PSEUDO RANDOM FORESTS FOR TUBE IDENTIFICATION

A THESIS IN

Computer Science

Presented to the Faculty of the University
of Missouri-Kansas City in partial fulfillment of
the requirements for the degree

MASTER OF SCIENCE

by
KENDALL LEE BINGHAM

B.S. University of Missouri-Kansas City, 2004

Kansas City, Missouri
2015

PSEUDO RANDOM FORESTS FOR TUBE IDENTIFICATION

Kendall Lee Bingham, Candidate for the Master of Science Degree

University of Missouri-Kansas City, 2015

ABSTRACT

Random forests are widely used in machine learning as they can potentially offer higher accuracy than individual decision trees by the averaging of multiple independent models. We propose a modification called “Pseudo-random forests” that combines stochastic feature selection with dynamic problem-specific feature generation. As proof of concept, we apply the method to the problem of edge detection and classification in radiographic images. In particular, we use the method to detect feeding tubes in pediatric patients, which are inserted to deliver food and medicine. Since multiple layers of tissues and medical objects are overlaid in a single image, these can be difficult to read on x-rays, even for trained radiologists. The placement of these tubes is critical to the well-being and care of the patient. Automating the recognition of these tubes can help confirm the correct placement of these tubes, as an improperly placed tube could delay treatment or jeopardize the health of the patient. It can also save time by enhancing the visibility of tubes for interpretation by

radiologists, as hospitals may have to validate tens to hundreds of these x-rays a day. We report an average recall of 85% for tube pixel identification by using Pseudo-random forests for classification, based on leave-out-one cross-validation. Further improvement is possible by post-processing for tube continuity and the incorporation of other techniques developed as part of the research.

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the College of Computing and Engineering, have examined a thesis titled, “Pseudo Random Forests for Tube Identification”, presented by Kendall L. Bingham, candidate for the Master of Science degree, and certify that in their opinion it is worthy of acceptance.

Supervisory Committee

Deendayal Dinakarpandian M.D., Ph.D., Committee Chair
UMKC School of Computing and Engineering

Sherwin Chan M.D., Ph.D.
Radiologist Children’s Mercy Hospital

Ghulam Chaudhry Ph.D.
UMKC School of Computing and Engineering

Vijay Kumar Ph.D.
UMKC School of Computing and Engineering

CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Table Page

ACKNOWLEDGMENTS

This Thesis and research would not have been possible without the assistance of many people. Dr. Deendayal Dinakarpandian has been a wonderful advisor. He has provided guidance in finding a direction, encouragement when the well has come up dry, flexibility in allowing me to tilt at my own windmills along with the ability to bring me back to a productive middle ground. He has been incredibly generous with his time and thoughts and I will be eternally grateful for his assistance.

The research could not have been possible without radiographs, advice and the medical knowledge of Dr. Sherwin Chan.

Most importantly, I would like to express thanks to my wonderful and beautiful wife. Her patience and assistance during the course of this process has truly been an oasis. She has always been my biggest supporter and cheerleader.

CHAPTER 1

INTRODUCTION

"On average, 236 chest radiographs are taken per 1000 patients each year [1]". The abundance, relative ease, and cost effectiveness makes x-rays the most prevalent tool in assessing placement of feeding tubes. The portable convenience of these tools make them indispensable in Intensive Care Units (I.C.U.), and Emergency Departments (E.D.) for fast and reliable assessments. Since the placement of these tubes is critical for patient care, they may require ongoing evaluation of their positions as well. Radiographs, although effective, can suffer from noise and difficult to evaluate sections where intensities are similar. Computer aided diagnosis (C.A.D.) software can help highlight the areas that a Radiologist may need to look closer at. In a busy hospital where staff has to evaluate hundreds of patient x-rays, the C.A.D. software could help to triage the riskiest candidates that need to be assessed sooner.

Complications from improper tube placement can range from serious to fatal. The correct position of a Nasal Gastric (NT) or Feeding Tube (FT) is in the stomach. The tubes are used to deliver medicine and food. They can also be used as a sump to remove contents from the stomach. When the tube follows the airway instead of the stomach, aspiration can result. Aspiration is the process of inhaling a foreign object into the airway. In the case of a NT or FT it would mean inhaling the food or medicine that is meant to be digested. When medicine or food is mistakenly introduced to the lungs complications such as pneumonia can adversely affect the condition of the placement. The lungs aren't the only location where a feeding tube can be incorrectly positioned. It can be in the esophagus or down into the

intestines. There has even been a case where it has been incorrectly located in the brain[2]. While validation of tube position after the initial placement is important, they need to continually be assessed, since they can move out of place. The re-evaluation of the tubes can be done prior to each feeding or delivery of medicine, and also each day.

The risk of an incorrectly position tube in pediatric settings has been evaluated as high as 20-43.5%.[3] With the cost to the safety and the health of the patient being from the proper placement and the high error rate at placing these tubes, it is imperative to correctly identify the proper position.

There are many methods used to confirm proper placement. Some of these methods include pH testing of the aspirate, which can determine if the placement has overshot into the intestines. Observation of the contents of the aspirate can also be used. Evaluating the presence of carbon dioxide can help determine if the placement is in the lungs or the stomach. Air pressure in the tube can also be measured to determine if has been placed improperly in the lungs. However, radiographs are considered to be the most reliable and are used by many ICU and ED regularly to assess this information.

The use of an x-ray helps to eliminate much of the uncertainty to where the tube is currently located. Even with this assistance, errors can happen. Radiographs can sometimes be very difficult to read. Many of these patients may have multiple tubes, wires and non-biological features that make diagnosis difficult. To further complicate the tube location identification, the features of the body can obscure the location of the tubes. Figure *1* shows an x-ray with many of these features that show the difficulty in assessing these images.

Figure 1—Complex radiograph of tubes.

Many radiologists in large hospitals read dozens or hundreds of radiographs that have to be checked to ensure the safety of the patient. CAD software can potentially be used to enhance the image for the technician. It can also notify a technician of a possible problem before they've had a chance to review the image for a patient. The faster and more accurately that the incorrectly placed tubes can be identified, the fewer complications and compromises to the health of the patient.

We've chosen to try to find efficient reliable methods of helping radiologists identify problem tube placements. Early detection will help identify patients that need assistance

immediately. Enhancing and highlighting errors for the radiologist to look at closer can help them quickly identify and diagnose problems.

CHAPTER 2

REVIEW OF LITERATURE

Background

There are two main approaches to automatic tube identification; algorithmic [4-6] and machine learning [1]. Table 1 shows a summary of previous studies, with relative sample sizes and reported results. The algorithmic methods generally start with a region of interest (R.O.I.). Usually the ROI is of the neck region since ET and FT tubes have to pass through the esophagus. *Figure 2* shows the roots of the current work, divided into the sub roots of Machine Learning and Algorithms.

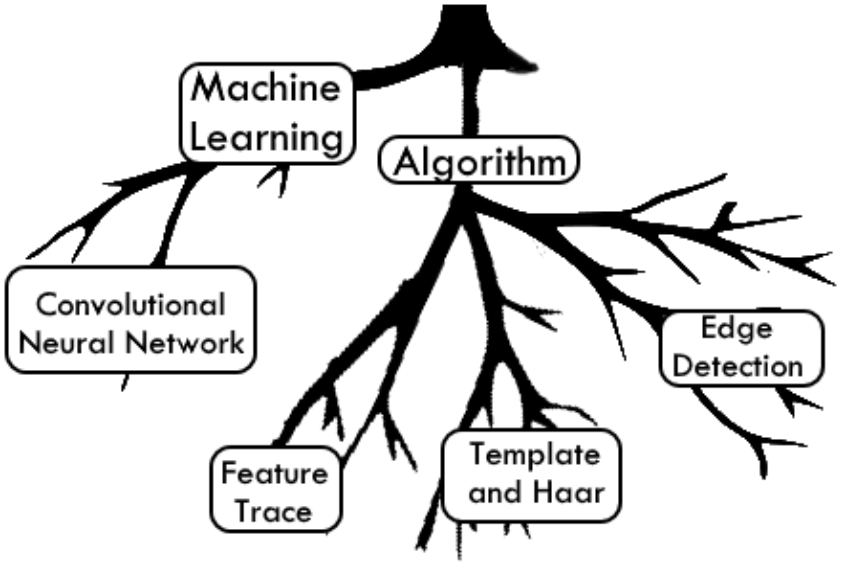


Figure 2-Current Approaches

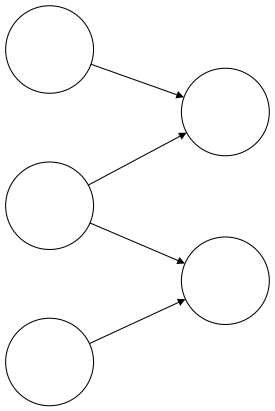
Table 1—Previous studies, sample sizes and results

Authors	Study	Sample Sizes	Result
Kao et al (2015)	Following Intensity based upon an initial seed point [3]	528 ET Tubes 816 w/o Tubes	94.3% Accuracy in seed point detection. 85.6% Accuracy in finding the ET tube tip within 5 mm.
Sheng et al (2009)	Hough Transform applied to horizontal slices. [4]	107 Images from 20 patients	94.0% ET detection 82.0% FT detection 92.0% NT detection
Cem Ahmet Mercan, Mustafa Serdar Celebi	Convolutional Neural Network with B-Splines [7]	21 w/ Tubes 247 w/o Tubes	99.9% Accuracy 59.13% Sensitivity 61 FP pixels/img 60 FN pixels/img
Ramakrishna et al (2012)	Detect Parallel lines with Hough transform in strips on the neck region. [5]	20 ET & NG 5 NG only 8 ET only 31 w/o Tubes	93.0% TPR 0.02 FPR ET 84.0% TPR 0.02 FPR NG
Huo et al (2008)	Haar-like template was used in the ROI. Templates were then used to enhance and detect the tubes. [6]	32 Training Images 121 Testing Images	91.0% Sensitivity Train 98.0% Sensitivity Test 87.0% Sensitivity LDA 92.0% Sensitivity QDA

Machine Learning Methods

Mercan and Celebi [1], use Convolutional Neural Networks and B-Splines to identify tubes in a Radiograph.

Convolutional Neural Networks (CNN) are a variation of artificial neural networks (ANN). CNN's are used in many image-processing tasks. They process images in the much same way that human eye and brain does. With a CNN there is no need for feature extraction since it works directly on the images. CNN's are not fully connected neural networks, but are sparsely connected with overlapping regions. Similar to ANN's, CNN's have multiple layers, many of which are designed with a specific task. There will always be a convolution layer, as shown in Figure 3, which can be thought of like a convolution kernel used in Edge Processing. However, since this is a learning algorithm there will be no convolution filter used; the weights of the input will be learned instead. There may be many convolution layers used in a CNN (see Figure 4). Besides convolution layers there can also be other types of layers, like pooling layers. These pooling layers pool the maximum, average, mean value of a feature over an area of the image. This helps to reduce the feature space of the image. Other layers are Rectifier Linear Units (ReLU layer), Dropout Layers, Loss Layers and more. These layers are combined in different orders and different configurations depending on the purpose.



Note: The layers are completely connected with all nodes.

Figure 3—Convolution Layer

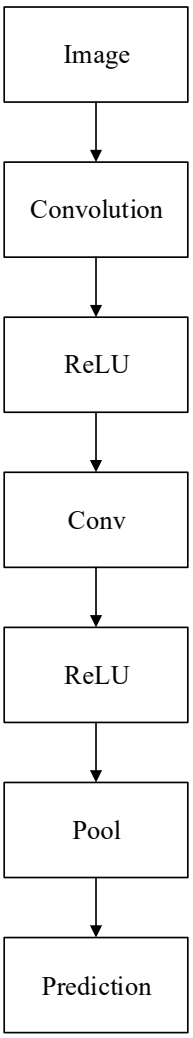


Figure 4—Convolutional Neural Network

In order to train the CNN, a pair of images for each X-ray are required. One is the original source image, and the other being the resulting identified tube. After tuning their CNN the results had breaks and sections missing from identified tubes. To connect these broken sections the authors attempted to connect the sections using Non-Uniform Rational B-Splines, or NURBS. Control points were selected by following along each continuous section and choosing every 36^{th} pixel. This study reported a pixel-by-pixel average recall of

59.46% on identifying Tube pixels. This is much different than their 99.9% accuracy for the whole process. Since there are so many more pixels that are not tubes than tube you could have an extremely high accuracy rate just by saying everything it saw wasn't a tube. The rate of correctly identifying a tube in this case was slightly less than 60% of the time. So out of 10 tube pixels they were correctly saying 6 were tubes, and missing 4.

Algorithmic Methods

Many of the algorithmic methods for finding tubes in radiographs begin by finding tube sections of seed points inside a Region of Interest (ROI). This differs from the Machine Learning method illustrated above, where they let the CNN itself find all possible tubes in the image regardless of the location. Since guiding a hand-made algorithm through all pixels of an image would be time consuming, it is typical to find a ROI first, and then after begin following the tube from that point.

Kao et al [4] found a ROI in the neck by finding the horizontal slice that was the thinnest section, which would likely be the neck. Their study also mentioned that finding the neck on the radiograph was complicated if the patient was not positioned correctly. After finding this slice they begin to determine the seed point of an Endotracheal Tube (ET). They applied a template based upon their examinations of ET Tubes. Using this template they scanned along the chosen slice, tilting the template from -30° to 30°. They convolved each point and rotation. Once they found the position that had the best match they would use that as their seed point. After using a line enhancing filter, they followed intensity of the seed point down. Choosing the next tube point by choosing the brightest of the pixels directly below the previous one. The end position of the ET tube was chosen by the greatest drop off

in intensity. They use a multiple threshold algorithm to find the largest drop in the intensity along the path and choose that as the end point of the ET tube. The result of this technique was 94.3 % accuracy in locating the seed point. They found the end of the ET tube within 5 mm of the actual end of the tube as identified by a radiologist 85.6% of the time. They surmised that much of this error was from incorrectly identifying the seed point.

In a few of the studies, the Hough Transform [7] was used to find sections of straight lines. The Hough transform is a popular technique for efficiently finding linear features in an image. Typically, the Hough transform is applied to a binary image. The desired result is to find the strong lines prevalent in the image. It would be inefficient and error prone to exhaustively trace through every possible line. The lines are usually not continuous either; they can have breaks and points missing. The Hough transform uses the idea of a parametrized space. Each binary point in the image can have an infinite number of lines passing through it. However, collinear points will share the same slope and intercept. Therefore,, we use an accumulator for all possible combinations of slope and intercept. Any significant lines in the image will have high values for a given slope and intercept, so all that needs to be done is to examine the image for local maximas in the accumulator. Since a vertical line has an infinite slope, it is common to use polar coordinates (angle and radius) from the origin as the parameters. Figure 5 shows an example of a Hough transform. The image on the left is the original image, the middle image is the Hough Transform space, and the image on the right is the image with the Hough transform lines found.

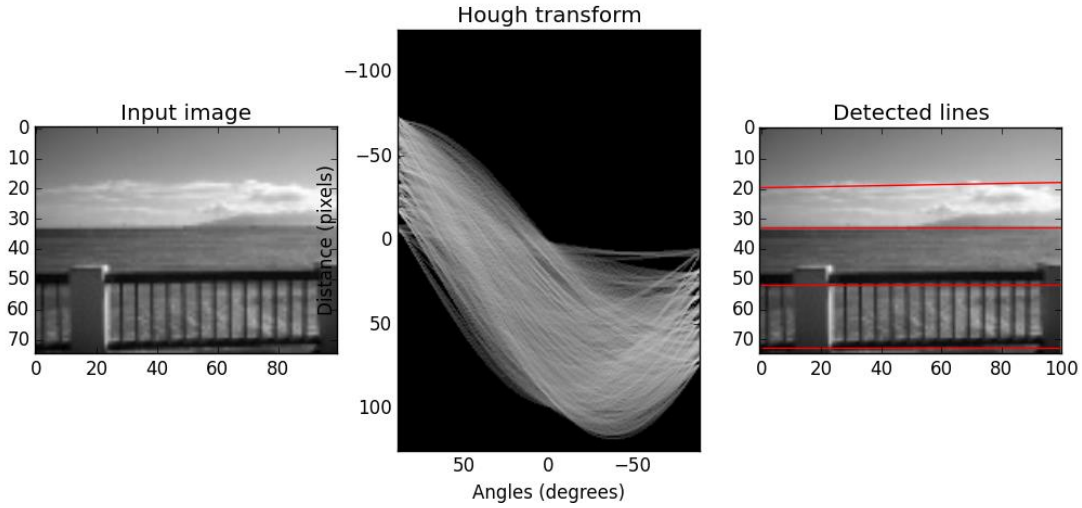


Figure 5—Hough transform example

Sheng et al [5] use the Hough transform to identify likely tubes. After initial pre-processing the image is divided into horizontal strips 100 pixels high. Each of these strips is processed with the Canny edge detection algorithm. The Hough transform is then applied to each strip to identify the left and right edges of possible tubes. The angle of the tubes in the segments that they look for is $\frac{\pi}{24}$. If a horizontal strip fails to find a tube, the strips on both sides of it are used to help continue the tube through the section without an identified tube. Invalid tubes are identified as tube walls that have a highly variable distance between the sides, indicating that they are not parallel. Also, if the tube walls don't match up close enough from section to section, then those indicate that a tube was not actually detected.

Once the tube has been tracked, the type of tube is determined by its position. If the position doesn't clearly indicate which tube is likely, then the width of the tube is used.

Their detection rate for FT and NT tubes were both 82%. They were able to do better at determining ET detection, which was 94%.

Another study also used the Hough transform to identify the lines of a tube. Ramakrishna et al [6] segments the image into bounding boxes for the neck, esophagus and abdomen. Once the next ROI is identified, it is divided into strips. Each strip has parallel lines identified as seeds. Once the seeds are generated the region is grown. By using the seeds as an origin, template matching [8] is used to continue the tube identification points. Doing this with the new points grows the tube from the neck region. The tube stops growing once the template matching score is below a given threshold. Tubes are then excluded if they don't have the correct length and location for that type of tube. ET tubes the location is the neck and esophagus, while the NG's are restricted to Neck, Esophagus and Abdomen. ET Tubes were identified with a TPR of 93% with 0.02 FP per image. NG tubes had 84% TPR and 0.02 FP's per image.

Haar-like [9] templates are used to aid in detection of tubes by Huo, Li, Chen and Wandtke[10]. Haar-like templates are used in a wide variety of image recognition processes. Viola and Jones[9] first introduced the Haar-like features as fast way to find features in an image. Rectangles are used for broad feature areas of an image to classify. These rectangular segmentations are a quick way to identify area. A common example of this is in facial recognition. Facial images are typically much darker around the eyes, by using a template with 2 dark rectangles for the eyes, we can compute the sum of the pixel values in these rectangles. We can slide this template over the image to quickly identify the regions where this template fits best.

The Haar-like image is passed over the ROI and rotates to match the direction of the spine ROI identified. The result of the Haar-like image and a generated edge image are used in tandem. The thresholded edge image is used to identify the sides of the tube, and the Haar-like results are used to identify the center of the tube. Once these points are identified as tubes they are linked and grown from the top and bottom until a threshold is met, signifying the end of the tube. Features from the tubes such as mean distance between walls, the curvature of the tube, and others are created. These features are used in Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA). The results from these algorithms help to identify and classify the tubes. With LDA the sensitivity of the study was 87% with 0.5FPs /image while QDA resulted in 92% sensitivity and 0.5Fps/image. The false positives per image prior to LDA and QDA was 2.9 FP's per image.

Limitations of Current Approaches

In many radiographs, the positioning of the patient is not consistent. The corresponding x-rays can be very diverse. Kao and E-Fong[4] used intensities in a horizontal slice to identify the neck region to begin searching for the tube. However, many images do not include a neck region to begin the search. Even when the neck ROI is in the frame, it isn't usually positioned in a conducive way to help identify the neck. Figure 6 illustrates some of the non-uniform radiographs that we encounter.

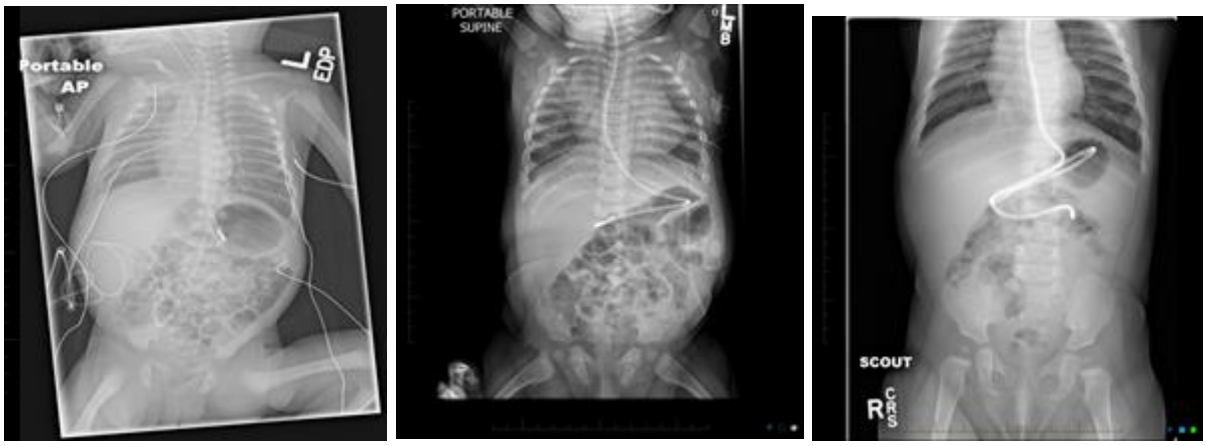


Figure 6— Non-distinct neck region x-ray

Some of the techniques make assumptions about the orientation of the tube being vertical. While this is true of an NG or FT tube in the esophagus it quickly begins looping, and bending when it reaches an open cavity. For the process of identifying a FT, NG, or ET, finding enough portions of it to classify it is probably sufficient. In order to assess its end location you would either need to trace the entire tube, or simply identify the tip. In either of those cases being able to identify horizontal sections of tube is a crucial step. It would also be ideal to be able to identify a general-purpose method so that identifying other non-anatomy structures can be performed in the future.

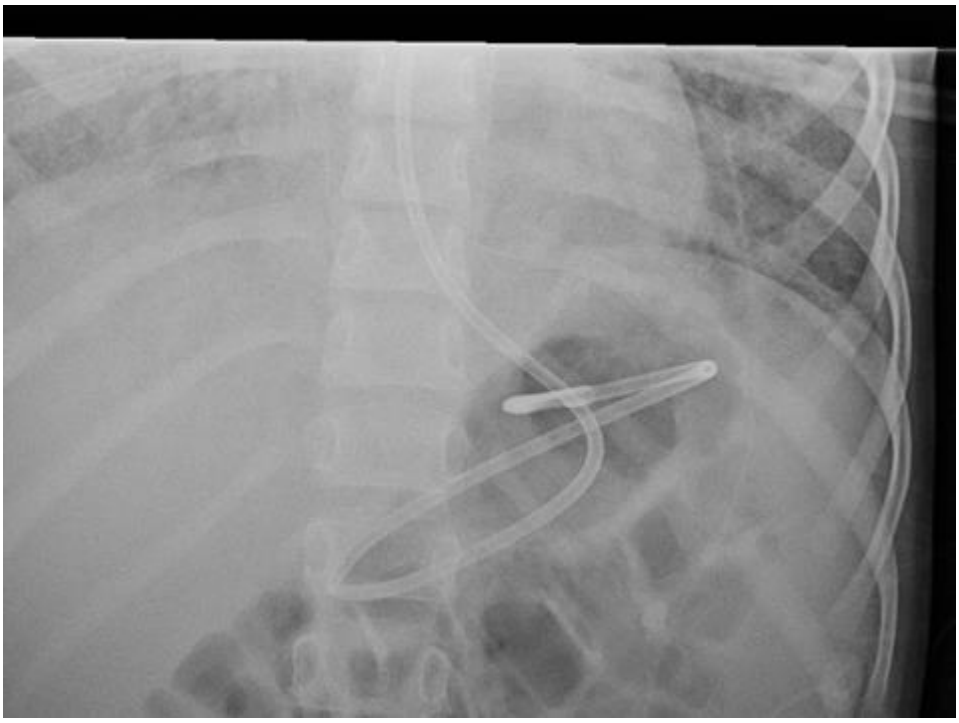


Figure 7—Coiled tube with horizontal sections

CHAPTER 3

METHODOLOGY

Data Collection and Tools

Radiographs used in this research project were provided by Dr. Sherwin Chan of Children's Mercy Hospital. The radiographs were a random sample containing NT Tubes only, NT Tubes with multiple other tubes, and leads. A search was performed in the radiology information systems (RIS) to find all chest or abdomen radiographs that mention "enteric tube" or "feeding tube." Radiographs that had the metal tip present in the radiograph were included.

We used Python [11] for our development process. The rapid prototyping and vast array of modules made it attractive for this project. Among many of Python's modules we used scikit-learn [12] for the machine learning needs. It has a full featured assortment of modern Machine Learning techniques. For complex number calculations using matrices and arrays the numpy [13] module has been an invaluable resource.

Our Process

We start with by doing some initial pre-processing - histogram adjustments and edge preserving filters to smooth noise. The image is then run through edge detection filters to get magnitude and orientation features for subsequent evaluation by a pre-trained Random Forest to find the likely edges of tubes. This process removes a lot of the false edges, but there are still many non-tube edges remaining. We begin associating edges with each other, discarding obvious sections that are not tubes. Obvious portions are detected edges with nothing around them. They are usually lone sections that are easy to discern as noise. Then

based upon finding parallel edges with a gradient mirrored back, we classify consistent edges as tube walls. These tubes can then be grouped and classified as the type of tube they are so that we can assess the location of the tip. Figure *8* shows the overview of our technique.

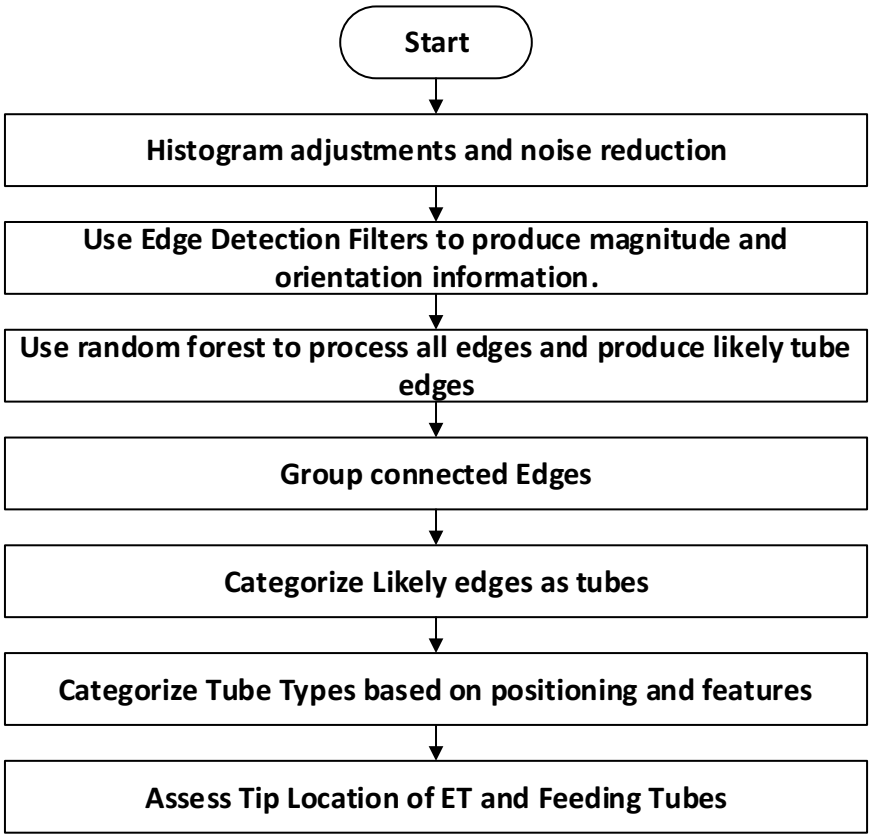
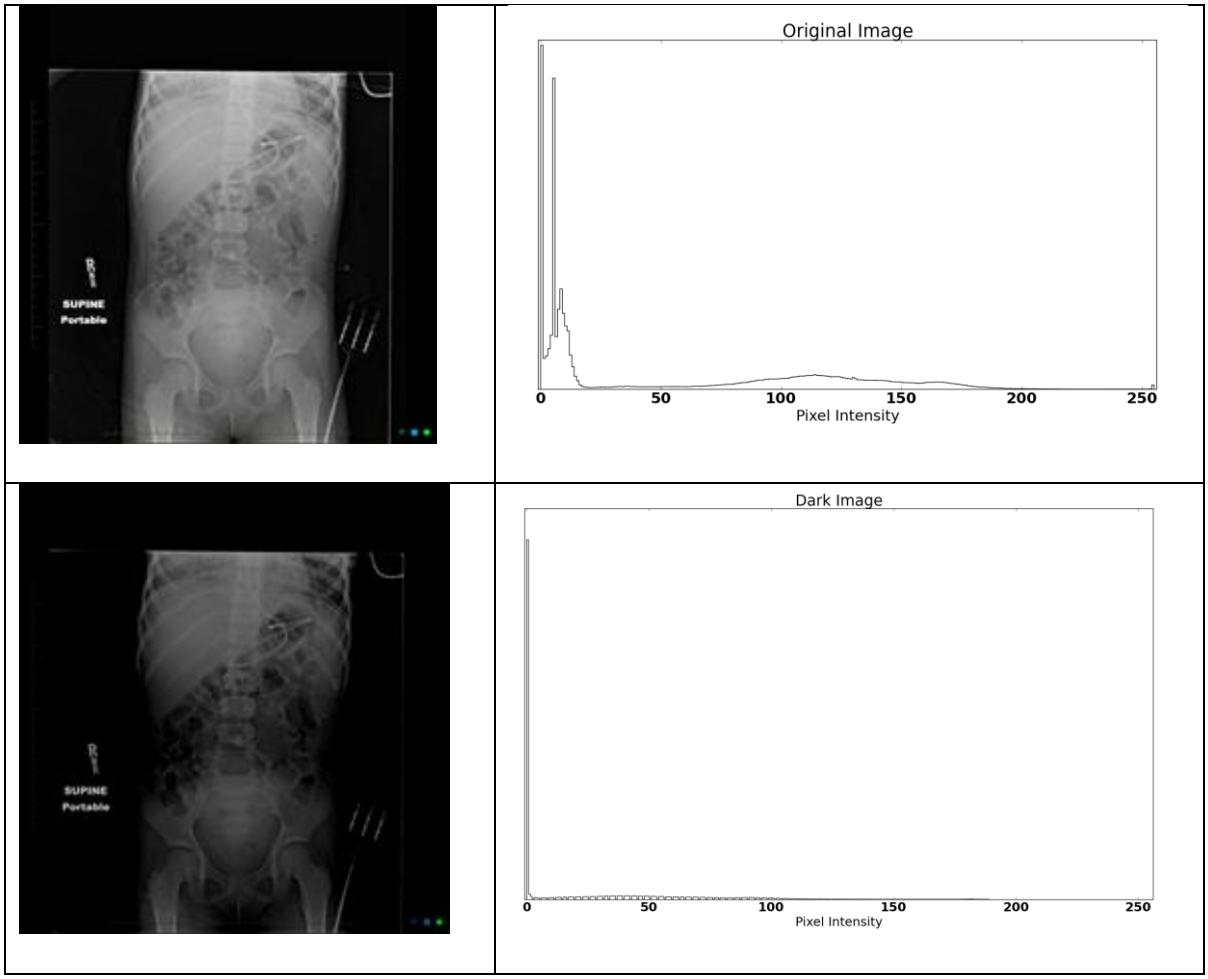


Figure 8—Random Forest Algorithm

Histogram Adjustments and Noise Reduction

Raw unprocessed images are rarely ideal in terms of brightness and distribution of intensity levels. Each pixel has an intensity value, and we can view a distribution of the intensity of the images with a histogram. If an image is too dark, then the histogram will

show most of the pixel intensities as being very low, and vice versa if the image is too bright. We can construct histograms of the image intensity to analyze the spectrum of intensity values. A well-balanced histogram has balanced intensities across the spectrum. Figure 9 shows the original image, a darkened image and a lightened image.



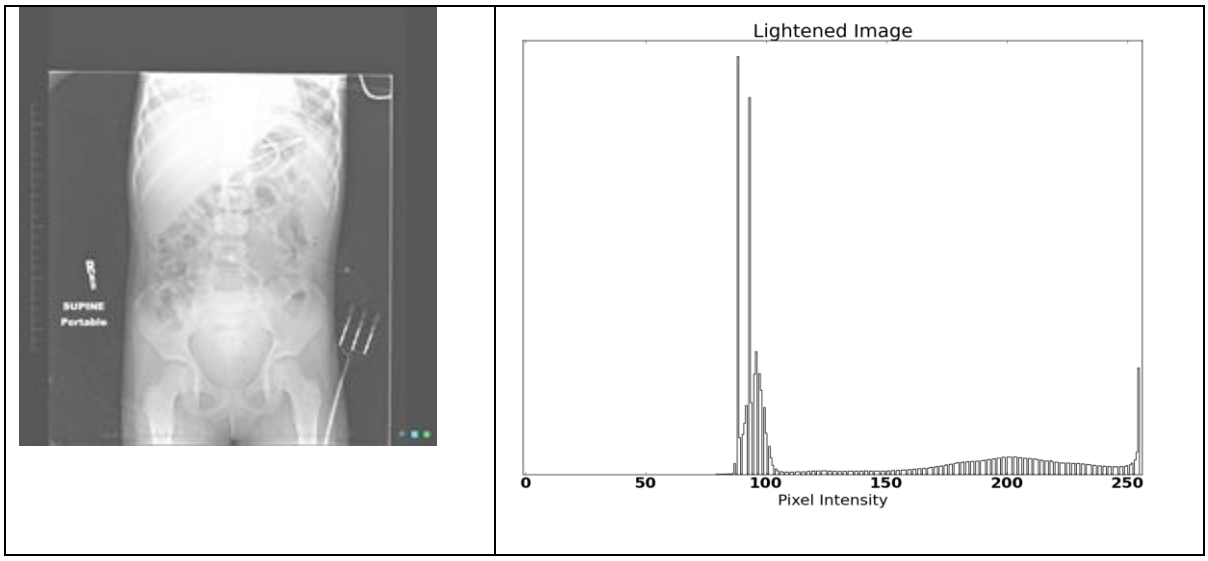
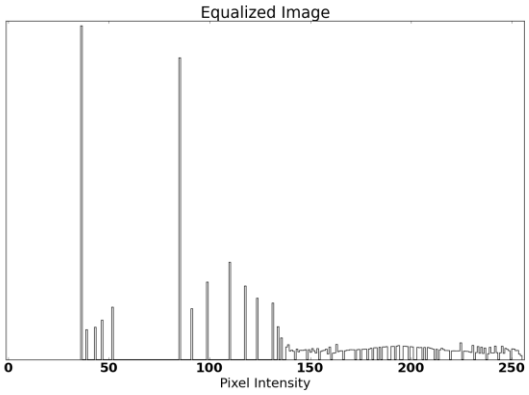


Figure 9—Histograms

Just lightening or darkening an image will just shift the intensities right and left. A more useful technique would be to use histogram equalization to give a uniform distribution of the intensity values in an image. Figure 10 shows an equalized histogram which strives to create a histogram with a more uniform distribution, Contrast Limited Adaptive Histogram Equalization (CLAHE[14]). Normal Equalization of the histogram works best when the image has a uniform sectioning of light and dark areas in the image. However, with images with areas that are light or dark it can further degrade those sections. An Adaptive Histogram Equalization will adjust a pixel by the area around it. This allows regions in an image that are extremely dark, or light to have their histograms adjusted to increase detail, rather than blowing out or reducing the detail in a section. Contrast Limited Adaptive Histograms simply create histograms for the region around a pixel, with the added construct that the distribution cannot increase too quickly. The Equalization shown in Figure 10 shows

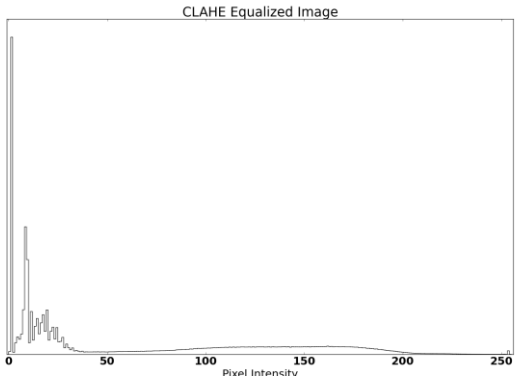
that some sections that were bright to begin with get blown out losing almost all of the detail. This is extremely detrimental when examining detail in a region. In part c) the regions of the image get adjusted, giving us the most detail we can get out of a particular region.

a) Equalized Intensities



b) Equalized Histogram

c) CLAHE



d) CLAHE Histogram

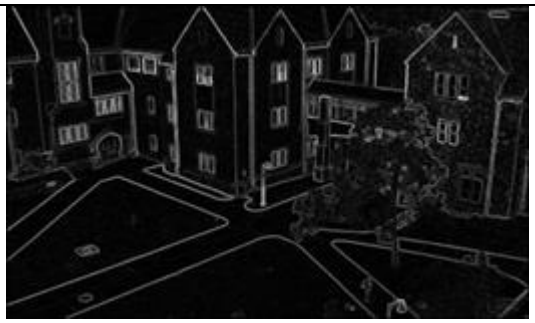
Figure 10—Equalized histograms

Radiographs are very noisy images and should be de-noised. There are many filters that can do just that, but many of those tend to destroy edges. Our algorithm makes use of edge detection, so we want to preserve those edges as much as possible. We tested with Total Variation [15] and Bilateral Filters [16]. With the Bilateral filter the weighted mean of pixels in proximity to the target pixel is used. The bilateral filter can be especially useful for images with multiple color channels since it can preserver the color balance across colors. However, in our use we do not have a need for multiple color channels. Bilateral filter still provides noise reduction without moving the edges around. Alternately, the Total Variation filter starts with the premise that noisy images have a lot of variation. It tries to find the signal that keeps the original signal while eliminating much of the variation.

Edge Detection

Edge detection is used in many image-processing systems. An edge is where there is a rapid intensity change in the image. Figure 11 shows the edges of an image.

a)



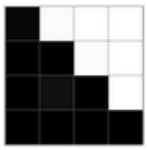
b)

Figure 11—UMKC edge detected image

This edge information contains both magnitude and a direction of the edge.

Figure *12* is an example of an image's edge and the resulting information. The original image is in the top right corner with the corresponding values of the image below. The figure on the right shows the edge information obtained. The arrows show the direction information obtained during the process.

Image



0	255	255	255
0	0	255	255
0	0	0	255
0	0	0	0

Edge

255	0	0	0
0	255	0	0
0	0	255	0
0	0	0	255

Figure 12—Image with resulting edge magnitude and orientation. On the left intensity values for original image. On the right, the edge detection value; magnitude and direction and acquired.

Obviously, if the edges in a radiograph for tubes were that stark the process would be fairly easy. In reality, much of the time the intensities are very similar. Figure *13* (bottom) shows a common area where identifying a tube would be difficult for an edge detector. It's easy to see at the top of the image the tube blends in with the surrounding intensities so much so that it disappears. There are also many other features in a radiograph that produce edges; ribs, vertebrae, etc.

Figure 13—Tube surrounded by high intensity areas.

There are many different edge detection algorithms with different advantages and disadvantages. Figure *14* shows some of the most Edge Detections algorithms.

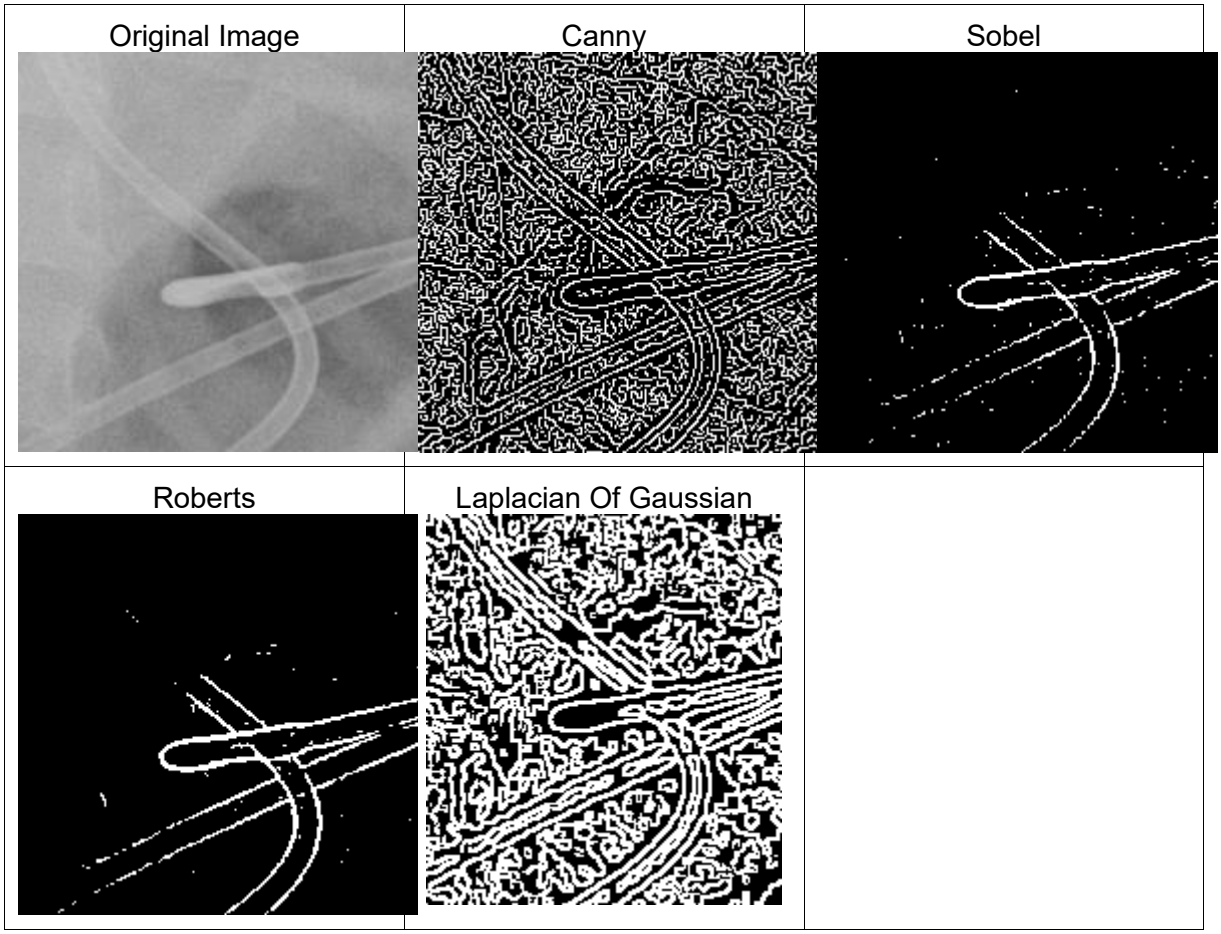


Figure 14—Comparison of edge detectors

Since Canny produces a singular line for each edge, I chose to use it to highlight the edges of tubes for training and analysis. While it would be ideal to use an expert system where multiple radiologists would choose the edges of the tubes we were looking at, this would be time consuming and impractical. In order to provide a ground truth we used the Canny algorithm to initially detect all the edges in an image and then manually highlighted the edge of the tube. While painting the images in this manner was still time consuming, it was much less laborious than identifying the pixels in the original image. Figure 15 shows

an image with the tube marked red and the tip marked blue. Initially 4 images were created in this manner. A set of 16 images was created in this manner.

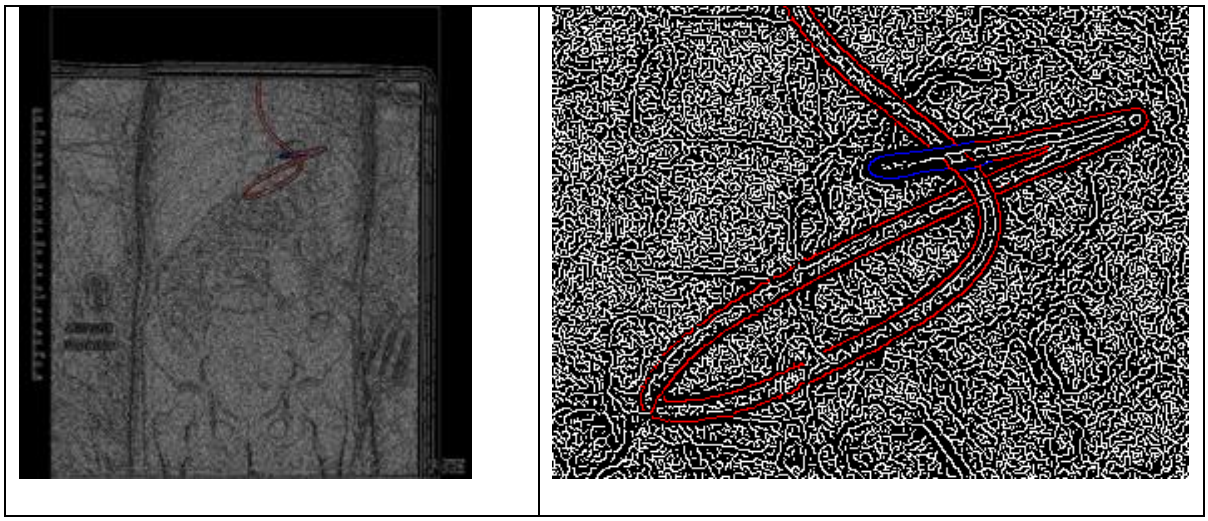


Figure 15—Painted ground truth

Having the tubes identified like this allowed us to analyze the properties of the tubes. Initially we took horizontal and vertical slices around the tube to see what common image intensities and edge magnitudes were commonly present in the tube.

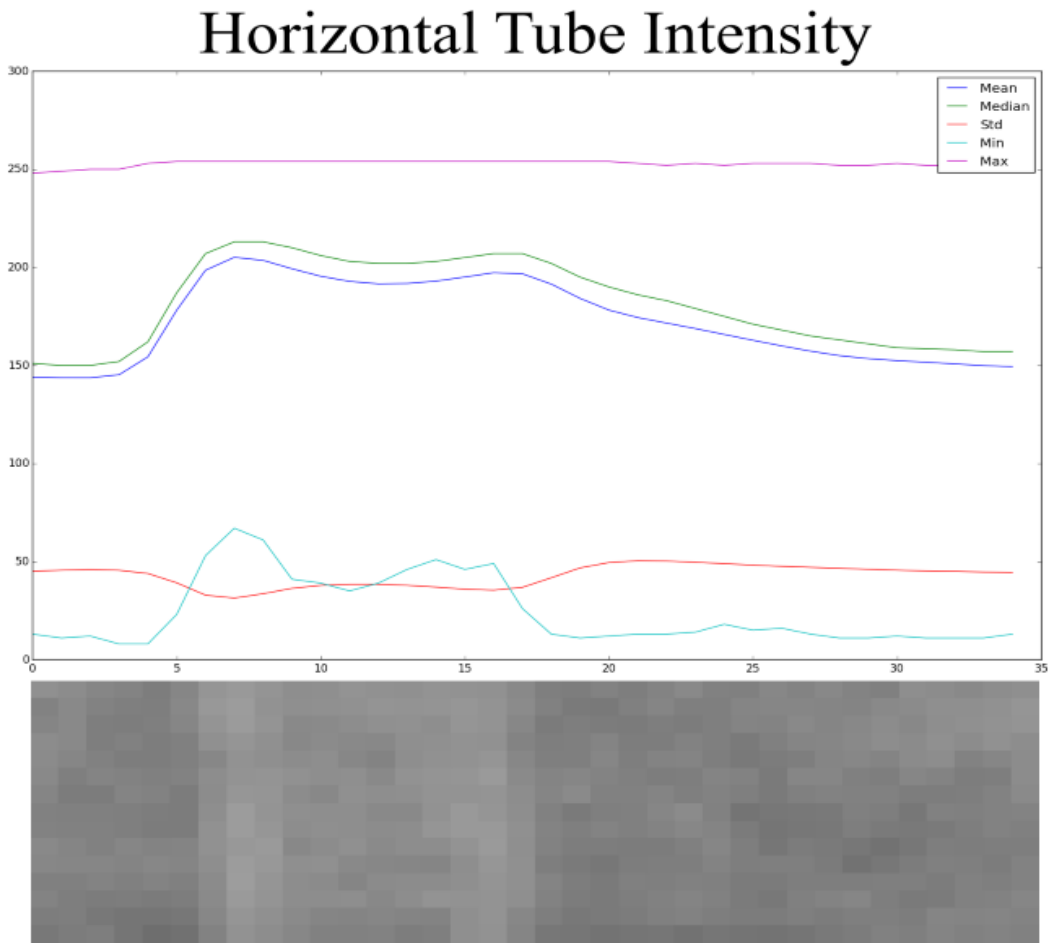


Figure 16—Horizontal tube intensity

Figure 16 shows the cross section profile from the aggregated tube information. The x-axis is along the horizontal slice of image. The relative ground truth pixel identified is pixel 5. We took 5 pixels to the left of the edge and 35 to the right. This is a good first look at some of the difficulties identifying the tubes. While the first thing we tend to notice is a nice Bactrian camel double-hump shape for the mean and median values of the tube, other factors begin to show some of the difficulties of this process. The biggest issue to confront is the standard deviation of the pixels at these positions. With a single standard deviation approaching 50 intensity levels out of 255, we can tell that the values for the tubes and the

tube walls can vary greatly. Even though the standard deviation dips slightly at the walls of the tube, matching the humps of the intensities, there is still a great amount of variation that these values can have.

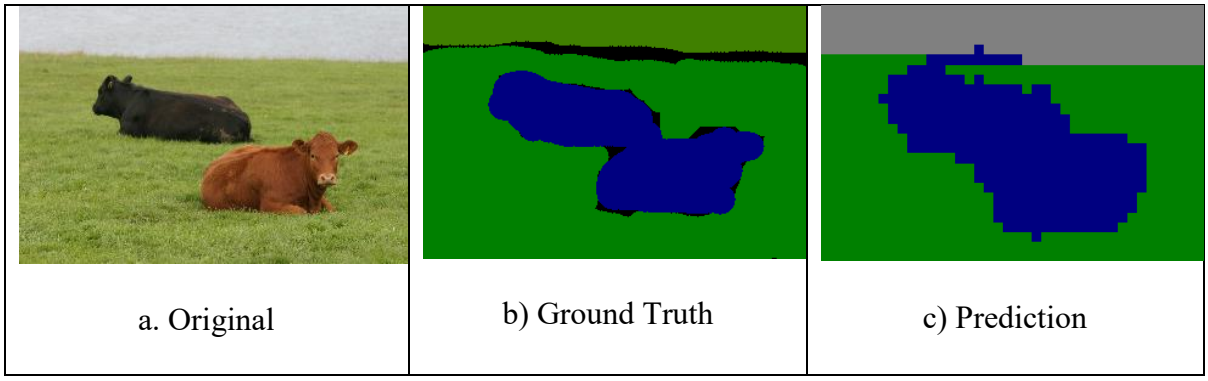
The maximum values, the purple line, illustrates that the intensity outside and inside the tube can exceed the mean value for the tube wall. There are indeed times when the tube is masked by other anatomy that shows as very bright in the x-ray. The minimum value, the cyan line, also shows that not only can the inside and outside of the tube have extremely low intensity, but the tube walls can as well.

Even though the median and mean of intensities shows a clear double hump, which would appear easy to identify in an actual image, the high variance of intensity values throughout the image indicates that identification of the tubes or edges of a tube can be a murky proposition. While our eyes naturally follow and pick up portions of where we think a tube should be, analysis of many of the areas indicate that our brains have filled in the missing gaps for us and the problem is much more difficult than it first appears.

Random Forest Training

There has been much research in using Semantic Texton Forests introduced by Shotton et al [17] for image classification. These techniques first build features by sliding a window over the image and randomly building features based upon the intensity of a single point, or the operation of two random points (add, subtraction and absolute value of subtraction). This textural information from 3 or more color channels are used as features in a Random Forest [18]. The random forest itself is then used in a Support Vector Machine

[19] to train and classify the image. Figure 17 show an example of an image and its final prediction by the Semantic Texton Forest method.



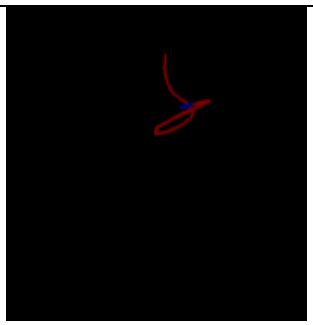
a. Original | b) Ground Truth | c) Prediction

Figure 17—Semantic Texton Forest Classification

Semantic Texton Forest a) is an example of the original image. b) is the hand identified regions of the image. c) is the predicted classified segmented image. The dark blue are identified as cows, the dark green is grass, and the gray is sky.

Radiographs do not contain as much textural data as a standard image would, they only contain one color channel, a gray scale level of intensity. We attempted to use the Semantic Texton Forest process on our x-rays. Using the same procedure we started with marking up several x-rays with ground truth information identifying where the tubes and other features are. Unfortunately, our results on radiographs with their high noise and limited textural information did not produce any desirable results.

a) Original Image



b) Ground Truth

c) Prediction

Figure 18—X-ray Semantic Texton Forest

Even though the Semantic Texton Forest wasn't successful in identifying the tube, we felt we could assist the process by using features of the intensity and the edge magnitudes and orientation. Since radiographs don't contain multiple channels of color information, we looked at some of the features that define the object we are looking for. The tubes would consistently contain an edge image. This edge would continue along the orthogonal of the edge direction. The change would mean that we would "root" the feature window around the tube edge that was being predicted. Instead of having a window passed over the image and features collected in each section, the window would be tilted at the angle of the edge. Figure 19 shows an example of a regularly tiled window that features would be taken from vs the window that is oriented along the edge of the target pixel. The orientation of the window would change for each feature selection.

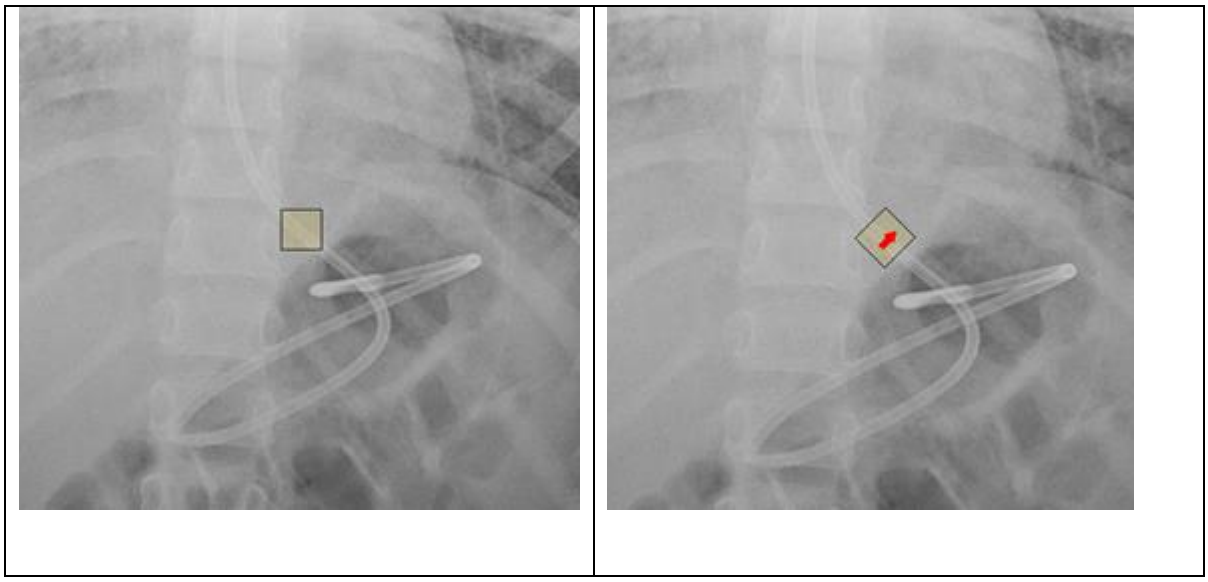


Figure 19—Sliding Windows vs Tilted Window

Since our features are dependent on the edges we tilt the feature collection along this orientation. The reason for this becomes obvious. Pixels orthogonal to the edge of a tube share the same intensity, magnitude and orientation as our source pixel. Figure *20* is an example of the orientation of these features on a tube. The blue arrow shows the direction of the edge and the red pixel is the target pixel we are currently collecting features for. The yellow area is the area we first began collecting features for. It would seem logical that moving orthogonal to the edge direction (along the edge) would have similar intensities and magnitudes as our source pixel. Not only would the intensities be similar, but the orientation of the edge would be in the same direction. Although our tube bends and is not straight, from one pixel to the next there normally wouldn't be a great deal of change. It would also make sense that the pixels just to the outside of the source pixel would be darker in intensity, as the tube is generally, but not always, brighter than the surrounding tissue on the x-ray.

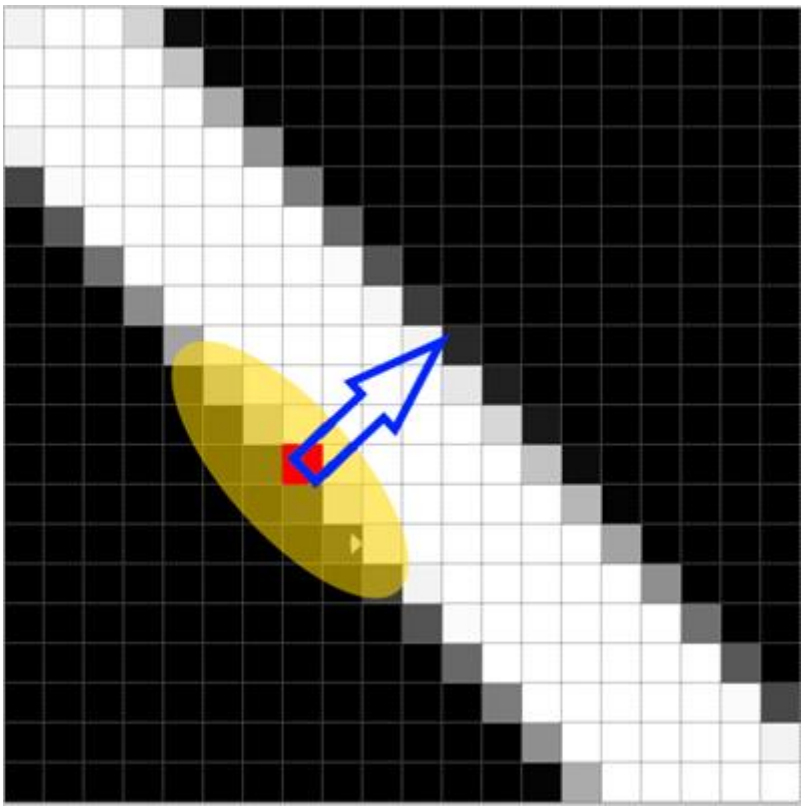


Figure 20—Tilted feature selection based on orientation

Consider now that there are 3 channels of information for each pixel in the image. We have the original intensity of the pixel in the image, the magnitude of the edge after edge processing, and the direction of the edge. Using our source pixel as a sample, we can collect features in relation to the edge we are wishing to classify. Since we are rotating our sampling window along the direction of the edge we are really modelling our tubes as all being consistently vertical as Figure 21 shows. Even when the opposite side of the tube was analyzed in this manner, its edge orientation initially would be directed opposite of what it was in Figure *20*, however from our perspective the tube would be flipped around and the tube would be running vertically with the edge direction pointed to the right.

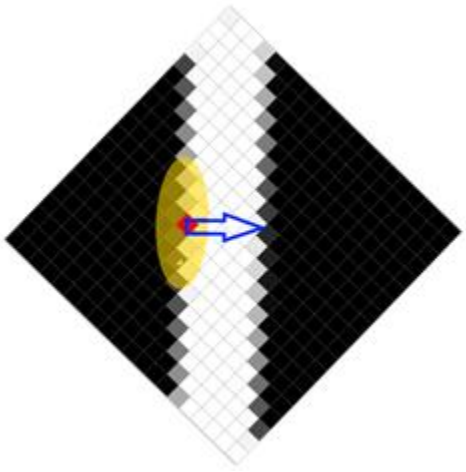


Figure 21—Rotated edge information window

From this new perspective we gather features from around the point we are attempting to classify. We consider our new channels of information in relation to this source located at (0, 0). The features built will be the actual values for the intensity, magnitude and orientation as well as the sum, difference, and absolute value of the difference with the source pixel. We will use the notation I(x,y), M(x, y) and O(x,y) for the Intensity, Magnitude and Orientation of the relative pixel with the source, always remembering that this is in relation to the rotated edge information. Therefore, I(0, 1) would indicate the intensity of the pixel directly below the source pixel. We choose an area around the source pixel to build features off of. Initially we chose a grid 3 wide by 7 high for features. Each of these would contribute their absolute intensity, magnitude and direction as well as the sum, difference and absolute value of the difference with the values of target pixel to the source pixel. For each pixel we get 3 channels with value, sum, difference and absolute difference,

so each pixel will generate 12 features. With 20 target pixels, not including the source we collect 240 features, not include the 3 channels for the source pixel.

We used these features on our initial grouping of 8 ground truth images using Leave-One-Out Cross-Validation per image to get the average TPR. In our images the viewer will quickly notice that the ratio of tube to non-tube portions is not equal. In order to give our Random Forest an even classification we adjusted the dataset to be about 50% non-tube and 50% tube pixels. Even removing so much of the non-tube data we gathered 57863 samples from the 8 images resulting in a 260MB file. Table 2 shows the initial results of using our technique to identify tubes.

Table 2-Random Forest 8 Image TPR's

	TPR	FPR
Non Tube	97.89%	2.12%
Tube	79.20%	20.79%

With a better TPR than the CNN technique discussed previously we saw some potential, and if a little bit of data is good, more data has to be better. We could increase the size of the patch around the source pixel, but the other feature that appeared to be interesting was the wall of the other side of the tube. The center of the tube was also had notable features. While the center was usually fairly noisy, the intensity was generally higher than pixels outside of the tube. This was particularly true in the tip of the tube which is metal and

is easier to identify in the x-rays. The template for feature collection ends up looking like an H. The left bar of the H collecting information around the edge of the source pixel. The right bar of the H is the data about the other wall from our source, and the horizontal bar are features from the middle of the tube. Figure 22 shows the general area of features collected in the yellow region.

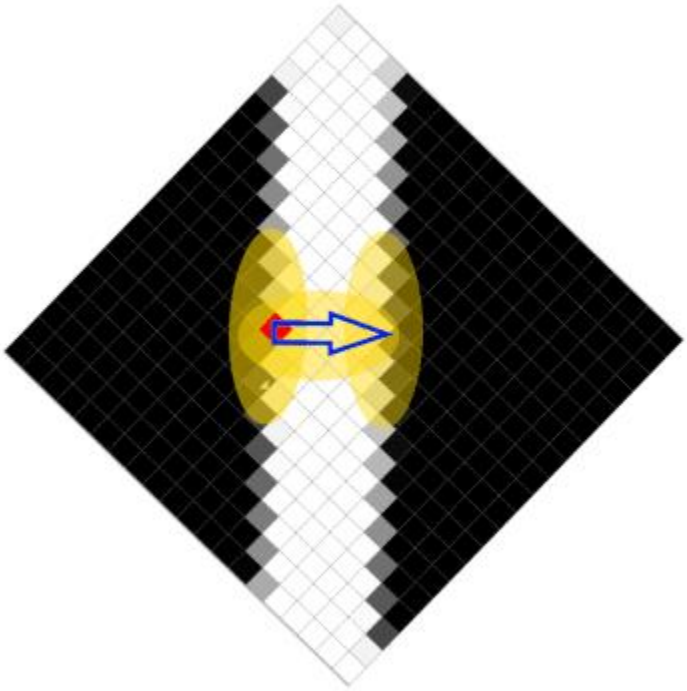


Figure 22—Full feature selection

Finding the other wall of the tube was problematic since FT's can be of different sizes. Using a fixed distance to the other wall wouldn't work consistently. During analysis

of the tubes it seemed consistent that the intensity of the other wall was consistent with our wall. We used that information to attempt to consistently find the other wall.

Once we were collecting all of these features, we increased the number of images used for validation of our process to 16. With 16 images the TPR increased to 85% with a TNR of 98%.

Table 3-Random Forest results with 16 images

	TPR	FPR
Non Tube	97.9%	2.1%
Tube	84.8%	15.2%

After training our random forest solution with all 16 images we used the technique to analyze the full image. Since our training data set had had tubes and non-tubes adjusted to roughly equal, and the actual data would be much greater for non-tube edges the resulting images were noisier than the FPR of the training/testing results would indicate. The Random Forest technique is very good at finding and highlighting strong edges. The tubes are strongly visible in the result, and indicate we've got a good initial start at finding the edges for tube with our learning algorithm. Figure *23* shows the original, canny edges and our predicted tube edges for one such image.

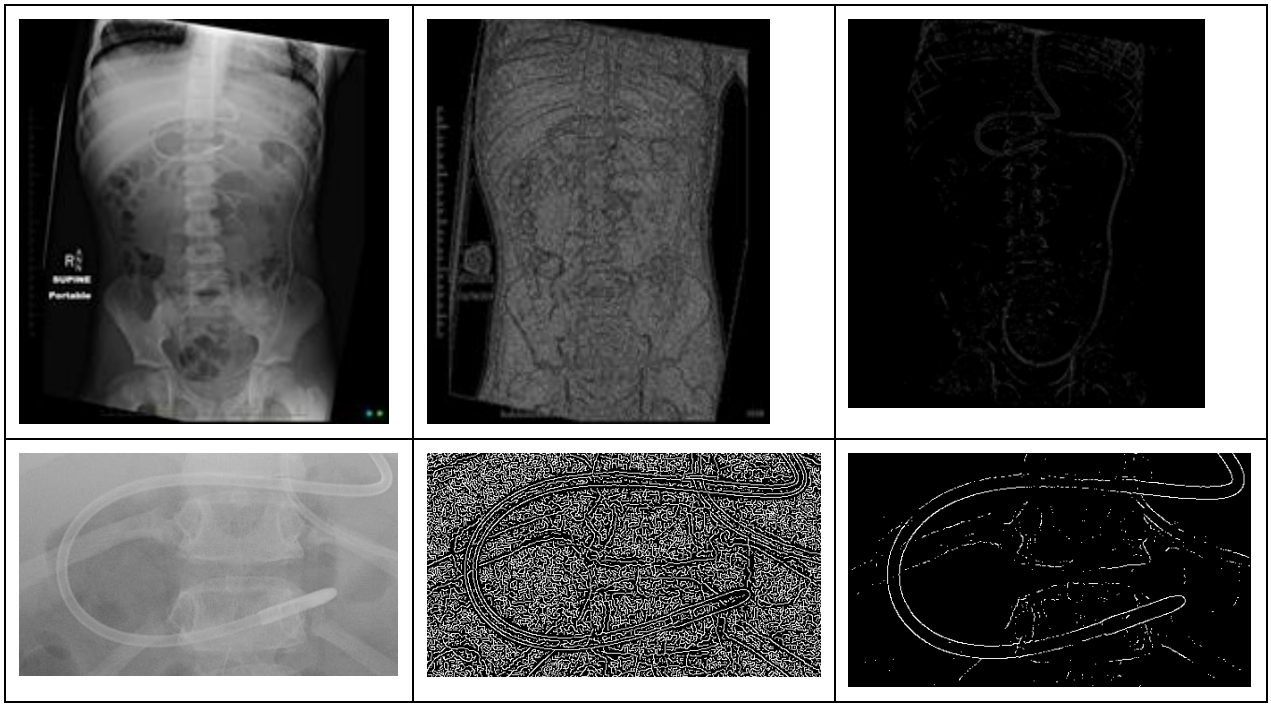


Figure 23—Original Image, Canny Edge Detected Image, Random Forest Image

Grouping Connected Edges

Our resulting image has many of the edge features from the image that are not part of any tube. We've removed a lot of noise, but we have yet to isolate the tubes. The next step is to group and follow connected edges. With each binary predicted edge we examine the orthogonal angles to the edge direction. If the orthogonal area has an edge with a similar orthogonal orientation then the two edge pixels are grouped together. We continue this processing, grouping like pixels together if they are running the same direction and are "partnered" together. Since the next pixel has to be reciprocating we help to eliminate the danger of adding noise to any connected edges. Figure 24 shows the results of grouping the tube pixels together. Much of the noise has been lost from the previous portion of the process.

Figure 24—After grouping image pixels

Tube Categorization

The contours of edges are grown and grouped together. We now want to determine if the edge is part of a tube. A tube will have 2 walls. So there should be 2 edges for a tube. Each side will have similar intensity and edge magnitude. The orientation of the edge will be opposite for each side wall. For each edge we shoot out in the direction of the edge to look for the corresponding other wall of the tube. If we find another edge within our max distance possible for a tube we check to see if the edge direction is opposite of the source edge. We chose a tolerance of 45 degrees. If the source and target angle are within 45 degrees opposite directions then we consider it a possible match. Each edge could possibly have 2 or more edges that could be the other tube wall. Most of the time there will only be one. We do this for each edge in our grouped edges. Once we have all the distances for a section of possible tube we use that information to decide if it a tube. In our analysis the Standard Deviation of the distance to another tube section can be as high as 3 or 4 pixels. We found that other areas of anatomy could also have as similar standard deviation. However, the standard deviation per edge pixel was much lower in a section of tube than in other parts of the image. We found that a tolerance of 0.01 std/edge pixel was a good initial choice. Using this technique we were able to find sections of the tube that long runs and a mirrored other wall. Figure 25 shows the results of this technique on a section of our image.

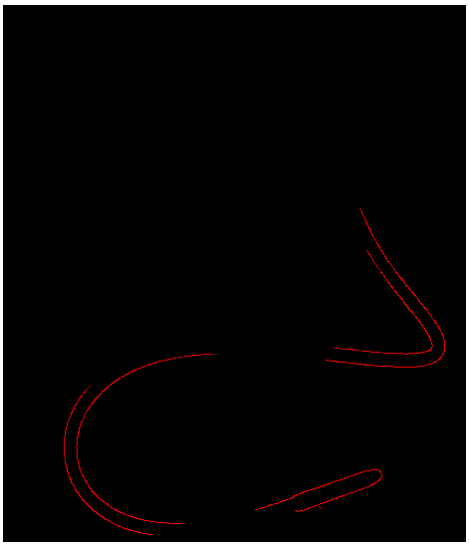


Figure 25—Tube categorization.

CHAPTER 4

EVALUATION

Important Features

The importance of any given feature can be computed with by the "mean decrease impurity" [20]. It is, in essence, the probability of reaching a node of a tree based upon the number of samples that will reach the node. In a random forest this will be the average of all such probabilities. If a feature is used on a node that only 2 out of 1000 samples will get to, its importance will be much lower than a feature that is at the first level of a tree that all samples will go through. Using these values we can see what features are considered to the most important features of our rotated feature window. Table 4 shows the top features for our random forest solution.

Table 4-Feature importance ranks

Rank	Operation	Explanation
1	G(0, -3) + G(0, 0)	Gradient 3 below source pixel added to gradient of source pixel.
2	G(0, 3)	Gradient 3 above source pixel
3	G(-1, 3) + G(0, 0)	Gradient 1 to the left and 3 above source pixel added to the gradient of the source pixel.
4	G(0, 3) + G(0, 0)	Gradient 3 pixels above source pixel added to gradient of source pixel.
5	G(1, 1)	Gradient 1 to the left and 1 above source pixel
6	G(-1, -1) + G(0, 0)	Gradient 1 left 1 below added to gradient of source pixel.

7	G(0, 0)	Gradient at source pixel
8	G(1, 1) + G(0, 0)	Gradient 1 right and 1 above added to gradient of source pixel.
9	G(1, 3) + G(0, 0)	Gradient 1 right and 3 above added to gradient of source pixel
10	G(-1, 1)	Gradient at 1 left and 1 above the source pixel
11	I(4, -2)	Intensity 4 to the right and 2 down. It is an intensity in the tube.
12	GW(0, 2) + G(0, 0)	Gradient at the wall 2 above where the wall was predicted added to the gradient of the source pixel.
13	GW(-1, 2) + G(0, 0)	Gradient 1 left and 2 up from predicted wall added to source gradient.
14	G(-1, -1)	Gradient 1 left and 1 down from source.
15	G(-1, 3)	Gradient 1 left and 3 up from source pixel
16	abs(GW(-1, 0) – G(0, 0))	Absolute value of gradient at source subtracted from gradient 1 left of predicted wall
17	Relative Y position	Relative y position of the source to the center of image
18	G(1, -1) + G(0, 0)	Gradient right and below source added to source of gradient.
19	I(3, 2)	Intensity 3 right and 2 up in tube.
20	I(3 -1)	Intensity 3 right and 1 down in tube.

Looking at the top 20 features we heavily weighted to the gradient channel. It seems to be using the amount of gradient heavily in the decision trees. Specifically, it uses the gradient of the relative pixel added to the gradient of the source pixel the most in that top 20. It's not too great a surprise that the magnitude of the edge would be a good predictive feature of the tube. It also shows a high affinity to the edge a few pixels away. Random noise, or

edge information of a feature that was highly curved or erratic would be more likely to be thrown out.

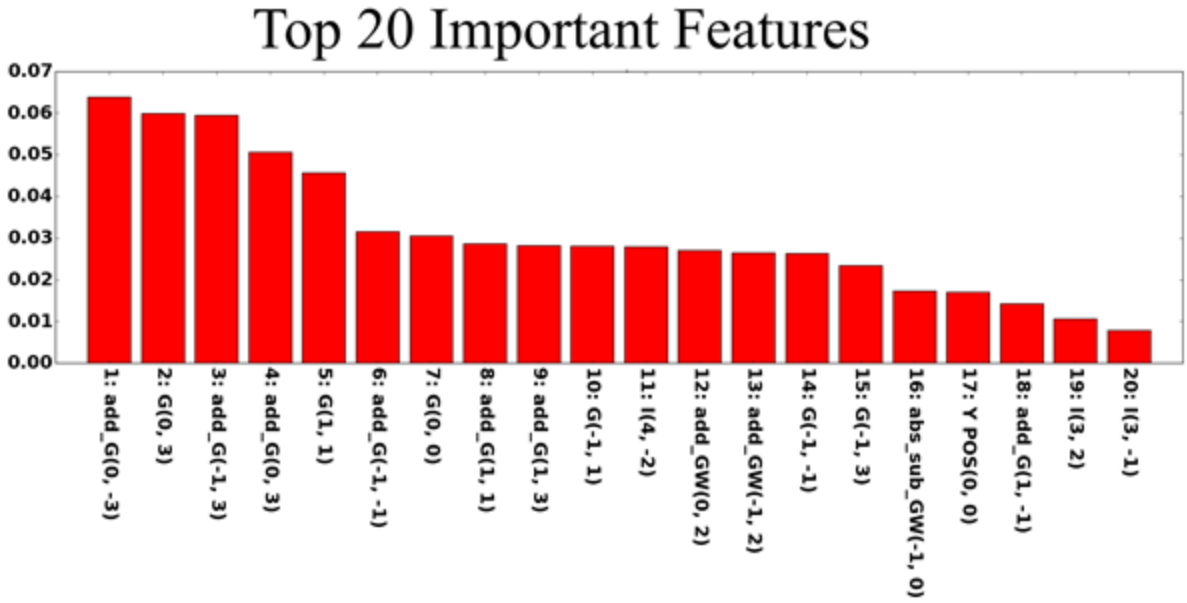


Figure 26-Top 10 features for random forest

Figure 26 shows the relative importance of the first 20 features. It shows the quick degrade in feature importance. Figure 27 shows a heat map of the relative pixel importance around our source pixel at (0,0). You can see that the source pixel itself has a fairly high value as well as their diagonals. It also seems to use the values heavily along the ridge of the edge itself, as far as 3 pixels away. The information around the predicted opposite tube wall is apparently valuable as well, and is used as part of the prediction in the top 20, and shows highly on the heat map.

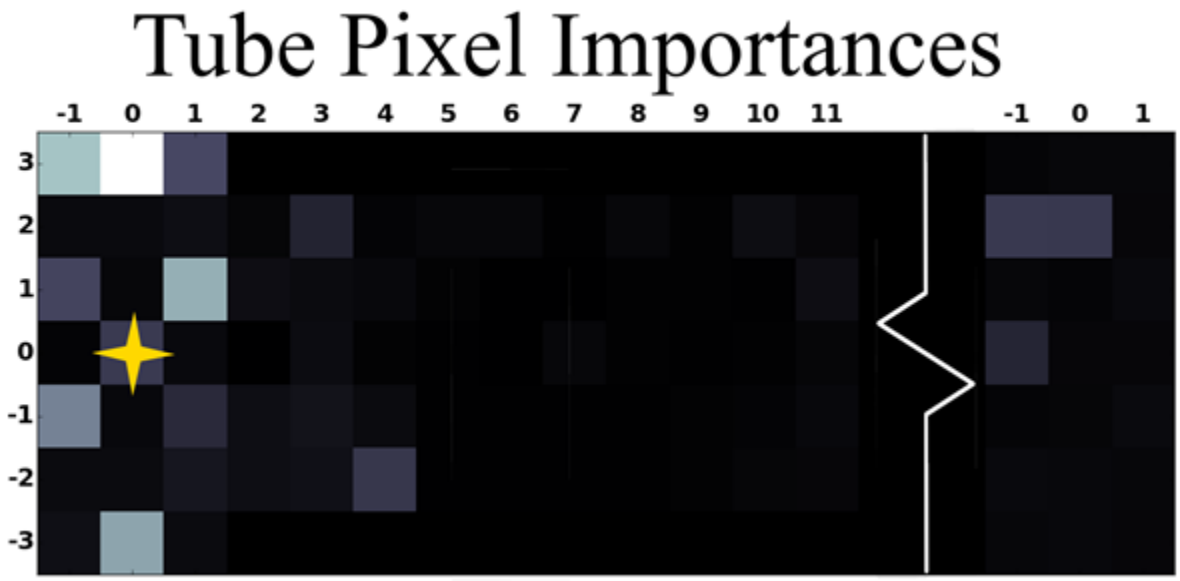


Figure 27--Tube pixel importance

Machine Learning Comparison

Table 5--Method Comparisons

Method	TPR	TPR Std	TNR	TNR Std
Random Forest	79.5%	25.2%	97.8%	1.9%
Naïve Bayes [21]	92.2%	13.6%	88.2%	8.5%
Linear SVC [22]	85.2%	16.8%	93.8%	4.6%
KNN Classifier	84.1%	25.6%	92.3%	4.8%

Table 5 show a comparison of the Random Forest with other learning methods. Initially it looks like the Naïve Bayes might be the better choice, however when put into use on images the results weren't as intuitively good as the Random Forest. This is likely

because the total accuracy for the random forest would be much higher, since in actual images the non-tube pixel weight would far outweigh the number of tube pixels. So the low TNR hurts Naïve Bayes and the other techniques more.

a) Random Forest

b) Naïve Bayes

Figure 28-- Random Forest vs Naive Bayes image results

Pre-Processing Results

Does histogram equalization and edge preserving filters impact our process? We tried our method with and without using CLAHE. We also wanted to evaluate the edge preserving filters to see how they performed.

Both Bilateral and Total Variations have parameters that vary the quality of the output. For bilateral we adjusted sigma range which is the range of the radius that pixels will be averaged across. Figure 29 shows the change in accuracy over a given sigma using the bilateral filter.

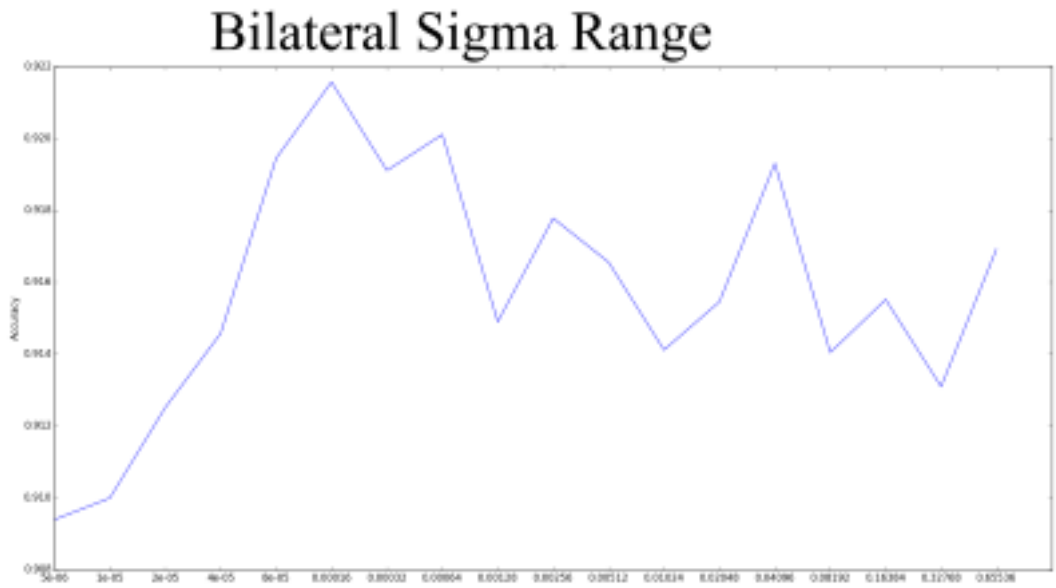


Figure 29--Bilateral accuracy vs sigma range

Next we did the same analysis with the total variation filter. The weight parameter adjusts the amount of de-noising.

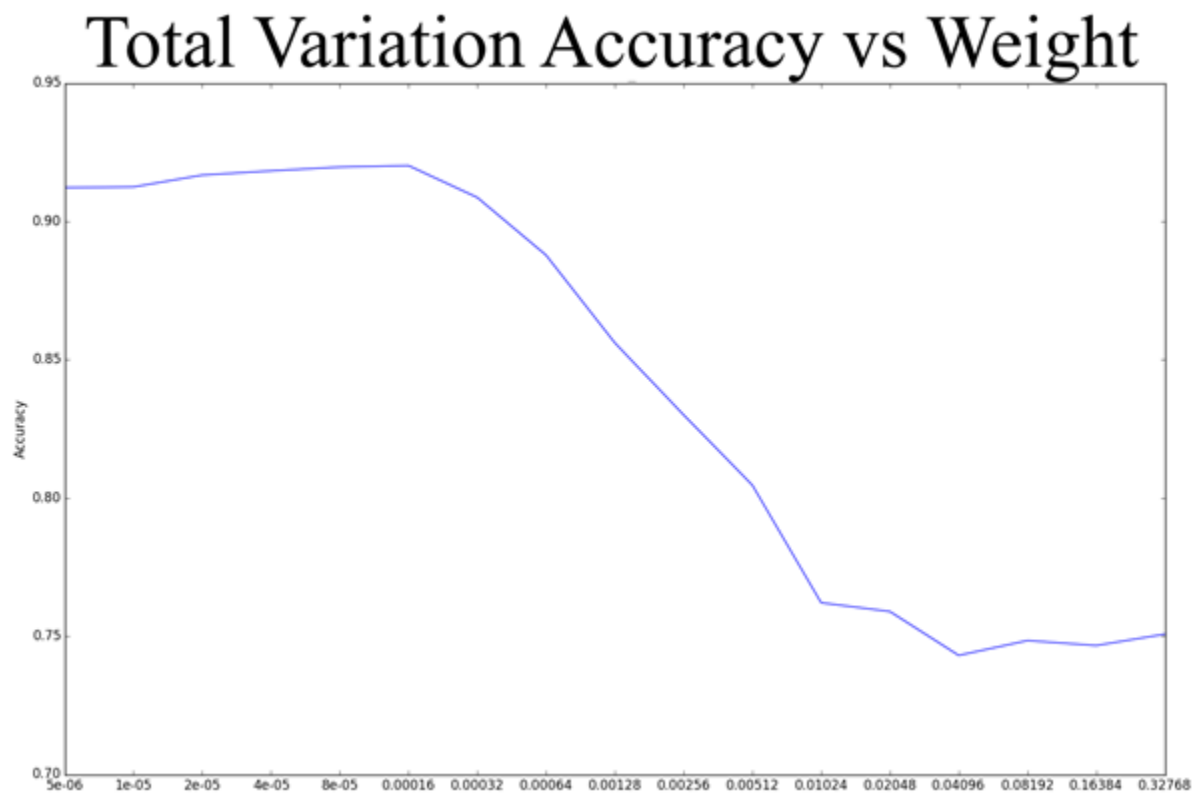


Figure 30--Total Variation accuracy vs weight

We chose the best value for bilateral filter and tested our data against all 3 options. Table 6 shows the results of these 3 options from our random forest technique.

Table 6--Results by pre-processing

Method	TPR	TPR Std	TNR	TNR Std
Original Image	79.9%	24.6%	97.9%	1.9%
CLAHE	79.5%	25.2%	97.8%	1.9%
Bilateral	79.9%	24.1%	98.13%	1.7%

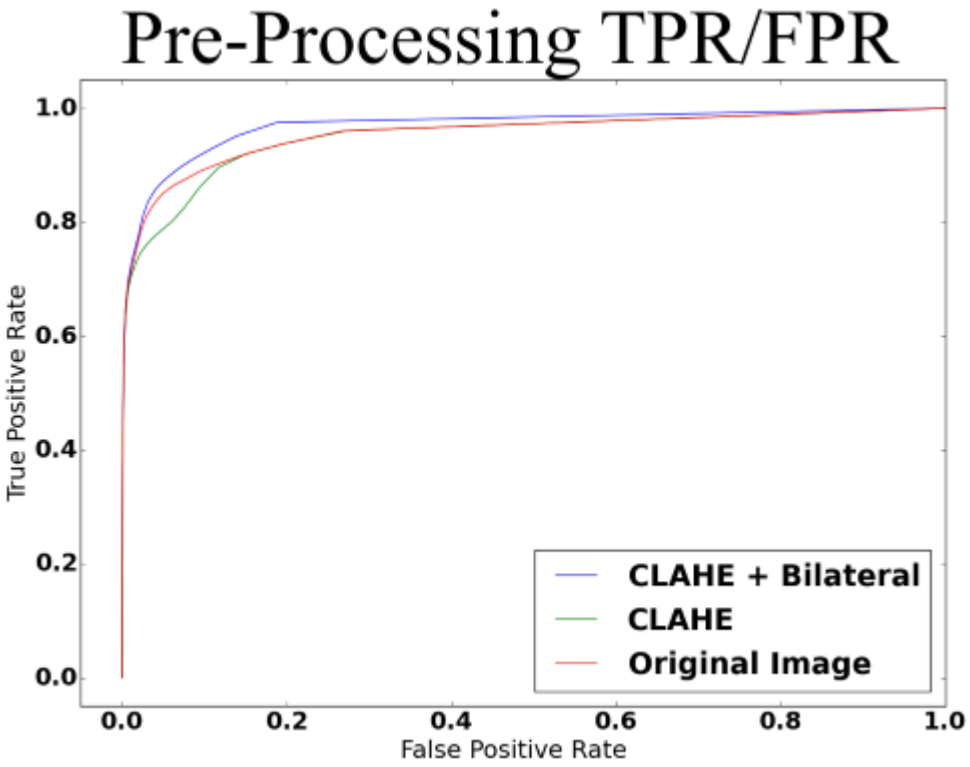


Figure 31--ROC Curve for pre-processing options

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

CONCLUSIONS

Transforming the feature window to orient along the direction of the edge provided a novel way to gather meaningful features. We would expect these edges to have consistent properties that could be picked up by a learning algorithm, and were pleased to see our initial idea perform so well. It was surprising how noisy and inconsistent x-rays were as opposed to normal images. The effect of many layers of soft tissues and tubes overlaid in such a fashion make it very difficult to produce solid reliable results, however, we feel we have found a very solid starting point.

Using Random Forests of trees we were able to create a tube edge detector that could remove much of the noise and artifacts from our Radiology images, leaving us with tube like features. Our initial results with 16 training images gave us an 85% TPR, which was better than other machine learning techniques we've studied. Our initial trials at putting these sections together to build connected sections is very promising and can only get better.

FUTURE WORK

While the work we've done has shown some great promise we know there are quite a few other avenues to explore. While we've gotten a great initial results out of the random forest technique, refinements to method of grouping the results would help even more.

Dr. Sherwin Chan is working approval for a short study on this technique. We are looking forward to implementing this in a trial and expanding the topic even further.

Additional preprocessing techniques could help to pre-identify areas to focus on to start the tube identification process. Currently we are analyzing the entire image. We've looked at template filters that can be used to isolate likely areas of tubes before we even begin. Template matching and other methods to segment and identify the throat or esophagus to limit the regions that we process could be used as well.

Adding the use of Splines to group results and fill in the gaps of tube sections. Very little effort was made to combine sections that were separated by missing portions. We have also attempted to trace the lines by the ridge created by the magnitude, this technique could also be used to fill in the gaps and sections as well.

Identifying the entire tube is useful for other applications besides FT's. Radiographs can have tubes and leads complicating the reading of x-rays when a radiologist is attempting to identify cancer, bone breaks, or other diseases. In this respect, our attempts to identify tubes and non-anatomical features could lead to help in other CAD assessments of x-rays. However, for the purpose of identifying proper FT and NG tube placement, we can largely assume the metal tip of the Feeding tube is the most important aspect. Future efforts could be put into simply identifying the location of this tip accurately. The tube has no standard

shape features that conventional image processing techniques work well on. Its general shape is never consistent and snakes and contorts in ways that make it unlikely to find with any template or shape techniques other than in small sections. However, the tip has a fairly standard shape which could be used to identify it.

We're looked at using generalized Hough transforms with a tube tip template to identify the tip. Generalized Hough can be used for general shapes and describes the image as distances to a binaries edge at given angles. The generalized hough transform can be modified to take into account scale and orientation since the tip can be in any orientation.

Other similar techniques can be used as well, including Active Shape Models. Genneken, Stegmann and Loog[23] used Active Shape Modules identify anatomical sections of an x-ray. This technique might be useful not only for helping us to tell what part of the anatomy the tip of the tube was in, but to also define a general shape for the tip of the tube.

REFERENCE LIST

[1] C. A. Mercan and M. S. Celebi, "An approach for chest tube detection in chest radiographs," *Image Processing, IET,* vol. 8, pp. 122-129, 2014.

[2] N. A. Metheny and M. G. Titler, "Assessing placement of feeding tubes," *AJN The American Journal of Nursing,* vol. 101, pp. 36-45, 2001.

[3] A. M. Creel and M. K. Winkler, "Oral and nasal enteral tube placement errors and complications in a pediatric intensive care unit*," *Pediatric Critical Care Medicine,* vol. 8, pp. 161-164, 2007.

[4] E.-F. Kao, T.-S. Jaw, C.-W. Li, M.-C. Chou, and G.-C. Liu, "Automated detection of endotracheal tubes in paediatric chest radiographs," *Computer methods and programs in biomedicine,* vol. 118, pp. 1-10, 2015.

[5] C. Sheng, L. Li, and W. Pei, "Automatic detection of supporting device positioning in intensive care unit radiography," *The International Journal of Medical Robotics and Computer Assisted Surgery,* vol. 5, pp. 332-340, 2009.

[6] B. Ramakrishna, M. Brown, J. Goldin, C. Cagnon, and D. Enzmann, "An improved automatic computer aided tube detection and labeling system on chest radiographs," in *SPIE Medical Imaging*, 2012, pp. 83150R-83150R-7.

[7] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM,* vol. 15, pp. 11-15, 1972.

[8] J. Lewis, "Fast normalized cross-correlation," *Vision interface,* vol. 10, pp. 120-123, 1995.

[9] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 2001, pp. I-511-I-518 vol. 1.

[10] Z. Huo, S. Li, M. Chen, and J. Wandtke, "Computer-aided interpretation of ICU portable chest images: automated detection of endotracheal tubes," in *Medical Imaging*, 2008, pp. 69152J-69152J-8.

[11] G. v. Rossum. (2001). *Python Reference Manual*. Available: http://www.python.org

[12] G. V. Fabian Pedregosa, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research,* vol. 12, pp. 2825-2830, 2011.

[13] P. F. D. d. Ascher, K. Hinsen, J. Hugunin and T. Oliphant. (2001). *Numerical Python*. Available: http://www.pfdubois.com/numpy

[14] K. Zuiderveld, "Contrast limited adaptive histogram equalization," in *Graphics gems IV*, 1994, pp. 474-485.

[15] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena,* vol. 60, pp. 259-268, 1992.

[16] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Computer Vision, 1998. Sixth International Conference on*, 1998, pp. 839-846.

[17] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *Computer vision and pattern recognition, 2008. CVPR 2008. IEEE Conference on*, 2008, pp. 1-8.
[18] L. Breiman, "Random forests," *Machine learning,* vol. 45, pp. 5-32, 2001.
[19] B. Schölkopf and A. Smola, "Support Vector Machines," *Encyclopedia of Biostatistics,* 1998.
[20] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*: CRC press, 1984.
[21] H. Zhang, "The optimality of naive Bayes," *AA,* vol. 1, p. 3, 2004.
[22] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *The Journal of Machine Learning Research,* vol. 9, pp. 1871-1874, 2008.
[23] B. Van Ginneken, M. B. Stegmann, and M. Loog, "Segmentation of anatomical structures in chest radiographs using supervised methods: a comparative study on a public database," *Medical image analysis,* vol. 10, pp. 19-40, 2006.

VITA

Kendall Lee Bingham was born on October 19, 1969 in Mexico, MO. He graduated from Mexico High School in 1988. He worked as a software engineer in Kansas City. He returned to school and completed his Bachelor of Science in 2004 from University of Missouri – Kansas City.

In 2012 he was accepted in the graduate program at University of Missouri – Kansas City and began working on his Masters as a part time graduate student while continuing to work full time. In 2013 he was given the opportunity to teach CS101 to incoming freshmen. In the summer of 2014 he resigned from his full-time software job to focus on his coursework while continuing to teach. He is a member of Phi Kappa Phi honor society as well as Upsilon Pi Epsilon the International Honor Society for Computing and Information Disciplines. In the spring of 2014 he received the Outstanding Student Award from the School of Computing and Engineering.

Upon completion of his degree, Mr. Bingham plans to continue pursuing an academic teaching position, as well as continuing research in medical image processing.