicker is designed for fully automated single particle picking in cryo-EM. Our framework contains two components: The first is a training particle-selection algorithm based on an unsupervised-learning (shown on the right side of Figure 5.1). The second is a single particle picking which utilizes supervised deep learning (shown on the left side of Figure 5.1). The first model “Automated training particles-selection and data generation based unsupervised learning scheme” has two sections: automated training particles picking, and automated training dataset generation. The first section of the automated training particles-selection is based on many steps.

First, the micrographs images are pre-processed using a set of advanced image processing tools to enhance and increase the quality of the micrographs. Second, each cryo-EM image is clustered using two different unsupervised learning clustering algorithms and then each clustered image is cleaned and used to detect and isolate each particle. Then, some irrelevant objects are removed. The second section of the automated training particles-selection is based on automatically evaluating each isolated particle sample and classifying it as a “good” or “bad” training sample.

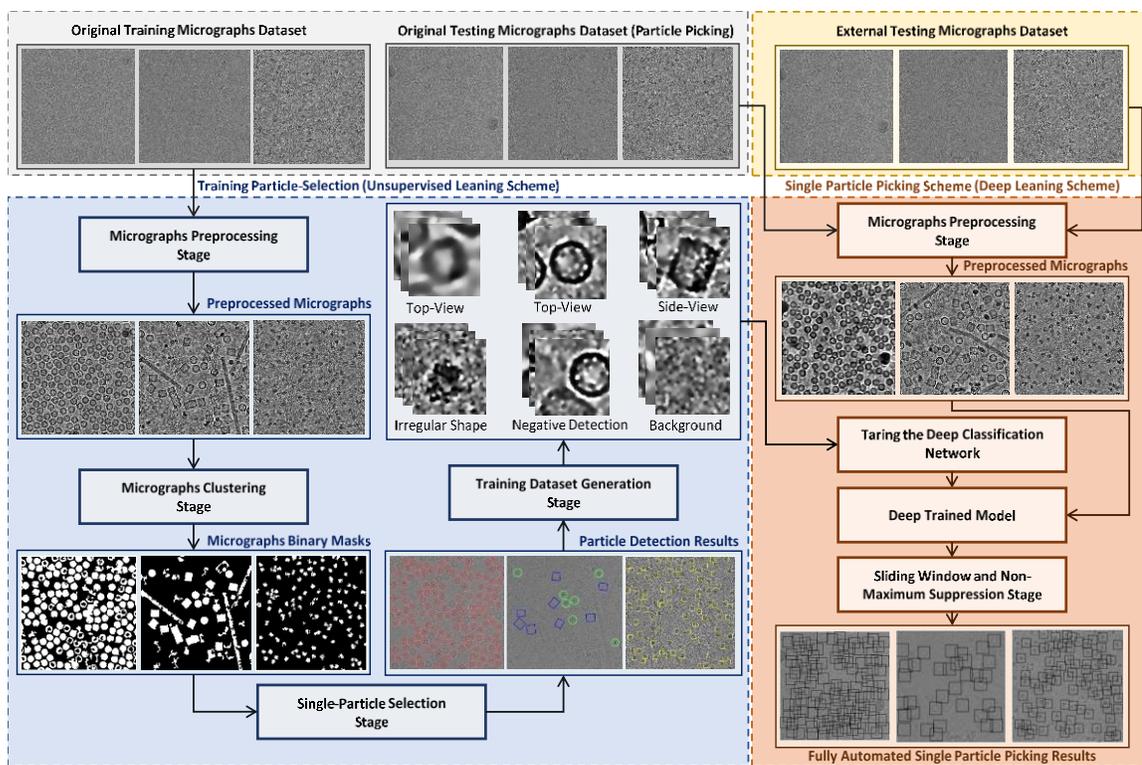


Figure 5.1: The general workflow of the training particle-selection based unsupervised scheme and single particle picking based on deep learning scheme. The gray part of the workflow shows the micrographs data collection. The blue part of the workflow shows the fully automated training particles-selection using clustering algorithms. The red part of the workflow shows the general flow of the single particle picking using deep classification network. The yellow part of the workflow shows the external testing part of the DeepCryoPicker

The second model we propose is the fully automated single particle picking. This model is based on a deep learning scheme which has many steps. The first step is designing and training a deep convolutional neural network using the training dataset that has been automatically generated using the first model of our framework. In the second step, the trained model is used to test every micrograph after pre-processing them using the same preprocessing stage that is used early on the training dataset. In the single particle picking model, two different micrograph testing datasets are used.

5.3.2 Model 1: Fully Automated Training Particles-Selection Based Unsupervised Learning Approach

In the first model, we develop a fully automated approach for training particles-selection and training dataset expanding (generation) based on the unsupervised learning scheme by using different image clustering algorithms. This model consists of two stages: (1) Stage 1: fully automated training particles-selection; (2) Stage 2: full automated perfect “good” training particles-selection and labelled training dataset generation.

5.3.3 Stage 1: Fully Automated Training Particles-Selection

The first stage of the training particles-selection model is to full automatically select all possible particles in each micrograph in the training dataset. Two different fully automated single particle picking approaches based on unsupervised learning, that have been proposed in our last two models (AutoCryoPicker [136] and SuperCryoPicker [137]), are used in this stage. AutoCryoPicker [136] and SuperCryoPicker [127] used the same preprocessing procedures to increase the SNR and the quality of each micrograph as shown in Figure 5.2 (green and gray rectangles).

More details about the micrographs pre-processing and quality improving using advanced image processing steps are provided in our first model AutoCryoPicker [11]. The results of the preprocessing procedures for Apoferritin [14], KLH [13], Ribosome [15], and Beta-galactosidase [16] images are shown in Figure 5.3.

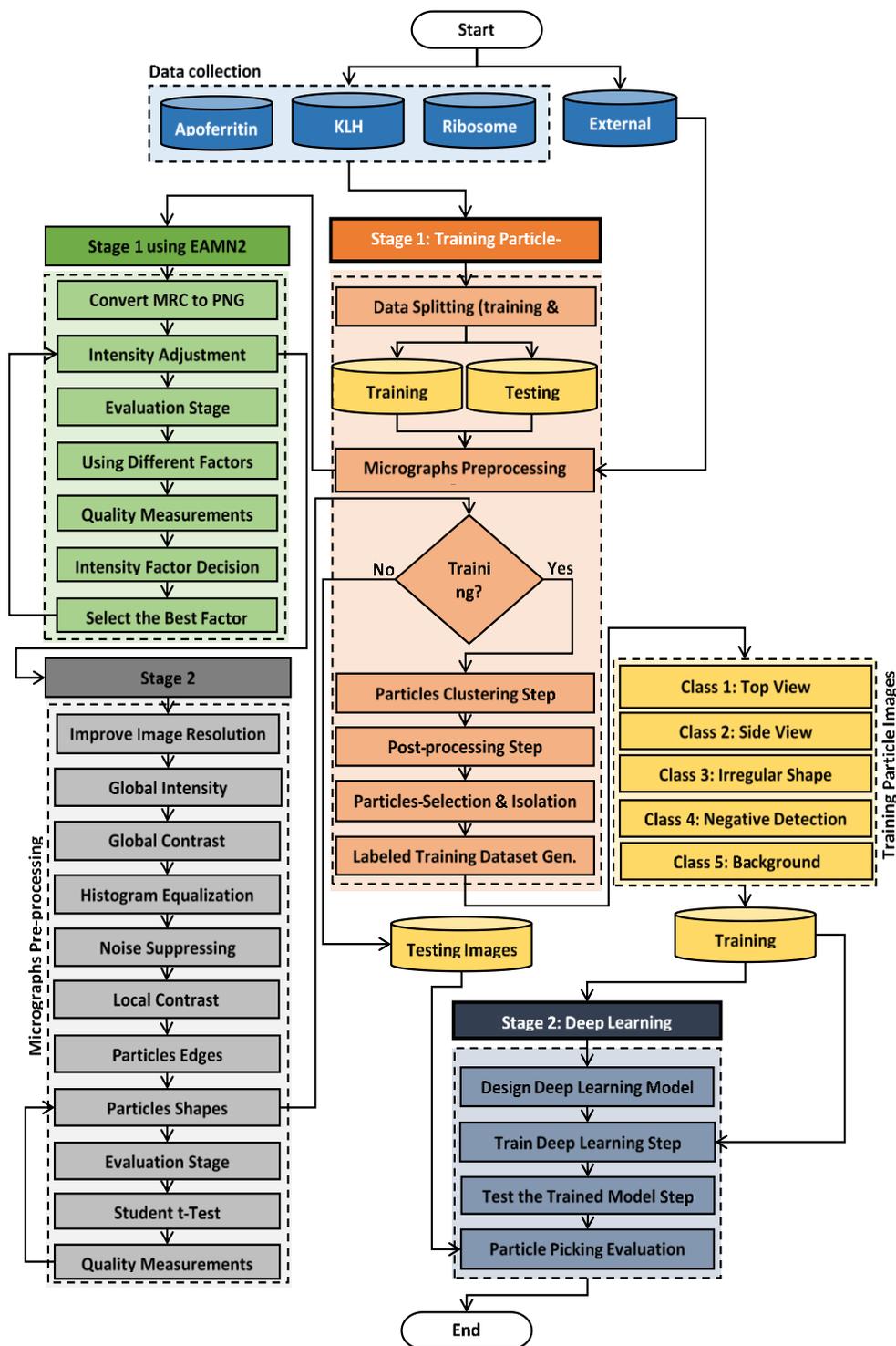
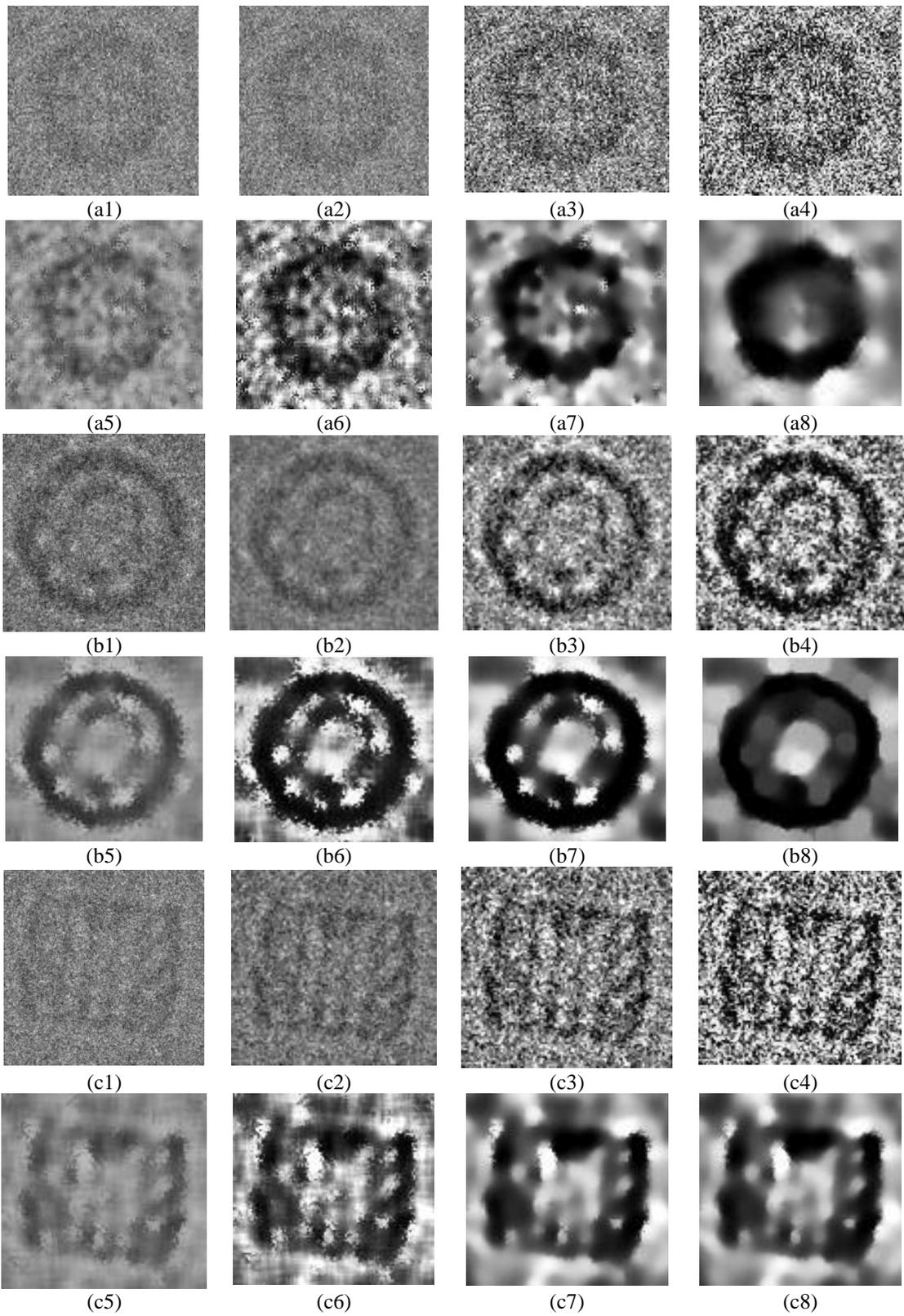


Figure 5.2: DeepCryoPicker architecture. The orange rectangle marks the first part of the fully automated approach “fully training particles-section and dataset generation”. The blue rectangle marks the second part “fully automated single particles picking”. The green and gray rectangles mark the first and second stage of the preprocessing step respectively.



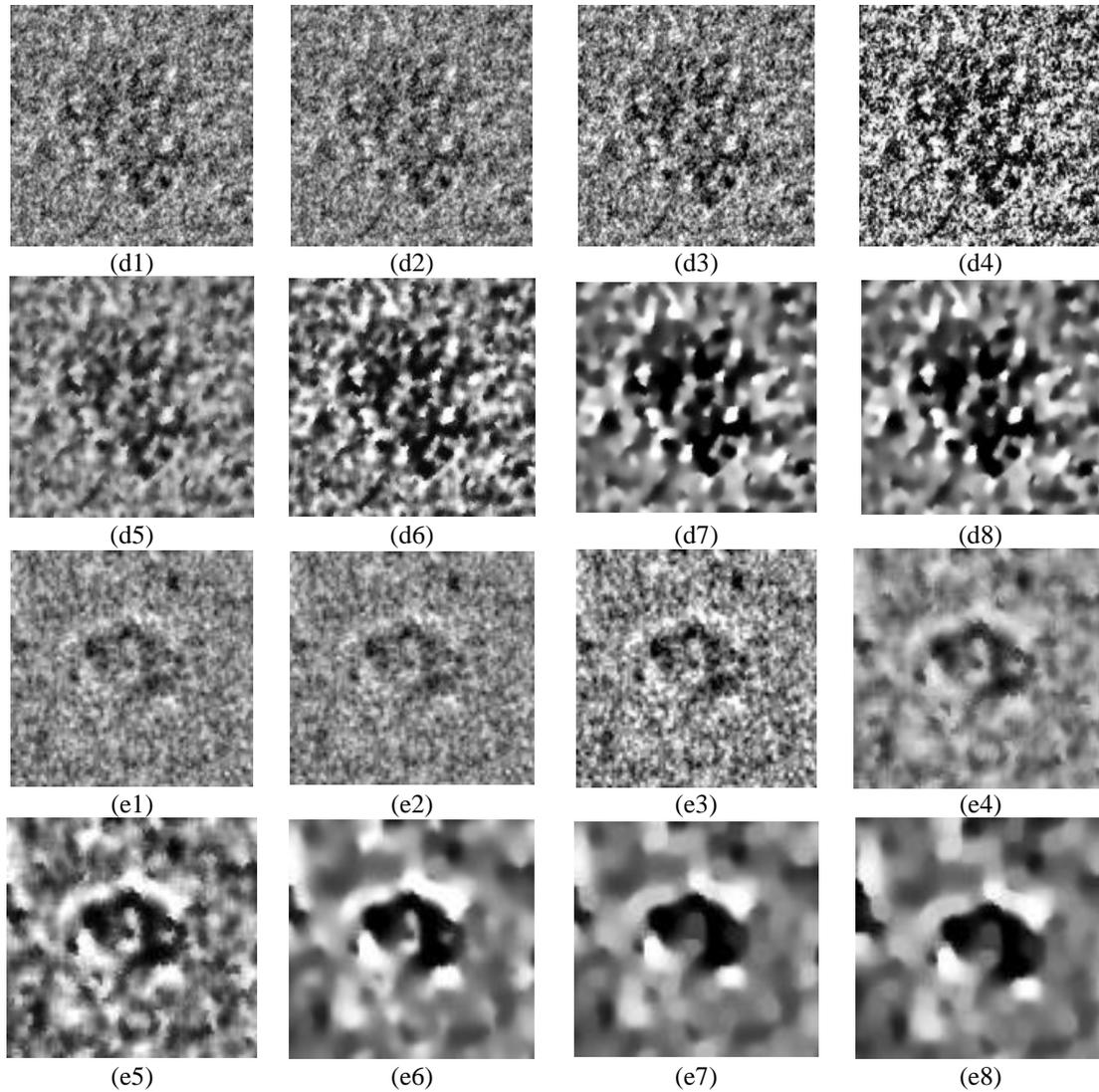


Figure 5.3: Illustration of effects of the cryo-EM image analysis on a zoom-in selected particle region using two different examples from two datasets. (a1), (b1), (c1), (d1), and (e1) original zoom-in particle regions (different shapes) are selected from different micrograph Apoferritin (top-view particle), KLH (top-view), KLH (side-view) Ribosome (irregular shape), and Beta-galactosidase (complex shape) respectively. (a2), (b2), (b2), and (e2) normalized single particle image region. (a3), (b3), (c3), (d3), and (e3) single particle region after applying the contrast enhancement correction (CEC). (a4), (b4), (c4), (d4), and (e4) single particle region after applying the histogram equalization. (a5), (b5), (c5), (d5), and (e5) single particle region after applying image resonation with Wiener filtering. (a6), (b6), (c6), (d6), and (e6) single particle region after applying the contrast-limited adaptive histogram equalization. (a7), (b7), (c7), (d7) and (e7) single particle region after applying image guided filtering. (a8), (b8), (c8), (d8) and (e8) single particle region after applying morphological image operation.

In order to pick each possible particle in the cryo-EM image, a binary mask that

clusters particles are needed. Both the base regular clustering algorithms (Intensity-Based clustering IBC [136]) as well the super clustering algorithm (SP-K-means) that were proposed in our first and second models [136] [137] are used to create the binary mask and pick each single particle in each micrograph. The protein particle shapes in the most cryo-EM datasets are either common shape – circle (top view), and square (side view), or irregular and complex shapes. Our first model AutoCryoPicker [136] has been basically designed to detect and pick the common shape (top and side view) perfectly while our second model SuperCryoEMPicker [137] has been basically designed to detect and pick the irregular and complex particle shapes perfectly.

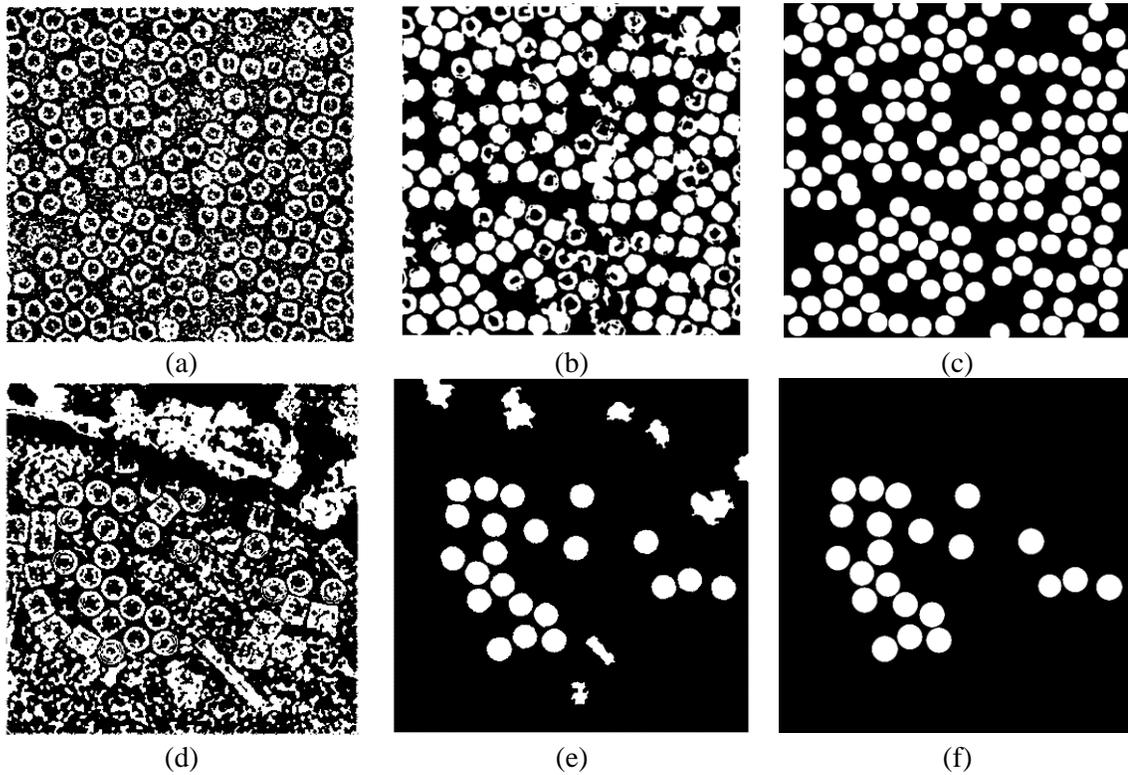
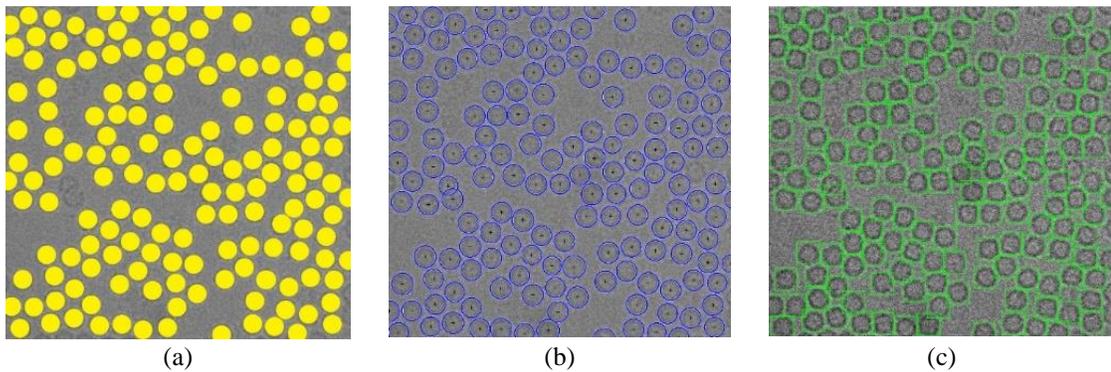


Figure 5.4: Top-view particle clustering using different cryo-EM image clustering results and the Intensity-Based Clustering Algorithm (ICB), (a) the Apoferritin micrograph clustering image (binary mask), (b) the Apoferritin micrograph binary mask image cleaning and small object removal. (c) the perfect circular particle shape generation on the Apoferritin micrograph binary mask image. (d) the KLH micrograph clustering

image (binary mask), (e) the KLH micrograph binary mask image cleaning and small object removal. (f) the perfect circular particle shape generation on the KLH micrograph binary mask image.

Figure 5.4(b) and (d) show an example of different cryo-EM clustering results using the intensity-based clustering method (ICB) with two cryo-EM datasets (Apoferritin [139] and KLH datasets [138]) that have top-view particle shapes. Figure 5.4(b) and (d) show the binary mask image cleaning and small object removal results. Figure 5.4(c) and (f) show the results of the perfect circular particle shape generation. After clustering, the top-view (circular) particles-selection based on the modified Circular Hough Transform algorithm (CHT) [136].

The detection and picking algorithm return the center and radius of each particle illustrated by a '+' sign and a blue circle as is shown in Figure 5.5(b), and (e) based on the clustering results of the ICB clustering algorithm for Apoferritin [139] and KLH [138] datasets to select the circular (top-view) particle shapes. A bounding box is drawn around all selected particle objects in the cryo-EM image (Figure 5.5(c) and (f)).



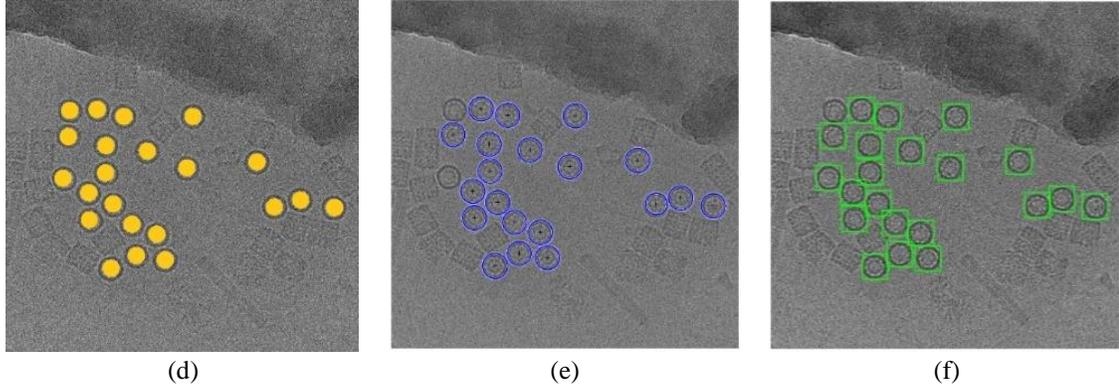


Figure 5.5: Top View (Circular) Particles Detection and Picking Results using Modified Circular Hough Transform (CHT). (a) The Ground truth (particles manually labelled) for the cryo-EM image from the Apoferritin dataset. (b) The center of each particle illustrated by the ‘+’ sign and the radius of each particle by the blue circle around each particle (ICB and Apoferritin dataset). (c) The bounding box for each particle object in the original cryo-EM image (ICB and Apoferritin dataset). (d) the ground truth (particles manually labeled) for the cryo-EM image from the KLH dataset. (e) the center of each particle illustrated by the ‘+’ sign and the radius of each particle by the blue circle (ICB and KLH dataset). (f) the bounding box for each particle in the original cryo-EM image (ICB and KLH dataset).

Perfect “good” Top-View Training Particles-Selection

We develop an additional step called “good top-view (circular) training particles-selection”. The good top-view particle training selection algorithm is described below.

Algorithm 5.1 Good Top-View (Circular) Training Particles-Selection

- 1: **input:** I_p /*particle binary sub-image */
 - 2: **return:** G_p /*good circular particle training samples */
 - 3: **for** $k = 1$ to total number of particles mask **do**
 - 4: $P \leftarrow \text{Trace}[\text{object}]$ /* Determine and extract the boundary particles mask’s pixels list by specifying the row and column coordinates of each point on the object boundary */
 - 5: Construct the output binary sub-image containing only the object circular boundary for each object.
 - 6: **for** $i = 1$ to each edge point **do**
 - 7: Draw a circle with centre (x, y) in the edge point with r where (x, y) is the image pixels with position x , and y , r is the circular radius.
 - 8: Increment all coordinates (x, y) that the perimeter of the circle passes through in the accumulator.
 - 9: Find one or several maxima in the accumulator
-

```

10:      Map the found parameters ( $r, a, b$ ) corresponding to the maxima back to
        the original image, where  $a$ , and  $b$  is the centre of the maxima.
11:      end for
12:      Generate a new particle mask's sub-image that has the perfect circle.
13:      end for
14:      for  $l = 1$  to each new particle mask sub-image do
15:          allAreas  $\leftarrow$  [Area(L(i))] /* Determine the region area of each sub-image's
        connected component (object) using MATLAB function (regionprops('Area'))
        */
16:          allPerimeter  $\leftarrow$  [Perimeter(L(i))] /* Determine the region perimeters of each
        sub-image's connected component (object) using MATLAB function
        (regionprops ('Perimeter')) */
17:          circularities  $\leftarrow$  (object) /* Determine the region circularities of each
        connected component (object) using*/
18:          Average_circularities  $\leftarrow \sum_1 \frac{\text{circularities}}{1}$  /*Calculate the average roundness
        class value */
19:      end for
20:      for  $n=1$  to each new particle mask sub-image do
21:          Determine the region area of each connected component (object) using
        MATLAB function (regionprops('Area'))
22:          Determine the region perimeters of each connected component (object) using
        MATLAB function (regionprops('Perimeter'))
23:          Determine the region circularities of each connected component (object) using
        Equation (1).
24:          If circularities < Max_Allowabl_Area then
25:              RoundObjects  $\leftarrow$  circularities > Average_circularities /* select the
        original particle image as a good training example*/
26:          else
27:              Select the individual particle as a bad training example.
28:          end if
29:      end for

```

This step is based on using the individual binary mask for each particle as shown in Figure 5.6 (d), (f), (h), and (j).

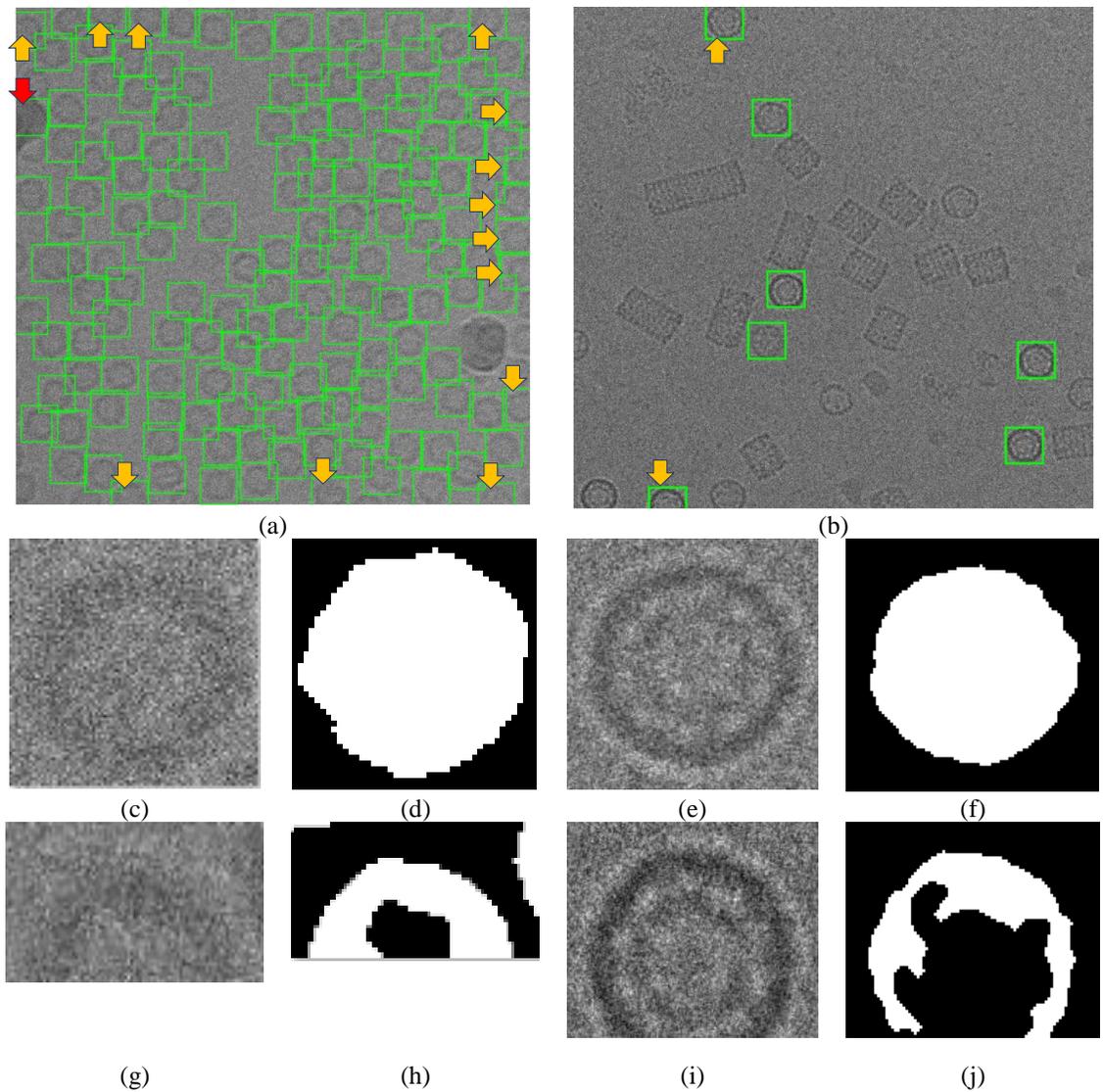


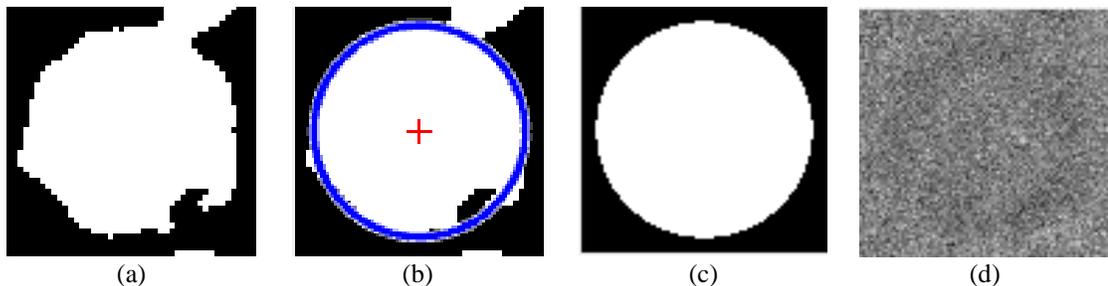
Figure 5.6: Top-view particles-selection results, (a) using cryo-EM micrographs from the Apoferritin [139] dataset, (b) using cryo-EM micrographs from the KLH [138] datasets, (c) Apoferritin good particle example, (d) Apoferritin good binary mask example, (e) KLH good particle example, (f) KLH good binary mask example, (g) Apoferritin bad particle example, (h) Apoferritin bad binary mask example, (i) KLH bad particle example, (j) KLH bad binary mask example.

Then, we use the modified Circular Hough Transform algorithm (CHT) algorithm that was proposed in our first model “AutoCryoPicker” [136] to generate a perfect circle on top of each particle’s mask. Then, we test each individual particle’s mask size and verify that it is a perfect full circle and label it as either a “good example” as a “bad example”.

We test each top-view particle by calculating the average roundness value for the whole top-view (circular) particles. This is determined by computing the area and perimeters using the connected component particle mask’s pixel index list and the circularity based on the Equation (5.1):

$$Circularities = \frac{allPerimeters^2}{4 \times pi \times allAreas} \quad (5.1)$$

where *allAreas* is the area of each selected particle and *allPerimeters* is the cemetery size of each particle. Then, each individual particle (circular) does achieve the average object roundness class is selected as a “good” training example unless is selected as a “bad” training example. Figure 5.7 shows the results of the good top-view training particles selection. Figure 5.7 (a) and (e) show individual top-view particle binary mask from the Apoferritin [139] and KLH [138] datasets. We notice that a perfect circle has been successfully drawn on top of the particle’s binary mask using the modified CHT algorithm as is shown in Figure 5.7 (b) and (f). Figure 5.7 (c) and (g) show the replaced artificial perfect circle binary masks that will used later to test the particles for (Apoferritin [139] and KLH [138]) datasets. Figure 5.7 (d) and (h) show the good Apoferritin [139] and KLH [138] top-view training particles selection. In contrast, Figure 5.7 (i), (l), (m), and (o) show other examples of the top-view particle’s binary masks that the modified CHT has failed to draw perfect circles on top of them. Figure 5.7 (j), (l), (n), and (p) show some bad top-view training particle examples.



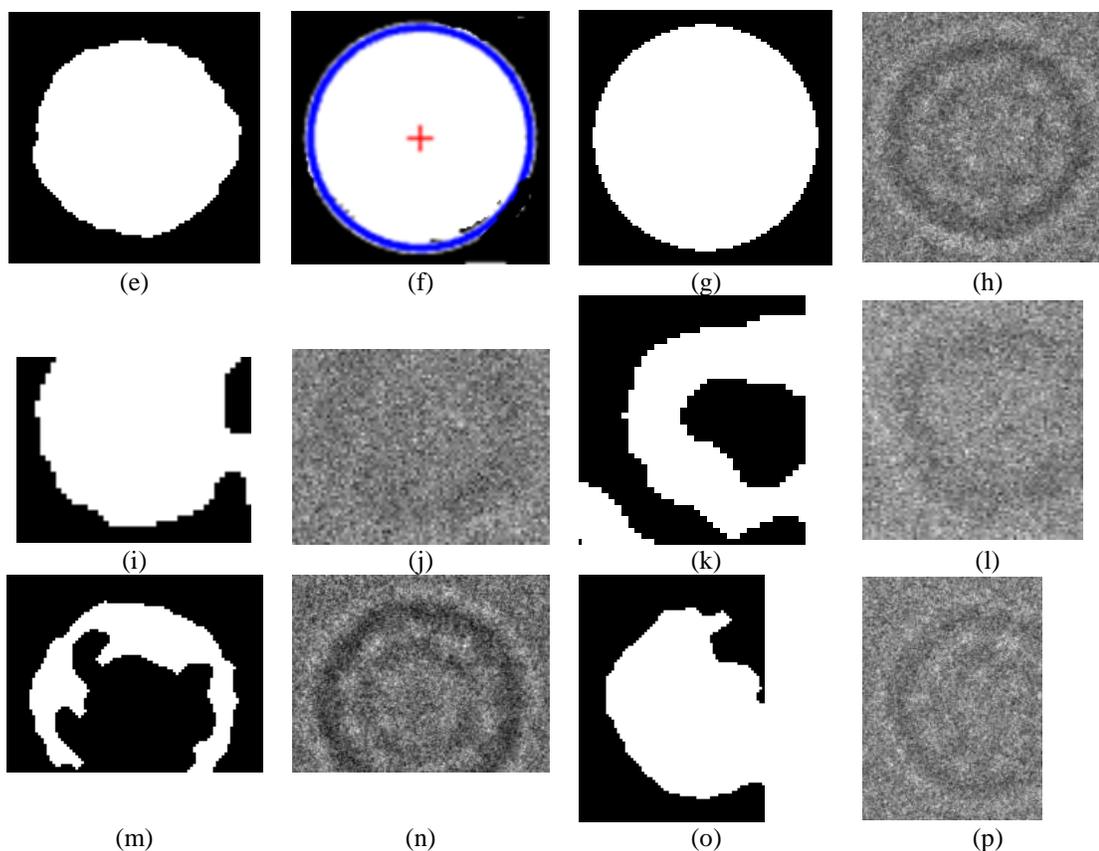


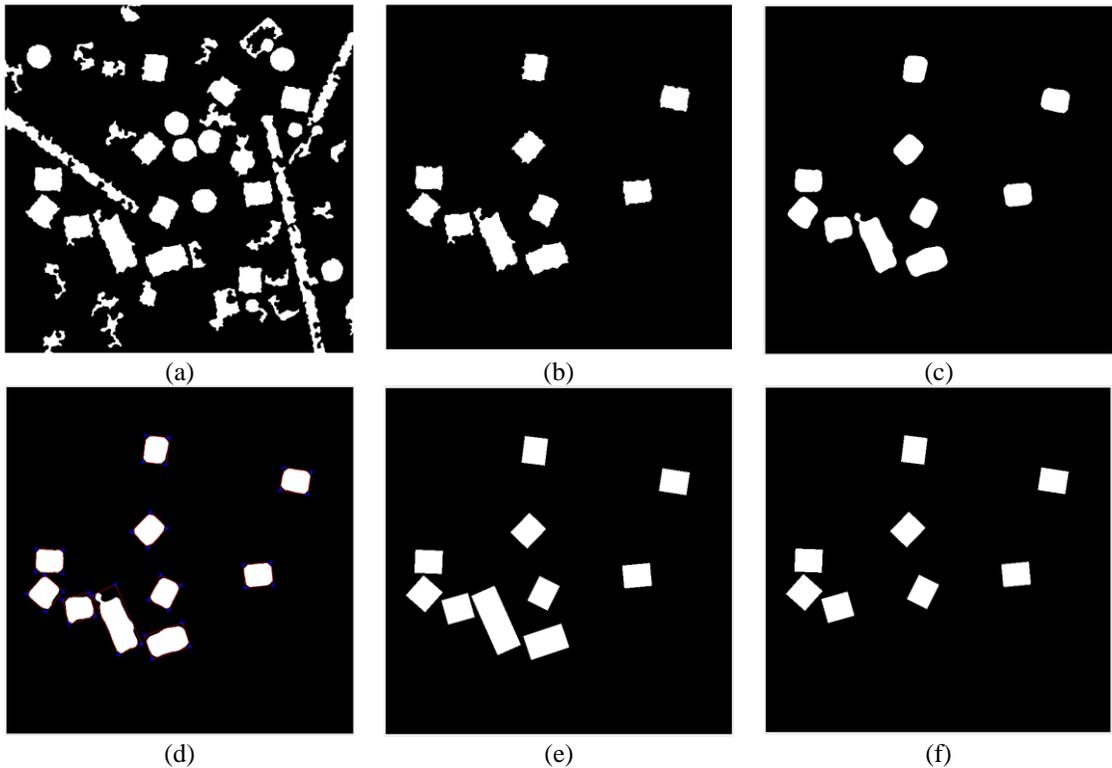
Figure 5.7: Good and bad top-view training particles-selection results. (a) and (e) individual top-view particle binary mask form the Apoferritin [139] and KLH [138] datasets. (b) and (f) CHT perfect circle on top of the particle’s binary masks. (c) and (g) the replaced artificial perfect circle binary after the CHT for the Apoferritin [139] and KLH [138] particle’s binary mask respectively. (d) and (h) the good Apoferritin [139] and KLH [138] top-view training particles selection. (i), (l), (m), and (o) other examples of the top-view particle’s binary masks that the modified CHT has failed to draw perfect circles on top of them. (j), (l), (n), and (p) bad top-view training particle examples.

Perfect “good” Side-View Training Particles-Selection

For the side-view particles picking, we don’t have issue with the overlapped particles selection since the only perfect side-view (square) particles are selected through the side view (square) training particle shapes-selection in cryo-EM based on using the “overlapped particles removal and perfect side-View particles-selection algorithm” that has been

proposed and used in the AutoCryoPicker model [136]. Figure 5.8 (a) and (g) in the supplemental document show different KLH cryo-EM clustering results using Intensity-Based Clustering Algorithm (ICB). Figure 5.8 (b) and (h) show the KLH cryo-EM clustered images after the circular and non-square object removal. The binary mask images have only the square particle shapes (side view) in the whole cryo-EM images.

Some overlapped particles still exist in the cleaned binary mask as is shown in Figure 5.8 (b). The overlapped particles are removed from the final cleaned masks (See Figure 5.8 (e) and (f)) after applying the overlapped particles removal using Feret diameter measures approach (see Figure 5.8 (d) and (j)). Figure 5.8 (f) and (l) show the same KLH binary mask images after the perfect side-view (square) particles shape generation on the of the cleaned binary masks.



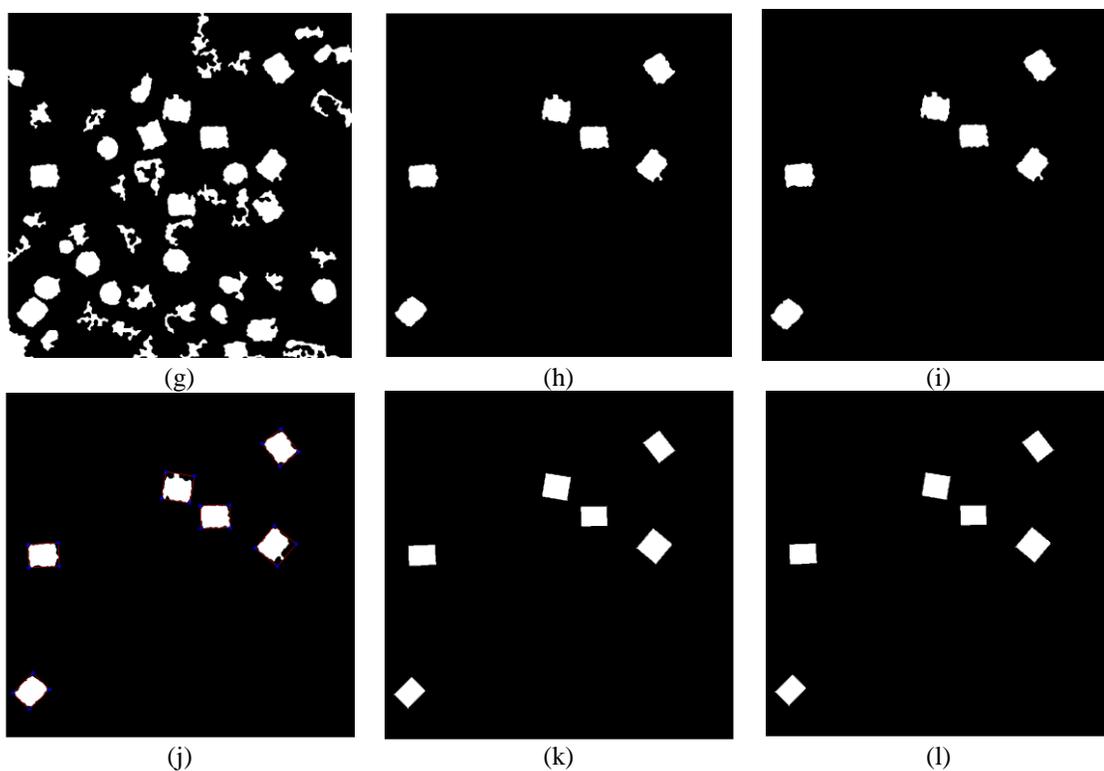
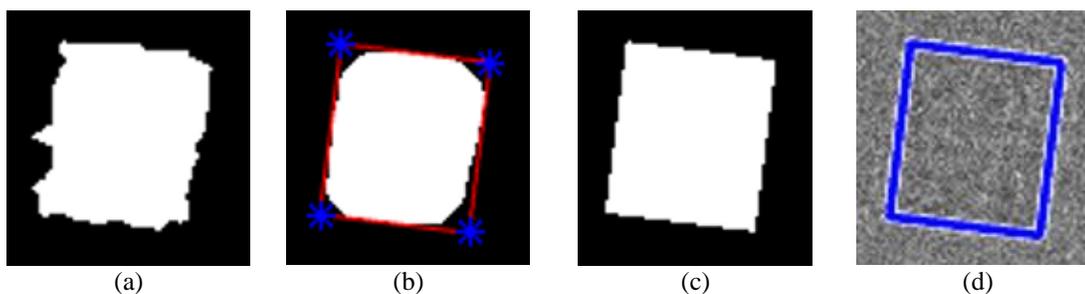


Figure 5.8: Side-view particles clustering using different cryo-EM image and Intensity-Based Clustering Algorithm (ICB), (a) and (g) different KLH micrograph clustering images (binary masks), (b) and (h) KLH micrograph binary mask images after image cleaning and small object removal. (c) and (i) particle objects smoothing. (d) and (j) Feret diameter measures for the particle objects. (e) and (k) perfect side-view (square) particle shapes generation on the top of the binary image of the KLH micrograph. (i) and (l) show the overlapped particles removal and perfect side-view particles-selection results.

Figure 5.8 and 5.9 in the supplemental document show a real-world example of the perfect side-view (square) particle selection. Figure 5.9 (a) and (d) show different cryo-EM micrographs from the KLH dataset [138].



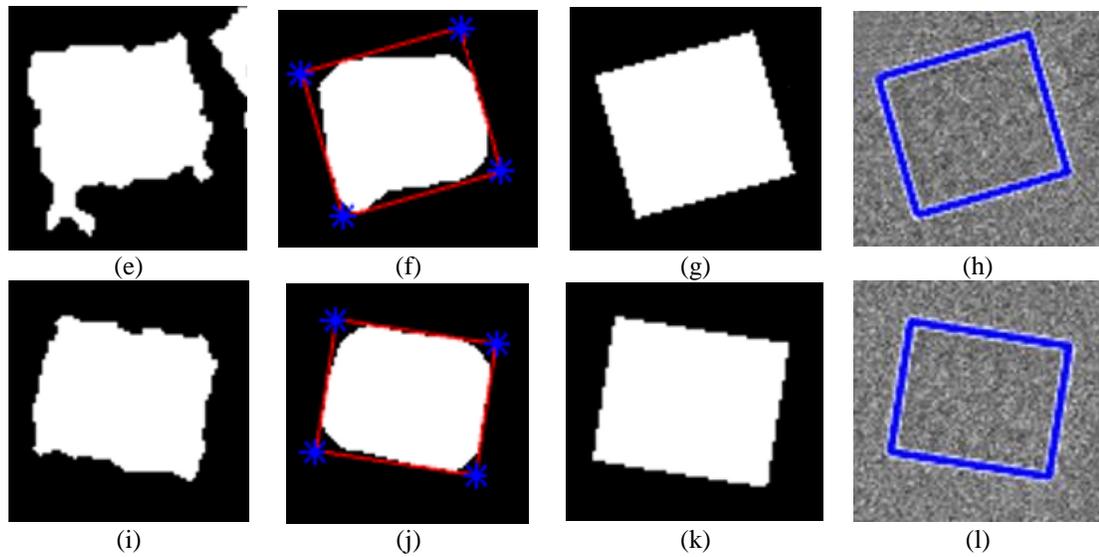
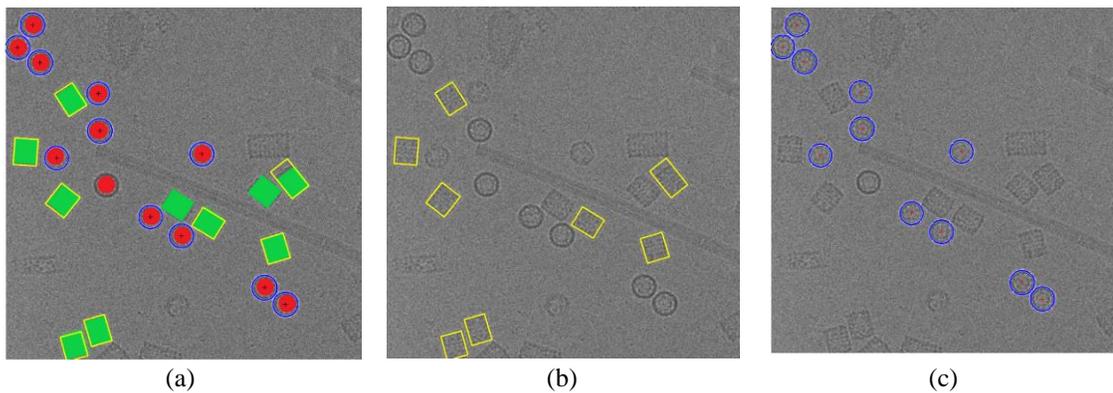


Figure 5.9: Good and bad top-view training particles-selection results. (a) and (e) individual top-view particle binary mask form the Apoferritin [139] and KLH [138] datasets. (b) and (f) CHT perfect circle on top of the particle's binary masks. (c) and (g) the replaced artificial perfect circle binary after the CHT for the Apoferritin [139] and KLH [138] particle's binary mask respectively. (d) and (h) the good Apoferritin [139] and KLH [138] top-view training particles selection. (i), (l), (m), and (o) other examples of the top-view particle's binary masks that the modified CHT has failed to draw perfect circles on top of them. (j), (l), (n), and (p) bad top-view training particle examples.

Figure 5.10 (b) and (e) show the final results of side-view particles-selection using different micrographs form the KLH dataset based [138] ICB clustering, and perfect square (side view) particle shapes detection using Feret object diameter. Figure 5.9 (c) and (f) show also the top-view particles-selection results based modified ICB clustering, and modified CHT [136].



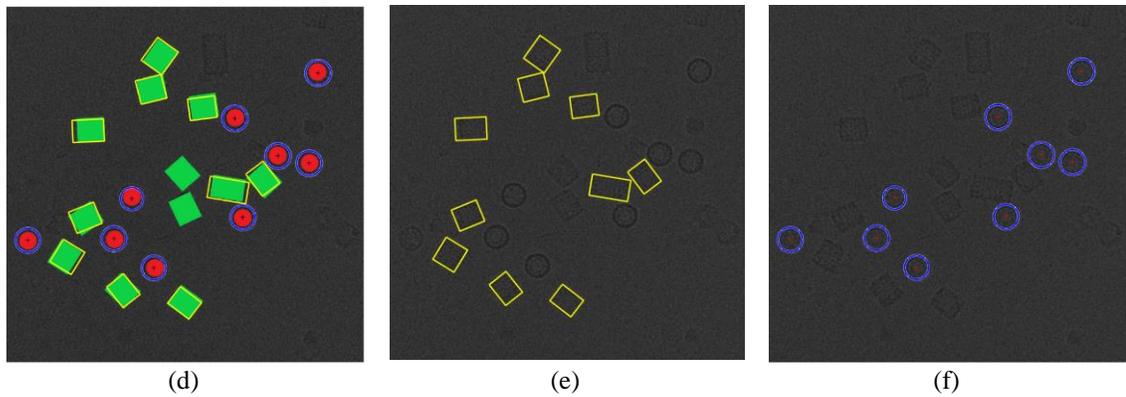


Figure 5.10: Side-View (square and circular) particles-selection. (a) and (b) The Ground truth (particles manually labelled) for the different cryo-EM images from the KLH dataset [13]. (b) and (e) side-view particles-selection results using IBC clustering and perfect side-view (square) particles-selection algorithm. (c) and (f) top-view particles-selection results using modified CHT algorithm (the red ‘+’ sign is the center of each particle, and blue circles around each particle are the radius of each particle by the blue circle around each particle).

Perfect “good” Irregular and Complex Training Particles-Selection

This step is also based on using the individual binary mask for each complex and irregular particle as shown in Figure 5.11 (b), (d), (f), and (h). Then, we test each individual particle’s mask size and determine if it is a usable training sample. We develop a “good irregular (complex) training particles-selection” algorithm (see Algorithm below) to test each irregular binary particle, by calculating the average area for the whole particle binary masks which is determined by computing the total number of white pixels in each particle using the connected component particle mask’s pixel index list.

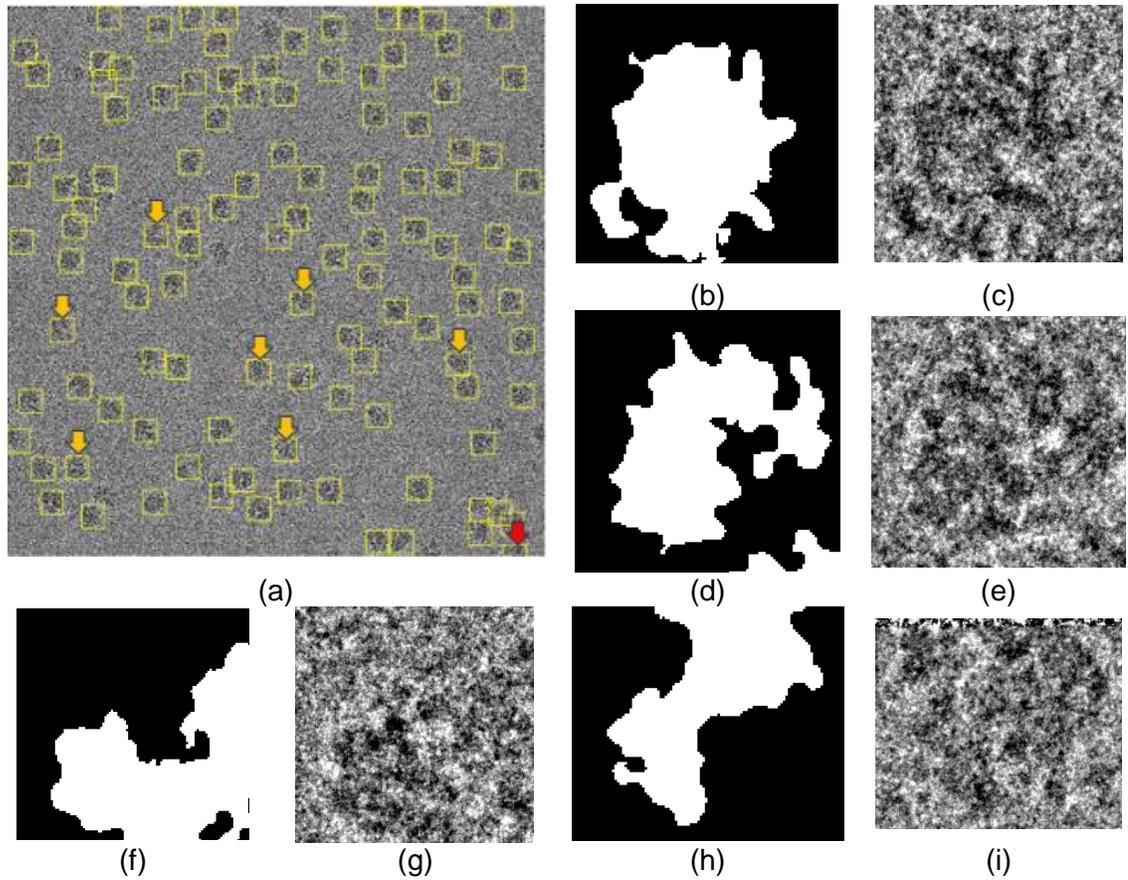


Figure 5.11: Irregular (complex) particles-selection results, (a) particle picking results using cryo-EM micrographs from the Ribosome [140] [141] dataset. (b) and (d) good particle binary mask examples, (c) (e) good training particle examples, (f) and (h) bad binary mask examples, (g) and (i) bad particle examples

Algorithm 5.2 Good Irregular (Complex) Training Particles-Selection

- 1: **input:** I_p /*particle binary sub-image */
 - 2: **return:** G_p /*good irregular particle training samples */
 - 3: **for** $k = 1$ to total number of particles mask **do**
 - 4: allAreas \leftarrow [Area(L(i))] /* Determine the region area of each sub-image's connected component (object) using MATLAB function (regionprops('Area')) */
 - 5: Average_Area $\leftarrow \sum_1 \frac{\text{allAreas}}{\text{number of particles}}$ /*Calculate the average area */
 - 6: **end for**
 - 7: **for** $n=1$ to each new particle mask sub-image **do**
 - 8: Determine the region area of each connected component (object) using MATLAB function (regionprops('Area'))
 - 9: **If** Area < Average_Area **then**
-

```

10:     Perfect_Objects_list ← Object[n] /* select the original particle image as a
      good training example*/
11:     else
12:         Select the individual particle as a bad training example.
13:     end if
14: end for

```

Then the average area as is shown in Equation (5.2) where l is the total number of particles in each cryo-EM image.

$$Area = \frac{\sum_l allAreas}{Total\ number\ of\ particles} \quad (5.2)$$

5.3.4 Model 2: Fully Automated Single Particle Picking Based on Deep Classification Network

The second model of the DeepCryoPicker is the particle picking based deep network as is shown in Figure 5.12. Model 2 of the DeepCryoPicker consists of many layers such as input layer, pre-processing layer, in addition to some convolutional layers, sub-sampling layers, two fully connected layers, and one output layer. The main architecture of the DeepCryoPicker contains in total thirteen layer. The first and second layers (input and the pre-processing layer) come from the first model of the DeepCryoPicker. The input layer takes the particles that have been already picked through the first model of the DeepCryoPicker. Each particle has been picked based on the preprocessed version of each of the micrographs. The rest are five convolutional layers, three max-pooling (subsampling) layers, two fully connected layers, and one output layer.

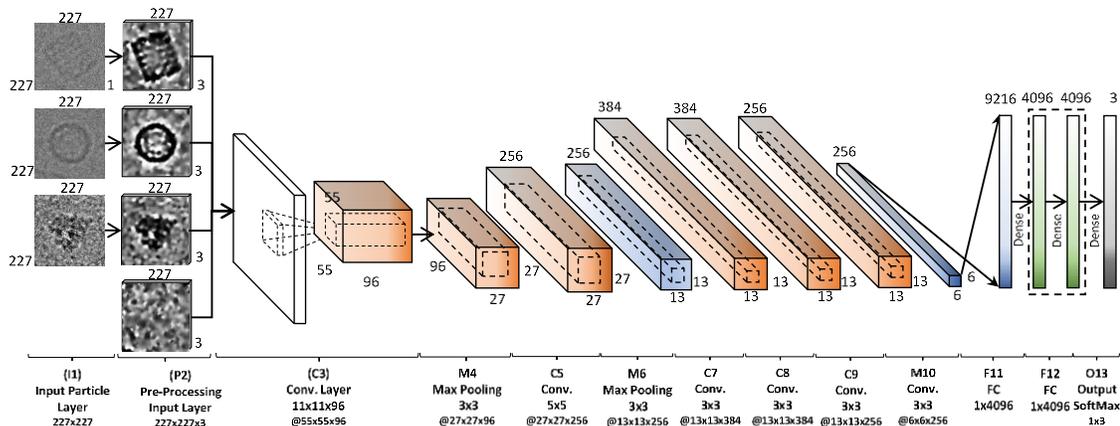


Figure 5.12: The architecture of the deep neural network used in DeepCryoPicker. The convolutional layer and the subsampling layer are abbreviated as C and S, respectively. C3:11x11x96 means that in the third convolutional layer (C3) is comprised of 96 feature maps, each of which has a size of 11×11 , also. C3:@27x27 means that output feature maps dimensions are 27x27 pixels.

In terms of using one deep network structure, we unify the variety of the particle sizes as is shown in Table 5.1 to one fixed size. In this case, after each particle is detected, a bounding box is drawn around each particle object in the cryo-EM image which is used to crop the particle image from the original micrograph.

Table 5.1: Training Particles-Selection Image Size using Good Training Particles-Selection form Apoferritin KLH, Ribosome datasets.

Criteria	Apoferritin Top-View	KLH Top-View	KLH Side-View	Ribosome Side-View
Particle Size	178x178	221x221	221x225	187x187
Size of Micrograph	1240x1240	2048x2048		4096x4096

In this case, we recalculate the bounding box dimension of each detected particle after calculating the center of each box and specifying the fixed size of each (width and height). Then, the input size of the first and second layer (input and the preprocessing) in our DeepCryoPicker structure is 277×277 . The third layer is the convolutional layer using 96 kernels with size 11×11 .

The first convolutional layer (third layer in the structure) produces 96 feature maps with dimensions 55×55 . The fourth layer is the max-pooling layer with kernel size 3×3 and the feature maps output dimension is 27×27 . The fifth layer is another convolutional layer using 256 kernels with size 5×5 . The fifth layer (convolutional) produces 256 feature maps with dimensions 27×27 . The sixth layer is another max-pooling layer with kernel size 3×3 and the feature maps dimensions output is 13×13 . The seventh, eighth, and ninth layers are convolutional layer using different number of kernels 384, 384, and 256 respectively. We use the same kernel size 3×3 for three convolutional layers. The output feature maps size for the last three convolutional layers 13×13 . The tenth layer is the third max-pooling later with kernel size 3×3 and output dimensions 6×6 . The last two layers are the fully connected layers to the final output (prediction layer) where the particle class is predicted based on the weight's matrix and the activation function. The convolutional and sub-sampling layers which is a core building of the CNN, produce feature maps. The kernels sizes are selected to establish the local connections while they expand through the entire particle image. The learnable kernels are convolved with each feature map from the previous layer. The convolutional layers (in the same convolutional operations) share the same local connective weights $W_{ij}^{[l]}$ based on the previous layer's weights $W_{ij}^{[l-1]}$, in which the feature maps in the current layer $X_j^{[l]}$ are produced based on Equation (5.3) [142]:

$$X_j^{[l]} = \text{sigmoid} \left(\sum_{i \in M_j} X_i^{[l-1]} W_{ij}^{[l]} + B^{[l]} \right) \quad (5.3)$$

where l represents the convolutional layer, W and B is the shared weights and bias, M is extracted feature maps (in the previous layer), j is the output feature maps. Then, the

feature maps are transformed to another layer by a non-linear activation function (sigmoid) as is given in Equation (5.4) [130]:

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (5.4)$$

To reduce the positional over-fitting, the subsampling (max-pooling) layer is designed to subsample the input feature maps by decreasing the actual size and reduce the number of the parameters [142]: The max-pooling (subsampling) after each particular layer is computed based on Equation (5.5) [142]:

$$X_{ij}^{[l]} = \frac{1}{MN} \sum_m^M \sum_n^N X_{iM+m.jN+n}^{[l-1]} \quad (5.5)$$

where I and j are the position of the output feature maps, M and N are the subsampling size. In the training process, the weights and bias are randomly initialized [0-1]. Then, they are updated during the training process. In our model, we used the cross-entropy loss function as the objective function Equation (5.6) [144]:

$$L(w) = \sum_{i=1}^N \sum_{c=1}^C -y_{ic} \log f_c(x_i) + \epsilon \|W\|_2^2 \quad (5.6)$$

where i is the sample number and c is its label, x represents the predicted probability of the class c . N is the total number of training samples, and C is the total number of classes. During the training process, the errors of the objective function is minimized propagating error via the backpropagation algorithm based stochastic gradient decent as follow [144] [145] [146].

$$\omega(l+1) = \omega(l) - \frac{\eta}{N} \sum_{k=1}^N \epsilon_n \frac{\partial \mathcal{E}_n}{\partial} \quad (5.7)$$

where \mathcal{E} is calculated as follow :

$$\mathcal{E}_n = \|t_n - y_n\| \quad (5.8)$$

where t_n is the label of the n^{th} training sample, and y_n is the value of the output layer corresponding to the n^{th} training sample. $\omega(l)$ and $\omega(l + 1)$ represents the training parameter before and after the update of each iteration. The learning rate, η , is initially set to 0.0001.

5.4 Experiments Results

5.4.1 Micrographs Data Collection

We consider three typical protein shapes in micrographs that are collected from different real-world micrographs datasets as is shown in Figure 5.11. The first protein shape is the top-view (circular) protein particles from the Apoferritin dataset [139]. The particle shapes in the whole micrographs images is circular shapes. The simulated top-view (circular) molecule shapes are shown in Figure 5.11 (b) while the real-world protein shape is shown in Figure 5.11 (c). The second protein shape is the side-view (square) protein shapes from the Keyhole Limpet Hemocyanin (KLH) dataset [138]. There are two main types of projection shape views in this dataset: the top view (circular particle shape) and the side view (square particle shape). The simulated top-view (circular) molecule shapes are shown in Figure 5.11 (d), and the simulated side-view (square) molecule shape is shown in Figure 5.11 (f) while the real-world protein shape of the top view is shown in Figure 5.11 (e), and the side view is shown in Figure 5.11 (g). The third protein shapes that is considered in our approach is complex (irregular) protein shapes. The simulated complex and irregular molecule shapes are shown in Figure 5.11 (h) and (j). In this case, we used two datasets Ribosome and Beta-galactosidase micrograph datasets [141]. The real-world irregular and

complex protein shapes are shown in Figure 5.13 (i) and (k).

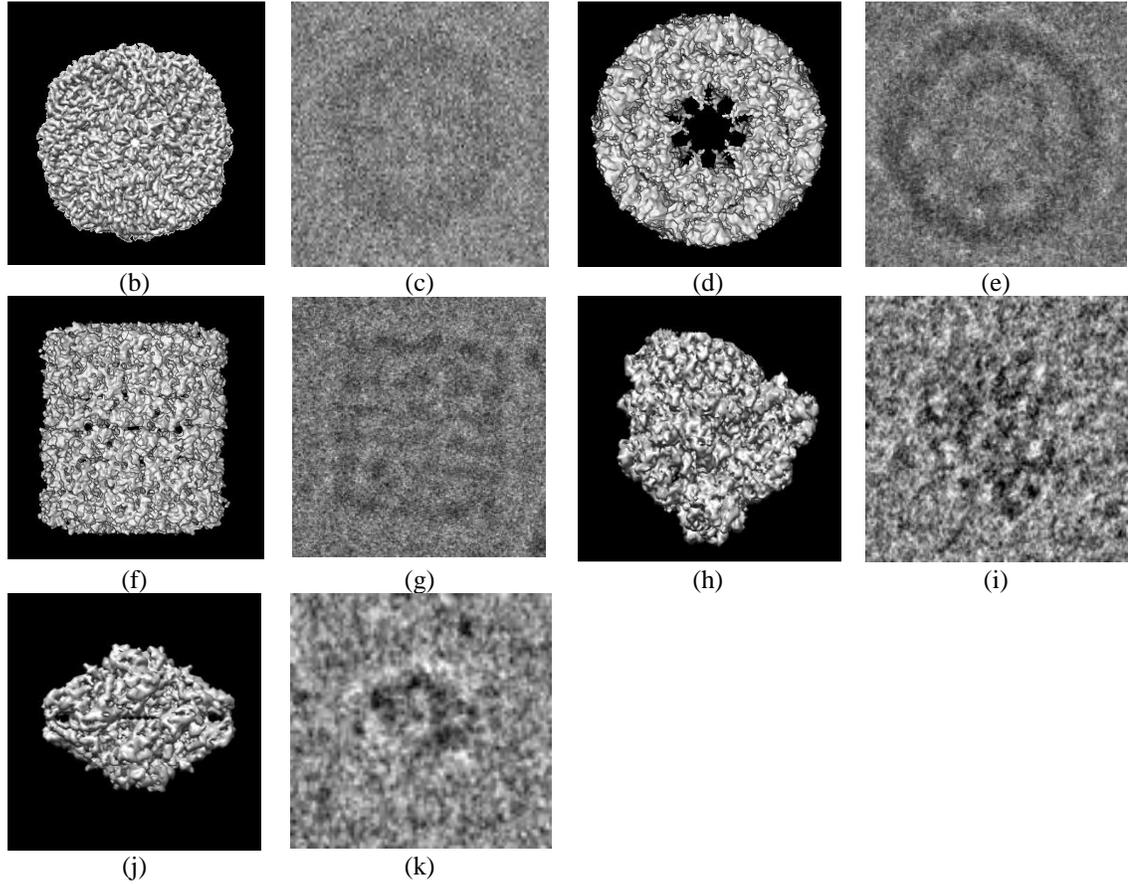


Figure 5.13: (b) simulated top-view (circular) of the Apoferritin molecule shapes, (c) Apoferritin real-world top-view (circular) protein shape, (d) simulated top-view (circular) of the KLH molecule shapes, (e) KLH real-world top-view (circular) protein shape, (f) simulated side-view (square) KLH molecule shape, (g) KLH real-world side-view (circular) protein shape, (h) simulated irregular (complex) Ribosome molecule shape, (i) Ribosome irregular (complex) protein shape, (j) simulated complex beta-galactosidase molecule shape, (k) beta-galactosidase complex protein shape.

5.4.2 Performance Evaluation Metrics

For the evaluation of the performance results we use one of the most popular evaluation metrics which is the precision-recall curve [143] that are defined by Equation (5.9) and (5.10).

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5.9)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5.10)$$

where TP is true positives of particles that are truly picked among the total particles number, FP is the false positives of other objects are detected as particles.

5.4.3 Experiments on Unsupervised Learning Framework for Fully Automated Training Particles-Selection

The first model of our DeepCryoPicker is “automated training particles-selection and data generation based unsupervised learning scheme” as is shown in Figure 5.1. The automated training particles-selection model has two steps: automated training particles picking, and automated training dataset generation. In the first step, 80% of the samples from the collected micrographs are used. Numerous particles are composed and picked from real-world micrograph images using the fully automated framework for particle picking based unsupervised learning approaches that we proposed in our previous models [136] [137]. Then, each single particle image is automatically isolated and evaluated as a “good” or “bad” training sample. The total number of particles for each dataset is shown in Table 5.2.

Table 5.2: Total Number of the Training Particles-selection using Fully Automated Good Training Particles-selection for Apoferritin KLH, Ribosome Datasets.

Criteria	Apoferritin Top-View	KLH Top-View	KLH Side-View	Ribosome Side-View
Total Particle Image Picking	356	878	356	1276
Particle Size	178x178	221x221	221x225	187x187
Perfect Particle Circle Shape Selection	337	874	337	1157
Non-perfect Particle Shape Selection	19	4	19	19
Number of Images	20	82		20
Size of Micrograph	1240x1240	2048x2048		4096x4096

5.4.4 Experiments on Automated Training Dataset Generation.

The final training dataset for the second part of the DeepCryoPicker is automatically expanded for five classes. Three classes that represent the original particle shapes (top-view, side-view, and irregular (complex) protein shapes) are automatically selected from the “good” particle examples after evaluating every single particle. The other two classes image samples are automatically generated from different micrograph’s background as a “background class” or automatically expanded and collected from the “bad” training samples as a “negative detection class”. Then, a certain number of image samples are randomly selected from each training class to expand the size of the training dataset. In addition to the negative detection, each sample is rotated 90, 180, and 270 degrees to generate three additional training sample copies. The total number of training particles before and after regeneration are shown in Table 5.3.

Table 5.3: The Whole Automated Training Particles-selection Dataset.

Dataset	Before Re-generation	After Re-generation
Apoferritin	921	1500
Ribosome	1157	1500
KLH (Top-View)	874	1500
KLH (Side-View)	337	1500
Negative detection	-	1500
Background	-	1500

5.4.5 Experiments on Training Deep Classification Model

To understand the impact of the number of classes on the classification model, we varied the number of classes in the training via three different experiments. In the first experiment, we used all five classes to train and validate the deep classification model. In the second experiment we remove the “background” class keeping the “top-view”, “side-view”, and “irregular protein”, and “negative detection” classes. In the third experiment, we kept the

“background” instead of the “negative detection” class. The corresponding precision-recall curve of each experiment is shown in Figure 5.14. The third case (using three main classes and background), yields the best improvement with an average precision of 100%. The average precision is reduced to 98% and 99% in the first and second case respectively.

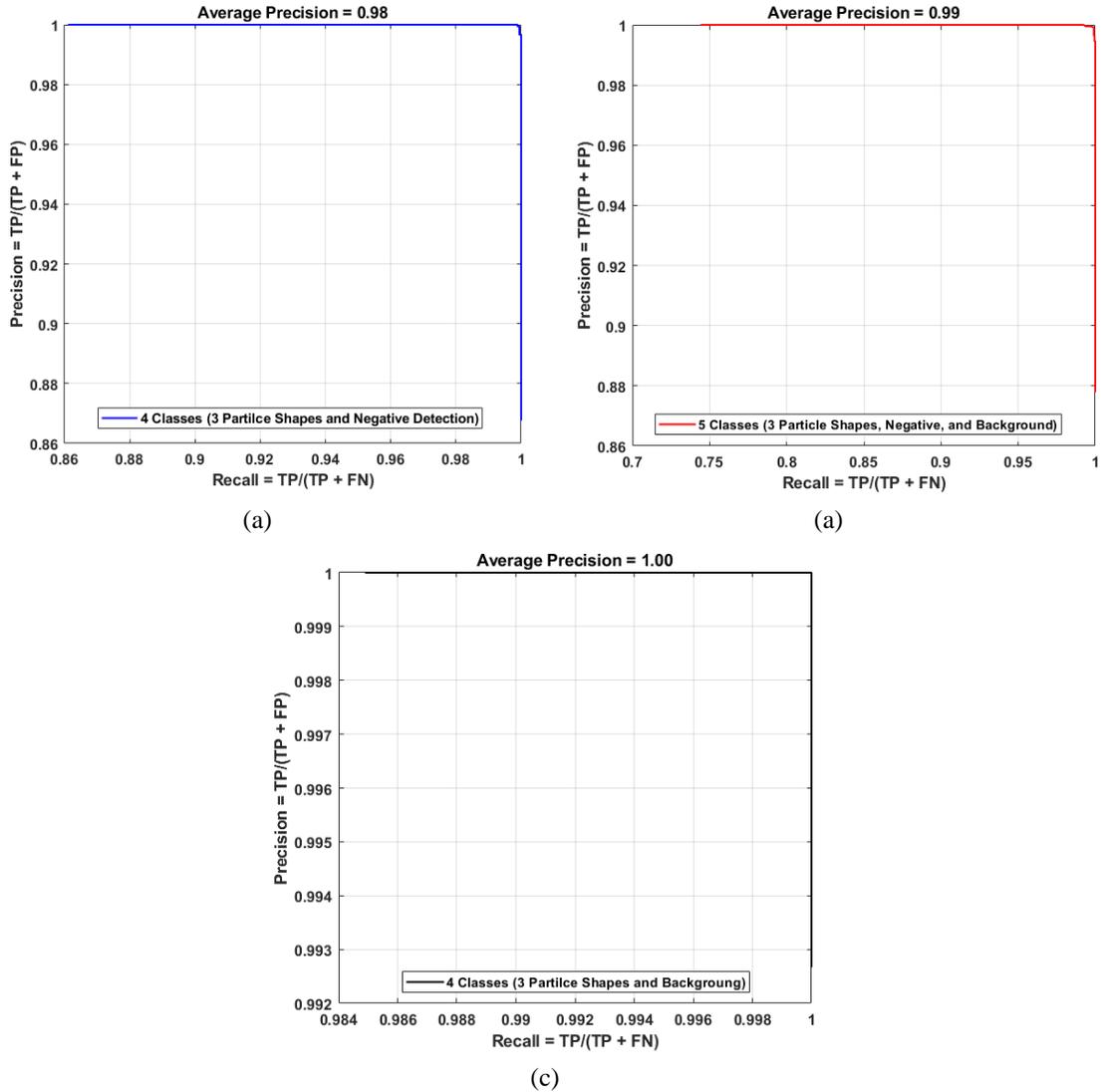
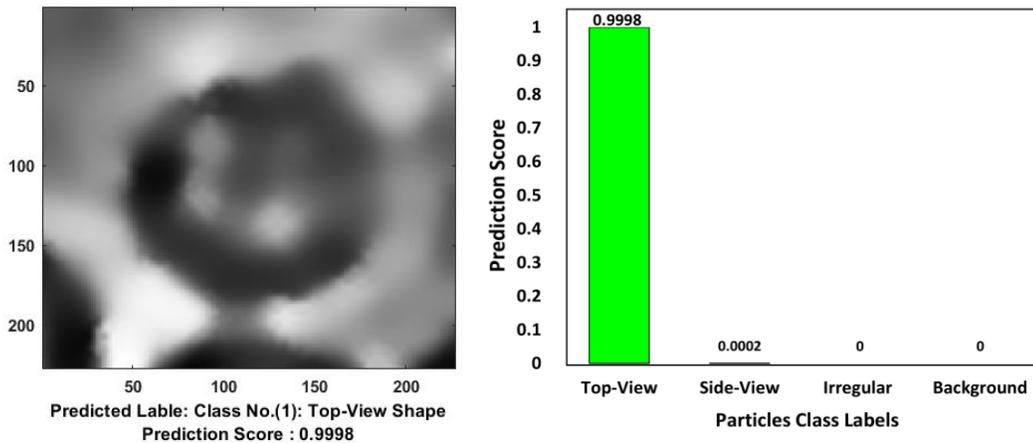


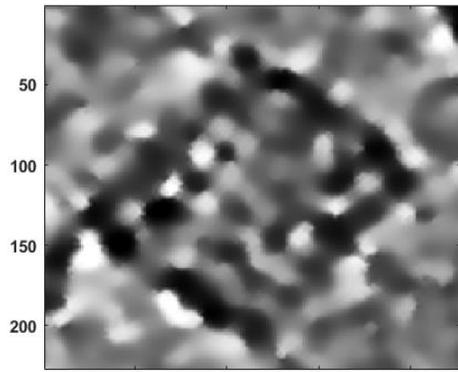
Figure 5.14: Impact of the number of the training classes on the precision-recall curve, (a), (c), and (c) red, blue, and black curves are obtained with the deep classification model training datasets and represents the precision-recall curves plotted for different single particle shapes picking using different micrographs datasets (KLH, Apoferritin, and Ribosome) including 5 classes, 4 classes (with negative detection), and 4 classes (with background) respectively.

5.4.6 Experiments on Testing Deep Classification Model

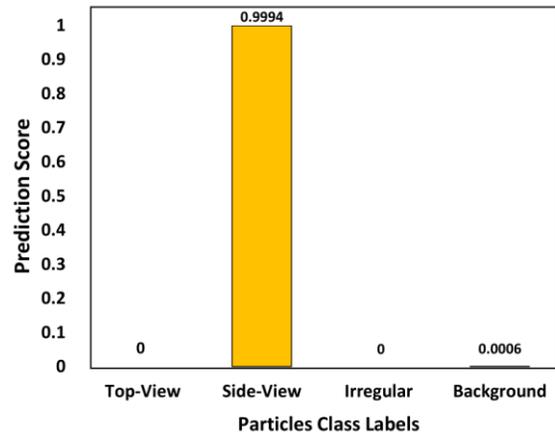
To evaluate the three different models, we split our dataset into a training, testing and validation set. Each class has 1500 particle images, we split the data to 80% training and validation (1050 particle images, 840 for training and 210 for validation) and 20% testing (450 particle images). The total number of the training particles using 5 classes in the first case is 5250 particles while the total number of the testing particles is 2250. For the second and third cases (background or negative) the training set contains 4200 particle images and the testing set contains 1800 particle images. The error or loss of the deep neural network was used as a feedback parameter to tune and adjust the weight and bias, including the number of the feature maps, kernel size of the convolutional layers, and the subsampling kernel size of the subsampling layer. Moreover, the training/testing cycles were tuned based on the hyper-parameters and updated the training datasets until the accuracy of the deep neural network reached a satisfactory level Figure 5.15 shows some testing examples of the deep classification network after training based on the third experiment type (three main particle shapes and background).



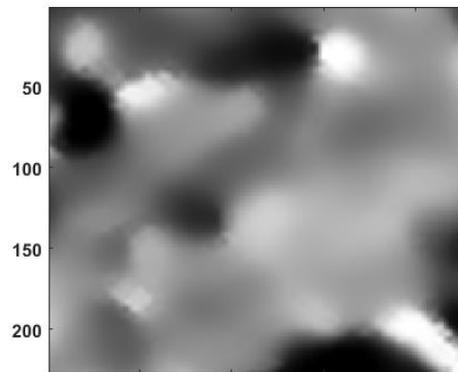
(a)



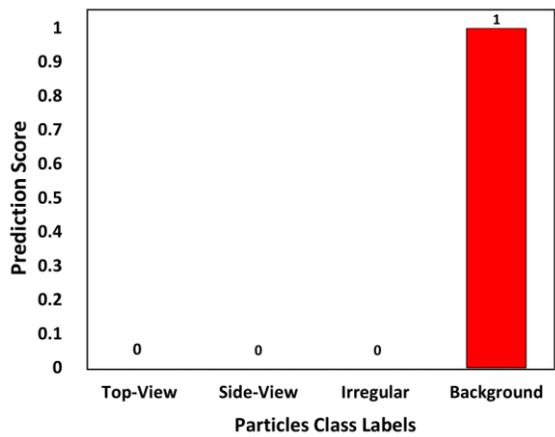
Predicted Class: Class No.(2) Side-View Shape
Prediction Score : 0.9994



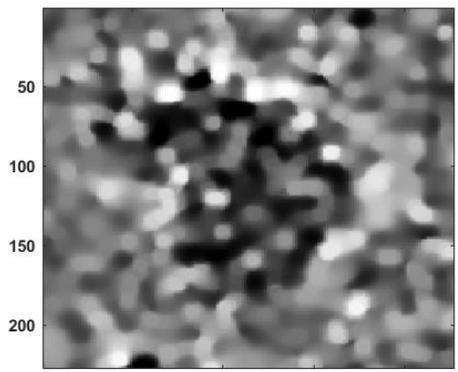
(b)



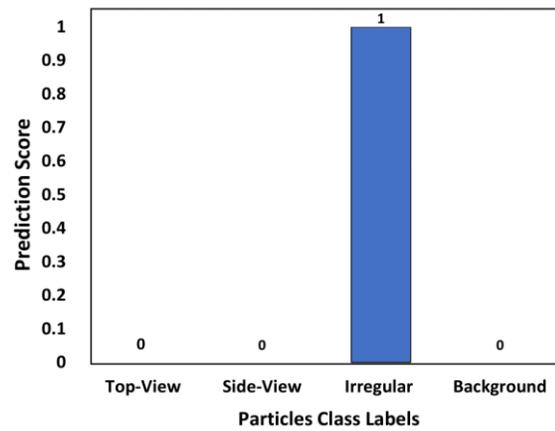
Predicted Class: Class No.(5) :Background
Prediction Score: 1



(c)



Predicted Class: Class No.(3) : Irregulare Shape
Prediction Score: 1



(d)

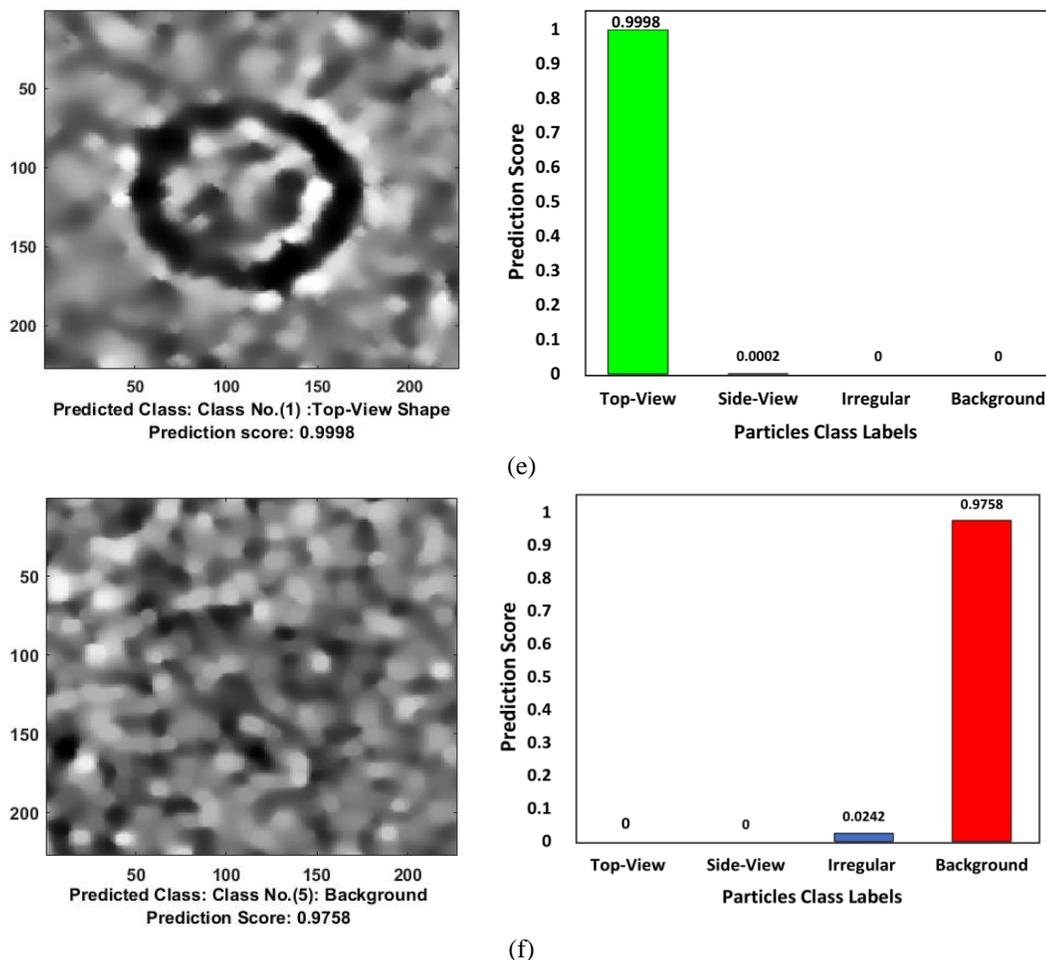


Figure 5.15: Different examples of the deep classification network results. (a) A typical testing image example showing high-density top-view particle's predicted label and prediction score of the Apoferritin micrograph dataset, (b) A typical testing image example showing high-density side-view particle's predicted label and prediction score of the KLH micrograph dataset, (c) A typical testing image example showing high-density background predicted label and prediction score, (d) A typical testing image example showing high-density irregular particle's predicted label and prediction score, (e) A typical testing image example showing high-density top-view particle's predicted label and prediction score of the KLH micrograph dataset, (f) A typical testing image example showing high-density background predicted label and prediction score.

The accuracy of the testing part of the deep classification network using different parameters is shown in Table 5.4. It is clear that the deep classification model achieves a higher accuracy 99.89% based on using the three classes with the background cases.

Table 5.4: Performance Results of the Deep Classification Network Testing using Different Parameters and Different Datasets

Criteria	Learning patch	epochs	Accuracy (%)
4 class “background”	16	20	99.83
	32		99.89
	64		99.72
4 class “negative”	16	20	97.83
	32		97.78
	64		97.83
5 classes	16	20	96.62
	32		95.96
	64		95.91

5.4.7 Experiments of Deep Learning Framework for Fully Single Particle Picking on cryo-EM Datasets

The second model of our DeepCryoPicker is the fully automated single particle picking based deep learning scheme. The particle picking model has three steps: scanning-test, scoring-cleaning, and filtering using non-maximum suppression step. In the first step, a sliding-testing window is used to scan each micrograph in the testing dataset from the top left to the bottom right corner with a constant step size. To determine the prediction parameter, a fixed size sliding window (square box) is chosen to be slightly larger than the particle size. During the scanning-testing step, each single patch is extracted and fed to the train deep classification network. After that each sliding window has a certain prediction value [0 1] as the deep network classification model predicts. The prediction scores represent the probability of considering the particles as a candidate at the current position. In this case, the prediction value is assigned to the center of the corresponding window. The second step of the fully automated single particle picking model is the scoring-cleaning step, in this step a scoring map is generated for each tested micrograph. The scoring map describes the likelihood scores distribution of the particles over the entire micrograph. In

fact, some detected object such as ice or noise can be easily defined as a false positive. To discard the false positive detection, a cleaning step is implemented which connects any two pixels scoring maps whose prediction scores are close and both above the threshold. Then, some connected areas (pixels) are regarded as a false positive if the connected area's size is larger than a cutoff value and removed from the candidate list. Finally, we used non-maximum suppression (NMS) [17] as an integrated step to refine the current particle candidates list. NMS is used to filter the detection boxes based on their Intersection over Union (IoU) between the detected boxes. Candidate particles filtering based on the NMS has three main steps: sorting, selecting, and repeating. First, all candidates' boxes for each given particle category are sorted based on their prediction scores (from high to low). Second, the candidate box that has the highest prediction score is selected as the final candidate box. Then, all other candidate boxes within the selected IoU are discarded. Third, within the remaining boxes the NMS repeats the first and second step until there is no remaining boxes in the candidate list.

A typical example of DeepCryoPicker result is shown in Figure 5.16, which illustrates the results of the particle picking using the fully automated framework and different micrographs from different datasets. The average precision-recall both reached 95%.

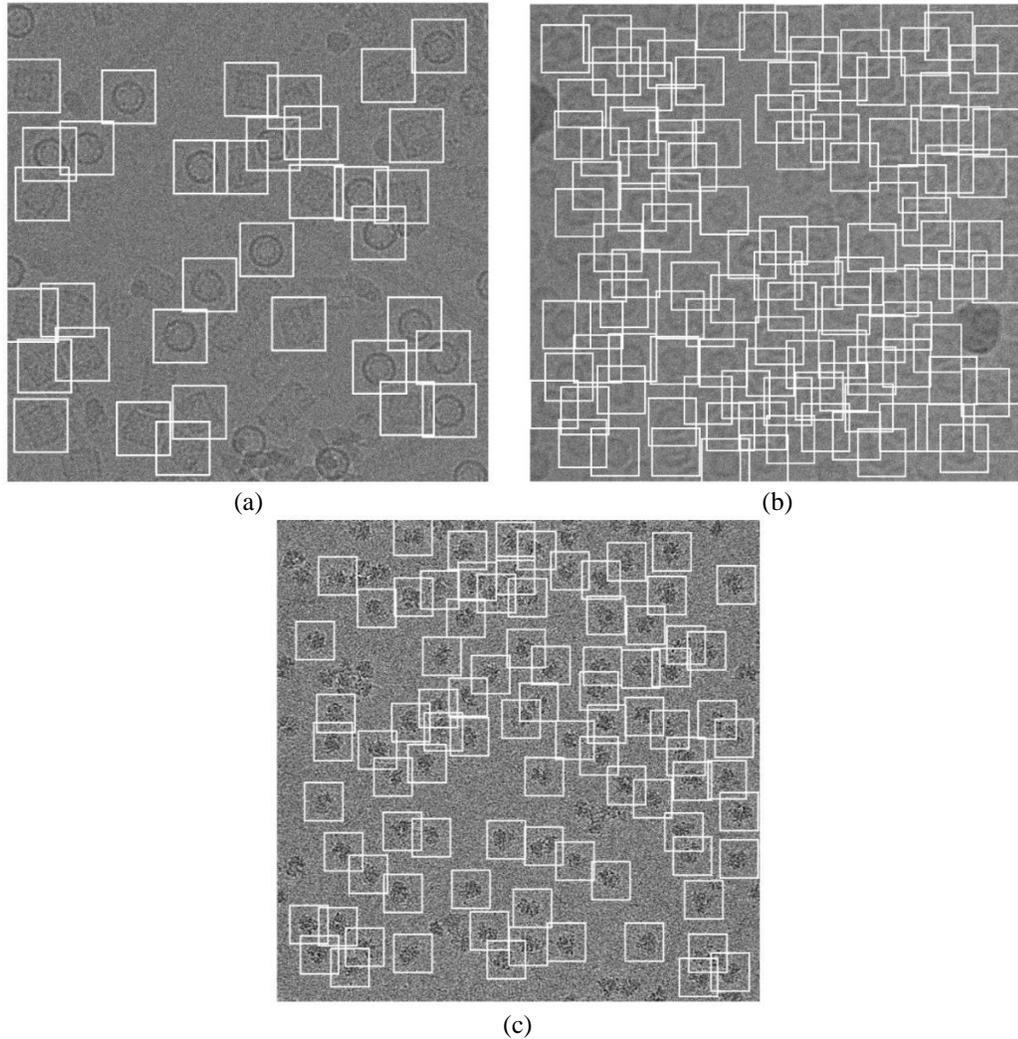


Figure 5.16: The DeepCryoPicker” fully automated single particle picking in cryo-EM images results for different micrographs datasets (a) typical micrograph showing the KLH Top and Side-View particles picking, (b) typical micrograph showing the Apoferritin Top-View particles picking, (c) typical micrograph showing the Ribosome irregular (complex) particles picking,

Figure 5.17 shows the precision-recall curves for each particle shapes individually using different datasets such as Apoferritin and KLH for the top-view particle shapes, KLH only for the side-view particle shapes, and Ribosome and beta-galactosidase for irregular and complex particle shapes. For instance, Figure 5.15 (a) shows the blue plotted curve of the recession-recall for top-view particle shapes picking, Figure 5.15 (b) shows the red plotted curve of the recession-recall for side-view particle shapes picking, and Figure 5.15

(c) shows the black plotted curve of the recession-recall for irregular and complex particle shapes picking.

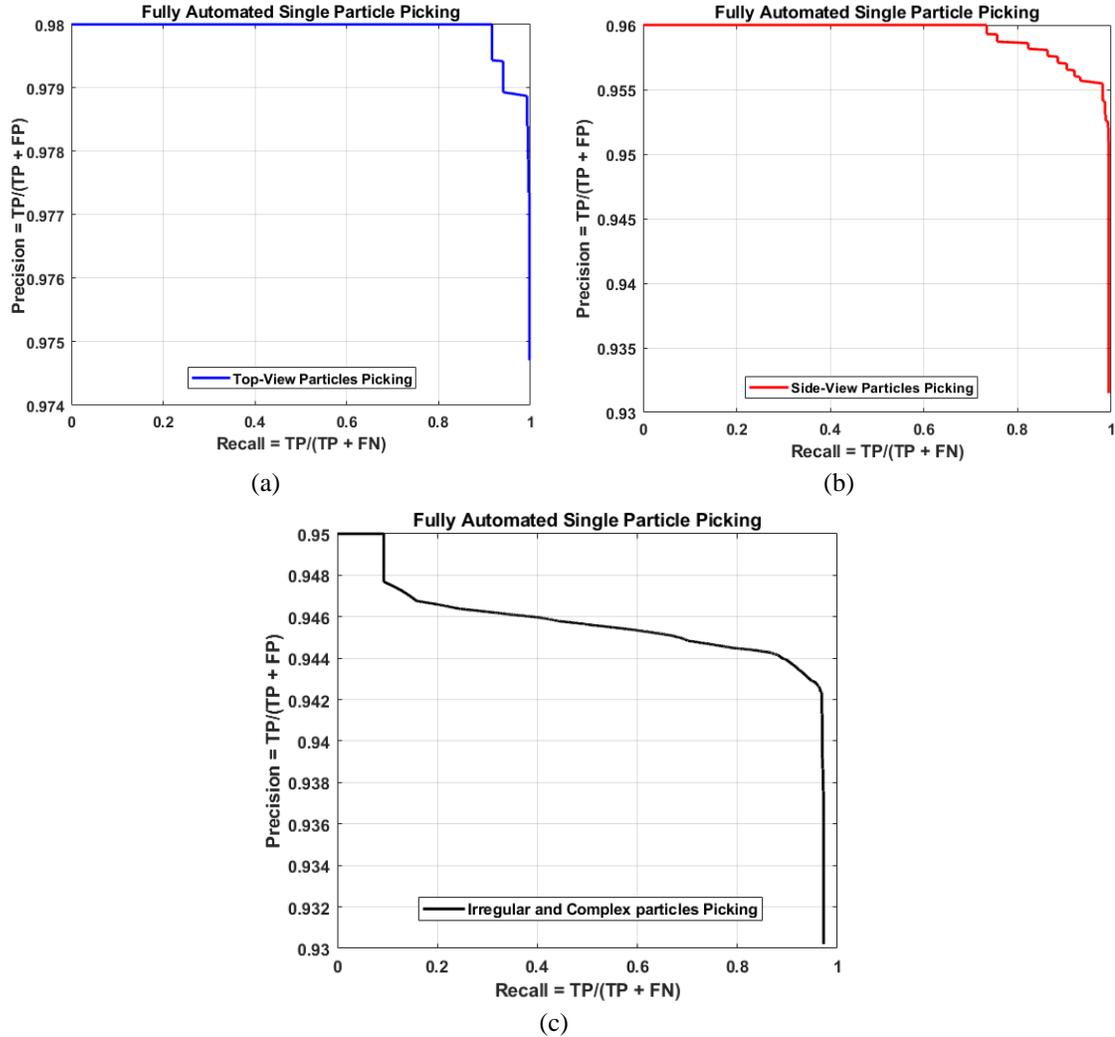
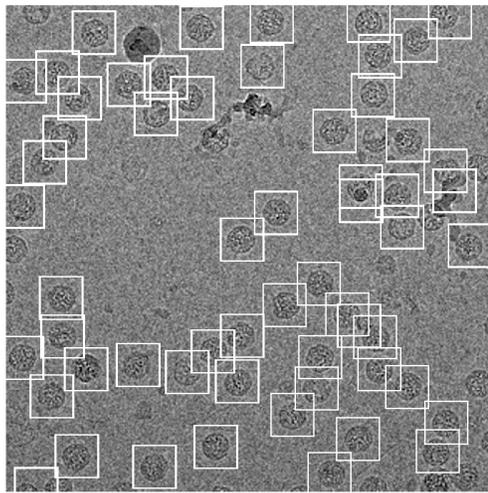


Figure 5.17: Precision-recall curves of the fully automated different single particle shapes picking result using deep classification network and different micrographs datasets, (a) precision-recall curve of the top-view particle shapes picking, (b) precision-recall curve of the side-view particle shapes picking, (c) precision-recall curve of the irregular and complex particle shape picking.

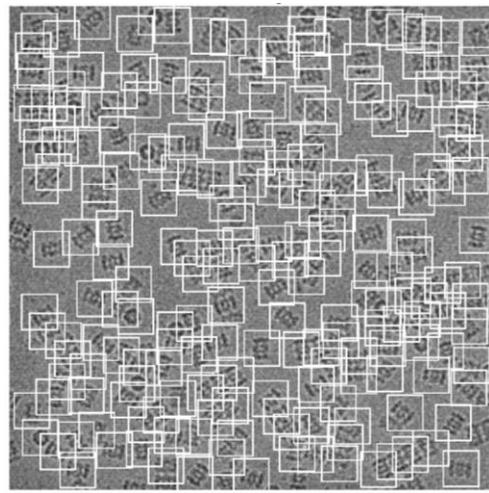
5.4.8 Experiments on External Testing Micrographs

In addition to test our model in different micrographs (testing set) that have been split from the whole collected datasets and in order to show the robustness of our DeepCryoPicker,

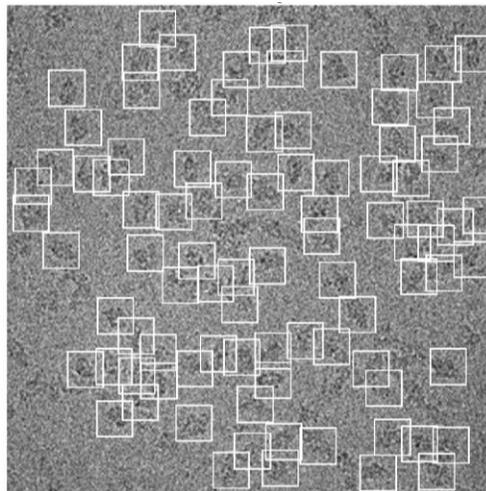
we test our model in three different micrographs (external testing micrographs) as is shown in Figure 5.18. The external testing micrographs are collected from different datasets that our system has trained on. The external testing micrographs have been selected based on the particle shapes that each micrograph has as is shown in Figure 5.18 (a), (b) and (c). For instance, Figure 5.18 (a), show the first external testing result using an external testing micrograph from the bacteriophage MS2 (EMPIAR-10075) [148] where the particle shapes are identical top-view. Figure 5.18 (b), show another external testing result using an external testing micrograph from the *T. acidophilum* 20 (EMPIAR- 10186) [149] where the particle shapes are top and side-view.



(a)



(b)



(c)

Figure 5.18: The External DeepCryoPicker testing results using different micrographs from different external testing datasets (a) typical external micrograph from the bacteriophage MS2 (EMPIAR-10075) [18] showing the Top-View particles picking, (b) typical external micrograph from the T. acidophilum 20 (EMPIAR- 10186) [19] showing the Top and Side-View particles picking, (c) typical external micrograph from the beta-galactosidase 2.2 Å (EMPIAR- 10061) [19] showing the irregular (complex) particles picking,

Finally, Figure 5.16 (c), show the last external testing result using an external testing micrograph from beta-galactosidase 2.2 Å (EMPIAR- 10061) [150] where the particle shapes are irregular shapes.

5.5 Discussion

The signal-to-noise-ratio (SNR) of original (2D) micrographs tends to be very low, with noise from a variety of causes including low contrast, particle overlap, ice contamination and amorphous carbon [130]. Hence, the task of single particle picking still presents major challenges. [130]. Many different computational methods have been proposed for the automated semi-automated single particle picking over the past decades. Single particle picking using template-based matching methods are very sensitive to noise [151] [152] [153] [154] [155] [131] [156]. Thus, some initial “good references” are selected in advance to ensure that those manual selected examples have less noise comparing with the other in the same (2D) micrographs. Similarly, the edge-based [157] [158] and feature-based methods [159] [160] [161] show significant reduction in performance since they are sensitive to the lower contrast of the (2D) micrographs [130]. Recently, three methods for single particle picking have been proposed based deep learning approach. EMAN2.21 (particle picking with convolution neural network [132], DeepEM [134], DeepPicker [133], FasetParticlePicker [134], and EMAN2 [121]. The four deep learning methods present significant contribution of the main particle picking and selection issue. However,

there are some challenges that those methods are facing such as lacking diversified training dataset, false-positive numerosity, and low-SNR micrographs accommodation.

To address these issues, we propose a fully automated deep neural network for single particle picking based fully automated training particle-selection using unsupervised learning algorithms. First, to generate such a sufficient training dataset, we design a fully automated training particle-selection based on unsupervised learning algorithms. In this approach, most of the regular protein shapes (top and side view) have been fully automated picked based on our IBC algorithm. The second super clustering algorithm (SP-K-means) has been proposed in our second model “a super clustering approach for fully automated single particle picking in cryo-EM” [137]. In this approach, most of the irregular and complex protein shapes (molecules) have been accurately picked based on a fully automated unsupervised learning approach using the proposed super k-means clustering algorithm. In this case, generation of the training set is fully automated thus eliminating the need for manual labeling or labor-intensive particle selection. Second, to accommodate the low-SNR, we have designed a general framework of micrographs preprocessing that has been used in both our last two models [136] [137]. In order to improve the quality of noisy cryo-EM images, we have selected a set of advanced preprocessing tools to improve the quality of the low-SNR micrographs. Those tools are tested on different cryo-EM datasets such as KLH [138], Apoferritin [139], Ribosome [140], and Beta-galactosidase [141]. In general, the preprocessing steps increase the particle’s intensity and pre-grouping the pixels inside each particle makes them easier to be isolated by the clustering algorithm.

Three main criteria have been based on to select the preprocessing tools. First, enhancing the global contrast of the micrographs. Second, enhancing the local contrast and

increasing the intensity level of each particle. Third, enhancing the particle shapes inside the micrographs. Third, to solve the considerable number of false-positive (FP) particle detection images, we use Non-Maximum Suppression (NMS) [147] during the testing phase in order to reduce the number of false-positive particle detections. Basically, some particle's image candidates are eliminated in fact different detection of the same particle object. In another word, removing multiple bounding boxes around the same particle object and keep the candidates for different particle objects. This step removes duplicates of bounding boxes centered around the same general region, consequently decreasing false-positive detections.

In the training particle-selection-based unsupervised learning scheme, we use the preprocessing stage which significantly improves the low-SNR micrographs, then use the preprocessed micrographs for the unsupervised micrographs clustering stage. Also, we combined two different fully automated particle picking approaches (AutoCryoPicker [136] and SuperCryoEMPicker [137]). In addition to that, we generated a fully automated approach for training dataset expanding and training particle images increasing. The fully automated training particle-selection is able to automatically distinguish between the “good” and “bad” training examples and isolate the selected particles to positive and negative detection examples. Later, in the second part which is the single particle picking, a deep neural network is designed and trained using the generated training dataset from the first part. Finally, for each testing micrograph, we used the same preprocessing stage to improve the quality of the low-SNR micrographs. Then, we use the trained deep neural network model and sliding windows to test every single sub-image based on using the NMS. In our experiments, we compare the results from the DeepCryoPicker with different

particle picking tools such as RELION [135], PIXER [128], DeepPicker [133], and DeepEM [130]. Figure 5.19 shows the quantity analysis on real micrograph datasets using a precision-recall curve. The green, yellow, black, blue, and red curves represent the precision-recall curves for RELION, DeepPicker, DeepEM, PIXER, and DeepCryoPicker respectively. The results indicated that DeepCryoPicker performance was comparable to the RELION method which is semi-automated and better than the other RELION method.

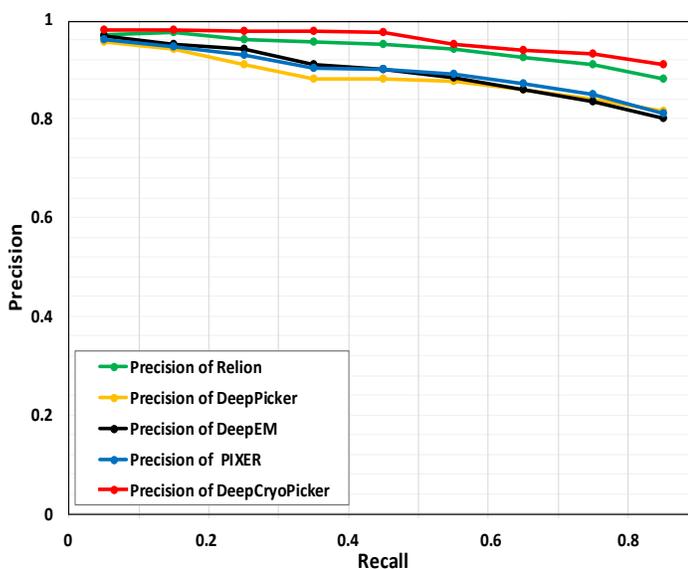


Figure 5.19: Quantity analysis on real datasets using a precision-recall curve of different single particle picking tools. The green, yellow, black, blue, and red curves represent the precision-recall curves for RELION, DeepPicker, DeepEM, PIXER, and DeepCryoPicker respectively.

Chapter 6

DeepCryoMap: Fully Automated cryo-EM Particles Alignment Approach for 3D Density Maps Reconstruction Based Deep Supervised and Unsupervised Learning Approaches

6.1 Background

The main component to building blocks and workhorses of the of biological organisms are the macromolecular complexes. A macromolecular complex (cell) is defined as a composed of different components such as protein and ribonucleic acids (RNA) [163]. For better understanding the macromolecular function, obtaining and determination the macromolecular structures is needed. Understanding fail of the macromolecular function sometimes leads to understand a common case of many diseases [164].

In general, protein is defined as a chain of amino-acid molecules [165]. Each amino-acid molecule has different backbone atoms that are connected to another atoms

(another chain of amino-acid molecules) or to another side-chain atoms (branch out from one atom in the backbone) as is shown in Figure 6.1 (a) [163] [166]. The amino-acid sequence constructs the protein primary structure, and the chain of the amino-acid molecules construct folds of the protein secondary structure (alpha helices or beta strands) (see Figure 1. (b), (c), and (d)) [1] [4]. Based on the tightly pack of the amino-acid one against each other in the secondary structure, the protein structures are folded which is refer to the protein tertiary structure as is shown in Figure 6.1, (e). Finally, different binding of protein to another produces different protein interactions which called the protein quaternary structure (see Figure 6.1, (f)) [1] [4]. The four different levels of the protein structure are shown in Figure 6.1.

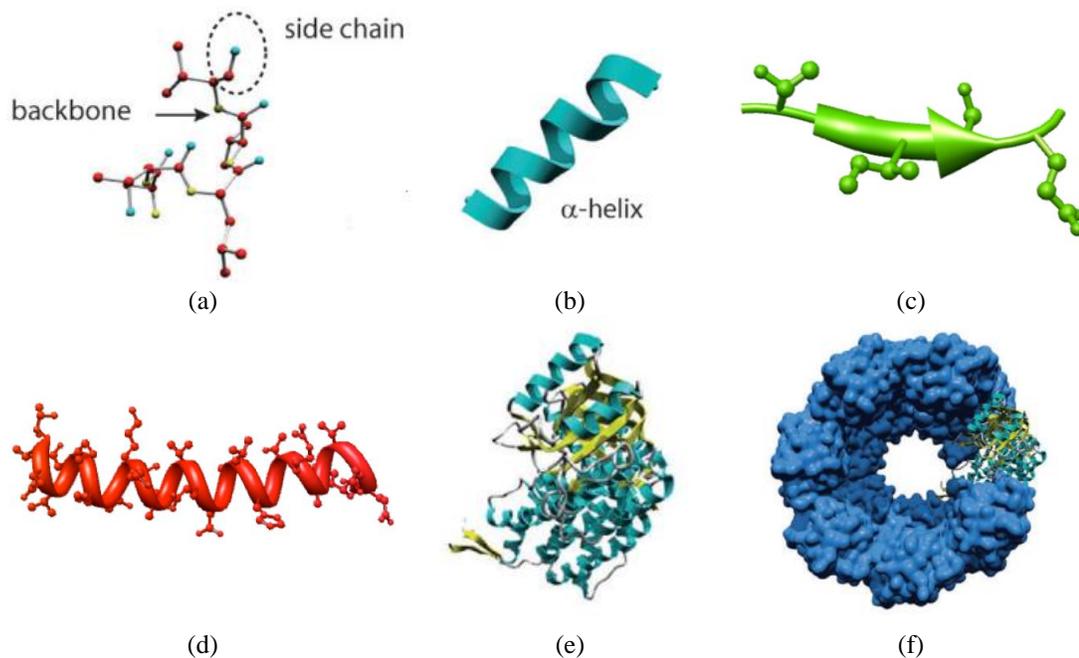


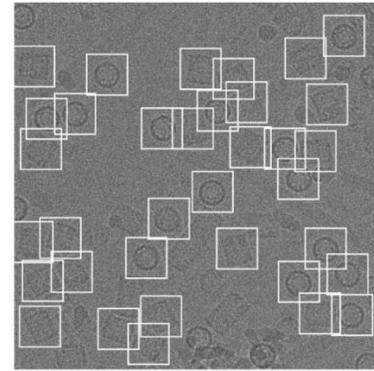
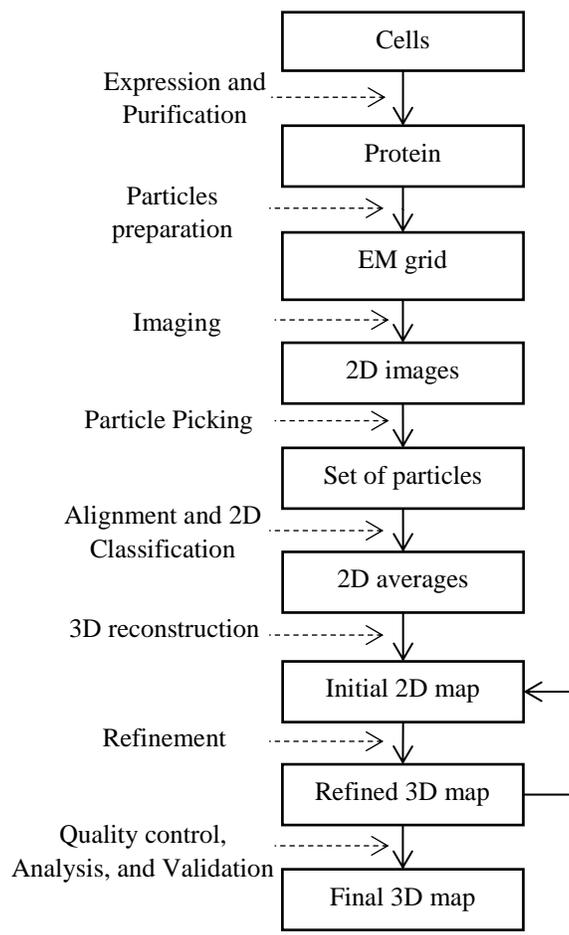
Figure 6.1: Different Levels of protein structures. (a) the protein primary structure (sequence of amino acids connected to form a long chain). (b) protein secondary structure. (c) alpha helical. (d) beta strand. (e) protein tertiary structure. (f) protein quaternary structure (Figure is retrieved from [163] [166]).

X-ray crystallography is a well-established method that used to understand and determination the macromolecular complexes. It is used to determine the atomic positions

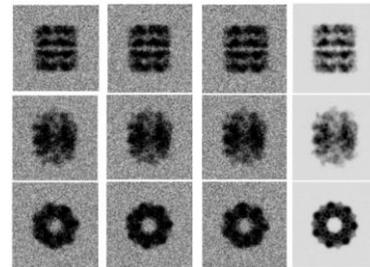
in a molecular complex [163] [167]. The x-ray crystallography uses the crystal diffraction patterns to reconstruct a 3D density maps (electron density maps). Typically, the position of the individual atoms can be accurately determined in a high-enough map resolution [167]. Therefore, the protein samples (complexes) must be crystallized first to apply the X-ray crystallography for understand and determination the macromolecular complexes [168]. However, the X-ray crystallography is universally method, which means it is not an accurate method to determine the dynamic, or complexes structures [169].

Cryo-EM (electron microscopy) method uses to uncovers the macromolecular complexes structures [170]. It depends in involving and purifying a solution onto a grid, cryogenines it (freezing it), and then using the electron microscope to image the frozen film [171]. Typically, the cryo-EM method here does not require the crystallization step. On the other hand, it can be applied to a wider of complexes. However, it requires a lot of processing (computation steps) to produce lower-resolution density map where the positions of the atomic will be hard to determine. On the other hand, the 3D density reconstruction map can be improved. It also gives insight into the structure to understand the molecular function that makes the cryo-EM density map for structure determination in increasingly used [172] [173]. During the cryo-EM (electron microscopy), many 2D images of the complex are produced in varying orientations [174]. So, the 2D images are classified or clustered corresponding to the same orientation first. Then, to improve the signal-to-noise-ration, the 2D images are averaged. Finally, the 3D volume (density map) is produced after back-propagate the averaged 2D images [163].

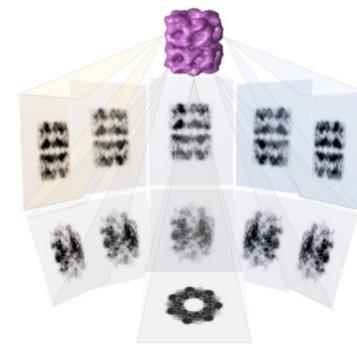
In cryo-EM, the standard procedure to produce a 3D density map using cryo-EM images, called single-particle-reconstruction that is illustrated in Figure 6.2 (a) [175].



(b)



(c)



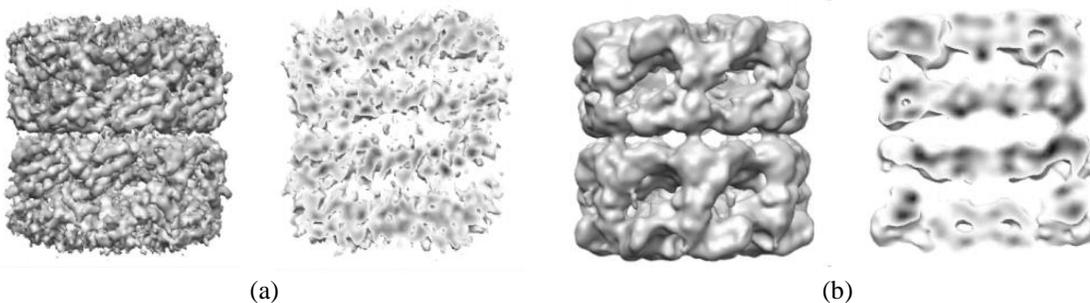
(d)

Figure 6.2: 3D Density Map Reconstruction using cryo-EM technique. (a) single-particle-reconstruction procedure (framework), (b) Identify the individual complex 2D image (particle picking) [163] [175].

Hundreds of thousands of the particle images (2D) are required to build and produce an efficient 3D density maps [176]. As is shown in Figure 6.2 (a), using many copies of the same complex are placed in a thin film (solution) to prepare the complex sample by cryogenically frozen it, then many 2D images of the same complex are picked. Later, the electron beam is used image the (captured the same complex in 2D cryo-EM image in different orientations) [163] [175] as is shown in Figure 6.2 (b) where the complex

(particles) are identified and picked with the red boxes [175]. After the particles are picked, the 2D complex (particle) images are clustered and averaged to improve the signal-to-noise-ratio and create a ‘cluster-average image’ as is shown in Figure 6.2 (c). Finally, the 3D volume (3D density map) is made (produced) by project back the 2D cluster-average images after compute the orientations of each average image with respect to this volume as is shown in Figure 6.2 (d).

EMAN2 [177], RELION [178], and SPIDER [179] are the recent methods that have been implemented and developed for 3D map reconstruction using cryo-EM. For those tools, an initial 3D model is required to build a decent 3D density map in addition to the manually particle picking issue. A density map is a 3D grid that each point has a certain density value. The density value reflects the electron density based on the corresponding point in the 3D space. The basic way to visualize the 3D density map is by project the 2D images to create a 3D volume by collecting triangles points [180]. An alternative way to visualize the density map is to show the 2D slice in the 3D volume. Each slice illustrates the density value (dark values represent higher densities). An example of different 3D density maps visualization is shown in Figure 6.3. It also shows different common software’s that are used to visualize the 3D density maps Iso-surface (see Figure 6.3 (a) and (b)), also slice (see Figure 6.3 (b) and (c)). Figures are revised from [163].



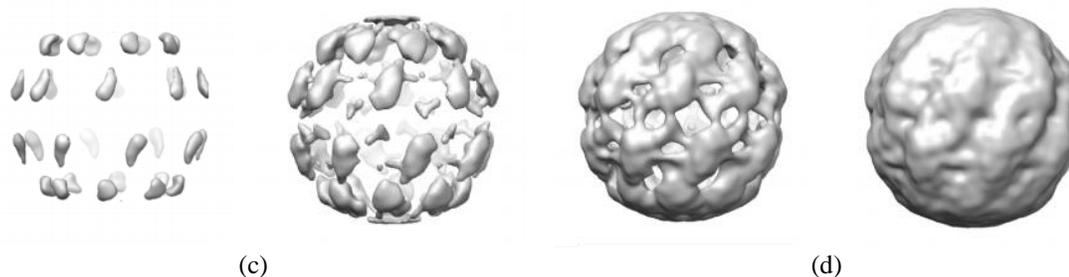


Figure 6.3: Different Complex cryo-EM density maps Visualization, the maps were captured from the EMDB (EMDB:5001 and 1042 respectively) [181] [182] and images are captured from [163]. (a) and (b) low- and high-density map resolution visualized 3D density map using so-surface of low- and high-density map resolution respectively, and slice (right) representations are shown for maps at two resolutions. Higher resolution density maps (lower number) have a greater amount of detail, while lower resolution (higher number) are smoother. cryo-EM density map of visualized with iso-surfaces at 4 different density thresholds. The surfaces shown are drawn at decreasing threshold values, with the surface on the left having the highest threshold. At higher threshold, the inner and denser parts of the complex are seen, while at lower thresholds a larger outer envelope of the complex can be seen.

6.2 Methods

We design DeepCryoMap a fully automated 3D cryo-EM density maps reconstruction based deep learning and unsupervised learning approaches. It is designed to reconstruct 3D density map for a structural single protein model. In another word, a structural single protein model that is illustrated as top and side view molecules as is shown in Figure 6.4 using cryo-EM images from two different cryo-EM datasets Apoferritin [183] and KLH [184] dataset. Figure 6.4 (a) shows picked particle from an apoferritin micrograph and (d) 3D Cryo-EM map. Figure 6.4 (b) shows top view picked particle from a KLH micrograph (circular particle), and (e) shows the 3D Cryo-EM map of KLH viewed. Figure 6.4 (c) shows side view picked particle from a KLH micrograph (square particle), and (f) shows the 3D Cryo-EM map of KLH viewed from the side.

The other protein model (unistructural molecular) such as irregular and complex cryo-EM particles is our future works.

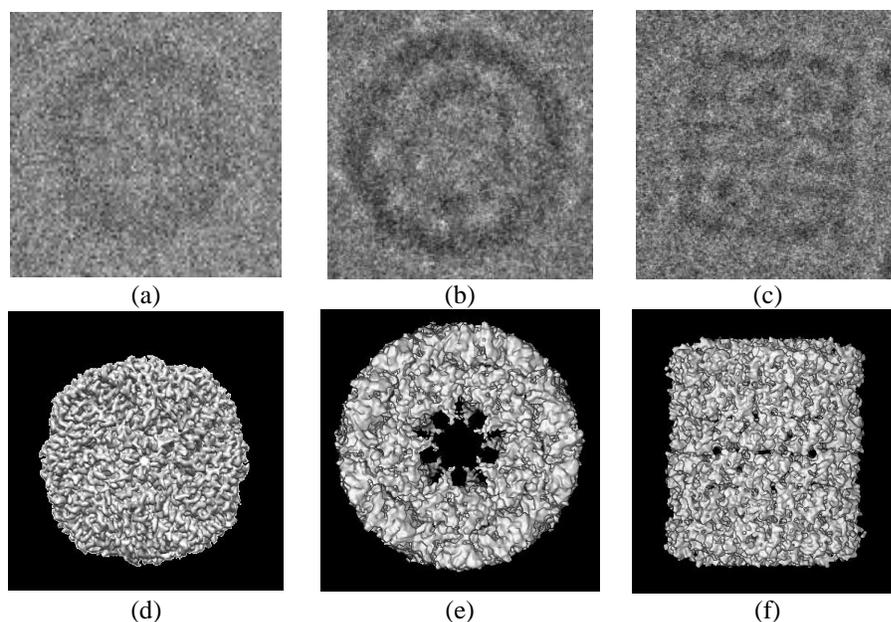


Figure 6.4: Structural cryo-EM 3D Density Maps examples (top and side view) (b) 3D Cryo-EM map of apoferritin, (c) picked particle from an apoferritin micrograph, (d) 3D Cryo-EM map of KLH viewed from the top, (e) picked particle from a KLH micrograph showing the top view (circular particle), (f) 3D Cryo-EM map of KLH viewed from the side, (g) picked particle from a KLH micrograph showing the side-view (square particle).

The main structure of the DeepCryoMap framework is shown in Figure 6.5. In general, the DeepCryoMap is designed for fully automated 3D density map reconstruction from cryo-EM data based fully automated single particle picking. Our framework contains five components: The first component is the processing step (shown on the green box of Figure 6.5). The micrographs are pre-processed using set of advanced image processing tools to enhance and increase the quality of the micrographs. In this component we used the same preprocessing steps that have been used before in our last three models AutoCryoPicker [185], SuperCryoPicker [186], and DeepCryoPicker [187]. The second component is the fully automated single particle picking based on deep neural network (shown on the brown box of Figure 6.5). In this component we used our last model DeepCryoPicker [187] for the fully automated single particle picking in cryo-EM.

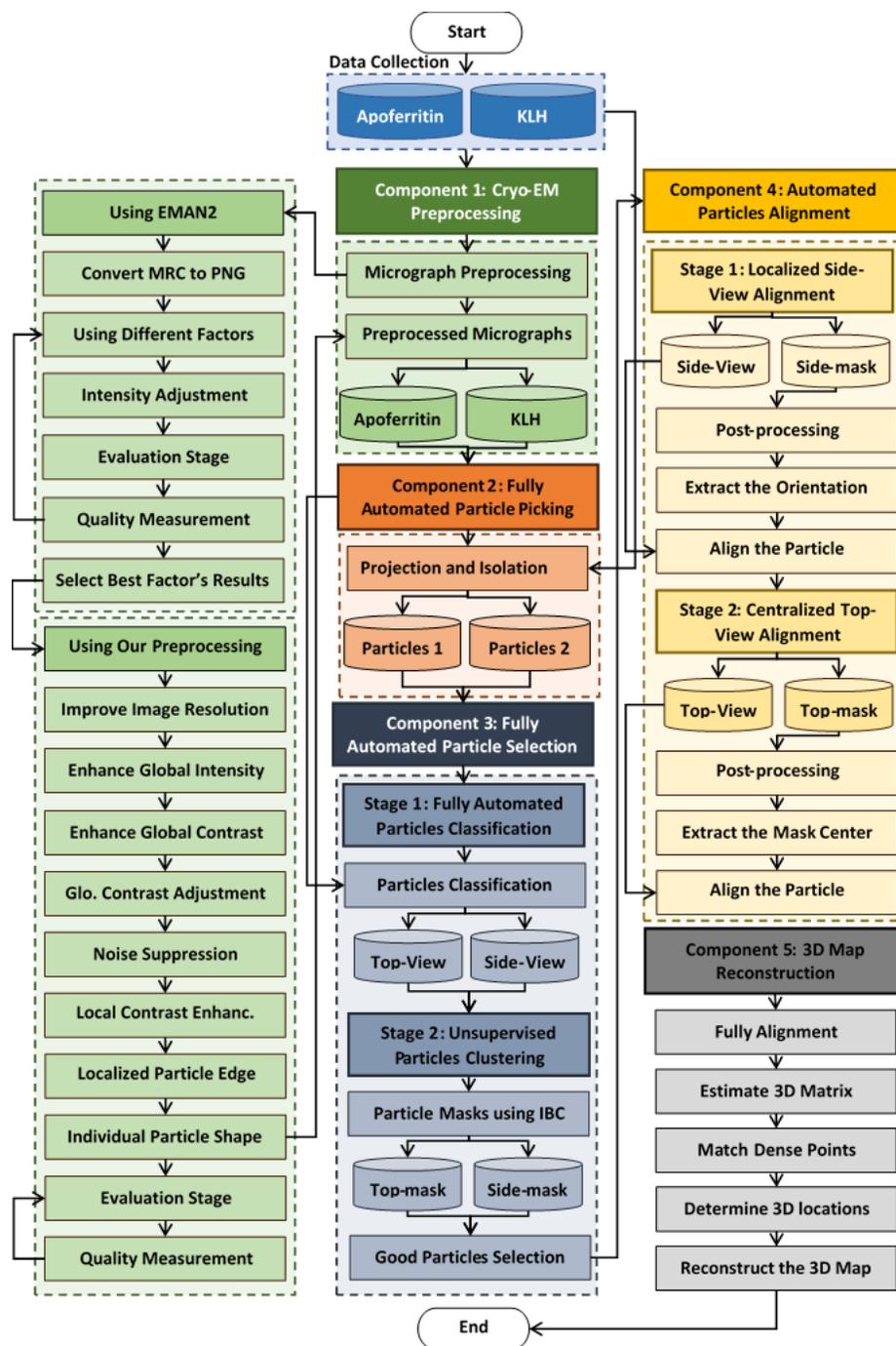


Figure 6.5: The general workflow of the DeepCryoMap based single particle picking based on deep learning scheme and unsupervised scheme. The green part of the workflow shows the micrographs data preprocessing. The red part of the workflow shows the fully automated single particles picking using DeepCryoPicker [25]. The blue part of the workflow shows the general flow of the good (perfect) 2D single particle-selection using unsupervised learning and deep classification network. The yellow part of the workflow shows the fully automated particle alignment, and gray part shows the 3D density map reconstruction using single-particle-reconstruction.

The third component is the fully automated particles selection (shown on the blue box of Figure 6.5). The third component has two stages: automated particles classification based supervised learning approach, and automated perfect (good) 2D particles selection based unsupervised learning approach. The first stage of the automated particles classification is based on deep classification model that has been used in the DeepCerypPicker [187]. In this stage, each particle is classified based on the same orientation. In the second stage, first, each individual particle image is clustered based unsupervised learning approach using our clustering algorithm Intensity-Based Clustering that has been proposed in our first model AutoCryoPicker [185] to generate set of particle binary masks.

Second, automatedly perfect (good) 2D particles are selected selection by evaluating each particle's mask and evaluated it as a "good" or "bad" 2D particle. The fourth component is the fully automated single particle alignment based on a using the unsupervised learning approach which has two stages. The first stage is designed for side-view particle alignment called localized side-view particle alignment. Some steps are implemented in this stage such as postprocessing which some irrelevant objects are removed and refence mask generation, apply the ferret dimeter and extract the particle orientation. Extract the difference angle value and align the original particle. The second stage is designed for the top-view particle center alignment which called centralized top-view particle alignment. In this stage, some steps are also implemented such as postprocessing step, mask enters extraction step, and mask alignment based centralized particle masks. The last component is the 3D density map reconstruction.

The final 3D density map is constructed in this component based on the single-particle-reconstruction. Different steps are implemented in this component such as first, a fully particle alignment in which the sparse set of points between the particle images are matching. Second, calculated the camera calibration in which estimate the fundamental matrix (camera parameters). Third, match and track the dense set of points between the particle images. Fourth, determine the 3D locations of the matched points using triangulate [163]. Finally, recover metric reconstruction resulting the actual 3D density map.

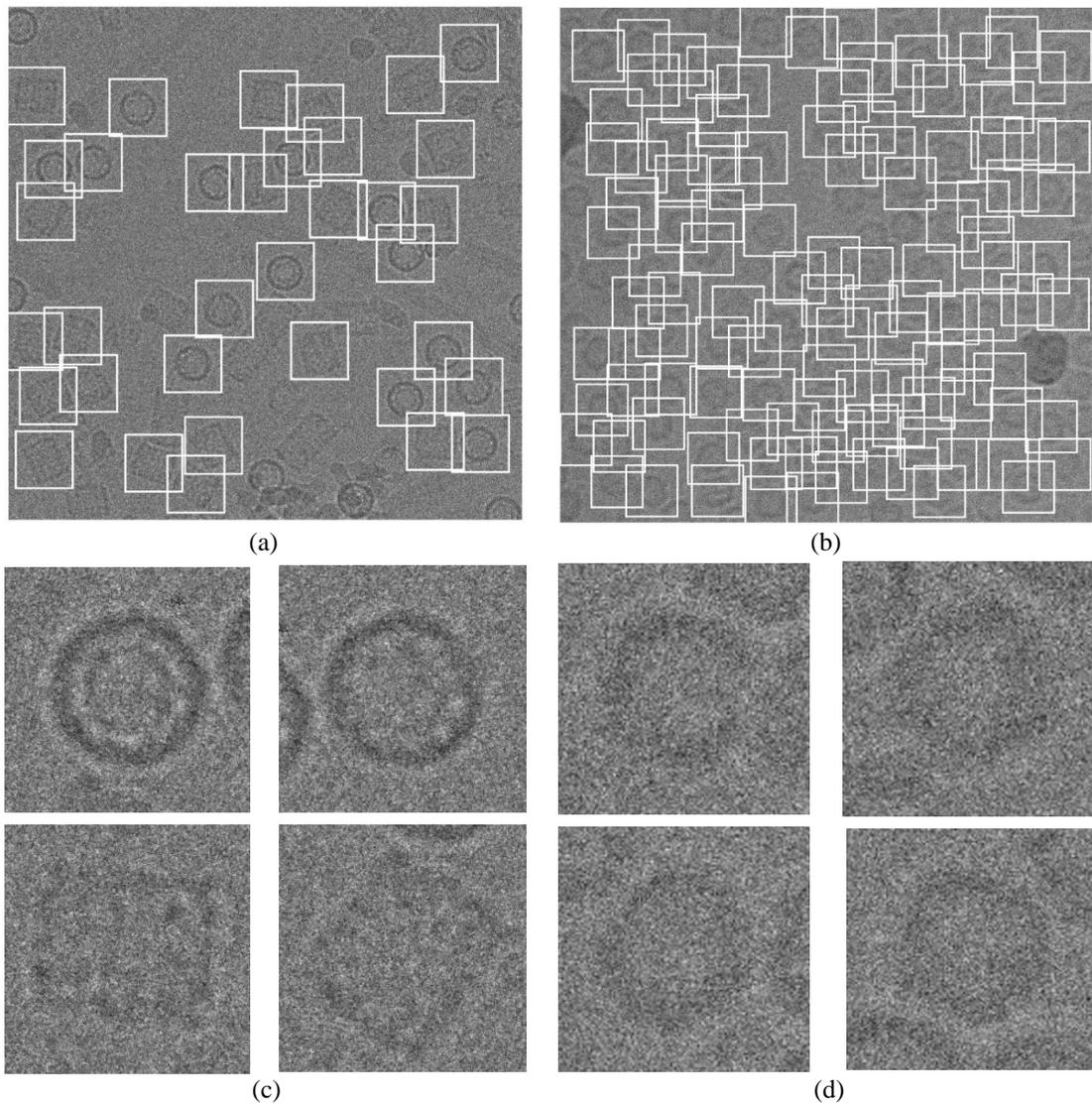
6.2.1 Component 1: Micrographs Particles Pre-processing

In this component, set of pre-processing steps are used such as image resolution, global intensity adjustment, global contrast enhancement-based histogram equalization, noise suppressing using Wiener filter, local particle contrast enhancement with the adaptive histogram equalization, edges enhancement using guided image filtering to improve the quality of the cryo-EM images and accommodate the low-SNR images, a general framework of micrographs preprocessing that has been used in our last three models [185] [186] [187] is applied to improve the quality of the low-SNR micrographs. In general, the preprocessing steps increase the particle's intensity and pre-grouping the pixels inside each particle makes them easier to be isolated.

6.2.2 Component 2: Fully Automated Single Particle Picking

This component is based on using our recent model DeepCryoPicker [187]. It is a fully automated deep neural network framework for single particle picking in cryo-EM. It is based on two components. First one is the fully automated training particle data generation using unsupervised learning algorithms. Second component is a deep neural network for

single particle detection, evaluation and picking. After particles are detected and picked using the DeepCryoPicker [187] using the preprocessed datasets, each detection result is projected back on its original one to pick two particle versions (original particle and the preprocessed one). In this case, two particles dataset are generated after this step. The first one is the preprocessing dataset 1, and the second one is the original particles dataset 1. Figure 6.6 shows the component 2 results after particle picking have been projected back on the original micrographs.



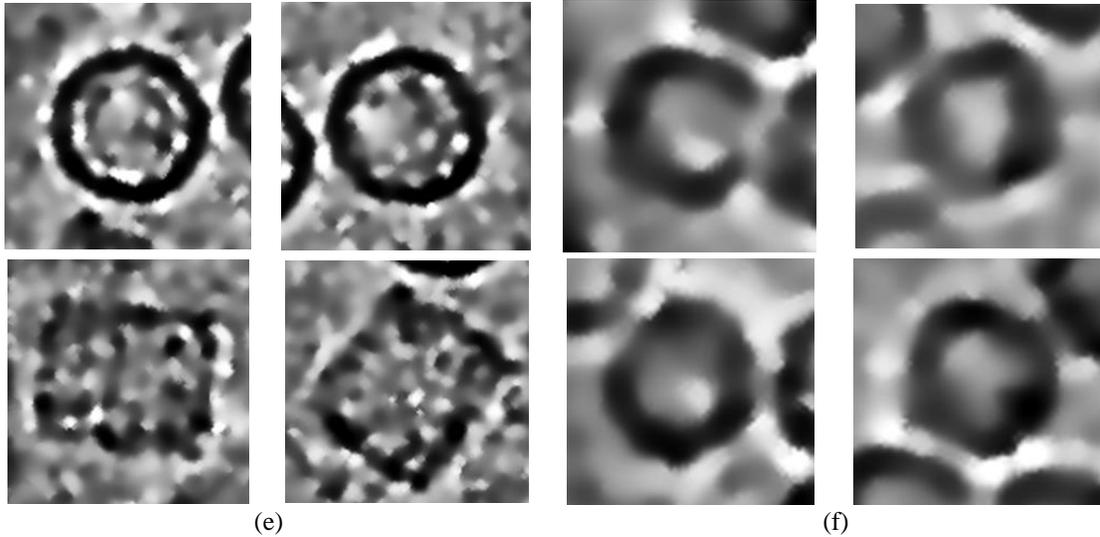


Figure 6.6: Fully automated particle picking experimental results (second component of the DeepCryoMap results). (a) and (b) the DeepCryoPicker [187] results using two datasets Apoferritin [183] and KLH dataset [184], (c) the original KLH particle picking results, (e) KLH preprocessing particle picking results, (d) the original Apoferritin particle picking results, (f) the Apoferritin preprocessing particle picking results.

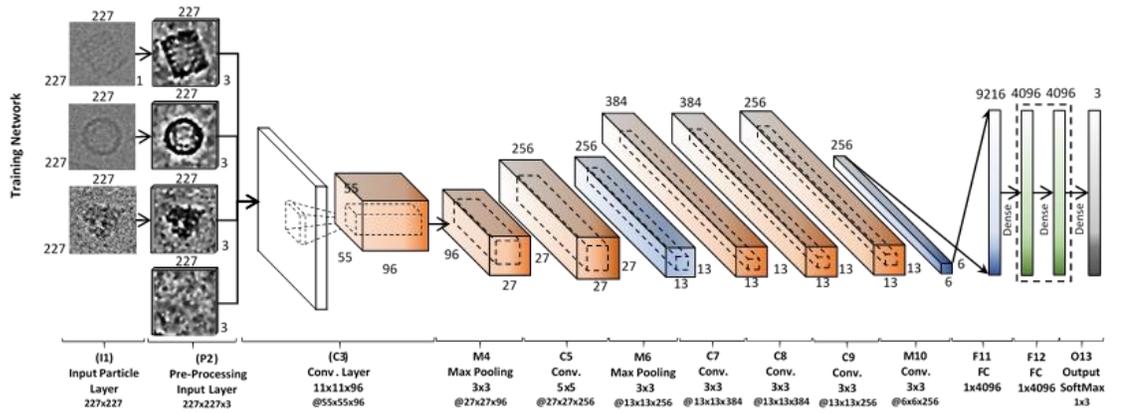
6.2.3 Component 3: Fully Automated Perfect 2D Particles-Selection

This component is designed to select perfect 2D particle images and generate perfect 2D particle mask for the next component which is the fully automated 2D particles alignment. Full automated perfect “good” 2D particle image evaluation, selection and perfect 2D binary mask particle generation is proposed for the fully particle alignment as a step to the 3D density map reconstruction. This component consists of two stages: (1) Stage 1: fully automated particle classification; (2) Stage 2: fully automated 2D particle mask generation based unsupervised learning approach.

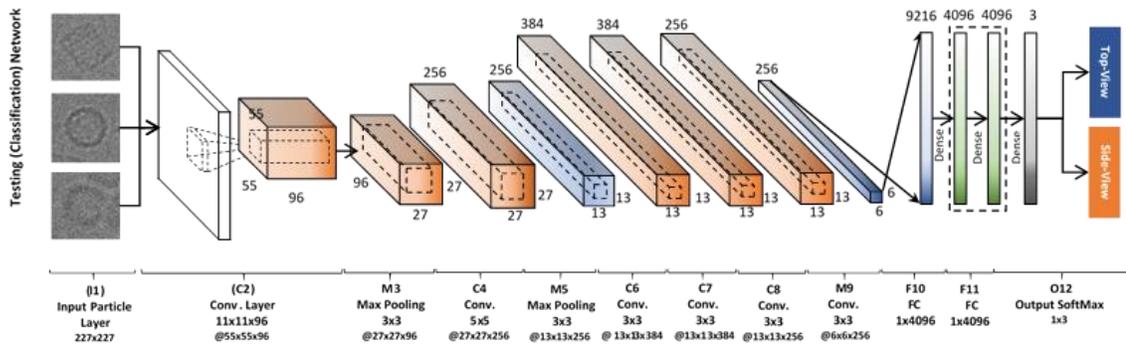
Stage 1: Fully Automated Particles Classification

The first stage of the fully automated perfect 2D particles-selection is the particles classification. In this stage, each individual particle image that is isolated from the second component is classified based deep network that is shown in Figure 6.7. The deep

classification network consists of many layers such as input layer, pre-processing layer, convolutional layers, sub-sampling layers, two fully connected layers, and one output layer. The main architecture of the DeepCryoPicker [187] has in total thirteen layers. The first and second layers (input and the pre-processing layer) come from the first the DeepCryoPicker [187]. The input layer takes the particles that have been already picked through the first model of the DeepCryoPicker [187]. Each particle has been classified based on the preprocessed version of each of the micrographs. The rest are five convolutional layers, three max-pooling (subsampling) layers, two fully connected layers, and one output layer. The results of the deep classification network are the label of each individual particle as is shown in Figure 6.7.



(a)



(b)

Figure 6.7: The architecture of the deep neural network used in DeepCryoPicker [25]. The convolutional layer and the subsampling layer are abbreviated as C and S, respectively. C3:11x11x96 means that in the third convolutional layer (C3) is comprised of 96 feature maps, each of which has a size of 11×11 , also. C3:@27x27 means that output feature maps dimensions are 27x27 pixels.

Stage 2: Fully Automated 2D Particle Mask Generation based Unsupervised Learning Approach

In order to evaluate each particle that has been picked through the DeepCryoPicker [187], a binary mask is generated base on using our clustering algorithms IBC [185] as is shown in Figure 6.8 (c). Then, each particle mask and its original particle image is evaluated and selected based tow different approaches.

Perfect “good” Side-view 2D Particles Images Selection

In terms of selection perfect 2D side-view (square) particle images, we develop an additional step called “good (perfect) side-view (square) 2D particle selection”. This step is based on using the individual binary mask for each particle as shown in Figure 6.8 (c). First, a binary mask of each clustered particle image is cleaned based on removal of the small and irrelevant objects. A cleaned binary image is generated first to generate the perfect particle mask as is shown in Figure 6.8 (d). The cleaned particle’s binary images have almost only the square objects (particles side view), in this case, we determine the connected components (for each objects) in the cleaned particle mask image, including a list of pixel area and locations for each one. Second, since some artifact object has a fully connected component and almost the same size the side view particle, we determine and extract the smallest rectangle region area. Third, determine the region area of each connected component (object) and keep the objects that are less that the smallest rectangle region area. Finally, bounding boxes are drawn around each discontinuous region

(rectangle region area) after determining the connected components (objects) in the image, including a list of pixel locations for each one and extract the centroid is the horizontal coordinate (or x-coordinate) and vertical. The perfect square (side-view) particle generation is shown below.

We use the Feret diameter measures approach [188] to measure and correct the particle object dimensions. New perfect particle shapes are generated based on the maximum and the minimum Feret diameter [188]. The maximum and minimum dimensions (width) of the particle object are used to identify the antipodal vertex pairs from the convex hull vertices set. Based on the new boundary box dimension, new perfect shapes are generated and inserted above each post-processed particle image. The last step eliminates the outliers object (overlapped particles) by defining the average particles size and eliminating the outliers that have particle size larger than the average size. Then, the new boundary box is drawn based on the dimension of new particle object shapes.

Algorithm 6.1 Perfect Side-View 2D Particles Shapes Generation and Selection

- 1: Label each connected components (objects) in the cleaned image and extract the total number of objects including a list of pixel locations for each one using MATLAB function (*bwlabel*).
 - 2: **for** each object in the cleaned clustered image **do**
 - 3: Smooth each object shape using Equation (13) with specific kernel size=5x5.
 - 4: **end for**
 - 5: Determine the connected components (objects) in the image, including a list of pixel locations for each one using MATLAB function (*bwlabel*).
 - 6: Measure set of properties specified by properties for each 8-connected component in the binary image using MATLAB function (*regionprops*).
 - 7: Calculate the ferret properties
 - 8: **for** each object in the cleaned clustered image **do**
 - 9: $P_{list} \leftarrow \text{PixelList}(\text{objects})$ /*Convert each object pixel to coordinates as an x-y order including a list of pixel locations for each one using MATLAB function (*PixelList*)*/
 - 10: $P_{hull} \leftarrow \text{PixelHull}(P_{list})$ /*Extract the pixel hull diamond shapes using MATLAB function (*PixelHull*) */
 - 11: $P_{pairs} \leftarrow \text{VerticPair}(P_{hull})$ /*Determine the maximum Feret diameter and its orientation (maximum diameter) */
-

```

12: Feterdimeter ← Min(Ppairs) /*Computes the minimum ferret diameter*/
13: Areabounding ← Min(Feterdim) /*Extract the minimum bounding box area*/
14: end for
15: for each object in the cleaned clustered image do
16: Determine the connected components (objects) in the image, including a list
    of pixel locations for each one and extract the centroid is the horizontal
    coordinate (or x-coordinate) and vertical coordinate (or y-coordinate) using
    MATLAB function (regionprops('centroid')).
17: Extract the bounding box dimension for each object (perfect square).
18: Determine and extract the 2D convex hull of the points (X, Y) for each object
    (particle) /* X and Y are column-vectors which presents a vector of point
    indices arranged in a counter-clockwise cycle around the hull */
19: end for
20: Construct the final cleaned output image containing only the perfect square
    objects classes occurring in each image by inserting a perfect square shape
    using the same dimension points(X, Y) and MATLAB function (Insert)
21: Determine and measure the connected components properties of all objects in
    the cleaned binary image including a list of pixel locations for each one using
    MATLAB function (bwlabel).
22: Determine and eliminate the outliers object using MATLAB function (outliers).
23: for each object in the cleaned clustered image do
24: Determine the average region area of each connected component (object)
    using MATLAB function (regionprops('Area')).
25: end for
26: for each object in the cleaned clustered image do
27: Determine the region area of each connected component (object) using
    MATLAB function (regionprops('Area')).
28: keep objects that are less that the average rectangle region area.
29: Extract each object position using MATLAB function (ismember).
30: end for
31: Determine the connected components (objects) in the image, including a list of
    pixel locations for each one and extract the centroid is the horizontal coordinate
    (or x-coordinate) and vertical coordinate (or y-coordinate) using MATLAB
    function (regionprops('centroid')).
32: Draw all bounding box for each discontinuous region (rectangle region area).
33: Construct the cleaned output image containing only the perfect square object
    classes occurring in each image.

```

The initial results of perfect particle shape generation are shown in Figure 6.8 (e). Sine some irrelevant object is still attached to the main particle masks, the perfect mask generated are not accurate based on the extracted bounding boxes are drawn around each discontinuous region as is shown in Figure 6.8 (f). To solve this issue, a binary mask of

each particle cluster image is post-processed based on removal of the small and irrelevant objects. A preprocessed binary image $I_{clustered}$ is generated first to shrink each object in the binary mask image by applying the morphological image operation on the original clustered image (binary) C_{Image} using image closing according to Equation (6.1)

$$I_{clustered} = (C_{Image} \oplus S_{sub_image}) \ominus S_{sub_image} \quad (6.1)$$

Where $I_{clustered}$ is the binary mask image, S_{sub_image} is the structural sub image using circular structure 5×5 , \ominus and \oplus denote erosion and dilation respectively. Then, the small objects and irrelevant ones are removed from the post-processed mask images as is shown in Figure 6.8 (g). The result of the perfect side-view particle shape generation is shown in Figure 6.8 (h).

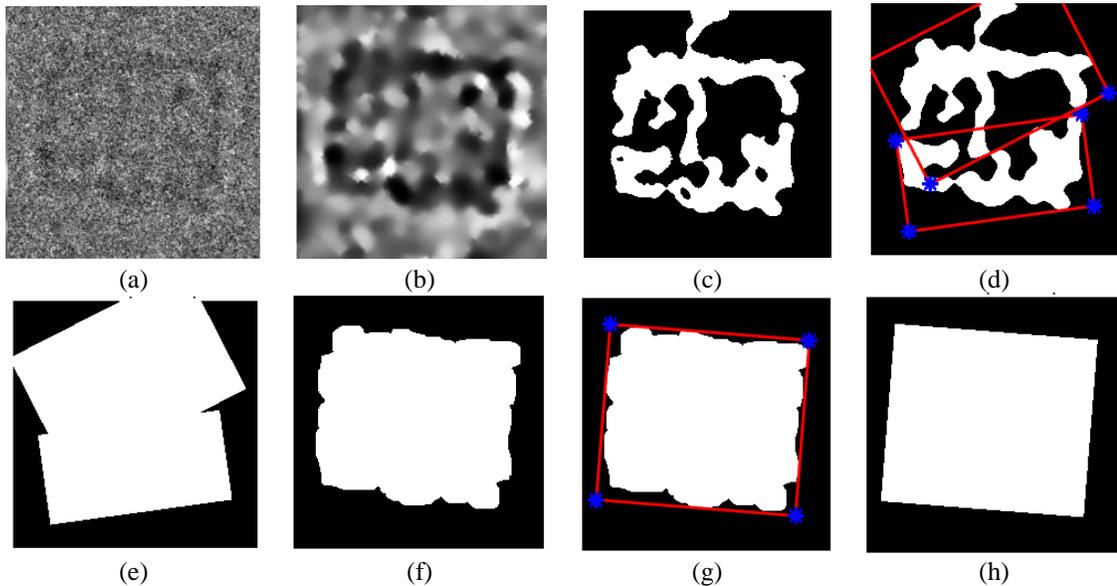


Figure 6.8: Perfect “good” side-view 2D particles images selection based perfect 2D mask generation. (a) original side-view particle image that is fully automated picked using the DeepCryoPicker [187] using KLH dataset [185], (b) preprocessed version of the original side-view particle, (c) initial binary mask of the (b) using IBC clustering algorithm in AutoCryoPicker [185], (d) Feret diameter detection of (c), (e) Initial binary mask generation of (a), (f) Postprocessing version of (c), (g) Feret diameter detection using (f), (h) Perfect binary mask generation.

The particle binary mask post-processing for particle image cleaning, small object, and irrelevant removal algorithm is shown below.

Algorithm 6.2 Particle Binary Mask Post-Processing

```

1: input:  $I_c$  /*cluster cryo-EM image */
2: return:  $I_{cc}$  /*cleaned cluster image */
3:  $I_{c1} \leftarrow imopen(I_c)$  /* Generate an intermediate clustered image by enlarge the
   small object using the image opening according to Equation (11) */.
4:  $L \leftarrow bwlabel(I_{c1})$  /* Label each object in the cluster image using MATLAB
   function (bwlabel) */.
5: for  $i=1$  to  $L$  do /* for each object in the intermediate clustered image*/
6:    $I_{object} \leftarrow state(L(k))$  /* determine the connected components (objects) in the
   image, including a list of indexing pixel locations for each one using
   MATLAB function (regionprops) */.
7:    $I_{object} \leftarrow bwareaopen(state(L(k)))$  /*remove the object that has not a fully
   connected edge using MATLAB function (bwareaopen)*/.
8: end for
9:  $obj_{number} \leftarrow ismember(I_{object})$  /*extract the number of object (particles)*/
10:  $L \leftarrow bwlabel$  /*label each object (particle)*/
11: for  $i=1$  to  $L$  do /* for each object (particles) */
12:   Do size filtering and roundness filtering
13:    $Areas \leftarrow [props.Area]$  /* Determine the region area of each connected
   component (object) using MATLAB function (regionprops('Area')) */
14:    $Threshold_{area} \leftarrow 50000$  /*determine the average objects value. */
15:    $keeperObjects \leftarrow threshold_{area}$  /* Keep objects that less than or equal to
   the average object's using MATLAB function (bwareaopen) */.
16:   Get actual index numbers instead of a logical vector
17:    $I_{c2} \leftarrow$  produce new binary image with only the small, round objects in it
18:    $I_{cc} \leftarrow bwareaopen(I_c)$  /*remove the object that has not a fully connected
   edge*/
19: end for
20: Construct the output image containing only the object circular “roundness”
   object classes in each image.

```

Perfect “good” Top-view 2D Particle Selection

Another regular and common shape of the particle protein is circle (top view), we develop an additional step called “good (perfect) 2D top-view (circular) image particle selection”.

This step is based on using the individual binary mask for each particle as shown in Figure 6.9 (c) and (k) using two different top-view particles that picked from Apoferritin [183]

and KLH [184] datasets. First, the same preprocessed versions of the particle images are used as is shown in Figure 8 (b) and (j). Then, the IBC clustering Algorithm [185] is used to generate an initial particle binary mask as is shown in Figure 6.9 (c) and (k). Second, the same post-processing steps are applied to generate cleaned binary masks as is shown in Figure 6.9 (d) and (l). Then, we develop an additional step called filed circular (top-view) object generation. This step is based on find the inner ring of the circular binary object (see Figure 6.9 (e) and (m)), and outer ring of the circular binary object (see Figure 6.9 (f) and (n)). The approximated points around the edges on each circular object is based on using the of convex hull of inner and outer ring points are used to extract the area between the inner and outer circular object ring as is shown in Figure 6.9 (g) and (o). The extracted circular object is smoothed using gaussian filter according to Equation (6.2):

$$g(m, n) = G_{\sigma}(m, n) \times f(m, n) \quad (6.2)$$

where $g(m, n)$ is the output smoothed image, $f(m, n)$ is the original input image (clustered), and G_{σ} is the gaussian kernel (mask) which is constructed according to Equation (6.3):

$$G_{\sigma} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{m^2+n^2}{2\sigma^2}\right)} \quad (6.3)$$

where σ is the sigma (which represents the signal width), m , and n is the image dimension. Then, the modified Circular Hough Transform algorithm (CHT) in AutoCryoPicker [173] is used to generate a perfect circle on top of each particle's mask as is shown in Figure 9 (h) and (p).

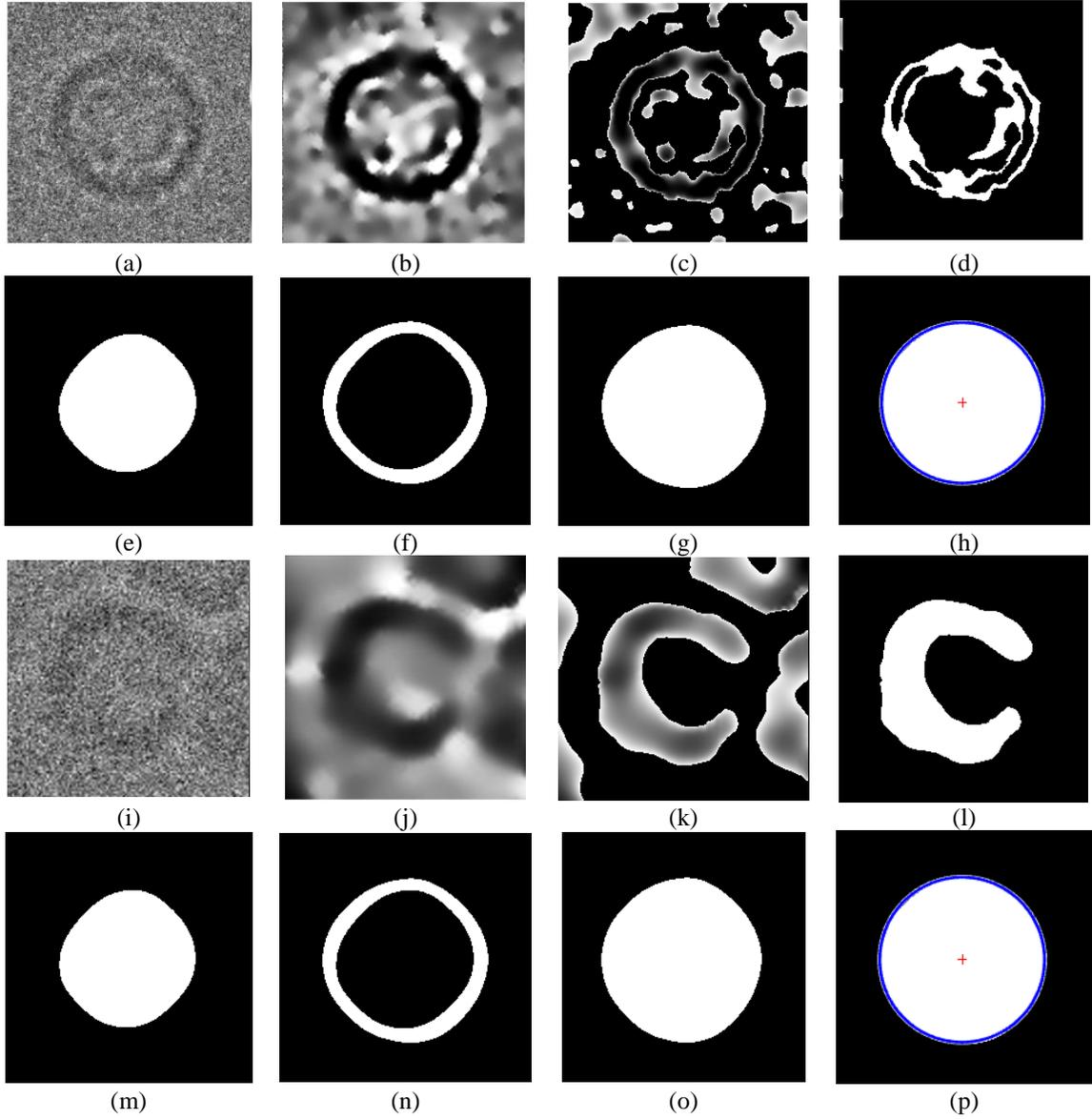


Figure 6.9: Perfect “good” top-view 2D particles images selection based perfect 2D mask generation. (a) and (i) two original top-view particle images that is fully automated picked using the DeepCryoPicker [25] using Apoferritin [21] and KLH dataset [23], (b) and (j) the preprocessed version of the original top-view particle images of (a) and (i) respectively, (c) and (k) the initial binary mask of the (b) and (j) using IBC clustering algorithm in AutoCryoPicker [23], (d) and (l) the cleaned circular clustered images of (c) and (l) respectively, (e) and (f) the inner and outer circular mask extraction of the (d). (m) and (n) the inner and outer circular mask of (l). (k) and (o) are the filled circular binary masks of (f) and (n) respectively. (h) and (p) perfect top-view binary mask generation of (g) and (o) respectively.

We test each individual particle’s mask size and verify if it is a perfect full circle and label it as either a “good 2D particle” or as a “bad 2D particle”. We test each top-view

particle by calculating the average roundness value for the whole top-view (circular) particles. This is determined by computing the area and perimeters using the connected component particle mask's pixel index list and the circularity based on the Equation (6.4):

$$Circularities = \frac{allPerimeters^2}{4 \times \pi \times allAreas} \quad (6.3)$$

The perfect “good” top-view 2D particle selection-based Hough Transform algorithm (CHT) is shown below [185].

Algorithm 6.3 Perfect “good” Top-view 2D Particle Generation

- 1: Extract the convex hull points of the outer circular object ring points
 - 2: Remove the outer circular object ring points based using morphological image operation “*imerode*” with structural size=10.
 - 3: Multiply the erosion image with the original binary mask
 - 4: Extract the convex hull of the inner circular object ring points
 - 5: Extract the area between the inner and the outer convex hull of the circular object
 - 6: Generate a gaussian kernel (mask) using $G_\sigma = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{m^2+n^2}{2\sigma^2}\right)}$, where σ is the sigma (which represents the signal width), m , and n is the image dimension.
 - 7: Smooth the input image with a Gaussian filter to reduce noise and unwanted details and textures by using $g(m, n) = G_\sigma(m, n) \times f(m, n)$ where $g(m, n)$ is the output image, $f(m, n)$ is the original input image, and G_σ is the gaussian kernel (mask).
 - 8: Compute gradient of $g(m, n)$ using any of the gradient operators to get M by using $M(m, n) = \sqrt{(g_m^2(m, n) + g_n^2(m, n))}$
 - 9: Where g_m is the gradient in the x-axis direction, g_n is the gradient in the y-axis direction.
 - 10: Threshold the gradient M by $M_T(m, n) = \begin{cases} M(m, n) & \text{if } M(m, n) > T \\ 0 & \text{Otherwise} \end{cases}$
 - 11: /* *Hough Transform Begin**/
 - 12: **for** each edge point **do**
 - 13: Draw a circle with centre (x, y) in the edge point with r where (x, y) is the image pixels with position x , and y , r is the circular radius.
 - 14: Increment all coordinates (x, y) that the perimeter of the circle passes through in the accumulator.
 - 15: Find one or several maxima in the accumulator
 - 16: Map the found parameters (r, a, b) corresponding to the maxima back to the original image, where a , and b is the centre of the maxima.
 - 17: **end for**
 - 18: /* *Hough Transform End**/
-

```

19: Use the detected centre to construct a perfect circular object with the average
    diameter.
20:  $L \leftarrow \text{bwlabel}(I_{c1})$  /* Label each object in the cluster image using MATLAB
    function ( $\text{bwlabel}$ ) */.
21: for  $i=1$  to  $L$  do /* for each object in the intermediate clustered image*/
22:    $I_{\text{object}} \leftarrow \text{state}(L(k))$  /* determine the connected components (objects) in
    the image, including a list of indexing pixel locations for each one using
    MATLAB function ( $\text{regionprops}$ ) */.
23:    $I_{\text{object}} \leftarrow \text{bwareaopen}(\text{state}(L(k)))$  /*remove the object that has not a
    fully connected edge using MATLAB function ( $\text{bwareaopen}$ )*/.
24: end for
25:  $\text{obj}_{\text{number}} \leftarrow \text{is member}(I_{\text{object}})$  /*extract the number of object (particles)*/
26:  $L \leftarrow \text{bwlabel}$  /*label each object (particle)*/
27: for  $i=1$  to  $L$  do /* for each object (particles) */
28:   Do size filtering and roundness filtering
29:    $\text{Areas} \leftarrow [\text{props.Area}]$  /* Determine the region area of each connected
    component (object) using MATLAB function ( $\text{region props('Area')}$ ) */
30:    $\text{Perimeters} \leftarrow [\text{props.Perimeter}]$  /* Determine the region perimeters of
    each connected component (object) using MATLAB function ( $\text{region props}$ 
    ( $\text{Perimeter}$ )) */
31:    $\text{Circularities} \leftarrow \text{allPerimeters}^2 / ((4 \times \pi \times \text{allAreas}))$  /* Determine
    the region circularities of each connected component (object) using
    Equation (12).
32:    $\text{Threshold}_{\text{area}} \leftarrow 50000$  /*determine the average objects "roundness"
    circularities value. */
33:    $\text{keeperObjects} \leftarrow \text{circularities} < 3 \ \& \ \text{Areas} < \text{threshold}_{\text{area}}$  /* Keep
    objects that less than or equal to the average object's "roundness"
    circularities value using MATLAB function ( $\text{bwareaopen}$ ) */.
34:   Get actual index numbers instead of a logical vector
35:    $I_{c2} \leftarrow$  produce new binary image with only the small, round objects in it
36:    $I_{cc} \leftarrow \text{bwareaopen}(I_c)$  /*remove the object that has not a fully connected
    edge*/
37: end for
38: Construct the output image containing the perfect circular "roundness" object

```

6.2.4 Component 4: Fully Automated Particles-Alignment

In terms of reconstruct the 3D density map the particles need be aligned and centered. This step provides more information that can helps and uses towards to reconstruct the 2D density map (three-dimensional model). Once the particles are picked and selected the prefect 2D mask for each particle is generated, the fully automated single particle

alignment is performed to perfectly align the particle images. This component consists of two stages: (1) Stage 1: fully automated side-view particle alignment; (2) Stage 2: full automated top-view (circular) particle alignment.

Stage 1: Fully Automated Side-View Particle Alignment

The first stage of the fully automated particle alignment is the side-view (square) particle alignment. Particles alignment basically relies on placing the image particle into a similar orientation [189]. Based on the relative plane of the two images, particles are shifted by $[x, y]$ or/and rotated by (φ) . Technically, image alignment needs to determine the correlation parameters $[x, y, \varphi]$ to map the images perfectly. Image registration aims to geometrically estimated and match two images based on different viewpoints [189].

Mathematically, the image registration bases on find the best geometrical transformation that match the same points on two images. Let assume that $I_F(x, y)$ is the fixed or the reference image and $I_M(x, y)$ is the moved image (image that needs to be aligned). The mathematical approach to estimate the geometrical transformation for the image registration $T(x, y)$ is based on the Equation (6.4) [190]:

$$T(x, y) = (T_1(x, y), T_2(x, y)) \quad (6.4)$$

Such that using the estimated geometrical transformation $T(x, y)$ to register the moved image $I_M(x, y)$ by using resulting a close image $I_c(x, y)$ to the reference image $I_R(x, y)$ suing the following Equation (6.5) [190]:

$$I_c(x, y) = I_M(T(x, y)) \approx I_R(x, y) \quad (6.5)$$

Thus, the image registration can be formatted as a maximalization problem-based optimizer function that is shown on the Equation (6.6) [190]:

$$I_c(x, y) = I_M(T(x, y)) \approx I_R(x, y) \quad (6.6)$$

Where T_{opt} denotes as the optimal geometrical transformation for $I_R(x, y)$ and $I_M(x, y)$ matching based on the selected metric of the measurement similarity (S) among the specific transformation (\mathcal{T}). Finally, the geometrical transformation $T(x, y)$ follows the 2D parametric model to estimate the continuous bivariate function to estimate the certain regularity conditions [21] based on the following Equation (6.7) [190] [191]:

$$\begin{cases} T_1(x, y) = \alpha(x \cos \Delta\phi + y \sin \Delta\phi) + \Delta x \\ T_2(x, y) = \alpha(x \sin \Delta\phi - y \cos \Delta\phi) + \Delta y \end{cases} \quad (6.7)$$

Where $(\Delta x, \Delta y, \Delta \phi)$ are the three geometrical (motional) parameters and α is the geometrical scaling parameter.

Intensity-based image registration is an image registration process which is based on the intensity image similarity to define the 2D geometrical transformation for minimizing or maximizing the similarity metric. It is basically based on the estimated the internal geometrical transformation matrix $T(x, y)$ after applying the image transformation (bilinear interpolation [192]) on the two images $I_F(x, y)$ and $I_M(x, y)$. The idea from applying the bilinear interpolation [192] on both images is that the bilinear interpolation [192] is one of the resampling techniques (image scaling) on the computer vision and the image processing which uses to transform that image to a specific transformation (\mathcal{T}). Then, the measurement similarity (S) of the transformed images to estimate the geometrical transformation $T(x, y)$.

To do the intensity-based image registration, we need to calculate two parameters, image registration optimizer (O) and the similarity metric (S). First, the image similarity metric (S) is calculated between two images using a mean square error (MSE) as a

confirmation metric that uses to measure the similar (S) between the two transformed images $I_R(x, y)$ and $I_M(x, y)$ based on the following Equation (6.8) [193]:

$$MSE = \frac{1}{m \times n} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} [I_c(x, y) - I_F(x, jy)]^2 \quad (6.8)$$

Then, the regular step gradient descent optimization [194] is used to estimate the optimizer parameters. The regular step gradient descent optimization [34] is a first order iterative algorithm uses to adjust the geometrical transformation parameters by following the gradient of the image similarity metric in the direction of the extrema [195]. It uses constant step size (length) between the computations along the gradient until the direction is changed. By default, the step size (length) is reduced according to the relaxation factor. Equation (6.9) shows the typical form of the gradient descent optimization is used to estimate the image registration optimizer [195].

$$X_{\eta+1} = X_{\eta} - \gamma \nabla F(X_{\eta}) \quad (6.9)$$

Where $\gamma \nabla F(X_{\eta})$ gradient factor that is a subtraction from X_0 to make it moves toward to the global minimum (stop condition), and X_0 is the local minimum of the main function F which is in our case the similarity metric (S). The two estimated parameters (image similarity metric and an optimizer) are used in the image registration function that is show in Equation (6.6) to register two images. The similarity metric defines how similar the two images are, and optimizer defines the methodology for minimizing or maximizing the step size (gradient factor) in Equation (6.8) based on the similarity metric. Finally, the image registration function maps each point in the moved image $I_M(x, y)$ into the corresponding point in the reference image $I_R(x, y)$ based on the estimated correlation parameters $[x, y, \varphi]$ from the similarity metric and optimizer functions.

Typically, different geometrical transformation can be used to register the two images such as translation, scaling, rotation, and affine transformation. In the translation transformation, the point $P(x, y)$ in the moving image $I_M(x, y)$ is translated to a new point $P(x', y')$ (see Equation (6.10) and (6.11)) by adding the translation correlation factor (d_x, d_y) (see Equation (6.12)) [196]:

$$P = T + P' \quad (6.10)$$

$$P = \begin{bmatrix} x \\ y \end{bmatrix}, P' = \begin{bmatrix} x' \\ y' \end{bmatrix}, T = \begin{bmatrix} d_x \\ d_y \end{bmatrix} \quad (6.11)$$

$$x' = x + d_x \text{ and } y' = y + d_y \quad (6.12)$$

Using the scaling transformation, the new point $P(x, y)$ is scaled along x and y axis to a new point $P(x', y')$ (see Equation (6.13) and (6.14)) by multiply x and y by the scaling factors S_x and S_y (see Equation (6.15)) [196]:

$$P = S \times P' \quad (6.13)$$

$$PP = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} \quad (6.14)$$

$$x' = S_x \times x \text{ and } y' = S_y \times y \quad (6.15)$$

By using the rotational transformation, the new point $P(x, y)$ is rotated around the origin to a new point $P(x', y')$ by an angle θ (see Equation (6.16), (6.17), and (6.18)) [36]:

$$x' = x \times \cos \theta - y \times \sin \theta \quad (6.16)$$

$$y' = x \times \sin \theta + y \times \cos \theta \quad (6.17)$$

$$P = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} \quad (6.18)$$

In some cases, the translation and rotation are not enough. However, the scaling is necessary to correct the points transformation in the $I_M(x, y)$. Therefore, the affine transform scales the translation and rotational points (see Equation (6.19)) based on using the two-dimensional shear transformation as is showing in Equation (6.20) [196]:

$$T_{Scale}(x, y) = \begin{bmatrix} x' \\ y' \end{bmatrix} = S \times \begin{bmatrix} x \\ y \end{bmatrix} \quad (6.19)$$

$$SH_x = \begin{bmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } SH_y = \begin{bmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.20)$$

where a and b are the proportionality constants along axis x and y , respectively [196]. Since in the second stage of the third component of our DeepCryoMap framework “fully automated perfect 2D particles-selection” which is “stage 2: fully automated 2D particle mask generation based unsupervised learning approach”, perfect binary masks are generated, we propose a fully automated approach for perfect side-view particle alignment based automatic intensity-based Image registration using the perfect generated particle binary masks. In terms of the fully automated approach, we use the binary masks instead of the original particle images for two reasons. The first one is it easy to automatically generate a reference image than manually select one. Second, it is very easy to find the correlation points (corresponding corners) in the generated mask than the original particle image since the signal-to-noise ratio is very low (low intensity value) that will not help our fully automated particles alignment-based intensity image registration. The main stapes to do the fully automated particle alignment-based intensity image registration using the perfect generated binary masks are as follow: First, we calculate the average binary particle object sizes and generate an artifice frontal view reference image (side-view) particle as is

shown in Figure 6.10 (a). Second, for each particle image, we use the original particle image and the generated binary mask as is shown in Figure 6.10 (b) and (c). Then, we use intensity-based automated image registration to align the perfect generated binary mask of each particle based on the generated reference binary mask (frontal view) using deferent geometrical transformation (see Figure 6.10 (e)-(i)). After the perfect alignment is done, we extract the angles of both aligned object and the original mask $\theta_{original}$ and $\theta_{aligned}$ which is the angle between the x-axis and the major axis of the object that has the same second-moments as the region (see Figure 6.10 (j) and (k)). Then, we extract the orientation angle $\theta_{orientation}$ based the on the difference between the aligned angle and the original angle. Finally, we use the orientation angle $\theta_{orientation}$ to rotate the original particle image as is shown in Figure 6.10 (l).

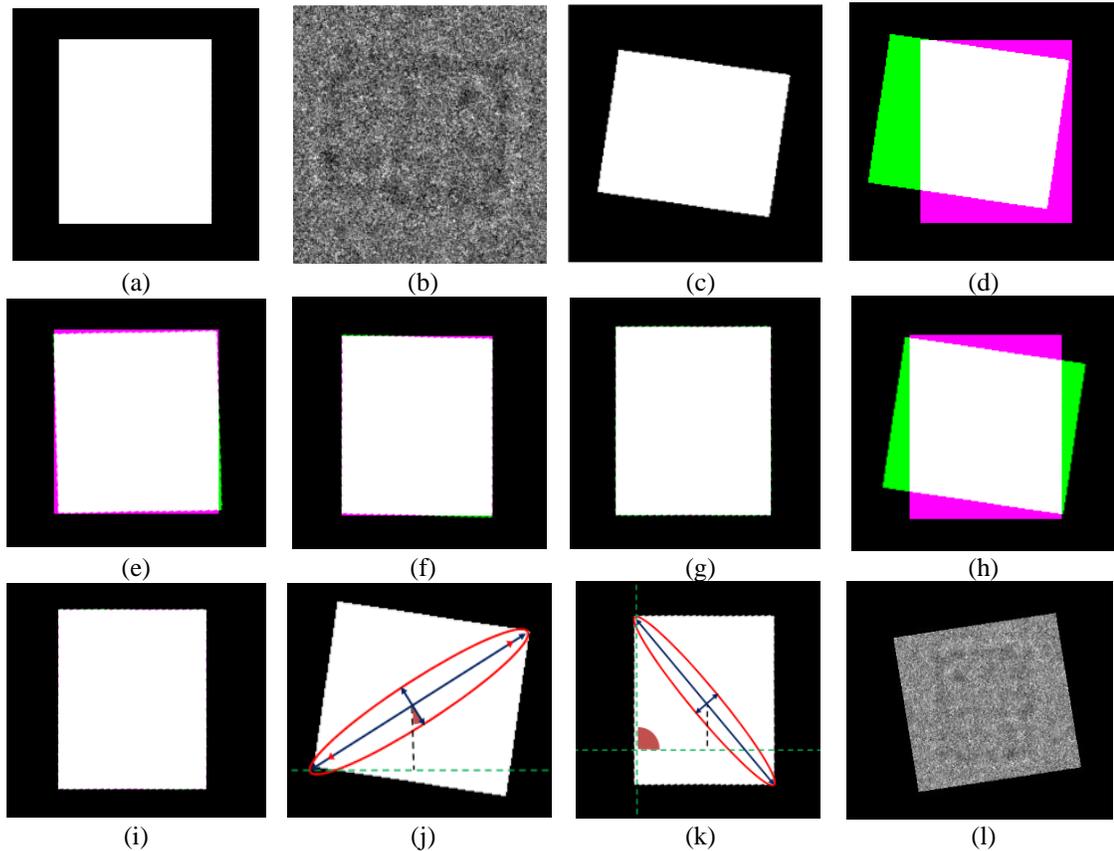


Figure 6.10: Fully automated side-view particle alignment using KLH dataset [184], (a) Artificial frontal view reference image generation based average binary particle object sizes (b) Original side view particle image (moving), (c) Perfect generated binary mask of (b). (d) Unalignment images projection (references (a) and moving (c)), (e) Default image alignment (initial registration). (f) Optimizer adjustment and metric configuration-based image registration. (g) Image registration based increasing the maximum iteration number. (h) Image registration-based optimization and rigid [189] transformation. (i) Image remigration using affine transform, (j) Original binary mask particle's orientation, (k) Aligned binary mask particle's orientation, (l) Final particle alignment result.

The higher resolution of the 3D reconstructed density map bases on much improved on the signal-to-noise-ratio (SNR). To improve the SNR needs to average over fewer particles to conduct the resolution. Although, the researchers used to manually pick particles and do the 2D image averaging to remove some false positive particles from the whole data. Instead of using the individual particles extraction from micrograph background that is proposed in RELION [178] which uses a manually user-defined radius a circle (normalisation procedure) to extract each particle image in a background area (outside the circle) and a particle area (inside the same circle) [198] and do the image averaging, we proposed a localized image averaging approach. The localized particle image is generated based on using the binary mask for each individual particle as is shown in Figure 6.11. In this case the original aligned particle image (see Figure 6.11 (a)) is multiply by the 2D aligned mask (see Figure 6.11 (b)) to generate the perfect 2D localized particle images (see Figure 6.11 (d)) based on the Equation (6.21) [193]:

$$I_{localized} = I_{original} \times I_{mask} \quad (6.21)$$

Where $I_{localized}$ is the localized 2D particle image, $I_{original}$ is the original 2D particle image, and I_{mask} is the 2D perfect mask.

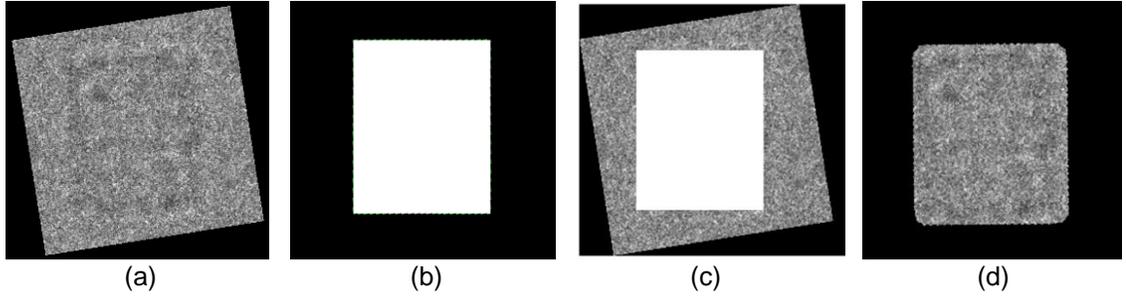


Figure 6.11: Localized 2D side-view aligned particle image generation. (a) Original aligned particle image, (b) Aligned binary mask particle of the original image (a), (c) Perfect particle binary mask image and original particle image projection, (d) localized 2D side-view aligned particle image.

The fully automated side-view particle alignment-based intensity image registration and particle masks is shown below and the whole framework of the fully automated side-view particle alignment-based intensity image registration is illustrated in Figure 6.12.

Algorithm 6.4 Automated Side-View Particle Alignment

```

1:  /*Reference mask image reconstruction*/
2:  Import the whole binary mask particle images
3:  for each particle image do /* for each binary mask image (particles) */
4:    /* Hough Transform Begin*/
5:    for each edge point do
6:      Draw a circle with centre  $(x, y)$  in the edge point with  $r$  where  $(x, y)$ 
      is the image pixels with position  $x$ , and  $y$ ,  $r$  is the circular radius.
7:      Increment all coordinates  $(x, y)$  that the perimeter of the circle passes
      through in the accumulator.
8:      Find one or several maxima in the accumulator
9:      Map the found parameters  $(r, a, b)$  corresponding to the maxima back
      to the original image, where  $a$ , and  $b$  is the centre of the maxima.
10:   end for
11:   /* Hough Transform End*/
12:   Compute the similarity metric using Equation (9)
13:   Specify the geometrical transformation.
14:   Do the intensity-based image registration using Equations (10)-(20).
15:   /* Particle Binary Mask Based Image Alignment Begin*/
16:   repeat
17:     Align the moving binary mask particle image  $I_{binary}(x, y)$  using
     Equation (6)
18:     
$$I_{binary\_aligned}(x, y) = I_{binary\_moved}(T_{opt} = \underset{T \in \mathcal{T}}{\operatorname{argmax}} S(I_{binary\_Reference}, I_{Moved\_original}(T)))$$


```

```

19:   until convergence or reach the maximum iterations number
20:   /* Particle Binary Mask Based Image Alignment End*/
21:   Extract the angle  $\theta_{aligned}$  between the x-axis and the major axis of the
      object that has the same second-moments as the region on the aligned binary
      image, returned as a scalar.
22:   Extract the orientation  $\theta_{original}$  angle between the horizontal dotted line and
      the major axis in the original binary mask
23:    $\theta_{orientaion} = |\theta_{aligned} - \theta_{original}|$ 
24:    $I_{original\_aligned}(x, y) = I_{original\_moved}(R_{\theta_{orientaion}}(x, y))$  /* use the
      extracted angle  $\theta_{orientaion}$  to rotate the original particle image
25: end
26:  $I_{localized} = I_{original} \times I_{mask}$  /*Construct the localized original aligned image
       $I_{aligned}$  by multiply the aligned particle image by the aligned particle mask*/

```

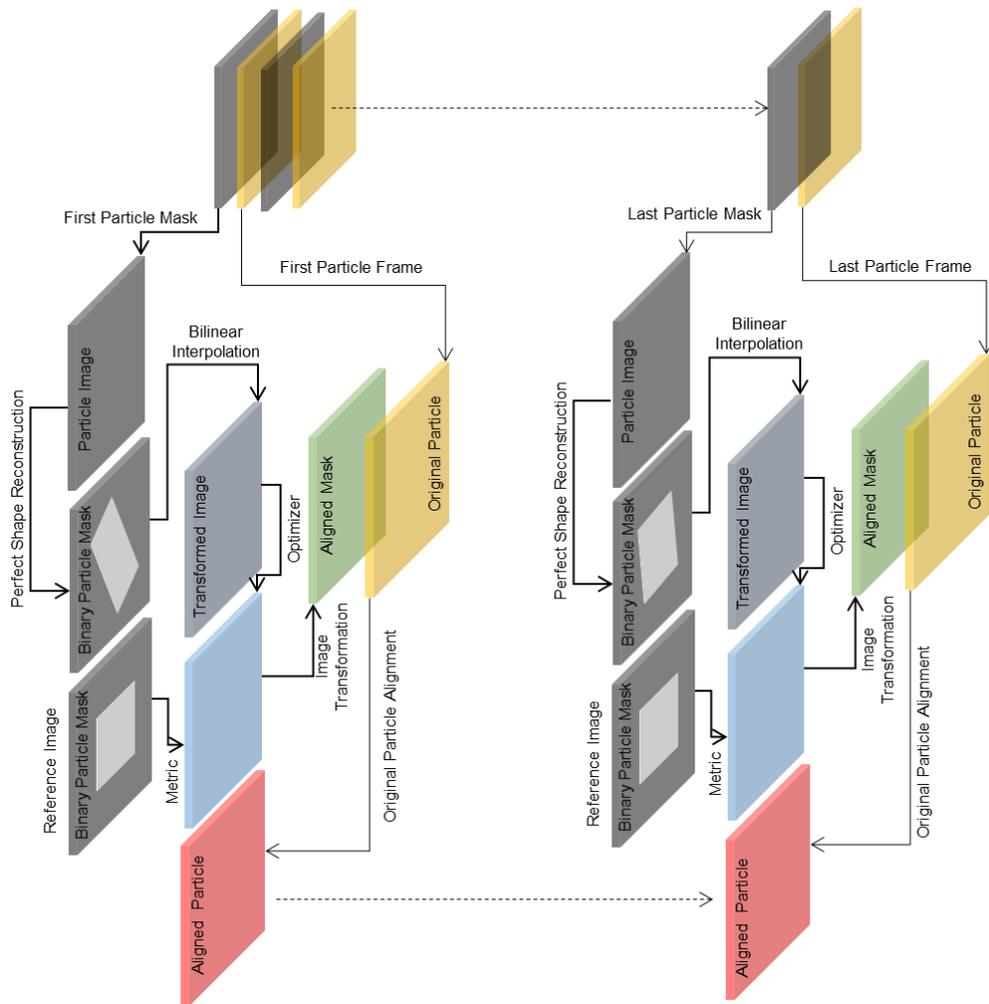


Figure 6.12: Fully automated side-view particle image alignment using intensity-based image registration.

Stage 2: Fully Automated Top-View Particle Image Alignment

The second stage of the fully automated particle alignment is designed to align the common type of the particle images top-view (circular) particle images. The top-view particle images are aligned based on centralize all particles together on the same point which is a common way to align the circular particle. Since one particle (original form) might be heavy noisy compared to another one, it is very hard to find the same centre point to centralized them. Also, circular particles can be mis centred especially those that have hollow in the particle ring. To come up with perfect fully alignment approach, we propose a localized top-view centralization approach for top-view (circular) particle alignment. First, we used the localization approach that is applied on the side-view particle images to produce localized top-view particle images (see Figure 6.13 (d)).

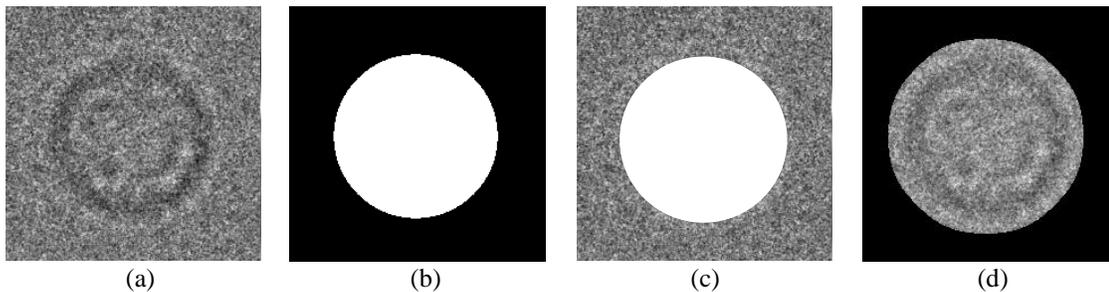


Figure 6.13: Localized 2D top-view aligned particle image generation. (a) Original aligned particle image, (b) Perfect 2D binary mask particle of the original image (a), (c) Perfect particle binary mask and original particle image projection, (d) localized 2D top-view aligned particle image.

Second, it is more accurate to find the same centre point (center of the binary circle) and the ring hollow is not a part of the centralization issue anymore. The centre of the circular object (binary) is defined as an average of all points in the circular shape [193]. Suppose that that circular shape consists of n points x_1, x_2, \dots, x_n (white pixels) as is shown in Figure 6.14 (a), the centroid (centre) of the circular white (binary) object is defined based on the Equation (6.22) [193]:

$$centerid = \frac{1}{n} \sum_{i=1}^n x_i \quad (6.21)$$

In our case, we use the modified CHT [23] to extract the exact center point of each particle's mask as is shown in Figure 6.17 (b) and (f). Then from the extracted center point we draw a new candidate box the takes the same dimension ($x_{width}, x_{heights}$). New bounding boxes are drawn around each top-view region (rectangle region area) after increase each object center (x, y) using the same factor vale and calculate the bounding boxes dimensions ($x_{width}, x_{heights}$). This approach allows the particles that have hollows (rings) to be accurately aligned based on the same particle mask extracted centre. Centralized based particle alignment based perfect binary mask generation allows the particles to be placed (aligned) in the same exact point (centre) which will help the 3D map reconstruction to overlap the particles in which they need to be aligned, shifted in the plane.

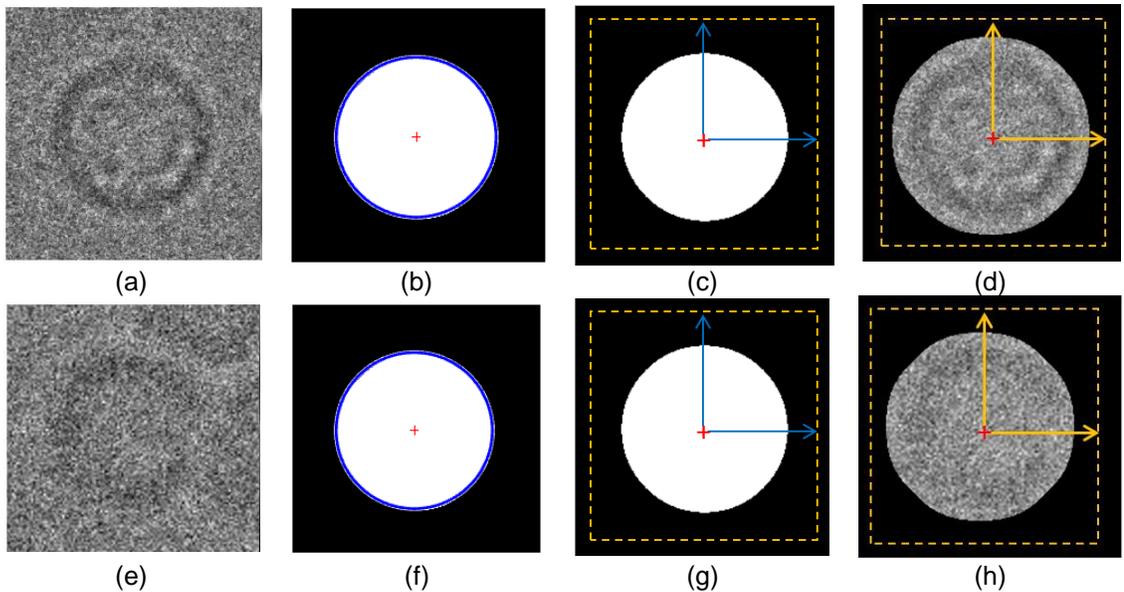


Figure 6.14: Fully automated top-view particle image alignment. (a) and (b) original top-view particle images from Apoferritin dataset [21], (b) and (f) Center point extraction using the modified CHT algorithm [23] using the generated perfect binary masks for (a) and (b) respectively, (c) and (g) Centralized top-view particle binary mask alignment result, (h) Centralized top-view of the localized particle alignment image result.

The extracted correlation point (center) that is determined based on extract the center of the perfect binary mask (see Figure 6.15 (b)) allows the particle to be shifted along the fixed center point (see Figure 6.15 (c)). In this case, all the particles are cross correlated to each other and shifted to as necessary to the same center point. Figure 8.18 shows an example of two top-view particles from two different datasets before and after the centralized alignment based perfect generated binary mask. The particle image is shifted as best as possible to be centralized aligned.

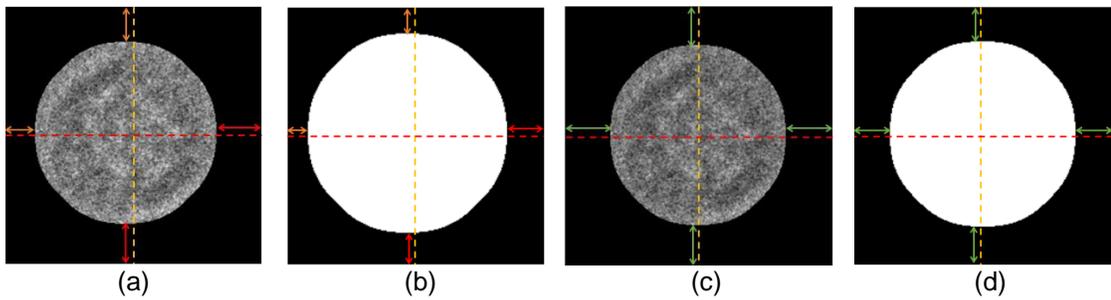


Figure 6.15: Localized 2D particle images before and after the centralized based particle image alignment, (a) localized 2D top-view particle image from Apoferritin dataset [21] before the centralized particle image alignment, (b) localized 2D top-view particle binary mask of (a) before the centralized particle image alignment, (c) localized 2D top-view particle image after centralized based particle image alignment, (d) localized 2D top-view particle mask image after centralized based particle image alignment.

The fully automated approach for centralized top-view particle alignment-based particle mask is shown below and illustrated in Figure 6.16.

Algorithm 6.5 Side-View Particle Alignment Based Intensity Image Registration

- 1: Import the binary mask particle images I_{mask}
 - 2: Import the original particle images $I_{original}$
 - 3: $I_{localized} = I_{original} \times I_{mask}$ /*Construct the localized original aligned image $I_{aligned}$ by multiply the aligned particle image by the aligned particle mask*/
 - 4: **for** each particle image **do** /* for each binary mask image (particles) */
 - 5: /* Hough Transform Begin*/
 - 6: **for** each edge point **do**
 - 7: Draw a circle with centre (x, y) in the edge point with r where (x, y) is the image pixels with position x , and y , r is the circular radius.
 - 8: Increment all coordinates (x, y) that the perimeter of the circle passes through in the accumulator.
-

-
- 9: Find one or several maxima in the accumulator
 - 10: Map the found parameters (r, a, b) corresponding to the maxima back to the original image, where a , and b is the centre of the maxima.
 - 11: **end for**
 - 12: */* Hough Transform End*/*
 - 13: $[x, y] \leftarrow Hough Transform(BinaryMask)$ */*Extract the centre of the binary circular binary object (particle mask) */*.
 - 14: $[x] \leftarrow x + factor, [y] \leftarrow y + factor$ */* increase the dimensions of the candidate box using the same factor value */*
 - 15: Draw all bounding box for each discontinuous region (rectangle region area).
-

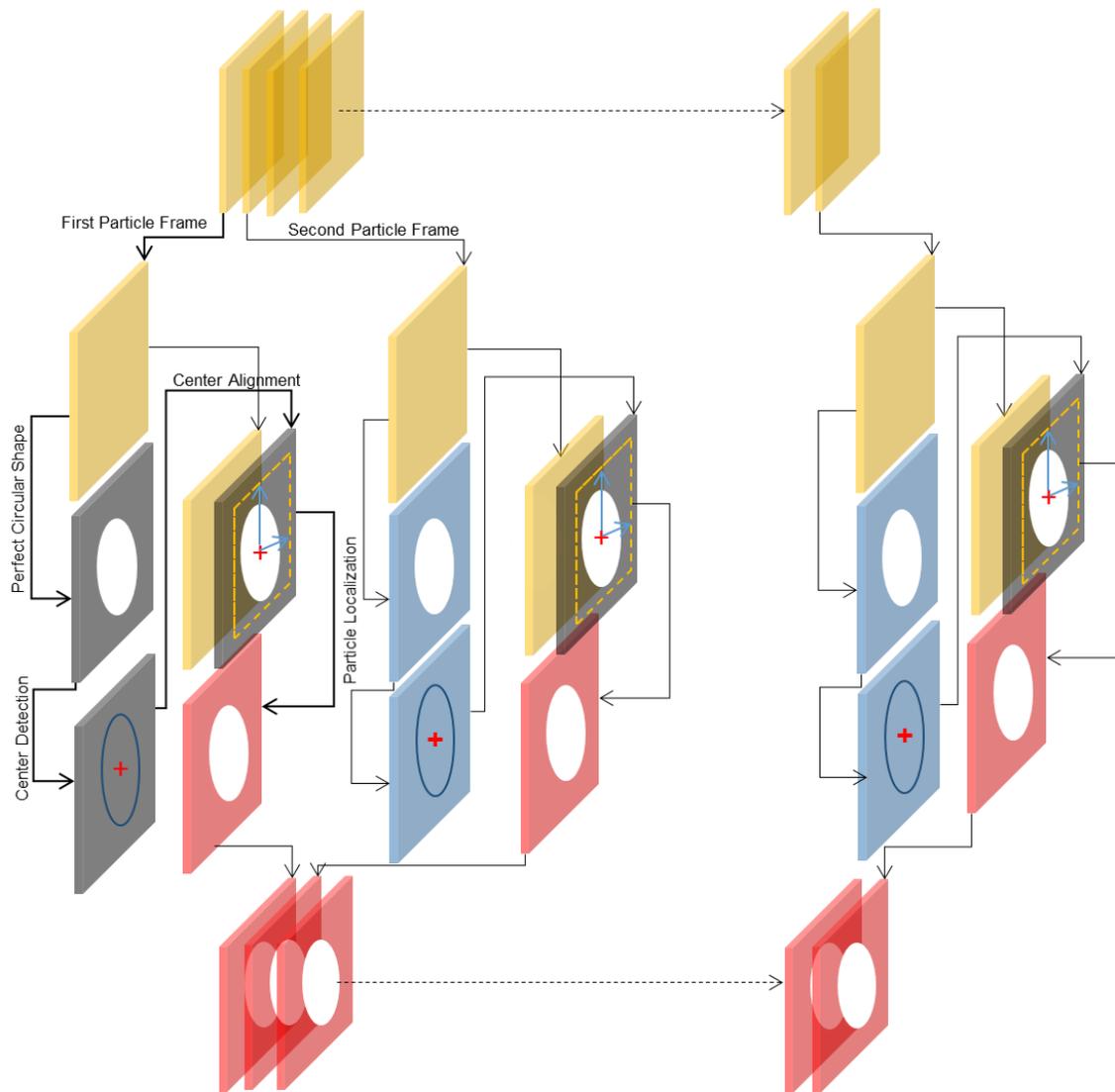


Figure 6.16: Fully automated localized 2D top-view particle image alignment using centralized image alignment based perfect binary 2D image.

6.2.5 Component 5: 3D Density Map Reconstruction

The basic idea of reconstructing the 3D density map based cryo-EM data bases of project the density depth of several thousands of 2D cryo-EM particles. Basically, the Fourier coefficient-based Fourier Transformation (FT) [199] is used to represent the 2D particles in another space (Fourier space). In which the structural of each 2D particle is represented in Fourier coefficients. In this case, the Fourier synthesis is used to reconstruct the 3D density map through its 3D Fourier transformation based the direction of the projection [175]. This approached requires huge number of 2D particles to build such significant Fourier coefficients the represent the particle object structure. To come up with the same descent 3D density map using a smaller amount of 2D particle images, we propose a new localized approach for 3D density map reconstruction based the 2D shape appearance of the real object. Localized based 3D density map reconstruction approach bases on extract the structural information-based particle object from every two 2D particle images.

Structural based motion information is a process that estimates the 3D structural (3D matrix) from set of 2D images [200]. Different steps are implemented in this component to achieve the 3D density map reconstruction-based particle structural motion information. First, match the sparse set of points between every two 2D particle images based perfect 2D image alignment. Second, estimate the fundamental matrix (3D matrix). Third, track a dense set of points between the two images that illustrates the estimated structure of the object (particle in the 3D). Then, determine the 3D locations of the matched points using triangulate. Finally, recover the actual 3D map based metric reconstruction. The 3D density map in this case is building based on only first two particle images. At the end the average of the all localized 3D density map represents the final 3D density map.

The whole framework of the localized 3D density map reconstruction is illustrated in Figure 6.17.

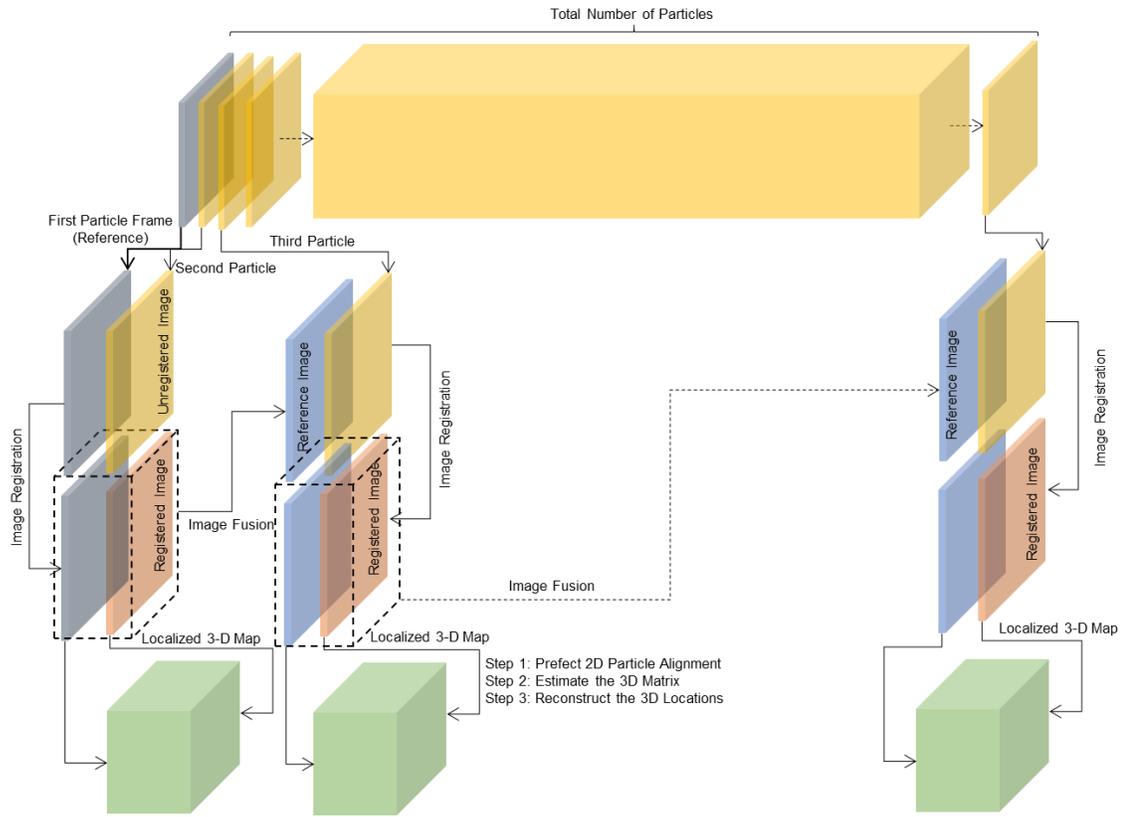


Figure 6.17: Localized 3D density map reconstruction framework using structural based motion information.

Step 1: Perfect 2D Particle Alignment

In terms of match a sparse set of points between the two 2D particle images, there are multiple ways of finding point correspondences between two 2D particle images by detecting corners in the first image and tracks them into the second image. In some cases (side-view) protein particles, we discover that our final localized 2D particle images are not aligned perfectly which causes the mis track the detected points (see Figure 6.18 (a) and (b)).

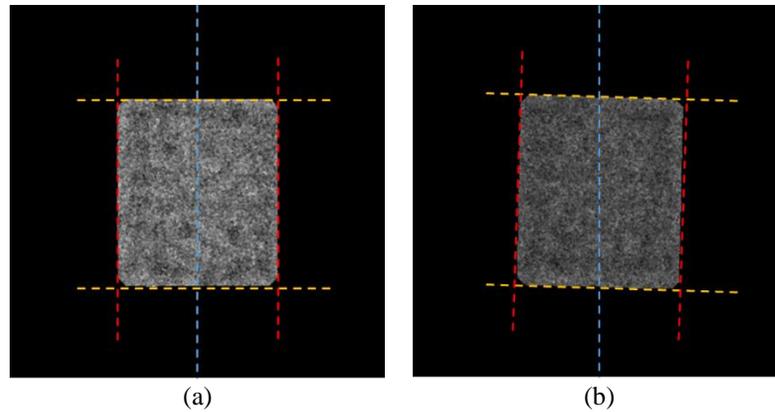


Figure 6.18: Some example from the localized 2D side-view particle image showing (a) and (b) two localized particle images are not perfectly aligned.

To solve this issue, we use the same fully automated side-view particle alignment algorithm directly on the aligned particle images. Different transformation functions are used to perfectly align the two particles images such as default alignment (initial registration) using affine transformation-based images scaling, rotation, and (possibly) shear (see Figure 6.19 (d)). We can notice that the default image alignment (initial registration) is very good. Thus, there are still some poor regions are not perfectly aligned. To improve the image alignment, we use the optimizer adjustment and metric configuration properties which is basically controls the initial step length (size) that is used to adjust the parameter space to refine the geometrical transformation (see Figure 6.19 (e)). By increasing the maximum iteration number during the image registration process that allows the image registration (alignment) to run longer and potentially to find significant registration results (see Figure 6.19 (f)). Image registration-based optimization works better than the initial registration. For this reason, we can improve the image alignment (registration) by starting with more complicated transformation such as ‘rigid’ [197] then the transformation result uses as an initial registration model by using the affine transform (see Figure 6.19 (g)). Another option that the initial geometrical transformation is used to

refine the image registration by using the affine transform with the similarity model. In this case, the refine model estimates the image registration result by including the shear transformation (see Figure 6.19 (h)).

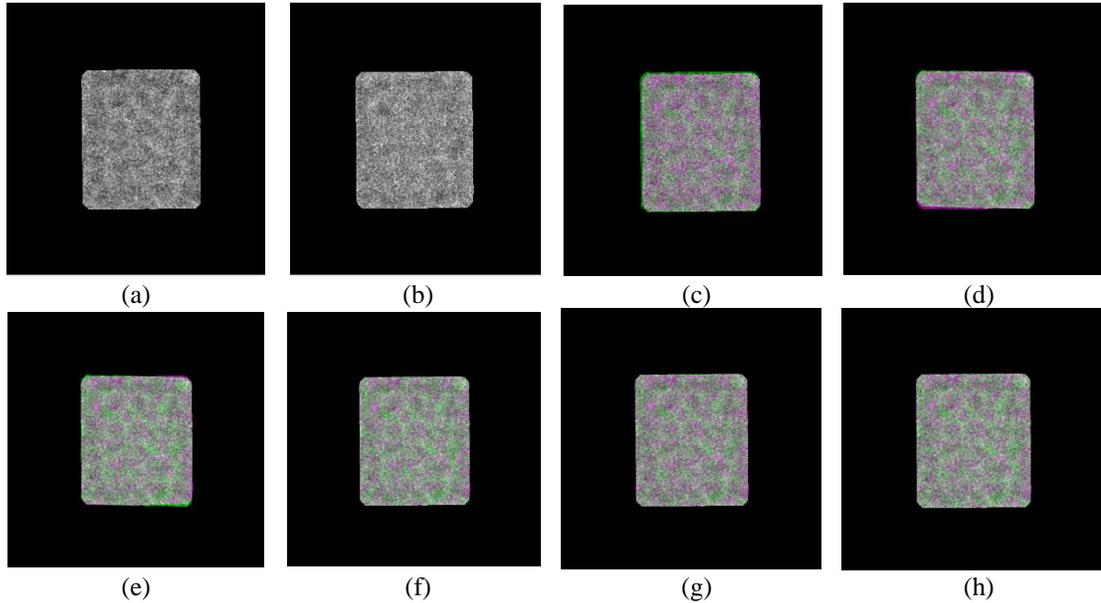


Figure 6.19: Fully automated perfect side-view particle alignment using KLH dataset [184], (a) and (b) two localized aligned particle images that are not perfectly aligned, (c) two particle image projection (a) and (b), (d) default image alignment (initial registration), (e) optimizer adjustment and metric configuration-based image registration, (f) image registration based on increasing the maximum iteration number, (g) image registration-based optimization and rigid [197] transformation. (h) image remigration using affine transform.

Step 2: Extract and Match Set of Sparse Points

After perfectly aligned all the particle images, the correlation points that will be extracted in this step will be accurately tracked because the interested points are on the space. There are many ways to find the correlation (corresponding) points between two particle images [201][202]. To extract the corresponding points, the first particle image is used as a reference image and detect the corner points (features) using the minimum eigenvalue algorithm developed by Shi and Tomasi [203] and MATLAB function ‘detectMinEigenFeatures’ [204] (see Figure 6.20 (b)). Then, the same extracted features

(detected points) are tracked on the second image using Kanade-Lucas-Tomasi (KLT), feature-tracking algorithm [205] [206] [207] [208] and MATLAB function “PointTracker” [209] (see Figure 6.20 (d)).

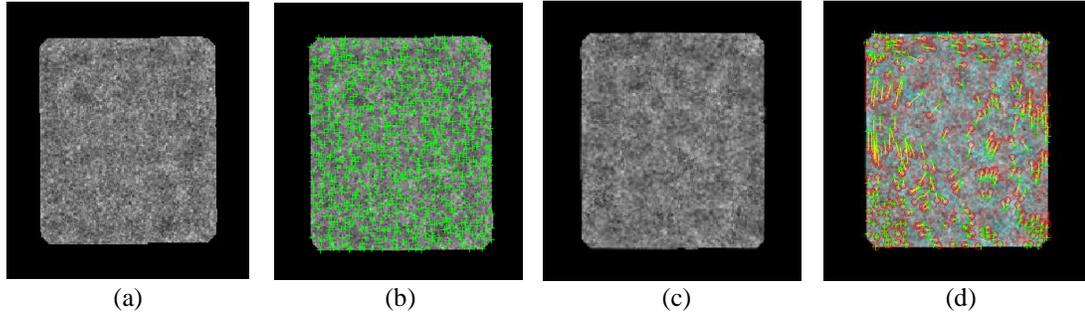


Figure 6.20: Sparse points matching and extraction, (a) first tested particle image, (b) features (corners) extraction using minimum eigenvalue algorithm [204], (c) second tested particle image, (d) correlation points detection and tracking using Kanade-Lucas-Tomasi (KLT), feature-tracking algorithm [205] [206] [207] [208].

Step 3: 3D Fundamental Matrix Estimation

The fundamental matrix is the estimated 3D matrix that relates to the corresponding points in two images [210] [211] [212] [213] [214]. The normalized eight-point algorithm [215] is used to estimate the 3D matrix based on using list of corresponding points in every two-particle image. The fundamental matrix is specified based on the following Equation (6.22) [215]:

$$[P_2] \times 3D_{Fundematal\ Matrix} \times [P_1] = 0 \quad (6.22)$$

Where P_1 is the point in the points list of the first image (list1) that is corresponding to P_2 which is the point in the point list of the second image (list2). The $3D_{Fundematal\ Matrix}$ estimates the outliers points based on using random-sample consensus algorithm (RANSAC) [212].

To compute and estimate the $3D_{Fundematal Matrix}$ different steps are implemented. First, the $3D_{Fundematal Matrix}$ is initialized by producing a 3×3 matrix of zeros $F_{initail}$. Second, a loop counter which iterated the whole process based on the specified number of trails N is initialized. Each trail represents the estimated outlier points in the 3D matrix. For each iteration, 8 Paris points are randomly selected from each point list in the two images (corresponding points) in list1 and list2. Then, use the selected 8 points to compute the fitness function f of the 3D fundamental matrix F by using the normalized 8-point algorithm [55] based on the following Equation (6.23) [215]:

$$(y')^T F y = 0 \quad (6.23)$$

Where y' and y are the corresponding selected points from list1 and list2 and F is the estimated 3D matrix, which can be similarly written as Equation (6.24) [215]:

$$f^T Y = 0 \quad (6.24)$$

Where f is denoted as the reshape version of $3D_{Fundematal Matrix} F$. After fitness function f is computed based on the corresponding points in the two images, if the fitness function f is better than the 3D fundamental matrix F , the 3D fundamental matrix F replace with the fitness function f . Then, the random number of trails N for every iteration is updated based on the RANSAC algorithm using Equation (6.25) [215]:

$$N = \min \left(N, \frac{\log(1 - p)}{\log(1 - r^8)} \right) \quad (6.25)$$

Where p is donated as the selected confidence parameters, and r is the calculated based on the Equation (6.26) [215]:

$$\sum_i^N \frac{\text{sgn}(du_i v_i, t)}{N} \quad (6.26)$$

Where $\text{sgn}(du_i v_i, t)$ is the distance function that follow the following Equation

(6.26) [215]:

$$\text{sgn}(a, b) \begin{cases} 1 & \text{if } a \leq b \\ 0 & \text{otherwise} \end{cases} \quad (6.27)$$

Two different types of distance (algebraic and Sampson) are used to measure the distance of pair points as Equations (6.28) and (6.29) show respectively [55]:

$$d(u_i, v_i) = (v_i F u_i^T)^2 \quad (6.28)$$

$$d(u_i, v_i) = (v_i F u_i^T)^2 = \left[\frac{1}{(u_i F u_i^T)_1^2 + (v_i F u_i^T)_2^2} + \frac{1}{(v_i F u_i^T)_1^2 + (v_i F u_i^T)_2^2} \right] \quad (6.29)$$

Where i is denoted as the index of the corresponding point and $(F u_i^T)_j^2$ is the square root of the j -th entity in the $F u_i^T$ vector. The 3D fundamental matrix estimation algorithm is shown below.

Algorithm 6.6 3D Fundamental Matrix Estimation

- 1: Initialize the 3D fundamental matrix F , 3-by-3 matrix of zeros.
 - 2: Import List1 and List /*corresponding points list in the first and second particle image*/
 - 3: Set the loop counter n , to zero, and the number of loops N , to the number of random trials specified
 - 4: **while** $i \leq N$ **do** /* Loop through the following steps */
 - 5: **for** each edge point **do**
 - 6: Randomly select 8 pairs of points from List1 and List2.
 - 7: Use the selected 8 points to compute a fundamental matrix, f , by using the normalized 8-point algorithm.
 - 8: Compute the fitness of f for all points in List1 and List2 using Equation (22) and (23).
 - 9: **If** fitness of f is better than F
 - 10: $F \leftarrow f$.
 - 11: **end if**
 - 12: update N using Equation (24)
 - 13: **end while**
 - 14: $n = n + 1$
-

Step 4: Reconstruct the 3D Matched Points Locations

In terms of build the localized 3D matrix (density map) using two corresponding images, the 3D locations of the matched points (corresponding) are estimated and calculated. In this step, a typical computer vision algorithm triangulates [216] is used to estimate and calculate the 3D locations of the corresponding points in the 3D space using the estimated 3D matrix from the previous step.

In general, the triangulation algorithm [216] refers to the process that a corresponding point between two images is determining in a 3D space [216]. In another word, triangulation reconstructs the 3D data based on a theory that says each point in an image is corresponding to one single line in a 3D space [217]. In this case, two or set of images can be projected in a common 3D point X [217]. The set of lines that are generated by the image points must intersect at the 3D point X . The algebra formulation of computing the 3D point X using the triangulation is showing in Equation (6.30) [218]:

$$X \sim \tau(y'_1, y'_2, C_1, C_2) \quad (6.30)$$

Where $[y'_1, y'_2]$ are the coordinates of the detected corresponding points in the image, and $[C_1, C_2]$ are the 3D estimated matrix. Mid-point method [219] is one triangulation method in which each corresponding point in the image y'_1 and y'_2 has one corresponding projected line L'_1 and L'_2 which can be determined by the 3D estimated matrix $[C_1, C_2]$ and computed based on the Equation (6.31) [219]:

$$d(L, X) = \text{Elicedian Distance}(L, X) \quad (6.31)$$

Where d is a distance function between the 3D line L'_1 and the 3D point x such that the X_{est} reconstruction point that joins the two projected lines can be calculated using the mid-point method based on the Equation (6.32) [219]:

$$d(L'_1, x)^2 + d(L'_2, x)^2 \quad (6.32)$$

The 3D reconstruction of the matched point locations algorithm is shown below.

Algorithm 6.7 Reconstruct the 3D Matched Points Locations

- 1: $I_R \leftarrow I[i]$ /*Import the first particle image from the whole dataset and Let assume the first particle image is the reference image I_R */
 - 2: $I_M \leftarrow I[i]$ /*Import the second particle image from the whole dataset and assume the next particle image is the moving image */
 - 3: $I_M \leftarrow I[i]$ /*Assume the next particle image is the moving image*/
 - 4: $List_1 \leftarrow Detect_{point}(I_R)$ /* Detect a sparse set of points (detects corners) in the reference image*/
 - 5: $List_2 \leftarrow Detect_{point}(I_M)$ /* Detect a sparse set of points (detects corners) in the moving image*/
 - 6: $Track_{points} \leftarrow KLTt_{point}(List_1, List_2)$ /* Match and track a sparse set of points between the two images using (KLT) tracking points algorithm*/
 - 7: $F[C_1, C_2] \leftarrow 3D_{Fundematal\ Matrix}(List_1, List_2)$ /*Estimate the fundamental matrix*/
 - 8: $Match_{points}[y'_1, y'_2] \leftarrow Detect_{point}(I_R, I_M)$ /*Match a dense set of points (Re-detect the corners) between the two images*/
 - 9: $X \sim \tau(y'_1, y'_2, C_1, C_2)$ /*Determine the 3D locations of the matched points using triangulate*/
 - 10: Recover the actual scale, resulting in a 3D metric reconstruction
-

Step 5: Metric Reconstruction and 3D Density Map Visualization

To visualize the localized 3D density map that is reconstructed based on the first two particle images, we use the MATLAB point cloud visualization function (pcshow) [220] to visualize and plot the point cloud of the first localized 3D density map of the single side-view protein model as is shown in Figure 6.24. For instance, Figure 6.21 (a) show the

density depth of the first localized 3D density map, while Figure 6.21 (b) shows the view of the same 3D density map.

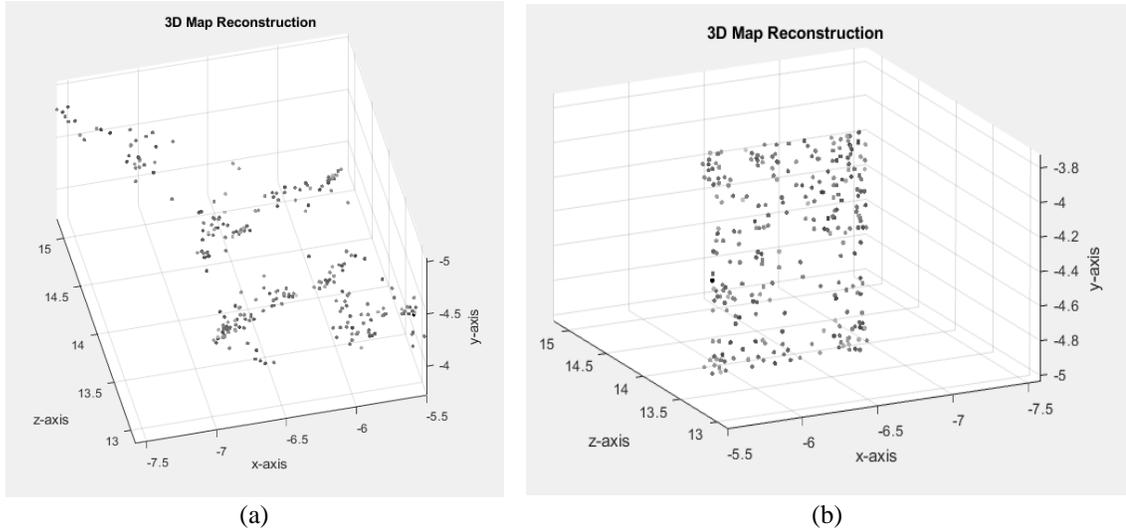


Figure 6.21: Localized 3D density map reconstruction and visualization, (a) localized 3D density map in a side view, (b) same localized 3D density map in a frontal view.

In terms of computing the second localized 3D density map that is reconstructed between the second and the third particle images, we must reconstruct a new reference particle image as is shown in main framework of the 3D density map reconstruction (see Figure 21). To reconstruct a new reference image that has the important information (corresponding points) between the two images, we use the image fusion [221] to gather the important information between the first two particle image. Image fusion is the process to combine two images and inclusion into new one image [221]. The new image is more accurate and informative that the individual two images since it gathers the corresponding important points (necessary information) between them [221].

The main purpose of doing the image fusion is the linear blend [221]. The traditional way (approach) is the linear blend [222]. It combines the two images after converting them to grayscale images and normalized the pixels values in way that the

darknets pixel value is represented by 0 and the lightest one (brightness) is represented by 1 using image Z-score normalization as shown in Equation (6.33) [223]:

$$x' = \frac{x - \bar{x}}{\sigma} \quad (6.33)$$

where \bar{x} is the mean of the intensity pixel values, and σ is the standard deviation.

Then, the image gradient is computed to detect the directional changing in the intensity value of an image as is shown in Equation (6.34) [222].

$$\nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial f} \\ \frac{\partial y}{\partial f} \end{bmatrix} \quad (6.34)$$

Where $\frac{\partial x}{\partial f}$ and $\frac{\partial y}{\partial f}$ are the gradient in the x and y direction respectively. Then, the

regions of the high special variance are combining across one image based on Equation (6.35) [222]:

$$G = \sqrt{g_y^2 + g_x^2} \quad (6.35)$$

An importance image information based weighted matrix W is calculated. The weighted matrix combines the input gradients $|G$ and gives an indication about the desired image output. The basic steps of the image fusion are described in the following algorithm.

Algorithm 6.8 Particle Image Reference Generation Based Image Fusion

- 1: $I_1 \leftarrow I_i$ /*Import the first particle image from the whole dataset*/
 - 2: $I_2 \leftarrow I_i$ /*Import the second particle image from the whole dataset*/
 - 3: $G_i \leftarrow \nabla f(I_i)$ /*Find gradient field of the two images*/
 - 4: $W_i \leftarrow |G_i|$ /*Compute importance image W_i from $|G_i|$ */
 - 5: **for** each pixel (x, y) **do**
 - 6: $G(x, y) \leftarrow \sum_i \frac{W_i(x,y)G_i(x,y)}{\sum_i W_i(x,y)}$ /* Compute mixed gradient field*/
 - 7: **end for**
 - 8: $I' \leftarrow |G_i|$ /*Reconstruct image I from gradient field G */
 - 9: $I' \leftarrow \sum_i W_i I_i$ /*Normalize pixel intensities in I to closely matched*/
-

Figure 6.22 shows some examples of particle image references generation-based image fusion. Figure 6.22 (a) and (b) show the first image (reference) and second aligned particle images (moving). Figure 6.22 (c) shows the blended overlay fused particle image, by scaling the intensities of the reference image (a) and aligned moving image (b) jointly as a single data set. Figure 6.22 (d) visualized the fused blended (overlay) image using red channel for the reference particle image, green channel for the aligned moving image, and yellow channel for the areas of similar intensity between the two images.

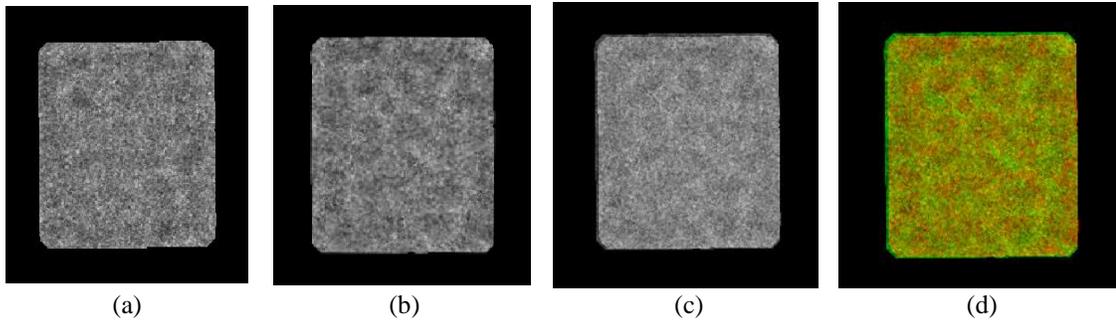


Figure 6.22: Reference image generation-based image fusion, (a) first original localized aligned particle image, (b) second localized perfect aligned particle image, (c) blended overlay fused particle image, by scaling the intensities of the reference image, (d) visualized the fused blended (overlay) image using red channel for the reference particle image, green channel for the aligned moving image, and yellow channel for the areas of similar intensity between the two images.

After the new particle reference image generation, the whole process for the second localized 3D density map reconstruction is repeated until the last particle image in the whole dataset is processed. Finally, we average the whole localized 3D density maps to produce the final 3D density map. The 3D reconstruction of the matched point locations algorithm is shown below.

Algorithm 6.9 3D Density Map Reconstruct

- 1: $I_R \leftarrow I[i]$ /*Import the first particle image from the whole dataset*/
 - 2: Let assume the first particle image is the reference image I_R
 - 3: **for** $i=2$ to the total number of the particle images N **do**
 - 4: $I_M \leftarrow I[i]$ /*Assume the next particle image is the moving image*/
 - 5: $List_1 \leftarrow Detect_{point}(I_R)$ /* Detect a sparse set of points (detects corners) in
-

```

the reference image*/
6:   $List_2 \leftarrow Detect_{point}(I_M)$  /* Detect a sparse set of points (detects corners) in
the moving image*/
7:   $Track_{points} \leftarrow KLTt_{point}(List_1, List_2)$  /* Match and track a sparse set of
points between the two images using (KLT) tracking points algorithm*/
8:   $F[C_1, C_2] \leftarrow 3D_{Fundematal\ Matrix}(List_1, List_2)$  /*Estimate the fundamental
matrix*/
9:   $Match_{points}[y'_1, y'_2] \leftarrow Detect_{point}(I_R, I_M)$  /*Match a dense set of points (Re-
detect the corners) between the two images*/
10:  $X_i \sim \tau(y'_1, y'_2, C_1, C_2)$  /*Determine the 3D locations of the matched points
using triangulate*/
11: Recover the actual scale, resulting in a 3D metric reconstruction
12:  $I_{new\_reference} \leftarrow Fusion(I_R, I_M)$  /*produce new reference image by fused the
reference and moving image in one image*/
13:  $I_R \leftarrow I_{new\_reference}$  /*Assign the new fused reference image by the old refence
image*/
14: end for
15:  $X_{final} = \sum_{i=1}^N \frac{X_i}{N}$  /*Reconstruct the final 3D density map by average the whole
localized 3D density maps*/

```

6.3 Results and Discussion

6.3.1 Datasets

Images from two datasets (Apoferitin dataset and Keyhole Limpet Hemocyanin (KLH) dataset) are used to evaluate AutoCryoPicker. The particles in the two datasets are regular shapes, which are ideal for testing AutoCryoPicker because it is designed to detect and pick regular (e.g. circular) particle shapes. Two common shapes of protein particles in cryo-EM images are circles and rectangles. Apoferitin dataset [183] uses a multi-frame MRC image format (32 Bit Float). The size of each micrograph is 1240 by 1200 pixels. It consists of 20 micrographs each having 50 frames at 2 electrons/ \AA^2 /frame, where the beam energy is 300 kV. The particle shape in this dataset is circular. The Keyhole Limpet Hemocyanin (KLH) dataset from US National Resource for Automated Molecular Microscopy [224] uses a single frame image format in a JPG file format. The size of each micrograph is 2048 by 2048 pixels. It consists of 82 micrographs at 2.2 electrons/ \AA^2 /pixel, where the beam

energy is 300 kV. There are two main types of projection views in this dataset: the top view (circular particle shape) and the side view (square particle shape). The KLH dataset [184] is a standard test dataset for particle picking. The KLH dataset is a challenging dataset because of different specimens (different particles) and confounding artifact (ice contamination, degraded particles, particle aggregates, etc.).

6.3.2 Evaluation Metrics

DeepCryoMap has four main components that need quality assessments. The first one is the first component in our DeepCryoMap which is the micrographs preprocessing. The second one is the second component which is the fully automated single particle picking. The third one is third component which is the fully automated particle selection. The last one is the fourth component which is the fully automated single particle alignment. Different evaluation metrics are used in different component. For instance, the micrographs preprocessing component the same criteria of the image preprocessing measurements that have been used in our last three models [185] [186] [187] are used in this component such as peak signal-to-noise ratio (PSNR), signal-to-noise ratio (SNR), and mean squared error (MSE) to evaluate the improvement of the quality of cryo-EM images [193]. The main results and their discussions have been described in our last three models [185] [186] [187]. For the fully automated single particle picking and selection (classification), the same criteria measurements that have been proposed in our last model DeepCryoPicker [187] uses the in this component such as accuracy, precision, recall and F1-score (i.e. the geometry mean of precision and recall).

For the fully automated single particle alignment (fourth component), we use different image quality assessment to ensure that the aligned images are still similar to their

original images but in different origination. In this case, we use the Structural Similarity Index (SSIM) [225] for measuring image quality after the fully alignment step. SSIM aims to assessing the perceptual image quality traditionally by quantify the errors (differences) between the distorted image (aligned image in our case) and the reference image (original reference image) using a variety of known properties of the human visual system [65]. SSIM bases on compute the image quality metric to assesses to the perceptual image quality based on three image characteristics luminance, contrast and structure as are shown in Equations (6.36), (6.37), and (6.38) respectively [225].

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (6.36)$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (6.37)$$

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_{xy} + C_2} \quad (6.38)$$

where $\mu_x, \mu_y, \sigma_x, \sigma_y$ and σ_{xy} are the local means, standard deviations, and cross-covariance for images x, y . The Structural Similarity (SSIM) Index quality assessment index in computed based on the three terms as is shown in Equation (6.39) [225]:

$$SSIM(x, y) = [l(x, y)]^\alpha \times [c(x, y)]^\beta \times [s(x, y)]^\gamma \quad (6.39)$$

6.3.3 Experiments on Fully Automated Single Particle Picking

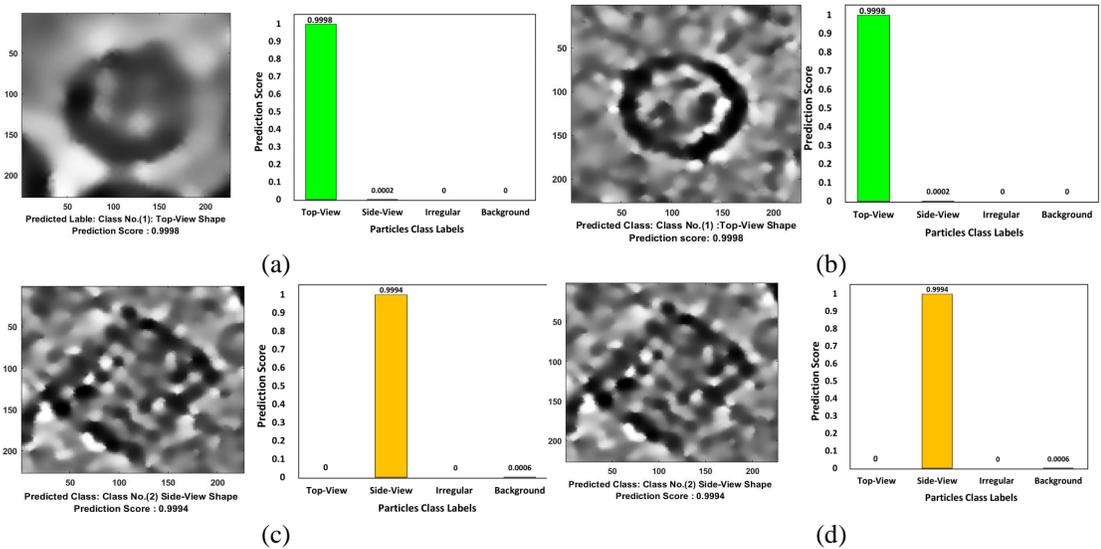
Numerous single particles are detected and picked from micrograph images using our DeepCryoPicker the fully automated deep neural network for single protein particle picking in cryo-EM [225]. Then, each single particle image is automatically isolated and evaluated as a “good” 2D particle sample. The total number of particles that are detected, picked and isolated is shown in Table 6.1.

Table 6.1: Total number of single particles picking using automated deep neural network for single protein particle and different datasets Apoferritin [183], and KLH [184] datasets.

Criteria	Apoferritin Top-View	KLH Top-View
Total Particle Image Picking	2611	2090
Number of Images	20	82
Size of Micrograph	1240x1240	2048x2048

6.3.4 Experiments on Fully Automated Perfect 2D Single Particles Selection

We split our original training dataset that has fully automated generation based on our last model DeepCryoPicker [187] into training, validation sets, while the testing dataset is every single particle that is detected and pick from the fully automated single particle picking component. Since, our framework is designed to reconstruct the 3D density map based on the structural images (top and side-view) particles, we reduce the classes number to 4 classes. Each class 1500 particle images, we split the data to 80% for training and validation (1200 particle images for training) and 20% for validation (300 particle images). Figure 6.23 shows some testing examples of the single particle classification using deep classification network.



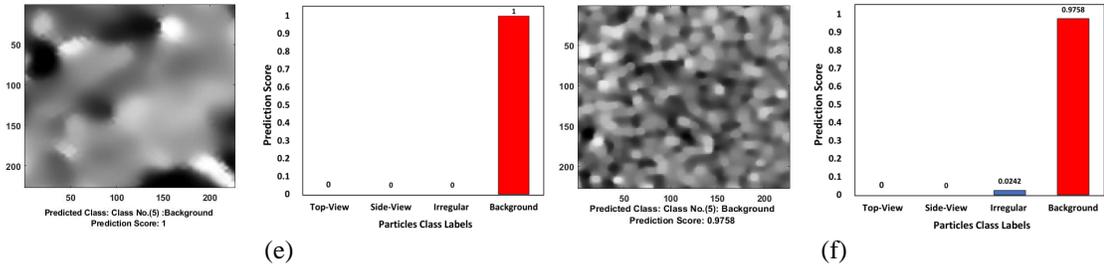


Figure 6.23: Different examples of the deep classification network results. (a) A typical testing image example showing high-density top-view particle’s predicted label and prediction score of the apoferritin micrograph dataset [183], (b) A typical testing image example showing high-density top-view particle’s predicted label and prediction score of the KLH micrograph dataset [184], (c) and (d) typical testing image examples showing high-density side-view particle’s predicted label and prediction score of the KLH micrograph dataset [184], (e) and (f) typical testing image examples showing high-density background predicted label and prediction score.

The testing accuracy of the deep classification networks using different parameters is shown in Table 6.2. It is clear that the deep classification model achieves a higher accuracy 99.89% based on using the three classes with the background cases.

Table 6.2: Performance results of fully automated perfect 2D single particles selection using the deep classification network using different parameters and datasets, learning patch size illustrates the number of subsection of an input image to the CNN which describe how many chunk of an image the been processed by the kernel at the time to estimate the good regularization property “small number of parameters” to be good across many regions of each image.

Deep Neural Classification Model	Learning patch	Epochs	Accuracy (%)
3 class “background”	16	20	99.83
	32		99.98
	64		99.72

6.3.5 Experiments of the Single 2D Particle Images Alignment

First experimental results of the perfect 2D particle image alignment is based on the perfect 2D mask (square and circular) shapes generation. Figure 6.24 shows some example results of the perfect square (side-view particles) particle shapes generation using perfect square particles shape generation of the KLH dataset [184].

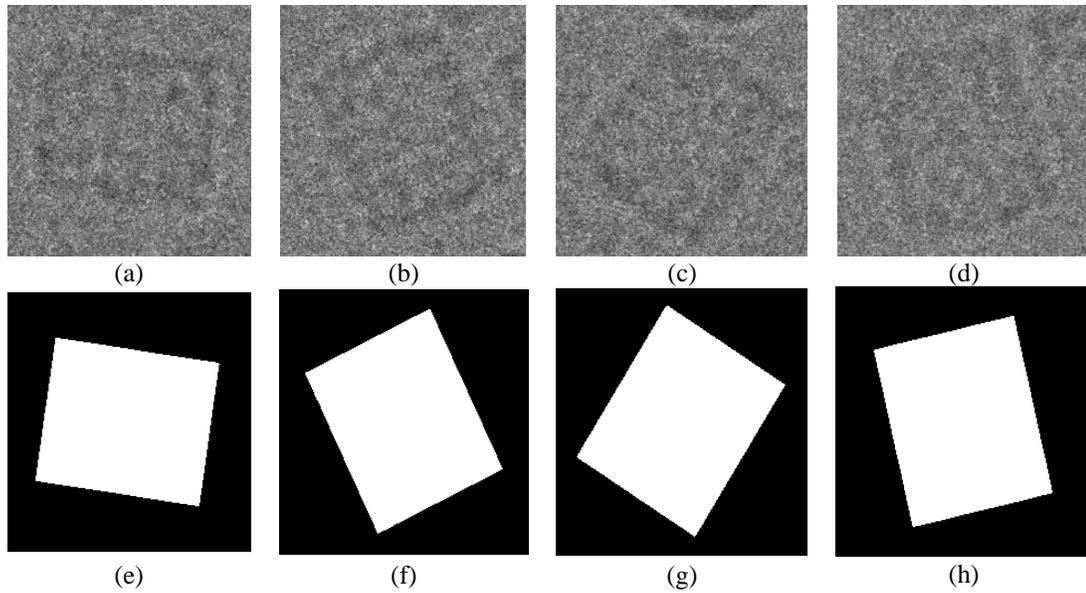
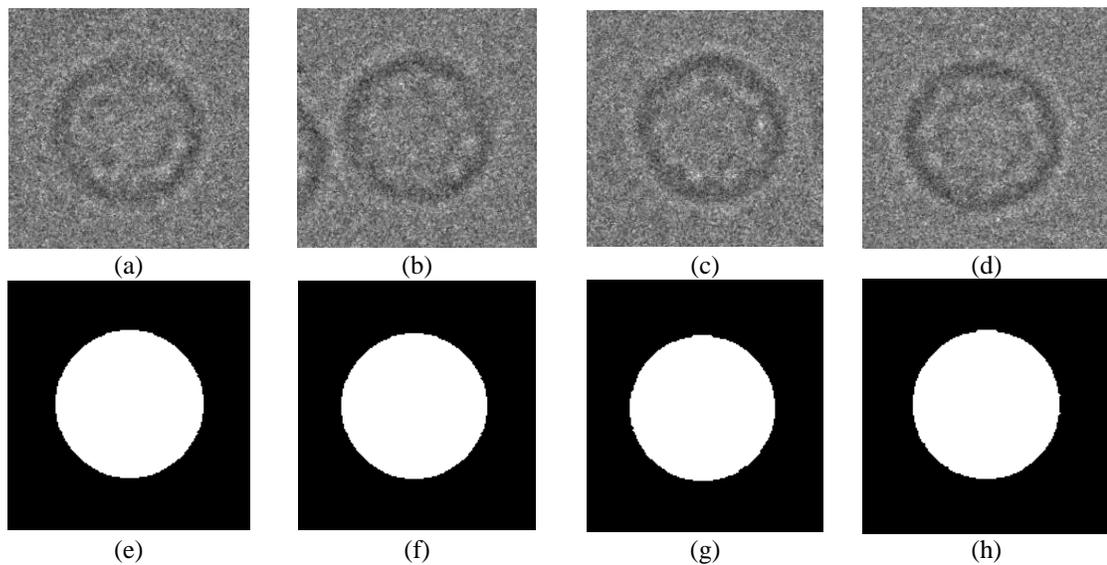


Figure 6.24: Perfect binary mask generation stage for “good” side-view 2D particles images selection. (a)-(d) are the original side-view particle image from KLH dataset [184], (e)-(h) are the perfect binary mask generation for the good 2D particle sample selection.

Figure 6.25 shows some example results of the perfect circular (top-view particles) particle shape generation and selection using perfect circular particles mask shape generation of the KLH [184] and Apoferritin [183] datasets.



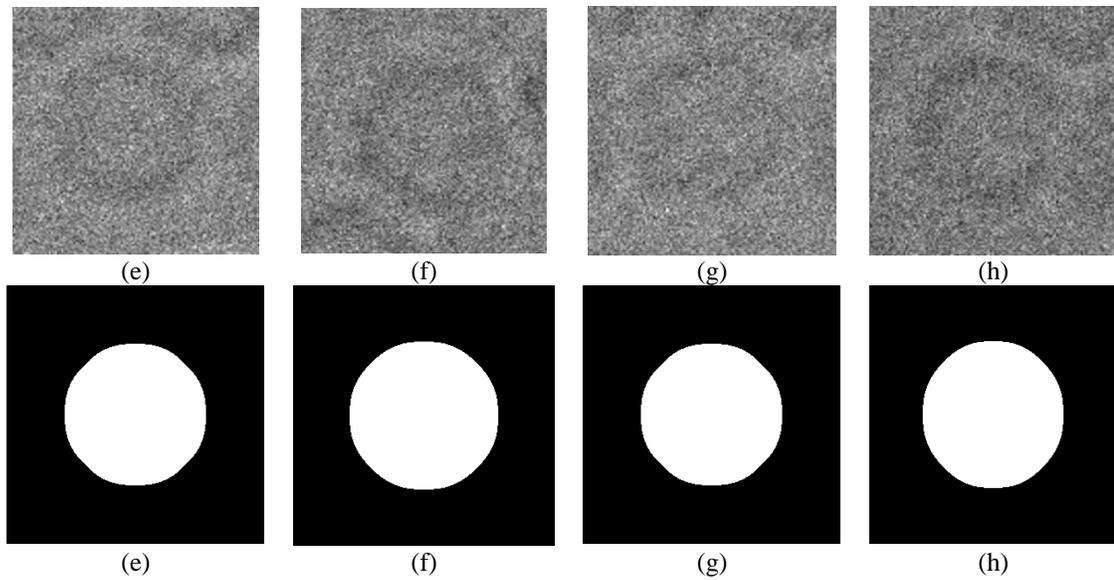
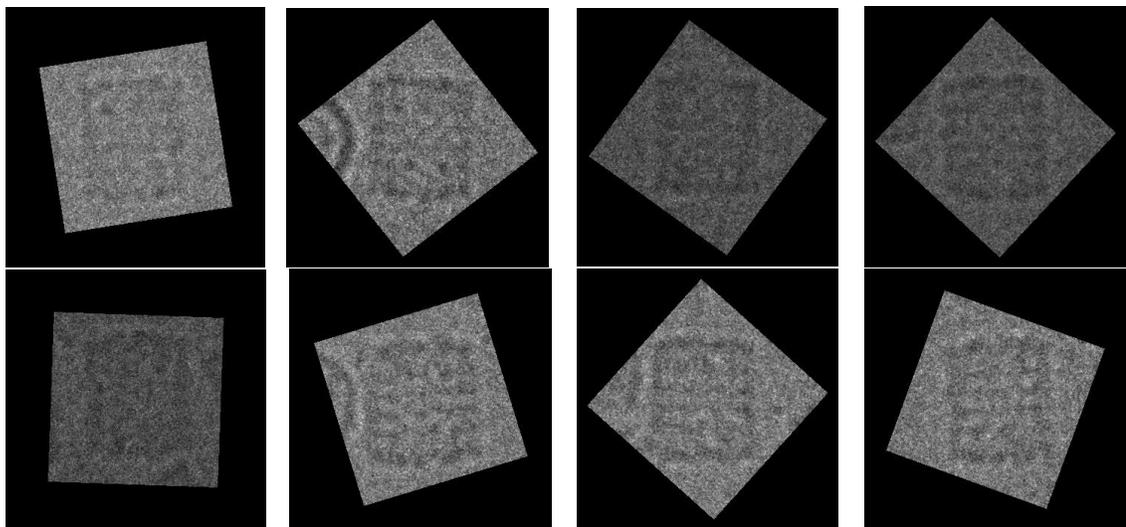


Figure 6.25: Different examples of the deep classification network results. (a) A typical testing image example showing high-density top-view particle's predicted label and prediction score of the apoferritin micrograph dataset, (b) A typical testing image example showing high-density side-view particle's predicted label and prediction score of the KLH micrograph dataset [184].

The second experiential result is a fully alignment results of the single 2D particle images. Figure 6.26 shows some example of the fully side-view particles alignment-based intensity-based registration and perfect generated particle masks.



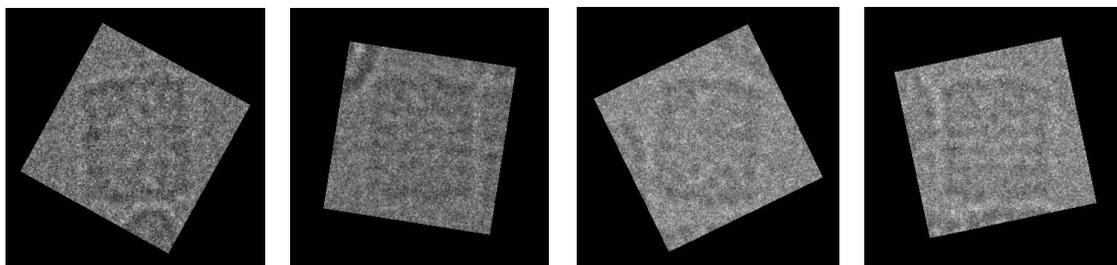


Figure 6.26: Different examples of the side-view fully alignment results-based intensity-image registration and perfect generated particle masks using KLH dataset [184].

Also, the experimental results of the localized 2D particle image alignment and generation are shown in Figure 6.27, 6.28, and 6.29. Figure 6.27 shows some examples of the localized 2D side-view particle image generation using the fully automated side-view particle alignment and perfect binary mask generation.

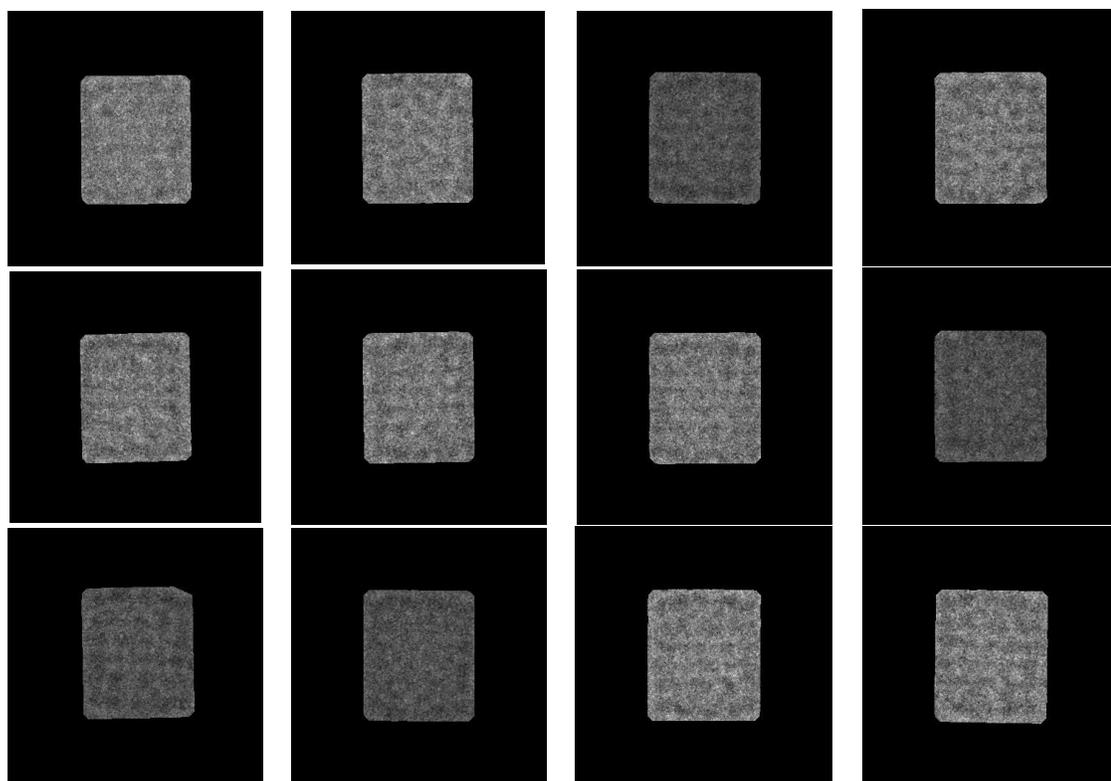


Figure 6.27: Different examples some examples of the localized particle image generation using the fully automated side-view particle alignment and perfect generated binary mask using KLH dataset [184].

Figure 6.28 shows some examples of the localized 2D KLH top-view perfect particle image using the fully automated side-view particle alignment and perfect binary mask generated.

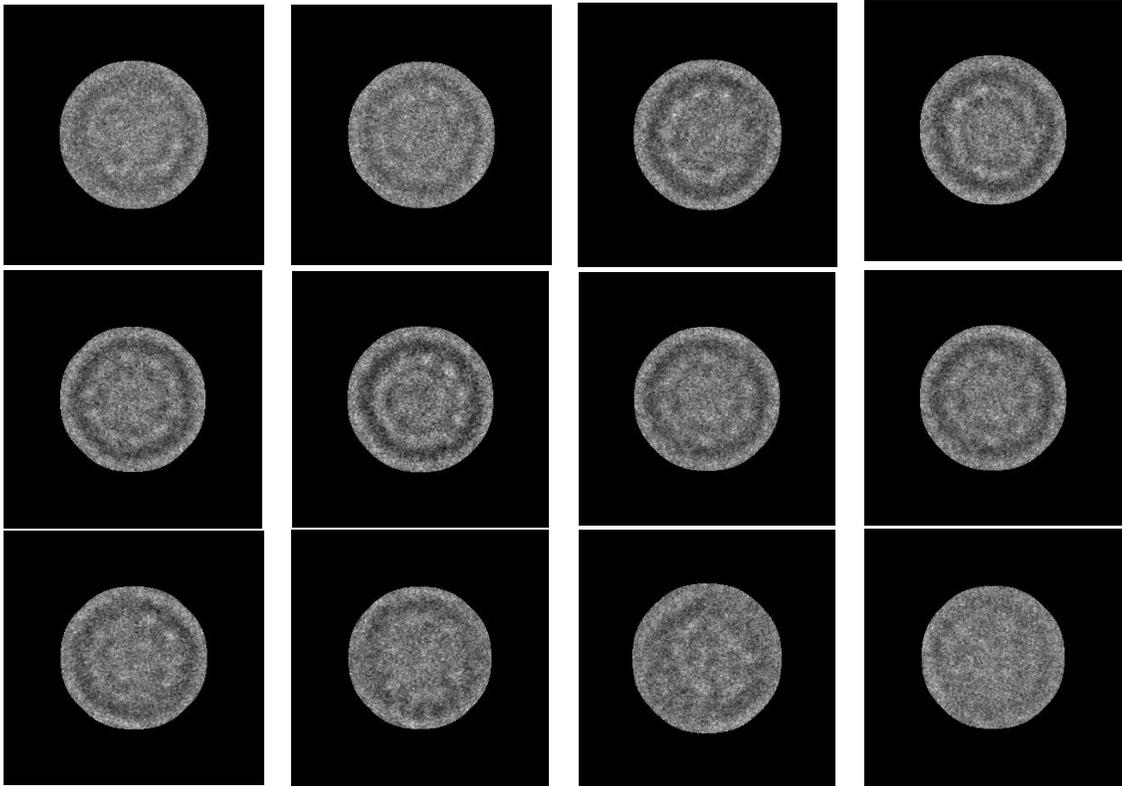


Figure 6.28: Different examples of the perfect localized 2D top-view particle alignment using KLH dataset [184].

The average similarity metric (SSIM) for the fully automated single particle alignment reaches to 99.819% using the adjusted initial radius image registration with maximum iteration number 300. But it consumed around 5.51 minutes to calculate the average similarity metric for the whole dataset. The SSIM results for the fully automated single particle alignment based different approaches are shown in Table 6.3.

Table 6.3: The average similarity metric scores (SSIM) for the fully automated single particle alignment.

SSIM Approach	Similarity	Time Consuming
Default Registration	99.648	2.19

Adjusted Initial Radius	99.754	2.06
Adjusted Initial Radius, Maximum Iterations	99.819	5.51
Similarity Transformation Model	99.561	5.16
Affine Model Based on Similarity Initial Condition	99.627	5.10

6.3.6 Experiments on Fully Automated 3D Density Map Reconstruction

Two different 3D density maps are reconstructed (top and side-view) for two single protein molecules Apoferritin [183] and KLH [184]. In this case, two different particle image versions (original and preprocessed versions) are used in the full automate 3D density map reconstruction components. The first 3D density map that is reconstructed for the side-view protein molecules using the original particles of the KLH dataset [184]. Figure 6.29 (b) shows serious of the original localized aligned particle images from the KLH dataset [184] and Figure 6.29 (a) shows the final average 3D density map reconstruction for the KLH side-view protein molecule.

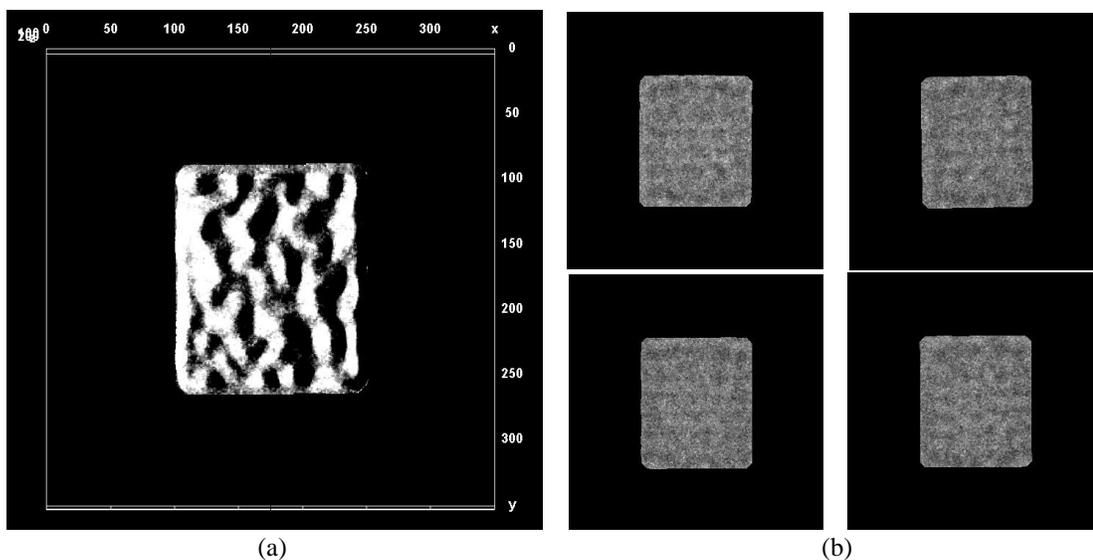


Figure 6.29: Fully automated 3D density map reconstruction for the KLH side-view protein molecule, (a) Final average 3D density map reconstruction, (b) localized alignment particle images from the KLH dataset [185].

The second 3D density map is the KLH top-view protein molecule using the preprocessed localized particle alignment images as is shown in Figure 6.30 (b). Figure 6.30 (a) shows the final average 3D density map reconstruction for the KLH side-view protein molecule.

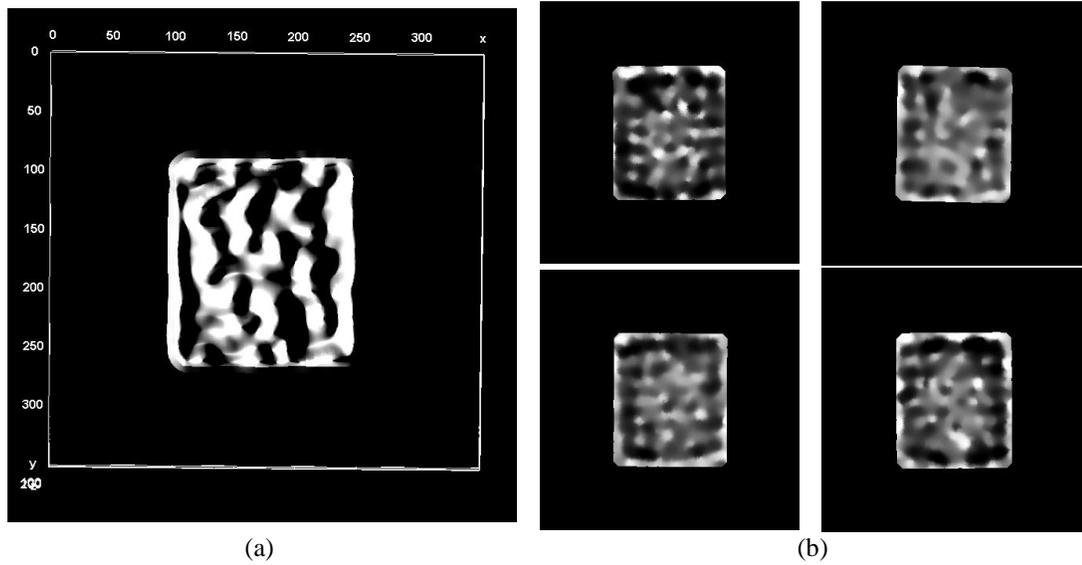


Figure 6.30: Fully automated 3D density map reconstruction for the KLH side-view protein molecule, (a) Final average 3D density map reconstruction, (b) localized aligned of the preprocessed particle images from the KLH dataset [185].

The other molecule for the KLH data [184] is the top-view protein molecule. Figure 31 (a) shows the final 3D density map of the KLH top-view protein molecule based the original particle images. Figure 31 (b) shows the preprocessed localized alignment images of the top-view particles.

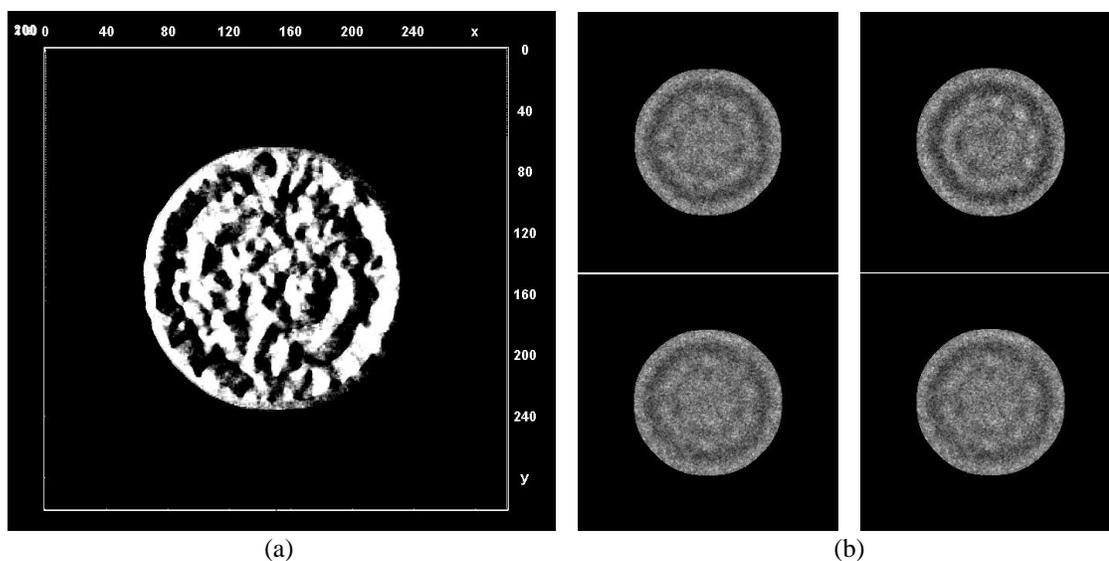


Figure 6.31: Fully automated 3D density map reconstruction for the KLH top-view protein molecule, (a) Final average 3D density map reconstruction, (b) original localized alignment particle images the KLH dataset [184].

Moreover, the preprocessed version of top-view particle images from the KLH dataset [184] (see Figure 6.32 (b)) are used to reconstruct the 3D density map as is shown in see Figure 6.32 (a).

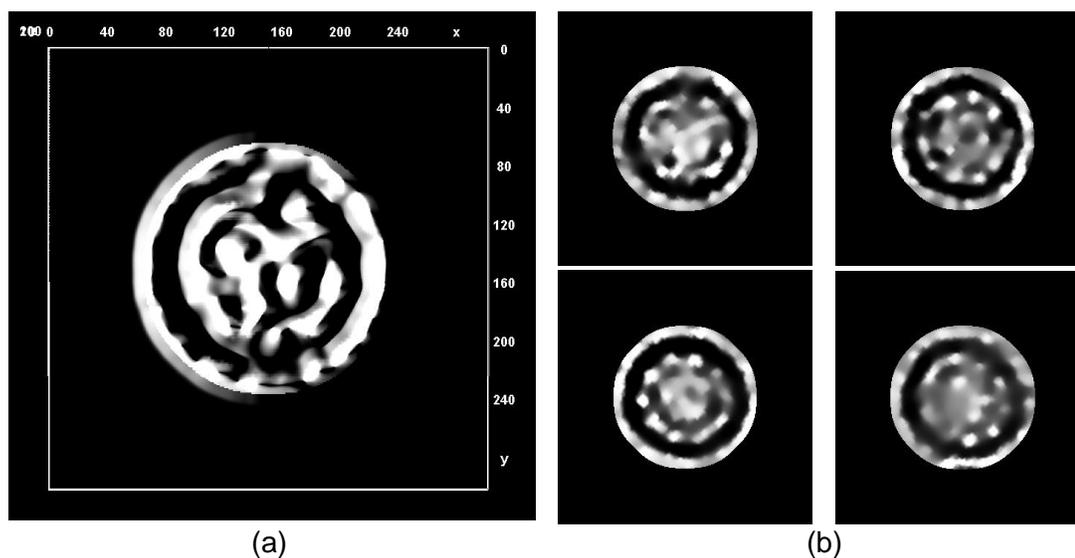


Figure 6.32: Fully automated 3D density map reconstruction for the KLH top-view protein molecule, (a) Final average 3D density map reconstruction, (b) preprocessed localized alignment particle images using KLH dataset [184].

The last 3D density maps are reconstructed for the Apoferritin top-view molecule (See Figure 6.33 and 6.33 (a)) based on using two different particle image version (original localized alignment (see Figure 6.33 (b)) and preprocessed localized alignment particle images (see Figure 6.33 (b))).

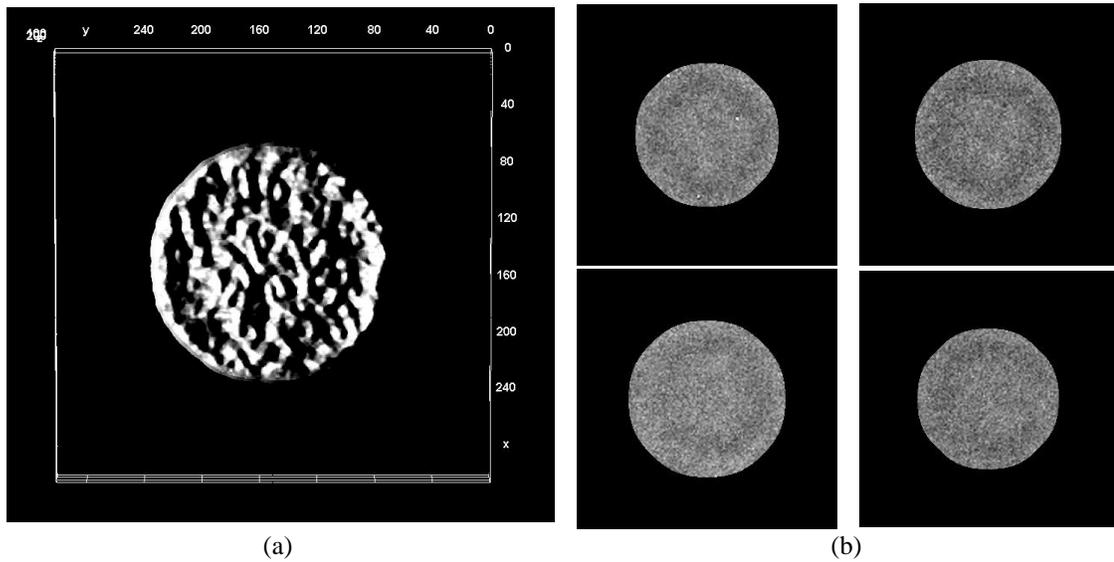


Figure 6.33: Fully automated 3D density map reconstruction for the Apoferritin top-view protein molecule, (a) Final average 3D density map reconstruction, (b) original localized alignment particle images using Apoferritin dataset [183].

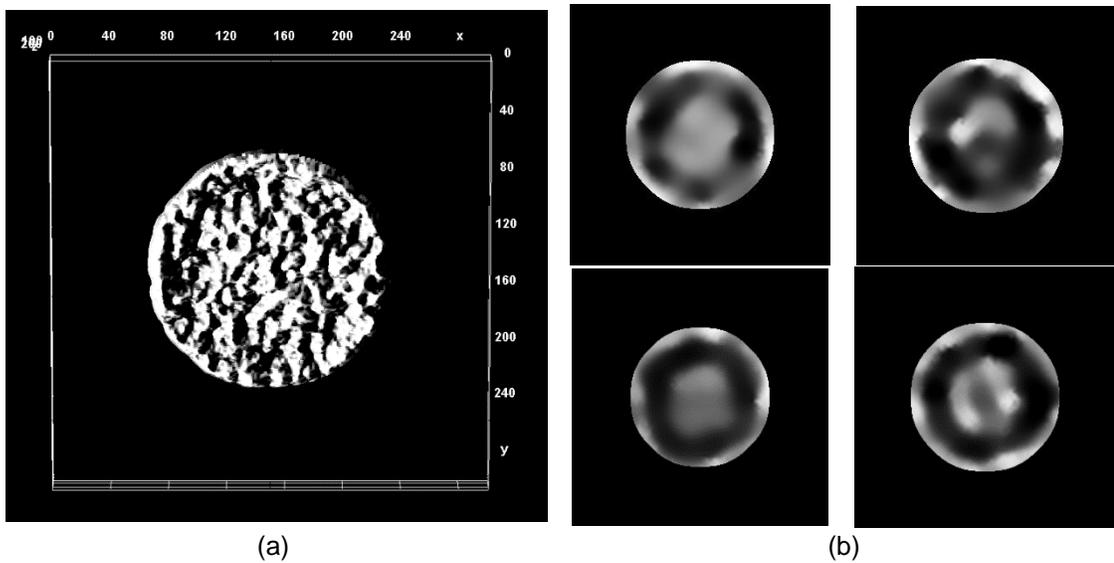


Figure 6.34: Fully automated 3D density map reconstruction for the Apoferritin top-view protein molecule, (a) Final average 3D density map reconstruction, (b) prepressed localized alignment particle images using Apoferritin dataset [183].

6.4 Conclusions

We introduce DeepCryoEM, a fully automated approach for cryo-EM 3D density maps reconstruction based deep supervised and unsupervised learning approaches. DeepCryoMap bases on different technique that the other tools used for the particle alignment. It uses the fully automated unsupervised learning algorithm ICB [185] to generate 2D particle shapes that are used mainly for the fully perfect particle alignment. Also, the perfect 2D particle image are used to produce localized aligned particle images that have only the particle and isolate the background. We show that the DeepCryoMap is able to do a fully and accurately alignment first for different particle shapes using different micrograph datasets. Second, based few thousands of particle images the DeepCryoMap is able to build a descent 3D density map comparing with the other software and tools that require a hundred of thousands of particle images. Finally, by using the preprocessed version of the particle images the DeepCryoMap reconstructs a better 3D density map than using the original particle images. We expect more sophisticated steps will be used to evaluate and improve the reconstructed 3D density map in the future.

Chapter 7

Tools for Fully Automated Single Particle Picking and 3D Density Map Reconstruction

7.1 Basic Dependencies

All the methods (fully automated single particle picking and 3D density map reconstruction) were developed in MATLAB. Users need to MATLAB 2018b/2019a installed before the executables for the program can be used. Download MATLAB from here: 2018b or 2019a. (mathworks.com/products/matlab/whatsnew.html). There are no minimum system requirements for running MATLAB can be found.

7.2 AutoCryoPicker

7.2.1 Installation

The source codes and the datasets in addition to the ground truth (hand labeling dataset) for AutoCryoPicket, an supervised learning approach for fully automated single particle

picking in cryo-EM are available at <https://github.com/jianlin-cheng/AutoCryoPicker>. The main repository of the AutoCryoPicker has different files such as:

- The first folder is the "cryo-EM Dataset" which has different sub folders such as:
 - Apoferritin_cryo-EM_Dataset_after_averaging_EMAN2: This folder the original cryo-EM dataset after converted them from the (*.MRC) files to (*.PNG) files using EMAN2 and apply the auto sane image averaging while the conversion process to increase the intensity of the cryo-EM images.
 - Apoferritin_cryo-EM_Dataset_without_averaging: This folder has the same dataset images in (*.PNG) files without using the auto sane averaging.
 - Apoferritin cryo-EM Ground Truth: This folder has the ground truth of the original dataset images in (*.PNG) files.
- The second folder is the "Main File" which has all the MATLAB code files that is required to run the system. This folder has two sub folders such as:
 - Pre-processing Stage: this folder has the MATLAB source code of the pre-processing stage implementation to preprocess the whole images dataset and plot the average results of the PSNR, SNR, and MSE, as well as to the student-t test experimental results.
 - Single Particle Detection_Demo: this folder has the main MATLAB code for the single particle picking without the GUI version.
 - Guide User Interface_GUI: this folder has the main MATLAB code for the single particle picking with the GUI version.

7.2.2 Usage

To run the tool (AutoCryoPicker), you have to download or clone the “Single Particle

Detection_Demo” or the “Guide User Interface_GUI” repository and go to the main MATLAB file "AutoPicker_Final_Demo.m" to run the comment line version, “Pre-processing Stage: to run the preprocessing stage, or the “AutoCryoPicking.m” to run the GUI version.

Pre-processing Stage

In case of running the preprocessing stage individually, you need to open the “CryoEM_Pre_processing_Step.m” and download the cryo-EM dataset “cryo-EM Dataset” folder and update the dataset folder directory in the and CLICK run in MATLAB. Everything will automatically run.

Single Particle Picking Demo without the GUI

In case of running the comment lines of the AutoCryoPicker version, you need to open the "AutoPicker_Final_Demo.m" file. In this case the program will ask you to select one single image from the downloaded cryo-EM dataset, then the program will automatically run and display the single particles detection and picking results.

Single Particle Picking Demo using the GUI

There is a GUI version called "Guide User Interface_GUI" which is all in one, you need to download the “Guide User Interface_GUI” and go directly to open the "AutoCryoPicking.m" in MATLAB and run it. As is shown in Figure 7.1, the GUI version has many commends (options) such as:



Figure 7.1: AutoCryoPicker GUI demo, showing the main GUI version that has five commands: load cryo-EM, preprocessing cryo-EM, cryo-EM clustering, particles detection and picking, and performance results.

- Load cryo-EM: for load any cryo-EM image for testing (single particle picking) as is shown in Figure 7.2.
- Pre-processing cryo-EM: for doing the preprocessing task for the tested image as is shown in Figure 7.2

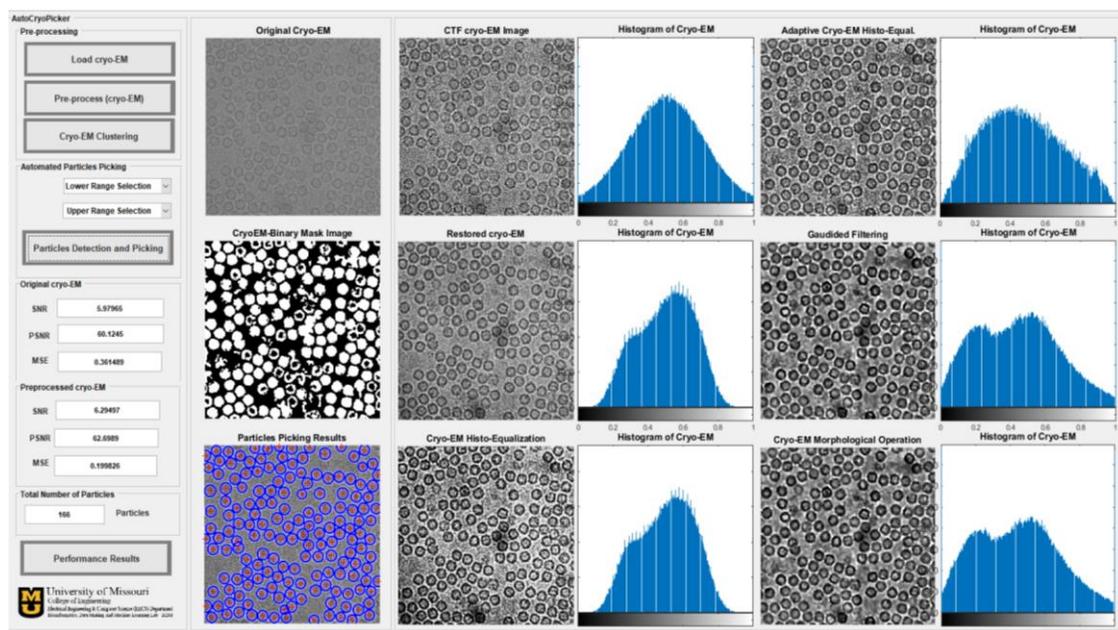


Figure 7.2: An example of preprocessed and particle detection using one cryo-EM image using Apoferritin cryo-EM dataset [21]

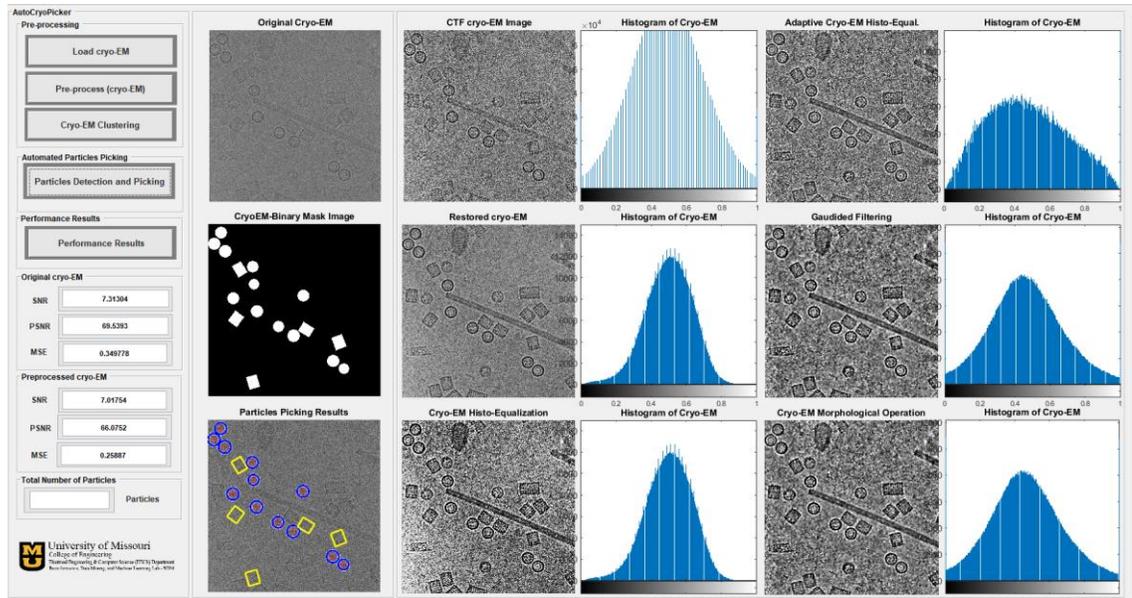


Figure 7.3: An example of preprocessed and particle detection using one cryo-EM image using KLH cryo-EM dataset [22]

- Particles Detection and Picking: for detect and picking the particles in the tested image as is shown in Figure 7.3
- Performance Results: in this case, if we want to get the accuracy results and other measurements you have to upload the GT image for each tested cryo-EM image (cryo-EM image with the yellow dots that have been uploaded to the GitHub repository),
 - Particles Picking Accuracy: in this case, after selecting the GT image the system will automatically calculate and display all the performance results once you click of the "Particles Picking Accuracy".
 - cryo-EM projection: This task is to extract the BOX for each single particle as is shown in Figures 7.4 and 7.5

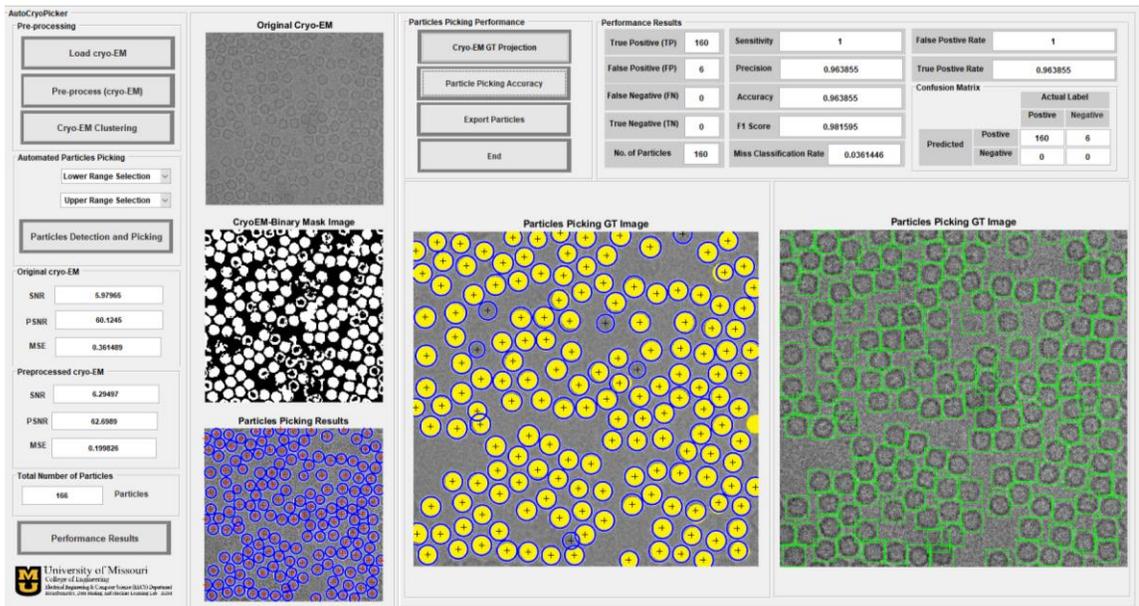


Figure 7.4: An example of particle detection and picking using one cryo-EM image using Apoferritin cryo-EM dataset [21]

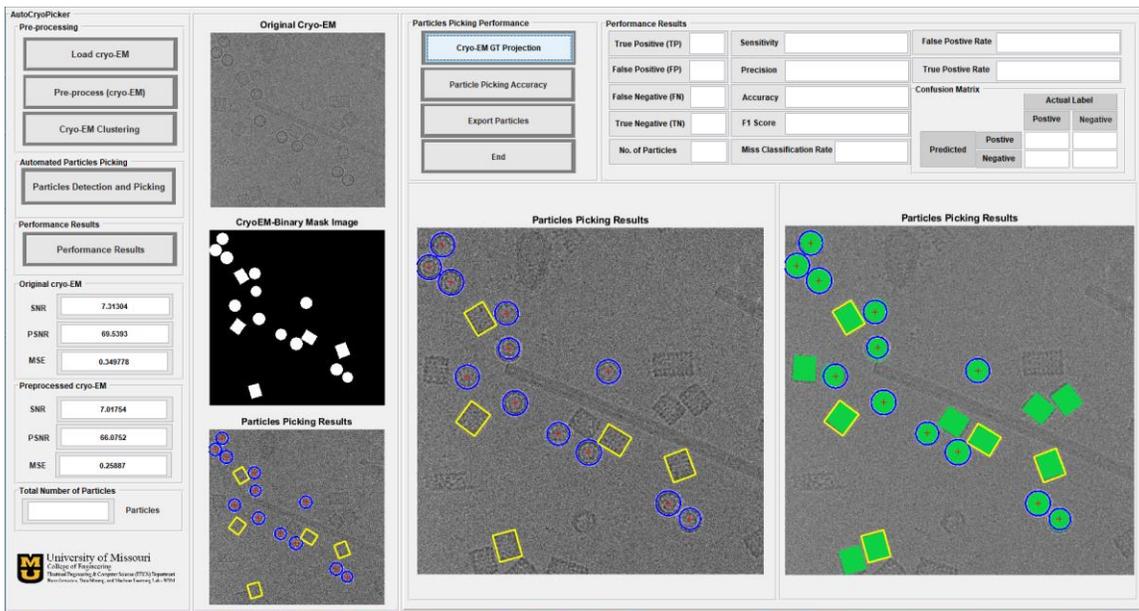


Figure 7.5: An example of particle detection and picking using one cryo-EM image using KLH cryo-EM dataset [22]

- **Export Particles:** this task is to extract the box dimension and the particle center information to *.TXT file as is shown in Figure 7.6.

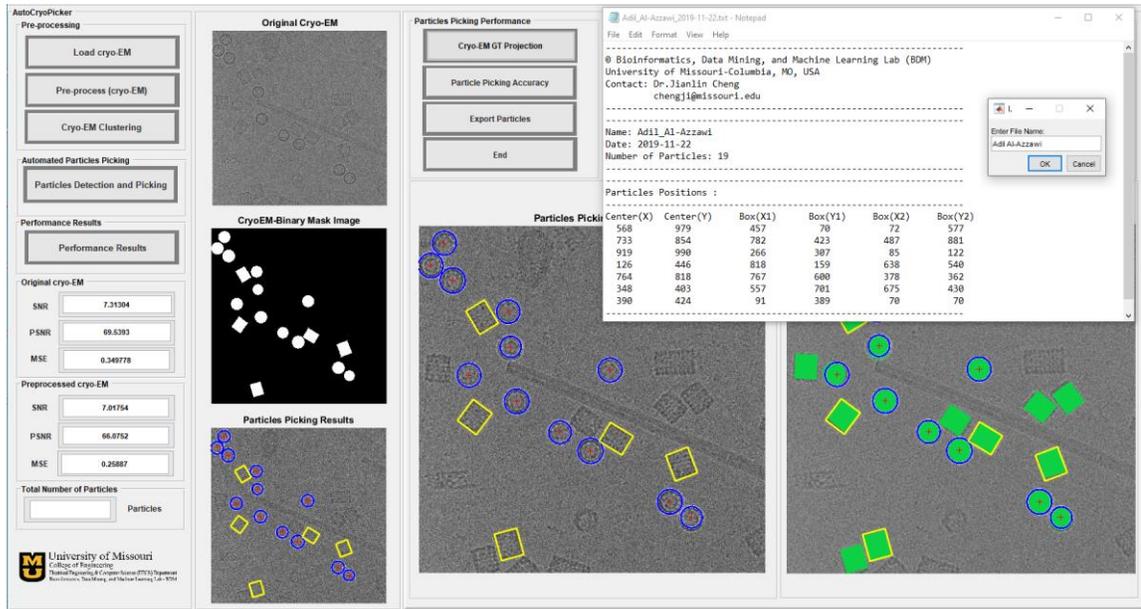


Figure 7.6: An example of exporting the particles indexing using one cryo-EM image using KLH cryo-EM dataset [22]

7.3 SuperCryoEMPicker

7.3.1 Installation

The source codes and the datasets in addition to the ground truth (hand labeling dataset) for SuperCryoEMPicker, a super clustering approach for fully automated single particle picking in cryo-EM are available at <https://github.com/jianlin-cheng/SuperCryoPicker>.

The main repository of the SuperCryoEMPicker has different files such as:

- The first folder is the "cryo-EM image Dataset" which has different cryo-EM datasets BDGAL and Ribosome DATASET in addition to the ground truth of the original dataset images in (*.PNG) files.
- The second folder is the " SuperCryoEMPicker" which has all the MATLAB code files that is required to run the system. This folder has many functions such as:

- Pre-processing Stage: this folder has the MATLAB source code of the pre-processing stage implementation to preprocess the whole images dataset and plot the average results of the PSNR, SNR, and MSE, as well as to the student-t test experimental results.
- FullyAutoCryo_Picker_Demo_Final: this file is the main MATLAB code for the single particle picking without the GUI version.
- Guide User Interface_GUI: this folder has the main MATLAB code for the single particle picking with the GUI version.

7.3.2 Usage

To run the tool (SuperCryoEMPicker), you have to download or clone the “SuperCryoPicker” repository and go to the main MATLAB file " SuperCryoEMPicker" to run the system.

Pre-processing Stage

In case of running the preprocessing stage individually, you need to open the “CryoEM_Pre_processing_Step.m” and download the cryo-EM dataset “cryo-EM Dataset” folder and update the dataset folder directory in the and CLICK run in MATLAB. Everything will automatically run.

Single Particle Picking Demo without the GUI

In case of running the commend lines of the “SuperCryoPicker” version, you need to open the "FullyAutoCryo_Picker_Demo_Final.m" file. In this case the program will ask you to select one single image from the downloaded cryo-EM dataset, then the program will automatically run and display the single particles detection and picking results.

Single Particle Picking Demo using the GUI

There is a GUI version called "SuperCryoEMPicker_Guide User Interface_GUI" which is all in one, you need to download the "SuperCryoEMPicker_Guide User Interface_GUI" and go directly to open the "SuperCryoPicker.m" in MATLAB and run it. As is shown in Figure 7.7, the GUI version has many commands (options) such as:



Figure 7.7: SuperCryoEMPicker GUI demo, showing the main GUI version that has five commands: load cryo-EM, preprocessing cryo-EM, cryo-EM clustering, particles detection and picking, and performance results.

- Load cryo-EM: for load any cryo-EM image for testing (single particle picking) as is shown in Figure 7.8 and 7.9.
- Pre-processing cryo-EM: for doing the preprocessing task for the tested image as is shown in Figure 7.8 and 7.9.
- Particles Detection and Picking: for detect and picking the particles in the tested image as is shown in Figure 7.10 and 7.11.
- Performance Results: in this case, if we want to get the accuracy results and other measurements you have to upload the GT image for each tested cryo-EM image

(cryo-EM image with the yellow dots that have been uploaded to the GitHub repository).

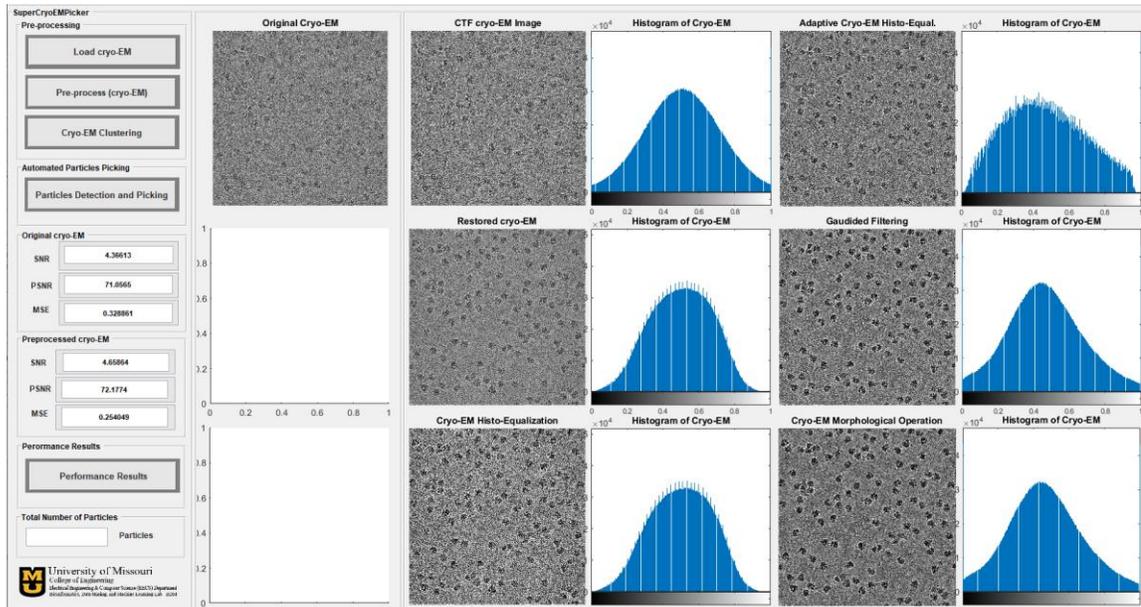


Figure 7.8: An example of preprocessed and particle detection using one cryo-EM image using Ribosome cryo-EM dataset.

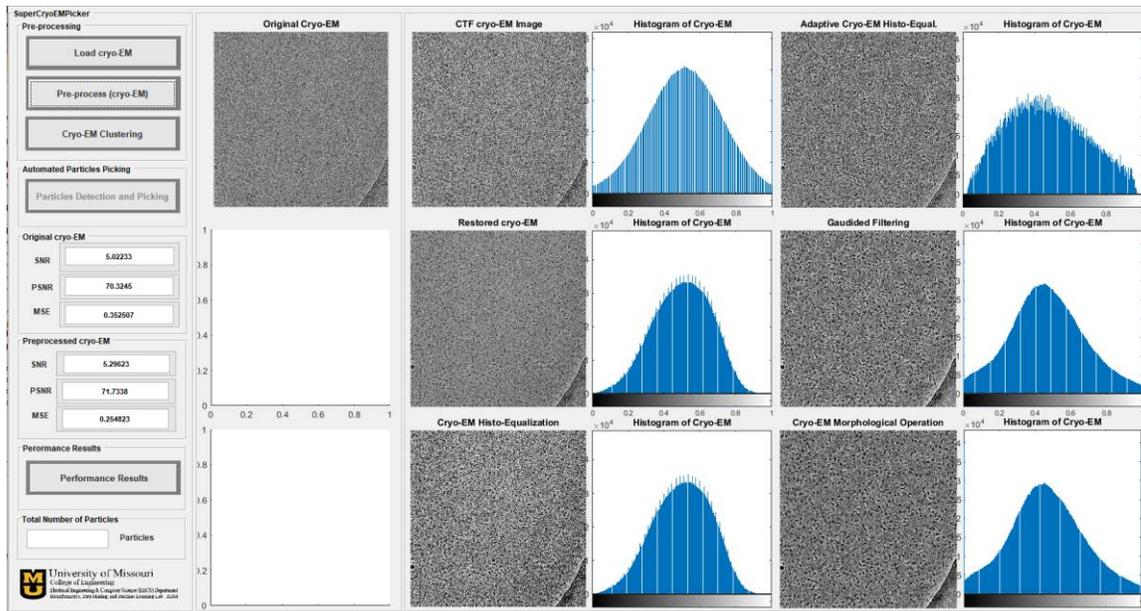


Figure 7.9: An example of preprocessed and particle detection using one cryo-EM image using Beta-galactosidase cryo-EM dataset.

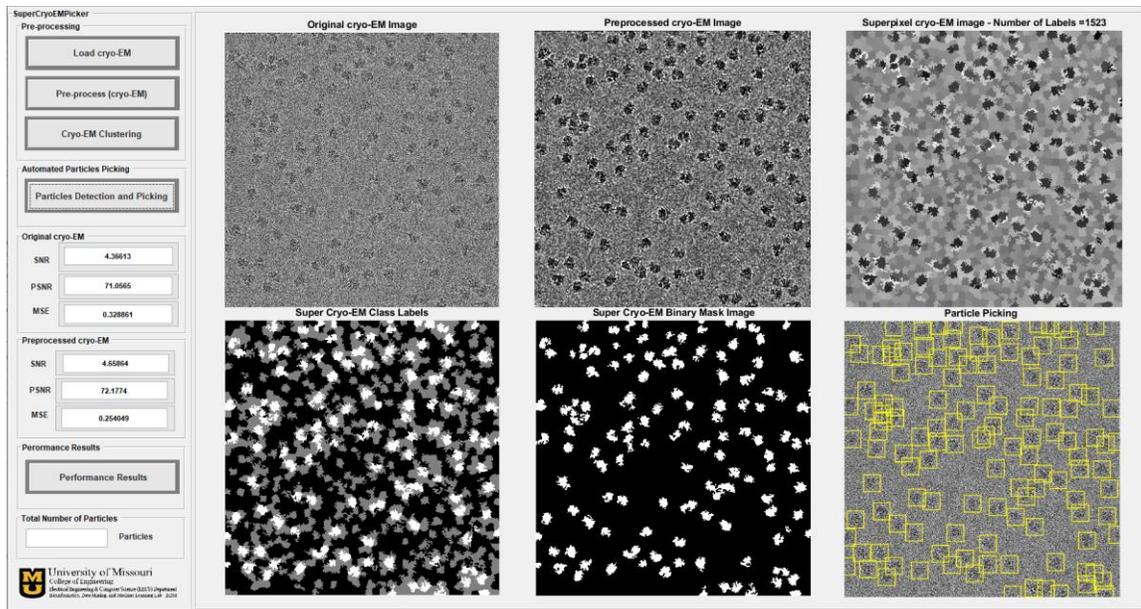


Figure 7.10: An example of particle clustering detection and using one cryo-EM image using Ribosome cryo-EM dataset.

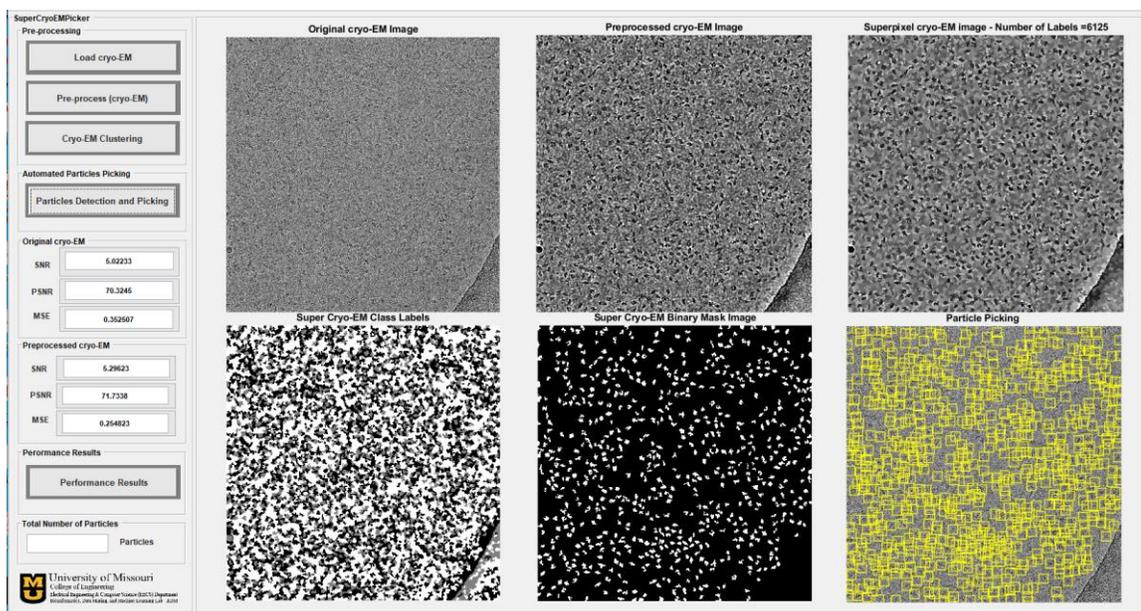


Figure 7.11: An example of particle clustering detection and using one cryo-EM image using galactosidase cryo-EM dataset.

- Performance Results: in this case, if we want to get the accuracy results and other measurements you have to upload the GT image for each tested cryo-EM image

(cryo-EM image with the yellow dots that have been uploaded to the GitHub repository).

- **Particles Picking Accuracy:** in this case, after selecting the GT image the system will automatically calculate and display all the performance results once you click of the "Particles Picking Accuracy" (see Figure 7.11).

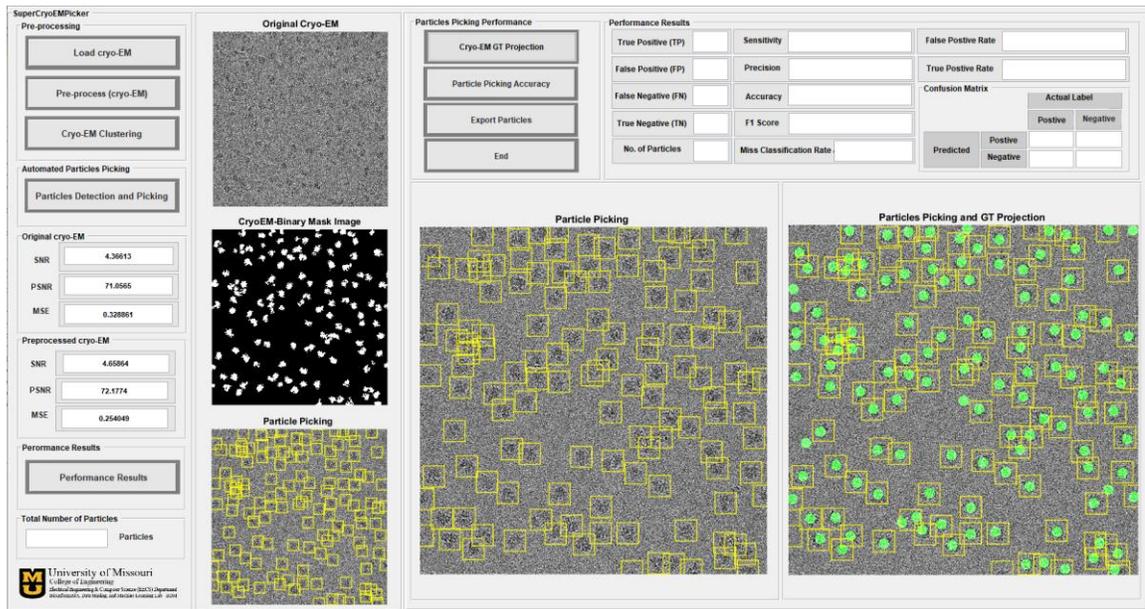


Figure 7.12: An example of particle detection and picking using one cryo-EM image using Ribosome cryo-EM dataset.

7.4 DeepCryoPicker

7.4.1 Installation

The source codes and the datasets in addition to the ground truth (hand labeling dataset) for DeepCryoEM, a fully automated deep neural network for single protein particle picking in cryo-EM are available at <https://github.com/jianlin-cheng/DeepCryoEM>. The main repository of the DeepCryoEM has different files such as:

- The first folder is the "cryo-EM Dataset" which has different cryo-EM datasets in (*.PNG) files.
- The second folder is the "Ground Truth" which has the ground truth of the original cryo-EM dataset.
- The Third folder is the "MATLAB Code" which has all the MATLAB code files that are required to run the system. This folder has sub folders such as:
 - Pre-processing Stage: this folder has the MATLAB source code of the pre-processing stage implementation to preprocess the whole images datasets in addition to evaluate the preprocessing stage and plot the average results of the PSNR, SNR, and MSE, as well as to the student-t test experimental results.
 - Single Particle PickingDetection Part: this file is the main MATLAB code for the single particle picking and for the training dataset generation, in this case, you need to
 - Single Particle Picking Evaluation Part: this folder has the implementation functions to evaluate and select the 'good and perfect training example selection'.

7.4.2 Usage

To run the DeepCryoPicker tool, you have to download or clone the "DeepCryoEM" repository and go to the main MATLAB file " Matlab Code" to run the system.

Component 1: Fully Automated Training Particle Selection

The first component of the DeepCryoPicker has three stages: (1) pre-processing cryo-EM micrographs; (2) Stage 1: fully automated training particle selection; (3) Stage 2: full

automated perfect “good” training particle selection and labelled training dataset generation.

- **Step 1: Pre-processing Stage:** The first stage in the DeepCryoPicker is running the preprocessing stage. In this case, you need to go to the “Matlab Code” and download the “Pre-processing Stage” folder, then we need to open the “CryoEM Preprocessing Step.m” and download the cryo-EM dataset “cryo-EM Dataset” folder and update the dataset folder directory in the and CLICK run in MATLAB. Everything will automatically run. The whole preprocessed cryo-EM datasets will be automatically saved in the “Pre_processed CryoEM” folder.
- **Stage 1: Fully Automated Single Particle Picking Stage:** in this stage, single particle picking using two functions from our previous models AutoCryoPicker [23] and SuperCryoEMPicker [24]. In this case, you need to go to the “Matlab Code” and download the “Component 1_fully_automated_training_particle selection” folder. You need to run the two functions “AutoPicker_Final_Demo.m” and “FullyAutoCryo Picker Demo Final”. Make sure that the directory of the preprocessed cryo-EM is changed and update and CLICK run in MATLAB. Everything will automatically run. The single particles will automatically save in different subfolders based on the name of the used cryo-EM dataset. This
- **Stage 2: Fully Automated Training Particles-Selection Stage:** the second stage is the perfect training sample selection. In this case, you need to go to the “Matlab Code” and download the “Component 1_fully_automated_training_particle selection” folder, then we need to use three different functions based on the particle shapes such as:

- Use the first function “Perfect top view training particle selection.m” to test each individual top-view particle’s mask size and verify if it is a perfect full circle and label it as either a “good example” or as a “bad example”.
- Use the first function “Perfect_side_view_training_particle_selection.m.m” to test each individual side-view particle’s mask size and verify if it is a perfect full square and label it as either a “good example” or as a “bad example”.
- Use the first function “Perfect irregular complex training particle selection.m” to test each individual irregular and complex particle’s mask size and verify if it is a perfect full particle and label it as either a “good example” or as a “bad example”.

Component 2: Fully Automated Single Particle Picking

The second component of the DeepCryoPicker is the particle picking based deep network. This component has two stages: (1) Deep neural network training; (2) Deep neural network testing for fully automated single particle picking.

- Stage 1: Deep Neural Network Training Stage: The deep classification neural network is training using the perfect “good” training samples. Go to the “Matlab Code” and download the “Component 2 fully automated single particle picking” folder, then you need to open the “Deep Classification Network Training Step.m”. Make sure that the directories of the cryo-EM particles are changed and update and CLICK run in MATLAB. Everything will automatically run.
- Stage 2: Deep Classification Particle Picking Testing Stage: The final stage of the DeepCryoPicker is doing the fully automated single particle picking using the

trained deep classification network. In this case, you need to go to the “Matlab Code” and download the “Component 2 fully automated single particle picking” folder” folder, then we need to open the “Deep Classification Network Testing Step.m. Make sure that the directories of the cryo-EM particles are changed and update and CLICK run in MATLAB. Everything will automatically run.

7.5 DeepCryoMap

7.5.1 Installation

The source codes and the datasets for the DeepCryoMap, the software package to align cryo-EM particles to create 3D density maps of proteins is available at <https://github.com/AlazzawiAdil/DeepCryoMap>. The main repository of the DeepCryoMap has different files such as:

- “Main code” folder which has all the MATLAB codes (functions) for the fully particles alignment and 3D density map reconstruction.
- “Particle Images” folder which has two main subfolders “Apoferritin” folder that has the top-view particle images, and “KLH” folder that has the top and side-view particle images. Each subfolder has another subfolder such as:
 - “Original Particle Images” subfolder which has the original particle images before the alignment results.
 - “Preprocessed Particle Images” subfolder which has the prepressed version of the original particle images before the alignment results.
 - “Binary Masks” subfolder which has the binary mask of each original particle images before the alignment results.

- “Alignment Results” folder which has two main subfolders “Apoferitin” folder that has the aligned top-view particle images, and “KLH” folder that has the aligned top and side-view particle images. Each subfolder has another subfolder such as:
 - “Original Particle Images” subfolder which has the original particle images after the alignment results.
 - “Preprocessed Particle Images” subfolder which has the preprocessed version of the original particle images after the alignment results.
 - “Binary Masks” subfolder which has the binary mask of each original particle images after the alignment results.

7.5.2 Usage

To run the DeepCryoMap tool, you have to download or clone the “DeepCryoMap” repository and go to the main MATLAB file " Matlab Code" to run the system. Make sure to change all the directories of the folder and subfolders. There is one function that does everything Alignment and 3D density map reconstruction. Go directly to open the “Density_3D_Reconstruction_New.m” function and just run it in MATLAB environment and everything will run automatically.

Chapter 8

Conclusion and Future Works

8.1 Conclusion

Micrographs (cryo-EM) have been widely used in the determination and understanding of the three-dimensional (3D) structures of macromolecules and proteins. Thousands of single particle images are extracted by researchers via two-dimensional (2D) cryo-electron microscopy and can be used to build reliable high-resolution (3D) reconstructions. This method has gained recent popularity in structural biology. However, because of the wide variety of different particle shapes found in micrographs, and the extremely high signal-to-noise ratio (SNR) of micrographs, single particle image picking still presents significant challenges and acquiring a sufficient quantity of high-quality particles requires excessive human labor. Accurate particle picking in cryo-EM images still requires substantial human intervention and, therefore, can be labor-intensive and time-consuming.

To address this challenge, we develop different models such as AutoCryoPicker –

a fully automated particle picking approach based on image preprocessing, unsupervised clustering and shape detection. SuperCryoEMPicker - a fully automated super particle clustering method for picking particles of complex and irregular shape in cryo-EM images. DeepCryoPicker – a Fully Automated Deep Neural Network for Single Particle Picking in cryo-EM. Although, we propose a fully automated approach for single particle picking based on two models. The first model is called the “fully automated training particles-selection and variety of training dataset generation based unsupervised learning approach”. The second model is the “fully automated single particle picking based on deep neural (classification) network”.

The experiments show that the approach can significantly improve signal to noise ratio in cryo-EM images and pick particles rather accurately. Therefore, the automated methods can relieve scientists from the laborious work of picking cryo-EM particles and help improve the efficiency and effectiveness of cryo-EM based protein structure determination. We conclude that AutoCryoPicker, SuperCryoEMPicker and DeepCryoPicker have the potential for being incorporated into the particle picking pipelines of other cryo-EM image processing software. Also, our moles are more accurate than two external semi-automated particle picking methods that require users to manually picking some reference particles for training. Therefore, they are useful and reliable tool for automated single particle picking in cryo-EM images. The experimental results indicate that our models compare favorably with other semi-automated methods such as DeepEM, DeepPicker and RELION.

8.2 Future Works

In the future work, we aim to reconstruct a better 3D density maps using the original

particle images, by developing and implementing more sophisticated steps. We also, aim to develop evaluation methods to improve the current reconstructed 3D density map. In order for the whole pipeline of the protein structure determination to be complete, we add the final step of predicting protein tertiary (three-dimensional) structure. Deep learning approach will be used for this step.

BIBLIOGRAPHY

- [1] Biswas, D. Ranjan, M. Zubair, and J. He. 837–843. "A dynamic programming algorithm for finding the optimal placement of a secondary structure topology in cryo-em data." *Journal of Computational Biology* 2015.
- [2] A.K. Jain, M.N. Murthy, and P.J. Flynn. 1999. "Data clustering, a review." *ACM Computing Surveys* 265–323.
- [3] Abdi, Hervé. 2010. "Normalizing Data." By Hervé Abdi. The University of Texas at Dallas: In Neil Salkind (Ed.), *Encyclopedia of Research Design*.
- [4] C., Alan V. Oppenheim and George. 2010. "Chapter 11 Wiener Filtering." In *Introduction to Communication, Control, and Signal Processing*, by Alan V. Oppenheim and George C., 195-210. Verghese: Spring.
- [5] C.O.S. Sorzano, E. Recarte, M. Alcorlo, J.R. Bilbao-Castro, C. San-Martín, R. Marabini, and J.M. Carazo. 2009. "Automatic particle selection from electron micrographs using machine learning techniques." *Journal of Structural Biology* 252 – 260.
- [6] Doerr, Allison. 2016. "Single-particle cryo-electron microscopy." *Nature Methods* 23. <https://www.nature.com/articles/nmeth.3700>.
- [7] Feng Wang, Huichao Gong, Gaochao Liu, Meijing Li, Chuangye Yan, Tian Xia, Xueming Li, and Jianyang Zeng. 2016. "DeepPicker: a Deep Learning Approach for Fully Automated Particle Picking in Cryo-EM." *Journal of Structural Biology* 325-336.
- [8] Zhang Y, Sun B, Feng D, Hu H, Chu M, Qu Q, Tarrasch JT, Li S, Kobilka TS, Kobilka BK. "Cryo-EM structure of the activated GLP-1 receptor in complex with a G protein". *Nature*. 2017;546(7657):248.
- [9] Parmenter CD, Cane MC, Zhang R, Stoilova-McPhie S. "Cryo-electron microscopy of coagulation factor VIII bound to lipid nanotubes". *Biochem Biophys Res Commun*. 2008;366(2):288–93.
- [10] Allison Doerr, "Single-particle cryo-electron microscopy", *Nature Methods* volume13, page23, 2016.

- [11] Fa Z, Yu C, Fei R, Xuan W, Zhiyong L, Xiaohua W. A two-phase improved correlation method for automatic particle selection in CryoEM. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. 2017;14(2):316–25.
- [12] Jingrong Zhang, Zihao Wang, Yu Chen, Renmin Han , Zhiyong Liu, Fei Sun, Fa Zhang, “PIXER: an automated particle-selection method based on segmentation using a deep neural network”, *BMC Bioinformatics* (2019) 20:41 <https://doi.org/10.1186/s12859-019-2614-y>.
- [13] Frank J. Three-dimensional electron microscopy of macromolecular assemblies. New York: Oxford U. Press; 2006.
- [14] Liu, H., Jin, L., Koh, S.B.S., Atanasov, I., Schein, S., et al., 2010. “Atomic structure of human adenovirus by cryo-EM reveals interactions among protein networks”. *Science* (New York, NY) 329 (5995), 1038–1043.
- [15] Scheres, S.H.W., 2012. “A bayesian view on cryo-EM structure determination”. *Journal of Molecular Biology* 415 (2), 406–418.
- [16] Daniel Cressey, Ewen Callaway, “Cryo-electron microscopy wins chemistry Nobel”, *Nature*, 04 October 2017.
- [17] Ewen Callaway,” The revolution will not be crystallized: a new method sweeps through structural biology”, *Nature*, 09 September 2015.
- [18] Nicola Jones, “Crystallography: Atomic secrets”, *Nature*, 29 January 2014.
- [19] Rebecca F. Thompson, Matt Walker, C. Alistair Siebert, Stephen P. Muench,a and Neil A. Ranson,”An introduction to sample preparation and imaging by cryo-electron microscopy for structural biology”, *Methods*. 2016 May 1; 100: 3–15.
- [20] Lau W.C.Y., Rubinstein J.L. “Single particle electron microscopy. *Methods Mol Biol*. 2013; 955:401–426.
- [21] Cheng Y., Grigorieff N., Penczek P.A., Walz T. “A primer to single-particle cryo-electron microscopy”. *Cell*. 2015; 161:438–449.
- [22] Egelman E.H. “Three-dimensional reconstruction of helical polymers. *Arch. Biochem. Biophys*”. 2015; 581:54–58.

- [23] Yifan Cheng, Nikolaus Grigorieff, Pawel A. Penczek, Thomas Walz, "A Primer to Single-Particle Cryo-Electron Microscopy", Cell. Author manuscript; available in PMC 2016 Apr 23.
- [24] Adil Al-Azzawi, Anes Ouadou, John J. Tanner, Jianlin Cheng, "AutoCryoPicker: an unsupervised learning approach for fully automated single particle picking in Cryo-EM images", BMC Bioinformatics, volume 20, Article number: 326 (2019).
- [25] Adil Al-Azzawi, Anes Ouadou, John J. Tanner, Jianlin Cheng, "Super Clustering Approach for Fully Automated Single Particle Picking", genes-547154, 2019.
- [26] Adiga PS, Malladi R, Baxter W, Glaeser RM. "A binary segmentation approach for boxing ribosome particles in cryo EM micrographs". J Struct Biol. 2004; 145:142–151.
- [27] Lucic V., Forster F., Baumeister W. "Structural studies by electron tomography: from cells to molecules". Annu. Rev. Biochem. 2005; 74:833–865.
- [28] Schenk A.D., Castaño-Diez D., Gipson B., Arbeit M., Zeng X., Stahlberg H. "3D reconstruction from 2D crystal image and diffraction data". Meth. Enzymol. 2010; 482:101–129.
- [29] Arbeit M., Castaño-Diez D., Thierry R., Gipson B.R., Zeng X., "Stahlberg H. Bacteriophages. Humana Press; Totowa, NJ: 2012". Image processing of 2D crystal images. pp. 171-194.
- [30] Joachim Frank, "Three-dimensional electron microscopy of macromolecular assemblies", Visualization of biological molecules in their native state, 2006.
- [31] GENERAL METHODS, "SINGLE PARTICLE RECONSTRUCTION", available at, https://www.emblhamburg.de/biosaxs/courses/embo2017/slides/EMBO_cryoEM_2_Bhushan.pdf.
- [32] Willy Wriggers, Ph.D., "Single Particle Reconstruction Techniques", School of Health Information Sciences <http://biomachina.org/courses/structures/09.html>.
- [33] Houston, "Cryo Electron Microscopy of Macromolecular Assemblies". Graduate Course offered at New York Structural Biology Center with David Stokes, Joachim Frank, et al., .

- [34] Joachim Frank, “Three-Dimensional Electron Microscopy of Macromolecular Assemblies”, 1996, Academic Press.
- [35] Robbie Ostrow, Trevor Tsue and Shalom Rottman-Yang, “Single-Particle Cryo-Electron Microscopy”, available at: http://cs371.stanford.edu/2018_slides/cryoem.pdf.
- [36] Orlova, E. V., & Saibil, H. R. (2011), “Structural Analysis of Macromolecular Assemblies by Electron Microscopy”, *Chemical Reviews*, 111(12), 7710–7748. doi:10.1021/cr100353t.
- [37] Parmenter CD, Cane MC, Zhang R, Stoilova-McPhie S. Cryo-electron microscopy of coagulation factor VIII bound to lipid nanotubes. *Biochem Biophys Res Commun*. 2008;366(2):288–93. 4.
- [38] Fa Z, Yu C, Fei R, Xuan W, Zhiyong L, Xiaohua W. A two-phase improved correlation method for automatic particle selection in CryoEM. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. 2017;14(2):316–25. 5.
- [39] Scheres SH. RELION: implementation of a Bayesian approach to cryo-EM structure determination. *J Struct Biol*. 2012;180(3):519–30. 6.
- [40] De la Rosa-Trevín J, Otón J, Marabini R, Zaldivar A, Vargas J, Carazo J, Sorzano C. Xmipp 3.0: an improved software suite for image processing in electron microscopy. *J Struct Biol*. 2013;184(2):321–8. 7.
- [41] Gatys LA, Ecker AS, Bethge M: A neural algorithm of artistic style. arXiv preprint arXiv:150806576 2015. 8. Shen D, Wu G, Suk H-I. Deep learning in medical image analysis. *Annu Rev Biomed Eng*. 2017; 19:221–48
- [42] Yu Z, et al. Detecting circular and rectangular particles based on geometric feature detection in electron micrographs. *J Struct Biol*. 2004; 145:168–80. 12.
- [43] Mallick SP, et al. Detecting particles in cryo-EM micrographs using learned features. *J Struct Biol*. 2004; 145:52–62. 13.
- [44] Sorzano COS, et al. Automatic particle selection from electron micrographs using machine learning techniques. *J Struct Biol*. 2009; 167:252–60.
- [45] Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL. Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans Pattern Anal Mach Intell*. 2018;40(4):834–48. 15.

- [46] Lawson CL, Patwardhan A, Baker ML, Hryc C, Garcia ES, Hudson BP, Lagerstedt I, Ludtke SJ, Pintilie G, Sala R. EMDDataBank unified data resource for 3DEM. *Nucleic Acids Res.* 2015;44(D1): D396–403.
- [47] Zhao J, et al.,” TMaCS: a hybrid template matching and classification system for partially-automated particle selection”. *J Struct Biol.* 2013; 181:234–42.
- [48] Langlois R, et al. Automated particle picking for low-contrast macromolecules in cryo-electron microscopy. *J Struct Biol.* 2014; 186:1–7.
- [49] Zhu Y, Ouyang Q, Mao Y. A deep convolutional neural network approach to single-particle recognition in cryo-electron microscopy. *BMC bioinformatics.* 2017;18(1):348.
- [50] Wang F, Gong H, Liu G, Li M, Yan C, Xia T, Li X, Zeng J. DeepPicker: a deep learning approach for fully automated particle picking in cryo-EM. *J Struct Biol.* 2016;195(3):325–36.
- [51] Xiao Y, Yang G: A fast method for particle picking in cryo-electron micrographs based on fast R-CNN. In: *AIP Conference Proceedings: 2017.* AIP Publishing: 020080.
- [52] Kim LY, Rice WJ, Eng ET, Kopylov M, Cheng A, Raczkowski AM, Jordan KD, Bobe D, Potter CS, Carragher B. Benchmarking cryo-EM single particle analysis workflow. *Front Mol Biosci.* 2018;5.
- [53] Merk, A., A. Bartesaghi, S. Banerjee, V. Falconieri, P. Rao, M.I. Davis, R. Pragani, M.B. Boxer, L.A. Earl, J.L.S. Milne, S. Subramaniam. 2016. “Breaking Cryo-EM Resolution Barriers to Facilitate Drug Discovery.” *Cell* 165(7):1698-1707.
- [54] Doerr, Allison. 2016. “Single-particle cryo-electron microscopy.” *Nature Methods* 23. <https://www.nature.com/articles/nmeth.3700>.
- [55] Jiang, J., B.L. Pentelute, R.J. Collier, Z.H. Zhou. 2015. “Atomic structure of anthrax protective antigen pore elucidates toxin translocation.” *Nature* 521(7553):545-9.
- [56] A. Bartesaghi, A. Merk, S. Banerjee, D. Matthies, X. Wu, J.L. Milne, S. Subramaniam. 2015. “2.2 Å resolution cryo-EM structure of beta-galactosidase in complex with a cell-permeant inhibitor”, *Science* 348(6239):1147-51.

- [57] Campbell, M.G., D. Veessler, A. Cheng, C.S. Potter, B. Carragher. 2015. “2.8 Å resolution reconstruction of the *Thermoplasma acidophilum* 20S proteasome using cryo-electron microscopy”, *Elife* 4.
- [58] Herzik, M.A., Jr., M. Wu, G.C. Lander. 2017. “Achieving better-than-3-Å resolution by single-particle cryo-EM at 200 keV.”, *Nat Methods* 14(11):1075-1078.
- [59] Yuanxin Zhu, Bridget Carragher, Robert M Glaeser, Denis Fellmann, Chandrajit Bajaj, Marshall. 2004. “Automatic particle selection: results of a comparative study”, *Journal of Structural Biology* 3 – 14.
- [60] Glaeser., William V. Nicholson, Robert M. 2001. “Review: Automatic particle detection in electron”, *Journal of Structural Biology* 90 – 101.
- [61] P.S Umesh Adiga, Ravi Malladi, William Baxter, Robert M Glaeser. 2004. “A binary segmentation approach for boxing ribosome particles in cryo EM micrographs”, *Journal of Structural* 142 – 151.
- [62] N.R. Voss, C.K. Yoshioka, M. Radermacher, C.S. Potter, B. Carragher. 2009. “DoG Picker and TiltPicker: Software tools to facilitate particle selection in single particle electron microscopy”, *Journal of Structural Biology* 205 – 213.
- [63] Jianhua Zhao, Marcus A. Brubaker, John L. Rubinstein. 2013. “TMaCS: A hybrid template matching and classification system for partially-automated particle selection”, *Journal of Structural Biology* 234 – 242.
- [64] Z. Liu, F. Guo, F. Wang, T.-C. Li, and W. Jiang. 2016. “a resolution cryo-em 3d reconstruction of close-packed virus particles”, *Structure* 319–328.
- [65] Ramin Norousi, Stephan Wickles, Christoph Leidig, Thomas Becker, Volker J. Schmid, Roland Beckmann, Achim Tresch. 2013. “Automatic post-picking using MAPPOS improves particle image detection from cryo-EM micrographs”, *Journal of Structural Biology* 59–66.
- [66] Grigorieff, James Z. Chen, Nikolaus. 2007. “SIGNATURE: A single-particle selection system for molecular electron microscopy”, *Journal of Structural Biology* 168 – 173.
- [67] Patwardhan, Richard J Hall, Ardan. 2004. “A two step approach for semi-automated particle selection from low contrast cryo-electron micrographs”, *Journal of Structural Biology* 19 – 28.

- [68] Penczek, Zhong Huang and Pawel A. 2004. “Application of template matching technique to particle detection in electron micrographs”, *Journal of Structural Biology* 29 – 40.
- [69] Robert Langlois, Jesper Pallesen, Joachim Frank. 2011. “Reference-free particle selection enhanced with semi-supervised machine learning for cryo-electron microscopy”, *Journal of Structural Biology* 353 – 361.
- [70] C. Sorzano, E. Recarte, M. Alcorlo, J.R. Bilbao-Castro, C. San-Martín, R. Marabini, J.M. Carazo. 2009. “Automatic particle selection from electron micrographs using machine learning techniques”, *Journal of Structural Biology* 252 – 260.
- [71] Pablo Arbez, Bong-Gyoon Han, Dieter Typke, Joseph Lim, Robert M. Glaeser, Jitendra Malik. 2011. “Experimental evaluation of support vector machine-based and correlation-based approaches to automatic particle selection”, *Journal of Structural Biology* 319 – 328.
- [72] Feng Wang, Huichao Gong, Gaochao Liu, Meijing Li, Chuangye Yan, Tian Xia, Xueming Li, Jianyang Zeng. 2016. “DeepPicker: a Deep Learning Approach for Fully Automated Particle Picking in Cryo-EM”, *Journal of Structural Biology* 325-336.
- [73] Yanan Zhu, Qi Ouyang, Youdong Mao. 2017. “A deep convolutional neural network approach to single-particle recognition in cryo-electron microscopy”, *BMC Bioinformatics* 2-10.
- [74] MacQueen, J. 1967. “Some methods for classification and analysis of multivariate observations”, in *Proc. 5th Berkeley Symp. on Math. Stat. and Probability*. Berkeley, CA. 281-297.
- [75] J. C. Dunn. 1973. “A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters”, *J. Cybern* 32-57.
- [76] G. Tang, L. Peng, P.R. Baldwin, D.S. Mann, W. Jiang, I. Rees & S.J. Ludtke. n.d. “EMAN2: an extensible image processing suite for electron microscopy”, *J Struct Biol.* 157 (PMID: 16859925): 38-46.
- [77] Guo F., Jiang W. (2014) Single Particle Cryo-electron Microscopy and 3-D Reconstruction of Viruses. In: Kuo J. (eds) *Electron Microscopy. Methods in Molecular Biology (Methods and Protocols)*, vol 1117. Humana Press, Totowa, NJ

- [78] Herv'e, Abdi. 2010. "Normalizing Data", By Herv'e Abdi. The University of Texas at Dallas: In Neil Salkind (Ed.), Encyclopedia of Research Design.
- [79] The MathWorks, Inc. 2018. Image Processing Toolbox™ User's Guide. Natick, MA: The MathWorks, Inc. <https://www.mathworks.com/help/images/contrast-adjustment.html>.
- [80] Woods, R. C. Gonzalez, R. E. 2018. "Digital Image Processing", 4th Edition. University of Tennessee.
- [81] Amit Singer," Mathematics for cryo-electron microscopy", arXiv:1803.06714v1 [physics. comp-ph] 12 Mar 2018.
- [82] Tejal Bhamre," Denoising and Covariance Estimation of Single Particle Cryo-EM Images", Preprint submitted to Journal of Structural Biology April 7, 2016.
- [83] Stark, J. Alex. 2000. "Adaptive Image Contrast Enhancement Using Generalizations of Histogram Equalization." IEEE TRANSACTIONS ON IMAGE PROCESSING 889-869.
- [84] Kaiming He, Jian Sun, Xiaoou Tang. 2013. "Guided Image Filtering." IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [85] Grant T, Rohou A, Grigorieff N. 2017. EMPIAR-10146. 07 12. Accessed 03 09, 2018. <https://www.ebi.ac.uk/pdbe/emdb/empiar/entry/10146/#&gid=1&pid=1>.
- [86] N.d.," K LH Dataset", available Online, <http://nramm.nysbc.org/>.
- [87] Atherton, D. Kerbyson, T. 1995. "Circle detection using Hough transform filters." Proc. 5th Int. Conf. Image Process, Appl., U.K. 370–374.
- [88] Steve on Image Processing, "Feret Properties – Wrapping Up", concepts, algorithms & MATLAB, <https://blogs.mathworks.com/steve/2018/04/17/feret-properties-wrapping-up/>.
- [89] Langlois, R. 2011. "A clarification of the terms used in comparing semi-automated particle selection algorithms in Cryo-EM", J. Struct. Biol 348-352.
- [90] Rohlfing, Torsten. 2012. "Image Similarity and Tissue Overlaps as Surrogates for Image Registration Accuracy: Widely Used but Unreliable." IEEE Trans Med Imaging 153–163.
- [91] Nogales, E.; Scheres, S.H. Cryo-EM: A Unique Tool for the Visualization of Macromolecular Complexity. Mol. Cell 2015, 58, 677–689.

- [92] Callaway, E. The revolution will not be crystallized: A new method sweeps through structural biology. *Nature* 2015, 525, 172–174.
- [93] Merk, A.; Bartesaghi, A.; Banerjee, S.; Falconieri, V.; Rao, P.; Davis, M.I.; Pragani, R.; Boxer, M.B.; Earl, L.A.; Milne, J.L.S.; et al. Breaking Cryo-EM Resolution Barriers to Facilitate Drug Discovery *Cell* 2016, 165, 1698–1707.
- [94] Jiang, J.; Pentelute, B.L.; Collier, R.J.; Zhou, Z.H. Atomic structure of anthrax protective antigen pore elucidates toxin translocation. *Nature* 2016, 521, 545–549.
- [95] Bartesaghi, A.; Merk, A.; Banerjee, S.; Matthies, D.; Wu, X.; Milne, J.L.; Subramaniam, S. 2.2 A resolution cryo-EM structure of beta-galactosidase in complex with a cell-permeant inhibitor. *Science* 2015, 348, 1147–1151.
- [96] Campbell, M.G.; Veessler, D.; Cheng, A.; Potter, C.S.; Carragher, B. 2.8 A resolution reconstruction of the *Thermoplasma acidophilum* 20S proteasome using cryo-electron microscopy. *Elife* 2015, 4, e06380.
- [97] Herzik, M.A., Jr.; Wu, M.; Lander, G.C. Achieving better-than-3-Å resolution by single-particle cryo-EM at 200 keV. *Nat. Methods* 2017, 14, 1075–1078.
- [98] Khoshouei, M.; Radjainia, M.; Baumeister, W.; Danev, R. Cryo-EM structure of haemoglobin at 3.2 Å determined with the Volta phase plate. *Nat. Commun.* 2017, 8, 16099.
- [99] Li, K.; Sun, C.; Klose, T.; Irimia-Dominguez, J.; Vago, F.S.; Vidal, R.; Jiang, W. Sub-3Å apoferritin structure determined with full range of phase shifts using a single position of volta phase plate. *J. Struct. Biol.* 2019, 206, 225–232.
- [100] Doerr, A. Single-particle cryo-electron microscopy. *Nat. Methods* 2016, 13, 23.
- [101] Chen, J.Z.; Grigorieff, N. A single-particle selection system for molecular electron microscopy. *J. Struct. Biol.* 2007, 157, 168–173.
- [102] Hall, R.J.; Patwardhan, A. A twostep approach for semi-automated particle selection from low contrast cryo-electron micrographs. *J. Struct. Biol.* 2004, 145, 19–28.
- [103] Huang, Z.; Penczek, P.A. Application of template matching technique to particle detection in electron micrographs. *J. Struct. Biol.* 2004, 145, 29–40.
- [104] Langlois, R.; Pallesen, J.; Frank, J. Reference-free particle selection enhanced with semi-supervised machine learning for cryo-electron microscopy. *J. Struct. Biol.* 2011, 175, 353–361.

- [105] Sorzano, C.O.S.; Recarte, E.; Alcorlo, M.; Bilbao-Castro, J.R.; San-Martín, C.; Marabini, R.; Carazo, J.M. Automatic particle selection from electron micrographs using machine learning techniques. *J. Struct. Biol.* 2009, 167, 252–260.
- [106] Arbeláez, P.; Han, B.G.; Typke, D.; Lim, J.; Glaeser, R.M.; Malik, J. Experimental evaluation of support vector machine-based and correlation-based approaches to automatic particle selection. *J. Struct. Biol.* 2011, 175, 319–328.
- [107] Wang, F.; Gong, H.; Liu, G.; Li, M.; Yan, C.; Xia, T.; Li, X.; Zeng, J. DeepPicker: A Deep Learning Approach for Fully Automated Particle Picking in Cryo-EM. *J. Struct. Biol.* 2016, 195, 325–336.
- [108] Zhu, Y.; Ouyang, Q.; Mao, Y. A deep convolutional neural network approach to single-particle recognition in cryo-electron microscopy. *BMC Bioinform.* 2017, 18, 348.
- [109] Adiga, P.U.; Malladi, R.; Baxter, W.; Glaeser, R.M. A binary segmentation approach for boxing ribosome particles in cryo EM micrographs. *J. Struct. Biol.* 2004, 145, 142–151.
- [110] Glaeser, W. V., Review: Automatic particle detection in electron. *Journal of Structural Biology*, 2001, 90 – 101.
- [111] Jianhua Z.; M. A., TMaCS: A hybrid template matching and classification system for partially-automated particle selection. *Journal of Structural Biology*, 2013, 234 – 242.
- [112] Voss, N.R.; Yoshioka, C.K.; Radermacher, M.; Potter, C.S.; Carragher, B. DoG Picker and TiltPicker: Software tools to facilitate particle selection in single particle electron microscopy. *J. Struct. Biol.* 2009, 166, 205–213.
- [113] Radhakrishna, A.; Appu, S.; Kevin, S.; Pascal, F.; Sabine, S. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Trans. Pattern Anal. Mach. Intell.* 2012, 34, 2274–2282.
- [114] J. MacQueen., Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, 281–297.
- [115] Dunn, J.C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *J. Cybern.* 1973, 3, 32–57.

- [116] Al-Azzawi, A.; Ouadou, A.; Tanner, J.; Cheng, J., AutoCryoPicker: an unsupervised learning approach for fully automated single particle picking in Cryo-EM images. *BMC Bioinformatics*, 2019, 1-26.
- [117] Rebecca F.; M. W., An introduction to sample preparation and imaging by cryo-electron microscopy for structural biology. *Methods*, 2016, 3-15.
- [118] Robert L.; J. P., Reference-free particle selection enhanced hanced with semi-supervised machine learning for cryo-electron microscopy. *Journal of Structural Biology*, 2011, 353 – 361.
- [119] Wong, W.; Bai X.; Brown, A.; Fernandez, I.; Hanssen E.; Condrón, M.; Tan, Y.; Baum, J.; Scheres, S. Cryo-EM structure of the Plasmodium falciparum 80S ribosome bound to the anti-protozoan drug emetine. *Elife* 2014, 3, e03080, PMID: 24913268, doi:10.7554/elife.03080.
- [120] Scheres, S. Beta-galactosidase Falcon-II micrographs plus manually selected coordinates by Richard Henderson. *J. Struct. Biol.* 2015, 189, 114–122, PMID: 25486611, doi: 10.1016/j.jsb.2014.11.010.
- [121] Tang, G.; Peng, L.; Baldwin, P.R.; Mann, D.S.; Jiang, W.; Rees, I.; Ludtke, S.J. EMAN2: An extensible image processing suite for electron microscopy. *J. Struct. Biol.* 2007, 157, 38–46, PMID: 16859925.
- [122] Pettersen, E.; Goddard, T.; Huang, C.; Couch, G.; Greenblatt, D.; Meng, E.; Ferrin, T.; J Comput, C. UCSF Chimera-a visualization system for exploratory research and analysis. *J. Comput. Chem.* 2004, 25, 1605–1612.
- [123] The Electron Microscopy Data Bank (EMDB) at PDBe, Available online: <http://www.ebi.ac.uk/pdbe/emdb/> (accessed on 01/2018).
- [124] Woods, G. Digital Image Processing, 4th ed.; University of Tennessee: Richard E. Woods, MedData Interactive, USA, 2018.
- [125] Scipion, Cryo em image processing framework. Integration, traceability and analysis. Available online: <http://scipion.i2pc.es> (accessed on 01/2019).
- [126] Zhang Y, Sun B, Feng D, Hu H, Chu M, Qu Q, Tarrasch JT, Li S, Kobilka TS, Kobilka BK. Cryo-EM structure of the activated GLP-1 receptor in complex with a G protein. *Nature*. 248 (2017).

- [127] Parmenter CD, Cane MC, Zhang R, Stoilova-McPhie S. Cryo-electron microscopy of coagulation factor VIII bound to lipid nanotubes. *Biochem Biophys Res Commun* 366, 288–93(2018).
- [128] Jingrong Zhang, Zihao Wang, Yu Chen, Renmin Han, Zhiyong Liu¹, Fei Sun, Fa Zhang. PIXER: an automated particle-selection method based on segmentation using a deep neural network. *BMC Bioinformatics* 20:41 (2019).
- [129] Frank J. Three-dimensional electron microscopy of macromolecular assemblies. New York: Oxford U. Press, (2006).
- [130] Yanan Zhu¹, Qi Ouyang, Youdong Mao. A deep convolutional neural network approach to single-particle recognition in cryo-electron microscopy. *BMC Bioinformatics*, 18:348 (2017).
- [131] Langlois R, et al. Automated particle picking for low-contrast macromolecules in cryo-electron microscopy. *J Struct Biol.*, 186:1–7 (2014).
- [132] G. Tang, L. Peng, P.R. Baldwin, D.S. Mann, W. Jiang, I. Rees & S.J. Ludtke. n.d. “EMAN2: an extensible image processing suite for electron microscopy”, *J Struct Biol.* 157 (PMID: 16859925): 38-46.
- [133] Wang F, Gong H, Liu G, Li M, Yan C, Xia T, Li X, Zeng J. DeepPicker: a deep learning approach for fully automated particle picking in cryo-EM. *J Struct Biol.* 195(3):325–36 (2016).
- [134] Xiao Y, Yang G: A fast method for particle picking in cryo-electron micrographs based on fast R-CNN. In: *AIP Conference Proceedings*: AIP Publishing: 020080 (2017).
- [135] Scheres SH. RELION: implementation of a Bayesian approach to cryo-EM structure determination. *J Struct Biol.* 2012;180(3):519–30.
- [136] Adil Al-Azzawi, Anes Ouadou, John J. Tanner, and Jianlin Cheng. AutoCryoPicker: an unsupervised learning approach for fully automated single particle picking in cryo-EM images. *BMC Bioinformatics*, accepted, (2019).
- [137] Adil Al-Azzawi, Anes Ouadou, Jianlin Cheng. A Super Clustering Approach for Fully Automated Single Particle Picking in Cryo-EM. *ICIBM International Conference on Intelligent Biology and Medicine*, Genes, accepted, (2019).
- [138] N.d.,” KLH Dataset”, available Online, <http://nramm.nysbc.org/>.

- [139] Grant T, Rohou A, Grigorieff N. 2017. EMPIAR-10146. 07 12. Accessed 03 09, (2018).
- [140] Wong W, Bai XC, Brown A, Fernandez IS, Hanssen E, Condrón M, Tan YH, Baum J, and Scheres SH, "Cryo-EM structure of the Plasmodium falciparum 80S ribosome bound to the anti-protozoan drug emetine", *Elife* 3, PMID: 24913268, DOI: 10.7554/elife.03080, (2014).
- [141] Scheres SH, "Beta-galactosidase Falcon-II micrographs plus manually selected coordinates by Richard Henderson", *J. Struct. Biol.* PMID: 25486611, DOI: 10.1016/j.jsb.2014.11.010, 189 114-122 (2015).
- [142] Waibel A, et al. Phoneme recognition using time-delay neural network. *IEEE Trans Acoustics Speech Signal Process.* 1989; 37:328–39.
- [143] Mallick SP, et al. Detecting particles in cryo-EM micrographs using learned features. *J Struct Biol.* 2004; 145:52–62.
- [144] https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html
- [145] Andrew Ng. et al. Feature extraction using convolution. <http://ufldl.stanford.edu/tutorial/supervised/FeatureExtractionUsingConvolution/>. 2015.
- [146] Rumelhart DE, et al. learning representations by back-propagating errors. *Nature.* 1986; 323:533–6.
- [147] Langlois R, et al. A clarification of the terms used in comparing semi-automated particle selection algorithms in Cryo-EM. *J Struct Biol.* 2011; 175:348–52.
- [148] Derui Wang, Chaoran Li, Sheng Wen, Surya Nepa, Yang Xiang. Daedalus: Breaking Non-Maximum Suppression in Object Detection via Adversarial Examples. arXiv:1902.02067v1 [cs.CV] 6 Feb (2019).
- [149] Koning RI, Gomez-Blanco J, Akopjana I, Vargas J, Kazaks A, Tars K, Carazo JM, Koster, AJ. Asymmetric cryo-EM reconstruction of phage MS2 reveals.
- [150] Herzik Jr MA, Wu M, Lander GC. T. acidophilum 20S proteasome core movies obtained using Talos Arctica operating at 200 kV equipped with a K2 – image shift used for exposure target navigation, *Nat. Methods* 14 1075-1078 (2017).
- [151] Bartesaghi A, Merk A, Banerjee S, Matthies D, Wu X, Milne JL, Subramaniam S. 2.2 Å resolution cryo-EM structure of beta-galactosidase in complex with a cell-permeant inhibitor, *Science* 348 1147-1151 (2015).

- [152] Roseman, A M. Particle finding in electron micrographs using a fast-local correlation algorithm. *Ultramicroscopy* 2003; 94:225–236.
- [153] Huang Z, et al. Application of template matching technique to particle detection in electron micrographs. *J Struct Biol.* 2004; 145:29–40.
- [154] Roseman, A M. FindEM- a fast, efficient program for automatic selection of particles from micrographs. *J Struct Biol* 2004; 145:91–99.
- [155] Rath BK, Frank J. Fast automatic particle picking from cryo-electron micrographs using a locally normalized cross-correlation function: a case study. *J Struct Biol.* 2004; 145:84–90.
- [156] Chen JZ, Grigorieff N, et al. SIGNATURE: a single-particle selection system for molecular electron microscopy. *J Struct Biol.* 2007; 157:168–73.
- [157] Scheres S. RELION: implementation of a Bayesian approach to cryo-EM structure determination. *J Struct Biol.* 2015; 180:519–30.
- [158] Adiga U, et al. Particle picking by segmentation: a comparative study with SPIDER-based manual particle picking. *J Struct Biol.* 2005; 152:211–20.
- [159] Woolford D, et al. SwarmPS: rapid, semi-automated single particle selection software. *J Struct Biol.* 2007; 157:174–88.
- [160] Yu Z, et al. Detecting circular and rectangular particles based on geometric feature detection in electron micrographs. *J Struct Biol.* 2004; 145:168–80.
- [161] Sorzano COS, et al. Automatic particle selection from electron micrographs using machine learning techniques. *J Struct Biol.* 2009; 167:252–60.
- [162] Grigore D. Pintilie, Segmentation and Registration of Molecular Components in 3-Dimensional Density Maps from Cryo-Electron Microscopy, Massachusetts Institute of Technology 2010.
- [163] L.D. Landau and E.M. Lifshitz, *Electrodynamics of Continuous Media*, Oxford: Pergamon, 1960.
- [164] C. Brändén and J. Tooze, *Introduction to protein structure*, New York (N.Y.): Garland, 1999.
- [165] <http://www.pdb.org>.
- [166] J.A. Kovacs and W. Wriggers, Fast rotational matching, *Biological Crystallography*, vol. 58, Aug. 2002, pp. 1282-6

- [167] A.M. Roseman, docking structures of domains into maps from cryo-electron microscopy using local correlation, *Biological Crystallography*, vol. 56, Oct. 2000, pp. 1332-40.
- [168] J.A. Kovacs, P. Chacón, Y. Cong, E. Metwally, and W. Wriggers, Fast rotational matching of rigid bodies by fast Fourier transform acceleration of five degrees of freedom, vol. 59, Aug. 2003, pp. 1371-1376.
- [169] Ranson NA, Clare DK, Farr GW, Houldershaw D, Horwich AL, Saibil HR., Allosteric signaling of ATP hydrolysis in GroEL-GroES complexes, *Nat Struct Mol Biol*. 2006 Feb;13(2):147-52. Epub 2006 Jan 22.
- [170] Mikel Valle, Andrey Zavialov, Jayati Sengupta, Urmila Rawat, Ma ñs Ehrenberg, Joachim Frank, Locking and Unlocking of Ribosomal Motions, *Cell*, Vol. 114, 123–134, July 11, 2003.
- [171] Zhou Y1, Morais-Cabral JH, Kaufman A, MacKinnon R., Chemistry of ion coordination and hydration revealed by a K⁺ channel-Fab complex at 2.0 Å resolution, *Nature*. 2001 Nov 1;414(6859):43-8.
- [172] Lander GC1, Evilevitch A, Jeembaeva M, Potter CS, Carragher B, Johnson JE., Bacteriophage lambda stabilization by auxiliary protein gpD: timing, location, and mechanism of attachment determined by cryo-EM, *Structure*. 2008 Sep 10;16(9):1399-406. doi: 10.1016/j.str.2008.05.016.
- [173] N.A. Ranson, D.K. Clare, G.W. Farr, D. Houldershaw, A.L. Horwich, and H.R. Saibil, “Allosteric signaling of ATP hydrolysis in GroEL-GroES complexes,” *Nat Struct Mol Biol*, vol. 13, Feb. 2006, pp. 147-152.
- [174] Willy Wriggers, Ph.D, Single Particle Reconstruction Techniques, School of Health Information Sciences <http://biomachina.org/courses/structures/09.html> [Access 2019].
- [175] Allison Doerr, Single-particle cryo-electron microscopy, *Nature Methods* volume13, page23, 2016.
- [176] S.J. Ludtke, P.R. Baldwin, and W. Chiu, “EMAN: semiautomated software for high-resolution single-particle reconstructions,” *Journal of structural biology*, vol. 128, Dec. 1999, pp. 82-97.

- [177] Scheres S. RELION: implementation of a Bayesian approach to cryo-EM structure determination. *J Struct Biol.* 2015; 180:519–30.
- [178] T.R. Shaikh, H. Gao, W.T. Baxter, F.J. Asturias, N. Boisset, A. Leith, and J. Frank. SPIDER image processing for single-particle reconstruction of biological macromolecules from electron micrographs, *Nature Protocols*, vol. 3, 2008, pp. 1941-1974.
- [179] W.E. Lorensen and H.E. Cline, Marching cubes: A high resolution 3D surface construction algorithm, *SIGGRAPH Comput. Graph.*, vol. 21, 1987, pp. 163-169.
- [180] Native, unliganded GroEL, D7 symmetrized, 4.2 Å resolution 0.5 criterion, <https://www.ebi.ac.uk/pdbe/entry/emdb/EMD-5001>.
- [181] ATP-bound states of GroEL captured by cryo-electron microscopy, available at: <https://www.ebi.ac.uk/pdbe/entry/emdb/EMD-1042>.
- [182] Grant T, Rohou A, Grigorieff N. 2017. EMPIAR-10146. 07 12. Accessed 03 09, 2018. <https://www.ebi.ac.uk/pdbe/emdb/empiar/entry/10146/#&gid=1&pid=1>.
- [183] N.d., "KLH Dataset", available Online, <http://nramm.nysbc.org/>.
- [184] Adil Al-Azzawi, Anes Ouadou, John J. Tanner, and Jianlin Cheng. AutoCryoPicker: an unsupervised learning approach for fully automated single particle picking in cryo-EM images. *BMC Bioinformatics*, accepted, (2019).
- [185] Adil Al-Azzawi, Anes Ouadou, Jianlin Cheng. A Super Clustering Approach for Fully Automated Single Particle Picking in Cryo-EM. *Genes (Basel)*. 2019 Aug 30;10(9). pii: E666. doi: 10.3390/genes10090666.
- [186] Adil Al-Azzawi, Anes Ouadou, Highsmith Max R, John J. Tanner, and Jianlin Cheng, DeepCryoPicker: Fully Automated Deep Neural Network for Single Protein Particle Picking in cryo-EM, *bioRxiv preprint first posted online Sep. 10, 2019*; doi: <http://dx.doi.org/10.1101/763839>.
- [187] Steve on Image Processing and MATLAB, <https://blogs.mathworks.com/steve/2018/04/17/feret-properties-wrapping-up>.
- [188] Yifan Cheng, Nikolaus Grigorieff, Pawel A. Penczek, Thomas Walz, A Primer to Single-Particle Cryo-Electron Microscopy, *Cell*. Author manuscript; available in PMC 2016 Apr 23.

- [189] Lisa Gottesfeld Brown, A survey of image registration techniques (abstract), ACM Computing Surveys archive, volume 24, issue 4, December 1992), pages 325 - 376
- [190] Chen Xing and Peihua Qiu, Intensity-Based Image Registration by Nonparametric Local Smoothing, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 33, NO. 10, OCTOBER 2011.
- [191] R.J. Althof, M.G.J. Wind, and J.T. Dobbins, "A Rapid and Automatic Image Registration Algorithm with Subpixel Accuracy," IEEE Trans. Medical Imaging, vol. 16, no. 3, pp. 308-316, June 1997.
- [192] <https://www.mathworks.com/help/images/intensity-based-automatic-image-registration.html>.
- [193] Press, William H.; Teukolsky, Saul A.; Vetterling, William T.; Flannery, Brian P. (1992). Numerical recipes in C: the art of scientific computing (2nd ed.). New York, NY, USA: Cambridge University Press. pp. 123–128. ISBN 0-521-43108-5.
- [194] Woods, R. C. Gonzalez, R. E. 2018. "Digital Image Processing", 4th Edition. University of Tennessee.
- [195] Dimitri P. Bertsekas, Nonlinear Programming, Athena Scientific 1999, 2nd edition, pp. 187.
- [196] Cauchy, Augustin. "Méthode générale pour la résolution des systemes d'équations simultanées." Comp. Rend. Sci. Paris 25.1847 (1847): 536-538.
- [197] Lambros S. Athanasiou, Dimitrios I. Fotiadis, Lampros K. Michalis, Validation Using Histological and Micro-CT Data: Registration and Inflation Using IVUS, Atherosclerotic Plaque Characterization Methods Based on Coronary Imaging, 2017, Pages 167-180
- [198] O. Bottema & B. Roth (1990). Theoretical Kinematics. Dover Publications. reface. ISBN 0-486-66346-9.
- [199] Sjors H. W. Scheres,* and José-María Carazo, Introducing robustness to maximum-likelihood refinement of electron-microscopy data, Acta Crystallogr D Biol Crystallogr. 2009 Jul 1; 65(Pt 7): 672–678.
- [200] Bailey, David H.; Swarztrauber, Paul N. (1994), "A fast method for the numerical evaluation of continuous Fourier and Laplace transforms" (PDF), SIAM Journal on

1110, CiteSeerX 10.1.1.127.1534, doi:10.1137/0915067.

- [201] Hartley, Richard, and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Second Edition. Cambridge, 2000.
- [202] Zhang, Z. “A Flexible New Technique for Camera Calibration”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 22, No. 11, 2000, pp. 1330–1334.
- [203] Heikkila, J, and O. Silven. “A Four-step Camera Calibration Procedure with Implicit Image Correction”, *IEEE International Conference on Computer Vision and Pattern Recognition*, 1997.
- [204] Shi, J., and C. Tomasi, "Good Features to Track," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 1994, pp. 593–600.
- [205] <https://www.mathworks.com/help/vision/ref/detectmineigenfeatures.html>
- [206] Lucas, Bruce D. and Takeo Kanade. “An Iterative Image Registration Technique with an Application to Stereo Vision,” *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, April, 1981, pp. 674–679.
- [207] Tomasi, Carlo and Takeo Kanade. *Detection and Tracking of Point Features*, Computer Science Department, Carnegie Mellon University, April 1991.
- [208] Shi, Jianbo and Carlo Tomasi. “Good Features to Track,” *IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.
- [209] Kalal, Zdenek, Krystian Mikolajczyk, and Jiri Matas. “Forward-Backward Error: Automatic Detection of Tracking Failures,” *Proceedings of the 20th International Conference on Pattern Recognition*, 2010, pages 2756–2759, 2010.
- [210] <https://www.mathworks.com/vision/ref/vision.pointtracker-system-object.html>
- [211] Kukulova, Z., M. Bujnak, and T. Pajdla *Polynomial Eigenvalue Solutions to the 5-pt and 6-pt Relative Pose Problems*. Leeds, UK: BMVC, 2008.
- [212] Nister, D. “An Efficient Solution to the Five-Point Relative Pose Problem.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Volume 26, Issue 6, June 2004.
- [213] Torr, P. H. S., and A. Zisserman. “MLE-SAC: A New Robust Estimator with Application to Estimating Image Geometry.” *Computer Vision and Image Understanding*. Volume 78, Issue 1, April 2000, pp. 138-156.

- [214] Bouguet, J.Y. "Camera Calibration Toolbox for Matlab", Computational Vision at the California Institute of Technology. Camera Calibration Toolbox for MATLAB.
- [215] Bradski, G., and A. Kaehler. Learning OpenCV : Computer Vision with the OpenCV Library. Sebastopol, CA: O'Reilly, 2008.
- [216] Richard I. Hartley (June 1997). "In Defense of the Eight-Point Algorithm". IEEE Transactions on Pattern Recognition and Machine Intelligence. 19 (6): 580–593. doi:10.1109/34.601246.
- [217] Hartley, R. and A. Zisserman. "Multiple View Geometry in Computer Vision." Cambridge University Press, p. 312, 2003.
- [218] Moons, Theo, Luc Van Gool, and Maarten Vergauwen. "3D reconstruction from multiple images part 1: Principles." Foundations and Trends in Computer Graphics and Vision 4.4 (2010): 287-404.
- [219] Vosselman, George, and Sander Dijkman. "3D building model reconstruction from point clouds and ground, International archives of photogrammetry remote sensing and spatial information sciences 34.3/W4 (2001): 37-44.
- [220] Griffiths, D. V.; Smith, I. M. (1991). Numerical methods for engineers: a programming approach. Boca Raton: CRC Press. p. 218. ISBN 0-8493-8610-1.
- [221] <https://www.mathworks.com/help/vision/ref/pcshow.html>.
- [222] M., Amin-Naji; A., Aghagolzadeh (2018). "Multi-Focus Image Fusion in DCT Domain using Variance and Energy of Laplacian and Correlation Coefficient for Visual Sensor Networks". Journal of AI and Data Mining. 6 (2): 233–250. doi:10.22044/jadm.2017.5169.1624. ISSN 2322-5211.
- [223] K. C. Rajini ; S. Roopa, A review on recent improved image fusion techniques, 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET).
- [224] Herv'e, Abdi. 2010. "Normalizing Data", By Herv'e Abdi. The University of Texas at Dallas: In Neil Salkind (Ed.), Encyclopedia of Research Design. N.d., "KLH Dataset", available Online, <http://nramm.nysbc.org/>.
- [225] Zhou, W., A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. "Image Quality Assessment: From Error Visibility to Structural Similarity." IEEE Transactions on Image Processing. Vol. 13, Issue 4, April 2004, pp. 600–612.

VITA

Adil Al-Azzawi was born in Iraq, Diyala, Baqubah. He obtained his bachelor's degree in Software Engineering (SE) from Software Engineering Department, Baghdad, Iraq in August 2001. He received his Higher Diploma's degree in Software Engineering (SE) from the University of Information Technology and Communication, Baghdad, Iraq in December 2002. He received his master's degree in Computer Science (CS) from the University of Technology, Iraq in December 2005. Adil Al-Azzawi received his second master's degree in computer engineering from the University of Missouri, USA in May 2017.

In his first master's thesis, Adil Al-Azzawi worked on the image denoising and noise types identification and classification using with machine learning based artificial intelligence and neural network. While at University of Missouri, Electrical Engineering and Computer Science Department, his research was focused on biometric data identification and recognition. He worked on solve the rotational identification issue of the biometric fingerprint identification and authentication by designing an robust spatially invariant models for latent fingerprint authentication approach. Also, he worked on the exists significant challenge and complex conditions of face recognition such as large expression, pose, illumination by designing a new approach in which a localized Deep-CNN structure is applied to demonstrate its' effectiveness and efficiency based deep learning. Moreover, he worked on finding the correlations of features inside the sub region of each learning face by designing a new Deep Learning structure called Localized Deep-Norm CNN.

He started his Ph.D. studies in the Department of Electrical Engineering and Computer Science at the University of Missouri-Columbia. With his research interest in bioinformatics, machine learning, and deep learning, he has proposed, and been published in reputable journals, including BMC Bioinformatics, Genes and Nature Communication, novel methods for protein particles picking in cryo-EM.

His long-term goal is to develop computational methods for particles alignment, 3D density map generation, and carbon alpha prediction for 3D protein structure and structure determination that can be used to study and extract useful information for macromolecular structure determination and prediction from experimental cryo-EM data.