AN IMAGE-CLASSIFICATION LEVERAGED OBJECT DETECTOR

A Thesis Presented to
the Faculty of the Graduate School
at the University of Missouri

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by

MIAO SUN

Dr. Tony X. Han, Thesis Supervisor

MAY 2014

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

An IMAGE-CLASSIFICATION
LEVERAGED OBJECT DETECTOR

presented by Miao Sun,

a candidate for the degree of Master of Science and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Tony X. Han

Dr. Zhihai He

Dr. Dmitry Korkin

ACKNOWLEDGMENTS

I would like to gratefully and sincerely thank my thesis committee: Dr. Tony X. Han, Dr. Zhihai He and Dr. Dmitry Korkin. Thanks for giving me the opportunity to present my research work. Special thanks to my advisor Dr. Tony X. Han for the continuous support of my Master study and research, for his patience, motivation, enthusiasm and immense knowledge. Besides my advisor, I would also like to thank Dr. Zhihai He and Dr. Dmitry Korkin for their encouragement, insightful comments, and hard questions.

Also, thanks to my fellow labmates: Xiaoyu Wang, Xutao Lv, Guang Chen, Shuai Tang, Yan Li, Hua Zhu, Nacy Yang Liu, Kaley Yang Liu, Haipei Fan, Zhixin Ren, Guobin Chen, Arthur Guang Chen, Ran Pan, Kai Huang, Benjamin Hotrabhavananda, Hussein Mohammed Abdulhussein and Ghadeer Hikmat Nadhim Shaaya. I have learned so much from you. Special thanks to Xiaoyu Wang, Xutao Lv, Guang Chen, Shuai Tang and Yan Li for their enlightening me the first glance of research and useful advices.

And, I would like to thank my roomate Chen Huang and Guang Chen. I am really having a great time with you.

Last but not the least, I would like my family: my parents Pingzhi Sun, Jiulian Zhang and my sister ShanShan Sun for their unconditional support and love.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

ABSTRACT

Currently, the state-of-the-art image classification algorithms outperform the best available object detector by a big margin in terms of average precision. We therefore propose a simple yet principled approach to leverage object detection through image classification on supporting regions specified by a preliminary object detector. Using a simple bag-of-words model based image classification algorithm, we leverage the performance of the deformable model objector by 5% in average precision, leading to a best known results on the standard PASCAL 2007 dataset.

Chapter 1

Introduction

To achieve the goal of automatic image understanding, computers should be able to recognize what objects are in an image and to locate where they are. If we give each class of objects a name(the class label), the task of recognizing what objects are in an image is called image classification. That is for each object class, predicting presence/absence of an example of that class in the image [12, 13]. The task of locating each object of a specific class is called object detection. It is widely accepted that the location of an object can be represented as a bounding box, according to the prestigious and influential PASCAL Visual Object Challenge (VOC) [12, 13]. Usually the object detection is regarded as a more difficult than image classification because object detection requires predicting not only the presence/absence of each object class but also the location of each instance. The results of the most recent PASCAL VOC support this argument [13]: in terms of average precision (AP), the winner of the image classification task [23, 20] achieve an mean AP of 81%; the winner of object detection task [15, 30, 1, 24] achieve an mean AP below 40%.

This big performance gap forces us to speculate: can we use the much better performed image classification to improve the object detection?

Furthermore, the available labeled training image data are quite unbalanced for image classification and object detection. Since most of the state-of-the-art image classification and object detection algorithms are supervised learning based, the quantity and the quality of the labeled data affect the performance heavily. This is another reason that we can achieve acceptable performance for image classification but not for object detection. We can easily tell the labor difference between annotating an image for image classification purpose and annotating an image for object detection purpose: for image classification, annotators only need to check a list of Yes/No check boxes of relevant object categories; for object detection, annotators have to label every instance of each object category with bounding boxes of various scales and aspect ratios. This labor difference is more salient for large scale image dataset: In the standard large scale ImageNet dataset [10], there are $14,197,122$ images of $21,841$ synsets (object categories) labeled for the image classification task. Among these large number of images with categorical labels, bounding box labels are only available for around $3,000$ popular synsets, of which the average number of bounding-box labeled images is merely 150 image per category [10]. We can save huge amount of human labor if we can train or improve an object detector with image data labeled for image classification.

Therefore, building an image classification leveraged object detector is quite desirable from the perspectives of performance as well as practical application cost.

However, there are several factors we need to consider in order to apply the available image classification algorithms to object detection. *First*, simply applying the state-of-the-art image classification algorithms [19, 21, 2, 3, 29, 25, 23, 26, 27] to each scanning window is in feasible due to the speed issue. Most of the aforementioned image classification algorithms [19, 21, 29, 25, 23, 26] uses one or several key classic components including BOW model of large size codebook, spatial pyramid matching (SPM), and feature pooling, which make the feature extraction very slow compared with the modern sliding window based detectors [4, 14, 30, 11] Usually a sliding win-

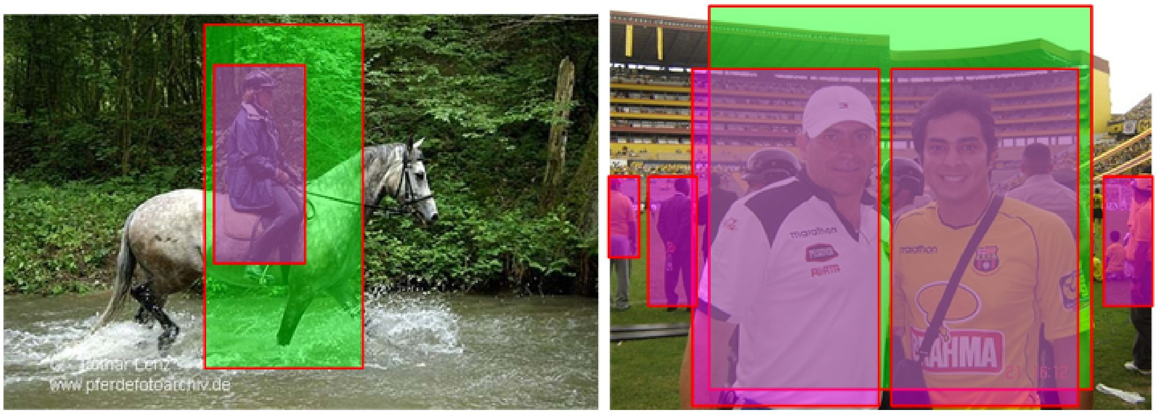

Figure 1.1: Supporting regions in the image-classification leveraged object detector. The red rectangular boxes are detection results from a preliminary object detector. Green regions are created by subtraction of two boxes. Both the magenta regions and green regions are called supporting regions, which will be the input for classification algorithm.

dow based object detector will scan hundreds of thousands sliding windows in order to detect every instances in the image. If we directly apply image classification algorithms to each scanning window, object detection in an image is equivalent to classify hundreds of thousands images. *Second*, if we apply image classification to selected candidate regions as what is done in [24], , the selective search on over segmented superpixels, the image classification algorithm should be robust to region cropping and clipping and should remain discriminative.

We therefore propose a simple yet principled approach to leverage object detection through image classification on supporting regions specified by a preliminary object detector. Using a simple bag-of-words model based image classification algorithm, we leverage the performance of the deformable model objector by 5% in average precision, leading to a best known results on the standard PASCAL 2007 dataset. An illustration of our idea is shown in Figure 1.1.

Chapter 2

Classification Leveraged Object Detector (CLOD)

This chapter describes the algorithm for our classification leveraged object detector. First, we talk about how to generated the detection bounding boxes using deformable part models and enhanced HOG-LBP features in Section 2.1. Then, the detailed classification algorithm is illustrated in Section 2.2. Section 2.3 talks about the context information used to boost the detection performance. Finally, how to combine the classification and detection to form CLOD is illustrated in Section 2.4.

2.1 Image Detection

Generic object detection is a fundamental challenge in computer vision research which aims at localizing all the objects of interest in an image. Recent approaches have been devoting major efforts to handling object deformations or speeding up an object detector. One of the most influential methods in generic object detection is the deformable part models (DPM) [15] and its extensions [16, 18, 30]

This section talks about one extension of deformable part models by integrating the enhanced HOG features and LBP features.

2.1.1 Enhanced HOG Features and LBP Featuers

As a dense version of the dominating SIFT [5] feature, HOG [4] has shown great success in object detection and recognition [4, 16, 6]. Histograms of Oriented Gradients(HOG) has been widely accepted as one of the best features to capture the edge or local shape information, while the Local Binary Pattern (LBP) operator [7] is an exceptional texture descriptors. It has been widely used in various applications and has achieved very good results in face recognition [9]. The LBP is highly discriminative and its key advantages, namely its invariance to monotonic gray level changes and computational efficiency, make it suitable for demanding image analysis tasks such as human detection. HOG performs poorly when the background is cluttered with noisy edges. LBP is complementary in this aspect. It can filter out noises using the concept of uniform pattern [7]. We believe that the appearance of a human can be better captured if we combine both the edge/local shape information and the texture information.

In this section, we start by reviewing HOG. Then, the enhanced HOG features and LBP features are described in details.

HOG Features

Let $\theta(x, y)$ and $r(x, y)$ be the orientation and magnitude of the intensity gradient at pixel (x, y) in an image. The gradient orientation at each pixel is discretized into one of p values using contrast sensitive B_1 or insensitive B_2, definition,

$$B_1(x, y) = round(\frac{p\theta(x, y)}{2\pi}) \mod p \tag{2.1}$$

$$B_2(x, y) = round(\frac{p\theta(x, y)}{5 \quad \pi}) \mod p \tag{2.2}$$

Then define a pixel-level feature map $F(x, y)$. Let $b \in \{0, \ldots, p-1\}$ range over orientation bins. The feature vector at (x, y) is

$$F(x,y)_b = \begin{cases} r(x,y) & \text{if } b = B(x,y) \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

Given F as a pixel-level feature map for a $w \times h$ image, let $k > 0$ be a parameter specifying the side length of a square image region. We could define a dense grid of rectangular "cells" and aggregate pixel-level features to obtain a ceil-based feature map C, with feature vectors $C(i, j)$ for $0 \leq i \leq \lfloor (w-1)/k \rfloor$ and $0 \leq j \leq \lfloor (h-1)/k \rfloor$. This aggregation provides some invariance for small deformations and reduces the size of a feature map.

Gradients are invariant to changes in bias and invariance to gain can be achieved by normalization. According to [4], four different normalization factors that feature vector $C(i, j)$. The factors are defined as $N_{\delta,gamma}$ with $\delta, \gamma \in \{-1, 1\}$

Let $T_\alpha(v)$ denote the element-wise trunction of a vector v by α. The HOG feature map is defined as

$$H(i,j) = \begin{pmatrix} T_\alpha(C(i,j)/N_{-1,-1}(i,j)) \\ T_\alpha(C(i,j)/N_{+1,-1}(i,j)) \\ T_\alpha(C(i,j)/N_{-+,+1}(i,j)) \\ T_\alpha(C(i,j)/N_{-1,+1}(i,j)) \end{pmatrix} \tag{2.4}$$

In this thesis, HOG features use $p = 9$ contrast insensitive gradient orientations, $k = 8$, and truncation $\alpha = 0.2$, which would lead to a 36-dimensional feature.

Enhanced HOG Features

The HOG features in the above section only use contrast insensitive gradient orientations, while in fact the contrast sensitive gradient would also contribute to capture more information. Let $p = 18$ in equation (2.2), and the total number of HOG dimension would be $4 \times (9 + 18) = 108$.

In practice, we use an analytic projection of these 108-dimensional vectors, defined by 27 sums over different normalizations, one for each orientation channel of F, and 4 sums over the 9 contrast insensitive orientations, one for each normalization factor. Therefore, the enhanced HOG feature would be 31 dimenional feature vector.

LBP Features

The local binary pattern(LBP) operator was first introduced as a complementary measure for local image contrast [7]. The original LBP operator forms labels for the image pixels by thresholding the 3×3 neighborhood of each pixel with the center value and considering the result as a binary number. The histogram of these $2^8 = 256$ different labels can be used as a descriptor.

Then, the LBP operator was extended to use neighborhoods of different sizes and a definition of uniform patterns, which is used to reduce the length of the feature vector and implement a simple rotation-invariant descriptor [7]. Those patterns that hold less than u $0-1$ transitions are called uniform patterns. We use the notation $LBP^{u}_{n,r}$ to denote LBP feature that takes n sample points with radius r, and the number of $0-1$ transitions is no more than u. In the computation of the LBP labels, each uniform pattern has a separate label and all the non-uniform patterns are labeled with a single label. LBP features are the histogram of LBP labels.

In this thesis, we use $LBP^{2}_{8,1}$: there are total of $2^8 = 256$ patterns , 58 of which are uniform patterns. Therefore, the LBP feature would be a 59 dimensional feature vector

PCA on HOG-LBP Features

The concatenated HOG-LBP is a 31+59=90 dimensional feature vector, which leads to that the detection speed is almost 2 times slower than the detection speed using only HOG features. However, speed issue is key problem in a exhaustive mult-scale sliding window search detection algorithm .

In this thesis, Principal Components Analysis(PCA) is used to reduce the HOG-LBP features to 40 dimensions without much loss of information. Another advantage of PCA is that it helps to highlight the similarities and differences.

2.1.2 Deformable Part Models

Sliding Window

For the sliding window detection approach, each image is densely scanned from the top left to the bottom right with rectangular sliding windows. For each sliding window, certain features such as edges, image patches, and wavelet coefficients are extracted and fed to a classifier, which is trained offline using labeled training data. The classifier will classify the sliding windows, which bound people, as positive samples, and the others as negative samples.

In this thesis, we apply the sliding window to different scales of original images, the feature is PCA reduced HOG-LPB feature(40 dimensional vector) and the classifier is called deformable part models [14]

Deformable Part Models (DPM)

A DPM object detector [14] consists of a coarse root filter and several higher resolution part filters. Here ”filter” means a set of weights. The score of a DPM at a particular position and scale within an image is the score of the root filter at the given location plus the sum of its part filters minus a deformation cost measuring the deviation of the part from its ideal location relative to the root filter:

$$score(p_0, ..., p_n) = \sum_{i=0}^{n} F_i' * \phi(H, p_i) - \sum_{i=1}^{n} d_i * \phi_d(dx_i, dy_i) + b \tag{2.5}$$

where

$$(dx_i, dy_i) = (x_i, y_i) - (2(x_0, y_0) + v_i) \tag{2.6}$$

gives the displacement of the i-th part relative to its anchor position and

$$\phi_d(dx_i, dy_i) = (dx, dy, dx^2, dy^2) \tag{2.7}$$

are deformation features. In Eq.(2.5), subindex 0 means the root while $i = 1...n$ means the part; $p_i = (x_i, y_i, l_i)$ specifies the level and position of the i-th filter ; F_i' is the filter weights reshaped in a row-major order; $\phi(H, p_i)$ is the feature map at p_i; d_i is a 4 dimensional weight vector of displacements; b is a real valued bias term.

The Eq.(2.5) can be expressed as:

$$score(p_0, ..., p_n) = \beta * \phi(H, z) \tag{2.8}$$

where β is a vector of model parameters

$$\beta = (F_0', ..., F_n', d_1, ..., d_n, b) \tag{2.9}$$

and Φ is a vector:

$$\Phi(H, z) = (\phi(H, p_0), ..., \phi(H, p_n) - \phi_d(dx_1, dy_1), ..., -\phi_d(dx_n, dy_n), 1) \tag{2.10}$$

The Eq.(2.8) indicates that the DPM parameters can be learned with a latent

SVM framework:

$$f_{\beta}(x) = \max_{z \in Z(x)} \beta * \Phi(x, z) \tag{2.11}$$

where z are latent values, the set $Z(x)$ defines the possible latent values for an example x

2.2 Image Classification

One of the state of the art image classication systems consist of two major parts: bag-of-features (BoF) [31] and spatial pyramid matching (SPM) [21]. SPM approach based on bag-of-features (BoF) requires nonlinear classifiers to achieve good image classification performance. Wang *et.al.* [25] present a simple but effective coding scheme called Locality-constrained Linear Coding (LLC) in place of Vector quantization (VQ) coding in traditional SPM so that LLC with linear classifier performs remarkably better than the traditional nonlinear SPM.

2.2.1 Bag of Features (BoF)

Bag-of-Features approach is motivated by analogy to learning methods using the Bag-of-Words representation for text categorization [32]. This idea of clustering descriptors of image patches has demonstrated impressive levels of performance [33, 34]. Usually a BoF framework contains following steps:

1. Extracting descriptors of image patches. Most used descriptors for image classification is SIFT [5].

2. Constructing a vocabulary(or dictionary) using clustering methods, where k-means is widely used due to its efficiency and scalability.

3. Assigning patch descriptors to a set of clusters (a vocabulary) with a vector quantization algorithm.

2.2.2 Spatial Pyramid Matching (SPM)

The BoF method represents an image as a histogram of its local features. It is especially robust against spatial translations of features, and demonstrates decent performance in whole-image categorization tasks. However, the BoF method disregards the information about the spatial layout of features, hence it is incapable of capturing shapes or locating an object.

By overcoming this problem, one particular extension of the BoF model, called spatial pyramid matching (SPM) [21], has made a remarkable success on a range of image classification benchmarks like Caltech101 [35] and PASCAL07 [12], and was the major compo nent of the state-of-the-art systems

SPM partitions an image into $2l \times 2l$ segments in different scales $l = 0, 1, 2$, computes the BoF histogram within each of the 21 segments, and finally concatenates all the histograms to form a vector representation of the image. In the case where only the scale $l = 0$ is used, SPM reduces to BoF.

2.2.3 Locality-constrained Linear Coding (LLC)

The traditional SPM approach based on bag-of-features (BoF) requires nonlinear classifiers to achieve good image classification performance. Based on [25], LLC and with linear classifier performs remarkably better than the traditional nonlinear SPM.

LLC uses the following criteria:

$$\min_{C} \sum_{i=1}^{N} \|x_i - Bc_i\|^2 + \|d_i \odot c_i\|^2 \tag{2.12}$$

Where x_i is the local descriptors extracted from an image; B is a pretrained dictionary(or codebook); c_i is the code for x_i and L1 norm of c_i is 1; $\odot$ denotes the element-wise multiplication; d_i is the locality adaptor, which can be defined as

$$d_i = exp(\frac{dist(x_i, B)}{\sigma}) \tag{2.13}$$

where $dist(x_i, B) = [dist(x_1, b_1), ..., dist(x_i, b_M)]^T$, and $dist(x_i, b_j)$ is the Euclidean distance between x_i and b_i. σ is used for adjusting the weight decay speed for the locality adaptor LLC incorporates locality constraint instead of the sparsity constraint due to that.

Compared with traditional sparse coding, Eq.(2.12) incorporates locality constraint instead of the sparsity constraint, which leads to better reconstrution, local smooth sparsity and analytical solution.

2.3 Context Information

Inspired by [15], we implemented a simple but powerful procedure to boost the performance: Let $(D_1, ..., D_k)$ be a set of detections obtained using k different models (for different object categories) in an image I. Each detection $D_i = (B, s)$ is defined by a bounding box $B = (x_1, y_1, x_2, y_2)$ and a score s. We define the context of I in terms of a k-dimensional vector $f_1(I) = (\alpha(s_1), ..., \alpha(s_k))$ where s_i is the score of the highest scoring detection in D_i , and $\alpha(x) = 1/(1 + exp(2x))$ is a logistic function for renormalizing the scores.

In our framework , we would have a classification score related to each detection box. Then we apply the same procedure as above so that we could get $f_2(I)$. So our context information for each box is a 46 dimension length feature: $[\alpha(d_i), \alpha(c_i), x_1, y_1, x_2, y_2, f_1(I), f_2($

2.4 Classification Leveraged Object Detector(CLOD)

In this section, we first define the supporting regions for classification as the Figure1.1 described. Then, we give the workflow as to how this classification procdure worked with the detection procedure.

2.4.1 Supporting Region for Classification

In this section, we will format our framework. Let $\overline{D}_i$ be the detection candidate boxes, i is from 1 to N for a single image. We sort the boxes so that the detection score of $\overline{D}_i$ is larger than $\overline{D}_j$, if $i < j$. Let B be the background region,

$$B = \bigcap_{i=1}^{N} \overline{D}_i \tag{2.14}$$

If there is no missing detections, the classification score of B would satisfy

$$f_c(B) < 0 \tag{2.15}$$

Then i is from 1 to N , that is, the boxes we want to classify are from high detection score to low detection score.

$$\begin{align} S_k &= B \cup \big(\bigcup_{i>k}(D_k \cap \overline{D}_i)\big) \tag{2.16} \\ &= B \cup \big(D_k - \bigcup_{i>k}(D_k \cap D_i)\big) \tag{2.17} \end{align}$$

This equation means the classification region for detection box k will only be affected by the the boxes whose detections scores are higher.

As we have mentioned above, there may be misdetection in the image, this would affect our results a lot. So we define a backbround region like

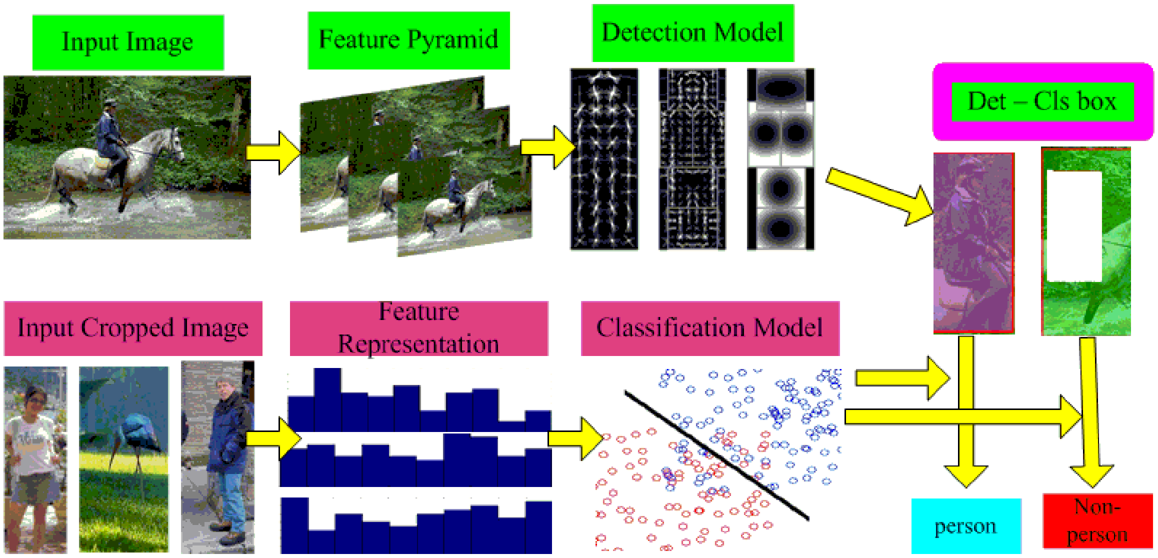

Figure 2.1: Workflow for classification leveraged Object Detector. Given the ground truth bounding boxes, we train one DPM detector as it is shown in the first row and we train one classification classifier as it is show in the second row. The supporting regions are obtained via subtraction of the detection bounding boxes, then the support regions are feed into the classification classifier to further decide whehter the original detection boxes contains target or not

$$B_i = D_i^c \cap (\bigcap_{i=1}^{N} \overline{D}_i) - D_i \tag{2.18}$$

In this equation, D_i^c is the box D_i with an extra margin.

2.4.2 Workflow for Classification Leveraged Object Dectector

For our Classification Leveraged Object Detector (CLOD), as shown in Figure 2.1, first we use the deformable part models to train a detection model for a detection dataset. Then we crop the ground truth in the dataset to form a cropped ground truth dataset, which is used for training a classification model. Later, we will explain why we choose this cropped ground truth dataset for the classfication model in Section 3.1.1.

With trained object detection model and classification model, we first apply the detection model to achieve detection candidate boxes. Then, each candidate box is given a supporting region for classification as we have defined in Section 2.4. Now, we can apply the classification model to those supporting regions, and we could get a classification score to help us rescore the original detection boxes

Chapter 3

Experiments and Discussion

3.1 Experiments and Discussion

To demonstrate the advantage of our approach, we adopt the very challenging PASCAL Visual Object Challenge 2007 (VOC2007) datasets [12]. First, we give a detailed description of VOC2007 dataset and the cropped dataset for our CLOD framework. Then, we evaluate our classification algorithm on PASCAL VOC2007. After that we compare the CLOD performance with the state the art detection peformance on PASCAL2007. Finally, the context information is incorporated into our framework to get the best performance mean AP 39.5%

3.1.1 Datasets and Metrics

PASCAL VOC2007 dataset

PASCAL VOC2007 datasets [12] has 20 categories, containing 9,963 images and 24,640 objects. This dataset is divided into "train", "val"and "test" subsets, which contains 2501, 2510 and 4592 images respectively. Parameters of the algorithm are

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow
Train	151	176	243	140	253	155	625	185	400	136
Val	155	177	243	150	252	114	625	190	398	123
Test	285	337	459	263	469	213	1201	358	756	244
	table	dog	horse	motor	person	plant	sheep	sofa	train	tv
Train	103	253	182	167	2358	248	130	124	145	166
Val	112	257	180	172	2332	266	127	124	152	158
Test	206	489	348	325	4528	480	242	239	282	308

Table 3.1: Statistics of ground truth bounding boxes

tuned via training on "train" set and evaluating on "val" set. The final model is trained on "train" + "val" sets and is applied on the "test" set to obtain the final results. This dataset are extremely challenging since the objects vary significantly in size, view angle, illumination, appearance and pose.

For the detection task, we do a staticstics of ground truth bounding boxes, as it shown in Table 3.1.

Region-level Dataset

Notice the the classification is applied on the supporting regions instead of the the whole image as it is shown in Section 2.4, so a region level (achieved by cropping the bounding boxes from the dataset) seems necessary for a satisfied classification classifier in CLOD.

We prepare a region-level dataset by cropping the detection ground truth boxes according the detection annotations. This cropped dataset also contains "train", "val" and "test" subsets. The positive examples are the ground truth bounding boxes, as shown in Table 3.1, while the negative examples are the ground truth bounding boxes from other categories. Although the Table 3.1 contains the "test" subsets, "test" subset is only to evaluate the final region-level classification classifier.

There is also another way to get classification classifiers: we can use detection false alarm boxes as the negative and the ground truth boxes as the positive. Notice that the false alarms boxes here are applied the supporting region technique, so that the false alarm does not have any part of the ground truth. In this way each category has a different kmeans codebook while features from other categories are not taken into consideration. For this task, the positive examples are the sum of ground truth in “train” and “val” in the Table 1, while negative examples are a random selection from the false alarms from the detection boxes in the “trainval” dataset. Remember, here each negative sample is changed by our CLOD methods. The number of negative samples is 2 times of the number of positive samples.

Metrics

Average Precision (AP) For the VOC2007 Challenge, the interpolated average precision [37] was used to evalute both classification and detection.

For a given task and class, the precision/recall curve is computed from a methods ranked output. Recall is defined as the proportion of all positive examples ranked above a given rank. Precision is the proportion of all examples above that rank which are from the positive class. The AP summarises the shape of the precision/recall curve, and is defined as the mean precision at a set of eleven equally spaced recall levels $[0, 0.1, ..., 1]$:

$$AP = \frac{1}{11} \sum_{r \in 0, 0.1, ..., 1} P_{interp}(r) \tag{3.1}$$

The precision at each recall level r is interpolated by taking the maximum precision measured for a method for which the corresponding recall exceeds r:

$$P_{interp}(r) = \max_{\hat{r}: \hat{r} \geq r} p(\hat{r}) \tag{3.2}$$

Where $p(\hat{r})$ is the measured precision at recall $\hat{r}$

Bounding Box Evaluation As noted, for the detection task, participants submitted a list of bounding boxes with associated score (rank). Detections were assigned to ground truth objects and judged to be true/false positives by measuring bounding box overlap. To be considered a correct detection, the overlap ratio a_o between the predicted bounding box B_p and ground truth bounding box B_gt must exceed 0.5 (50%) by the formula

$$a_o = \frac{B_p \cap B_gt}{B_p \cup B_gt} \tag{3.3}$$

where $B_p \cap B_gt$ denotes the intersection of the predicted and ground truth bounding boxes and $B_p \cup B_gt$ their union

3.1.2 Classification Classifier

In this section, we first tune the parameters of our classification algorithm using the image-level dataset and compare the performance with other state-of-the-art classfiication algorithm. Then we fix those parameters and apply the classification algorithm in our CLOD framework to compare the image-level classifier and region-level classifier.

Image-level classification

For our classification method, we choose dense SIFT and LBP as features and BoF+SPM+LLC system. For both dense SIFT and LBP, we adopt a multi-scale technique, in which the patch size for dense SIFT is $8 \times 8, 16 \times 16, 25 \times 25, 36 \times 36$ and the patch size for LBP is $12 \times 12, 16 \times 16, 20 \times 20, 24 \times 24$. The stride for dense SIFT is 4 and LBP is using 50% overlap stride. After extracting the dense SIFT and LBP features, a codebook is trained separately by kmeans. The codebook size for each feature is 10240 and the spatial pyramid matching is using $1 \times 1, 1 \times 2$, and 2×3. Therefore, each image would have a 184320-dimension feature. We can see the performance of our classification classifier on PASCAL VOC 2007 and compare it with other state-of-the-art classification algorithms on Table 3.2.

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	
INRIA Genetic	77.5	63.6	56.1	71.9	33.1	60.6	78.0	58.8	53.5	42.6	
SuperVec	79.4	72.5	55.6	73.8	34.0	72.4	83.4	63.6	56.6	52.8	
INRIA 2009	77.2	69.3	56.2	66.6	45.5	68.1	83.4	53.6	58.3	51.1	
TagModal	87.9	65.5	76.3	75.6	31.5	71.3	77.5	79.2	46.2	62.7	
CODC	82.5	79.6	64.8	73.4	54.2	75.0	87.5	65.6	62.9	56.4	
our dSIFT+LLC	73.1	61.2	49.1	65.5	26.0	55.0	75.7	56.9	51.7	36.1	
our dLBP+LLC	74.8	54.3	40.7	65.1	20.9	53.0	69.9	54.8	50.7	31.8	
our dSIFT+dLBP	77.2	64.3	52.7	70.4	27.2	60.3	77.3	61.0	54.6	40.2	
	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	mAP
INRIA Genetic	54.9	45.8	77.5	64.0	85.9	36.3	44.7	50.6	79.2	53.2	59.4
SuperVec	63.2	49.5	80.9	71.9	85.1	36.4	46.5	59.8	83.3	58.9	64.0
INRIA 2009	62.2	45.2	78.4	69.7	86.1	52.4	54.4	54.3	75.8	62.1	63.5
TagModal	41.4	74.6	84.6	76.2	84.6	48.0	67.7	44.3	86.1	52.7	66.7
CODC	66.0	53.5	85.0	76.8	91.1	53.9	61.0	67.5	83.6	70.6	70.5
our dSIFT+LLC	46.8	39.5	76.1	61.9	81.6	25.5	42.3	52.2	73.9	50.25	55
our dSIFT+LLC	40.8	42.6	72.9	46.8	80.3	22.2	34.8	43.7	72.7	39.06	50.6
our dSIFT+dLBP	53.8	46.9	77.2	62.4	84.0	26.8	44.1	54.2	77.2	51.4	58.2

Table 3.2: Classification Performance on PASCAL VOC 2007.

From the Table 3.2, we could see that our classifier is not the best one, but later we will prove that even with this below-average classification classifier, our CLOD approach would still be able to boost the detection a lot.

Region-level Classification

From Section 3.1.1, there are two kinds of region-level dataset.With the exact same experiment setup, we train our region-level classifier on the "train" + "val" subsets of the cropped ground truth dataset and support-region dataset. We evaluate it on the "test" subset, the performance is listed in Table 3.3.

In Table 3.3, only the LBP feature is used due to the speed issue. We can see that the image-level classifier is the worst and the region-level classifier from the pure ground-truth-box set is the best. Since our CLOD actually applied the classification on the supporting regions instead of the whole images, it is reasonable that the image-level classifier does not work well. But it is quite interesting that the classifier from the pure ground-truth-box dataset is better than the classifier from ground-truth-false-alarms set. Recall that our classifiers from ground-truth-false-alarms set just

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	
Det	35.7	59.8	11.8	19.6	31.0	51.8	58.7	29.3	23.4	28.7	
CLOD-I	36.4	59.8	11.8	19.6	31.0	51.8	58.8	29.3	23.6	28.7	
CLOD-Rg	37.0	60.1	12.2	20.6	31.9	53.4	59.6	32.3	24.0	31.4	
CLOD-Rf	36.5	59.9	12.0	20.0	31.1	52.3	58.7	30.4	23.5	29.5	
	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	mAP
Det	26.0	15.5	60.1	50.5	44.1	13.3	27.7	37.6	48.8	45.3	35.9
CLOD-I	26.0	15.5	60.1	50.5	44.1	13.5	27.7	37.6	48.8	45.3	36.0
CLOD-Rg	29.8	17.2	61.7	53.0	44.4	15.1	27.8	40.6	49.8	45.3	37.3
CLOD-Rf	26.6	16.5	60.6	51.0	44.2	14.4	27.7	37.8	48.9	45.7	36.3

Table 3.3: Comparison of CLOD with different type of classification classifiers. Det means the performance of preliminary detection resutls. CLOD-I means CLOD using classification classifiers trained on image-level set. CLOD-Rg means CLOD using classification classifiers trained on region-level pure ground-truth-box set. CLOD-If means CLOD using classification classifiers trained on region-level ground-truth-false-alarms set.

used twice times negative samples as the positive samples, while the classifier from pure-ground-truth-box set has almost 20 times the number of negative examples. The reason why negative samples for the classifiers from the ground-truth-false-alarms set are much less is mainly the time consideration. For the classifier from the cropped dataset, we need extract features from 12,608 images (total sum of the ground truth in “train” and “val” set), while for the classifier from the false alarms, there would be almost $12,608 \times 20 = 252160$ images because each classifier would have different negative samples. What’s more, the classifier from the ground-truth-false-alarms set requires 20 times more k-means than the the classifier from the pure-ground-truth dataset. Therefore, by taking all of the above into consideration, we choose the classification classifier from the pure-ground-truth dataset as our classification classifier in our CLOD framework.

3.1.3 Supporting Regions

From Section 2.4, the supporting regions are defined as the subtractions of bounding boxes from detection classifiers. In fact given different detection threshold, there will be different number of detection bounding boxes. In most of the object categories, if the detector threshold is set to -1.1, usually we would have more than 10,000 detection

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow
Box Num	2326	1763	1524	999	1225	1660	2657	1354	1204	2212
	table	dog	horse	motor	person	plant	sheep	sofa	train	tv
Box Num	2254	1844	1466	858	8132	923	715	756	844	1164

Table 3.4: Detection box number for classification.

candidate boxes for each category, which is too large to adopt any complicated and time-consuming classification algorithm. To reduce the candidate boxes for each class, we set threshold to -0.95 for all the categories, which would lead most categories to contain less than 2,000 candidate boxes. The details can be see from Table 3.4. The experiments show that even with this much less candidate boxes, we can still achieve very good performance (mean AP = 39.5%).

3.1.4 Leverage Detection with Classification and Context Information

Now, we have already discussed the datasets and details for the CLOD framework. In the Section 2.4, we showed that each box has a detection score and classification score. Notice that the classfication score is not achieved by the whole bounding box region but the supporting region. The Figure 3.1 compares the performance of using detection scores, supporting region classification scores and their combination on the PASCAL V0C2007 dataset.

From the Figure 3.1, we can see that if the detection curve is always far better than the classification curve, classification score would not boost the performance that much. But if the classification is similar or just a little worse than the detection curve, then det+cls curve would have a large improvement compared to the detection curve.

Also, you may notice that there is a big drop of the curve at the high recall part. This is due to the fact that we just apply our classification to the detection boxes whose detection scores are larger than -0.95. For the rest of detection boxes, we

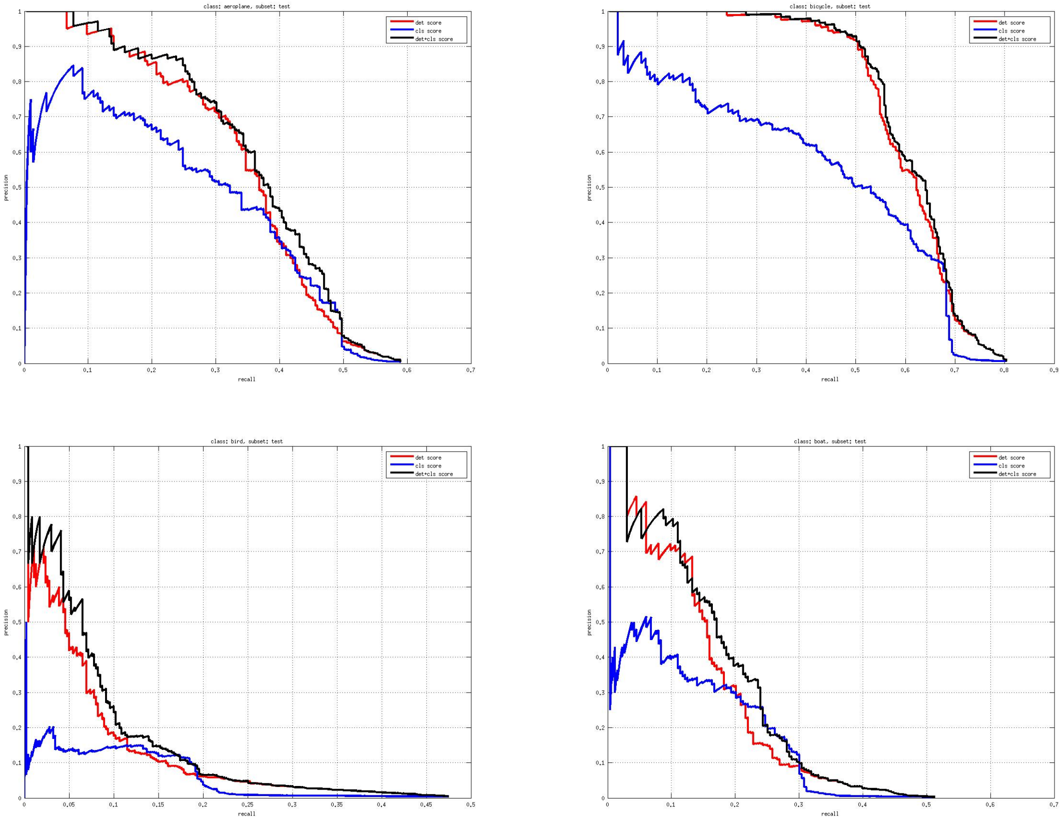

Figure 3.1: AP using detection scores, supporting region classification scores and their combination(PASCAL 2007 category 1-4).

simply apply -10 as the classification score.

For the context rescore, we choose number of positive examples as it is shown in the Table 3.4 and the number of negative samples is 3 times the number of positive examples for each category.

The Figure 3.2 shows the average precision for 20 categories in PASCAL VOC2007.

3.1.5 Discussion

Due to the complexity and time consuming classification algorithm, it is impossible for us to evaluate all the supporting regions. For those detection boxes which don't have the classification score, we just assign a constant negative value to them, which

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	mAP
Leo [30]	29.4	55.8	9.4	14.3	28.6	44.0	51.3	21.3	20.0	19.3	25.2	12.5	50.4	38.4	36.6	15.1	19.7	25.1	36.8	39.3	29.6
UCI2009	28.8	56.2	3.2	14.2	29.4	38.7	48.7	12.4	16.0	17.7	24.0	11.7	45.0	39.4	35.5	15.2	16.1	20.1	34.2	35.4	27.1
CMO [22]	31.5	61.8	12.4	18.1	27.7	51.5	59.8	24.8	23.7	27.2	30.7	13.7	60.5	51.1	43.6	14.2	19.6	38.5	49.1	44.3	35.2
INRIA2009	35.1	45.6	10.9	12.0	23.2	42.1	50.9	19.0	18.0	31.5	17.2	17.6	49.6	43.1	21.0	**18.9**	27.3	24.7	29.9	39.7	28.9
UoC2010	31.2	61.5	11.9	17.4	27.0	49.1	59.6	23.1	23.0	26.3	24.9	12.9	60.1	51.0	43.2	13.4	18.8	36.2	49.1	43.0	34.1
Det-Cls [23]	38.6	58.7	**18.0**	18.7	31.8	53.6	56.0	30.6	23.5	31.1	**36.6**	20.9	62.6	47.9	41.2	18.8	23.5	41.8	**53.6**	45.3	37.7
Oxford [1]	37.6	47.8	15.3	15.3	21.9	50.7	50.6	30.0	17.3	33.0	22.5	**21.5**	51.2	45.5	23.3	12.4	23.9	28.5	45.3	48.5	32.1
NLPR [28]	36.7	59.8	11.8	17.5	26.3	49.8	58.2	24.0	22.9	27.0	24.3	15.2	58.2	49.2	44.6	13.5	21.4	34.9	47.5	42.3	34.3
Ver.5 [17]	36.6	62.2	12.1	17.6	28.7	54.6	60.4	25.5	21.1	25.6	26.6	14.6	60.9	50.7	44.7	14.3	21.5	38.2	49.3	43.6	35.4
MOCO [38]	**41.0**	**64.3**	15.1	19.5	33.0	**57.9**	**63.2**	27.8	23.2	28.2	29.1	16.9	63.7	53.8	**47.1**	18.3	28.1	42.2	53.1	**49.3**	38.7
nms05	35.7	59.8	11.8	19.6	31.0	51.8	58.7	29.3	23.4	28.7	26.0	15.5	60.1	50.5	44.1	13.3	27.7	37.6	48.8	45.3	35.9
nms05+cls	37.4	60.3	12.5	21.0	31.7	54.0	59.8	32.9	24.1	32.3	31.5	18.1	62.6	54.1	44.6	15.3	28.9	42.0	50.3	45.7	38.0
nms05+contex	37.7	61.8	16.2	22.2	32.1	52.9	60.1	32.0	24.7	30.9	31.2	18.6	62.5	53.8	45.2	17.9	28.9	40.0	50.3	47.5	38.3
nms05+contex+cls	38.9	62.4	16.5	**22.7**	**32.2**	54.8	60.9	**34.0**	**25.4**	**33.4**	34.2	20.0	**63.8**	**55.1**	45.7	18.6	**30.4**	**42.6**	51.4	47.8	**39.5**

Table 3.5: Comparison with the state-of-the-art performance of object detection on PASCAL VOC 2007.

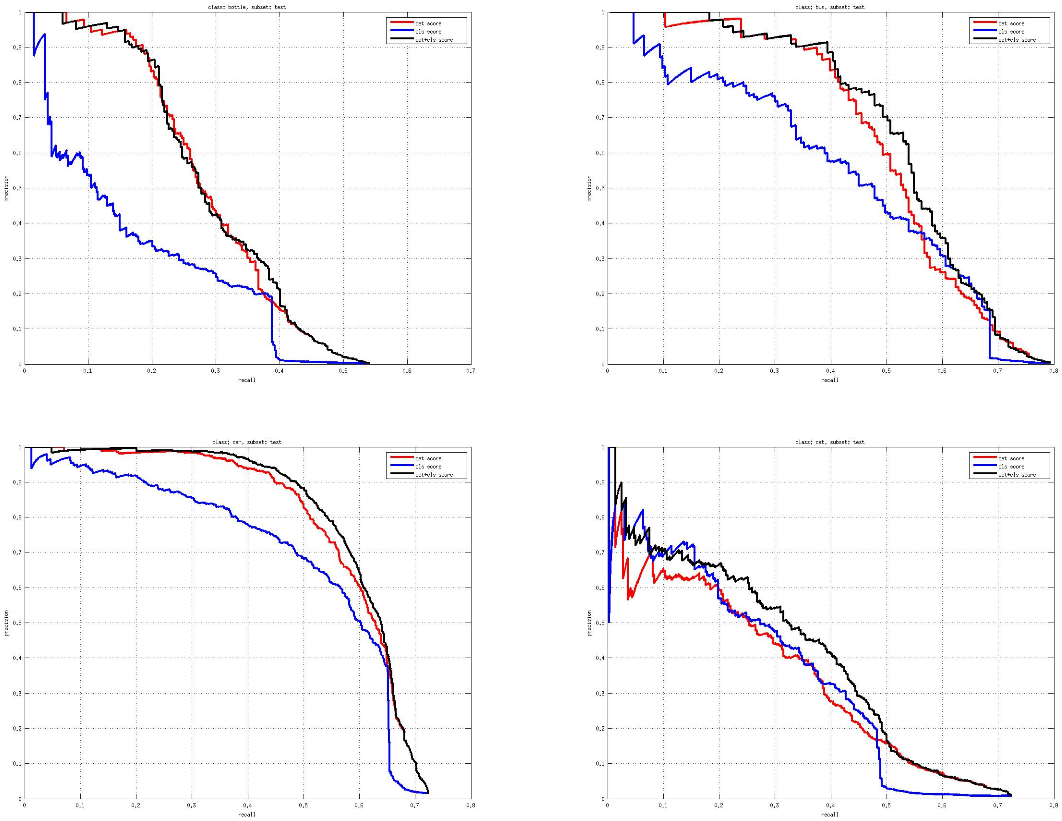

Figure 3.1: AP using detection scores, supporting region classification scores and their combination(PASCAL 2007 category 5-8).

is not a good technique. In the future, we would design a more efficient classification algorithm, even if this algorithm may not perform as well as the complex and time-consuming classification algorithm. We can apply the complex and time-consuming classification algorithm to the higher detection score boxes and the efficient classification algorithm to lower detection score boxes. Currently, this classification algorithm is just to help to rescore the detection boxes. But later, we would use the classification algorithm to find the misdetections.

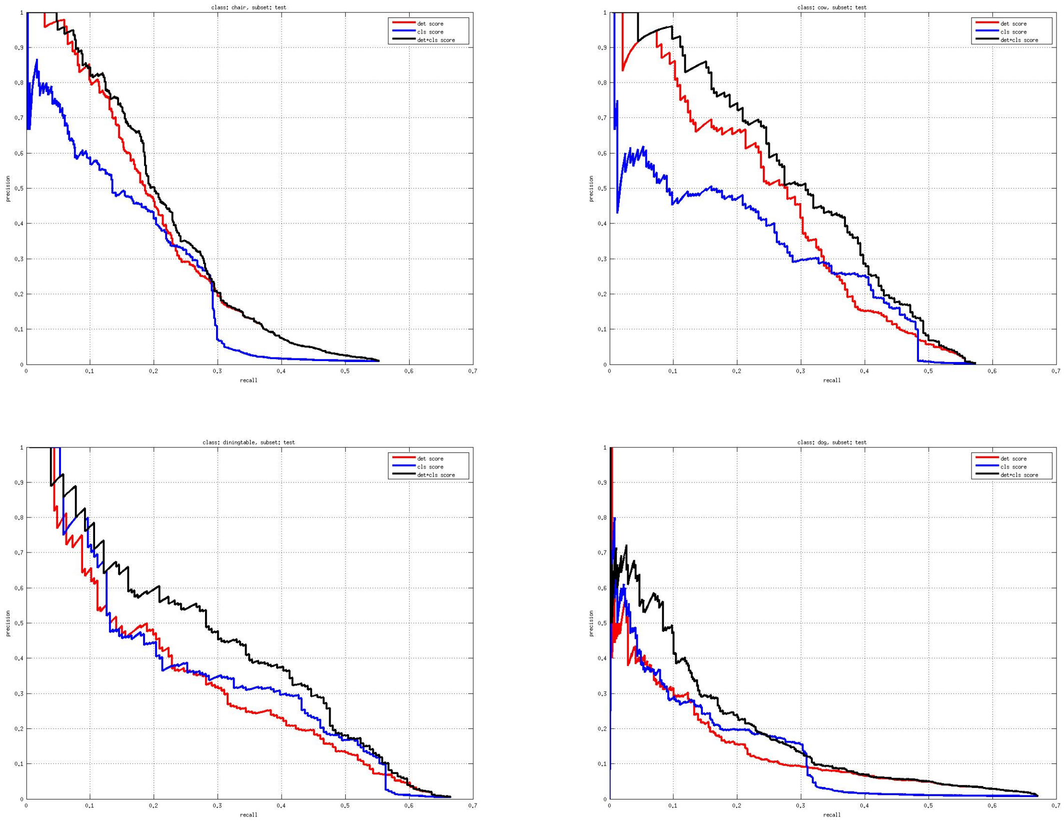

Figure 3.1: AP using detection scores, supporting region classification scores and their combination(PASCAL 2007 category 9-12).

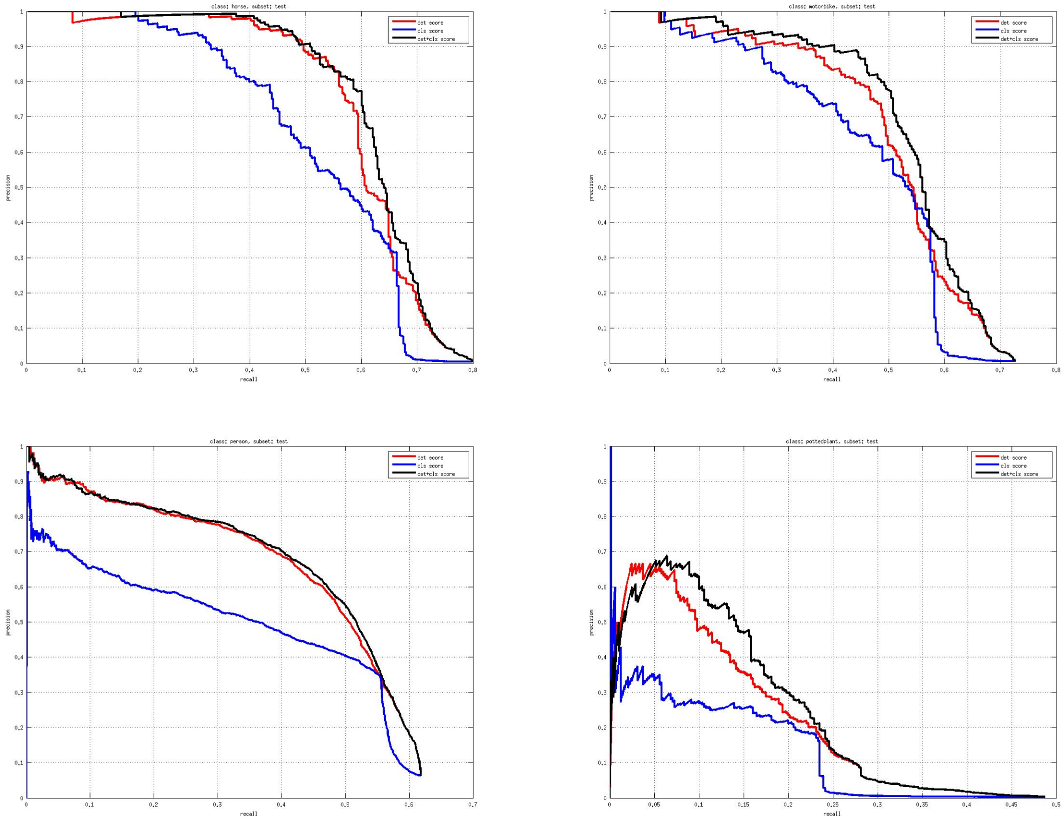

Figure 3.1: AP using detection scores, supporting region classification scores and their combination(PASCAL 2007 category 13-16).

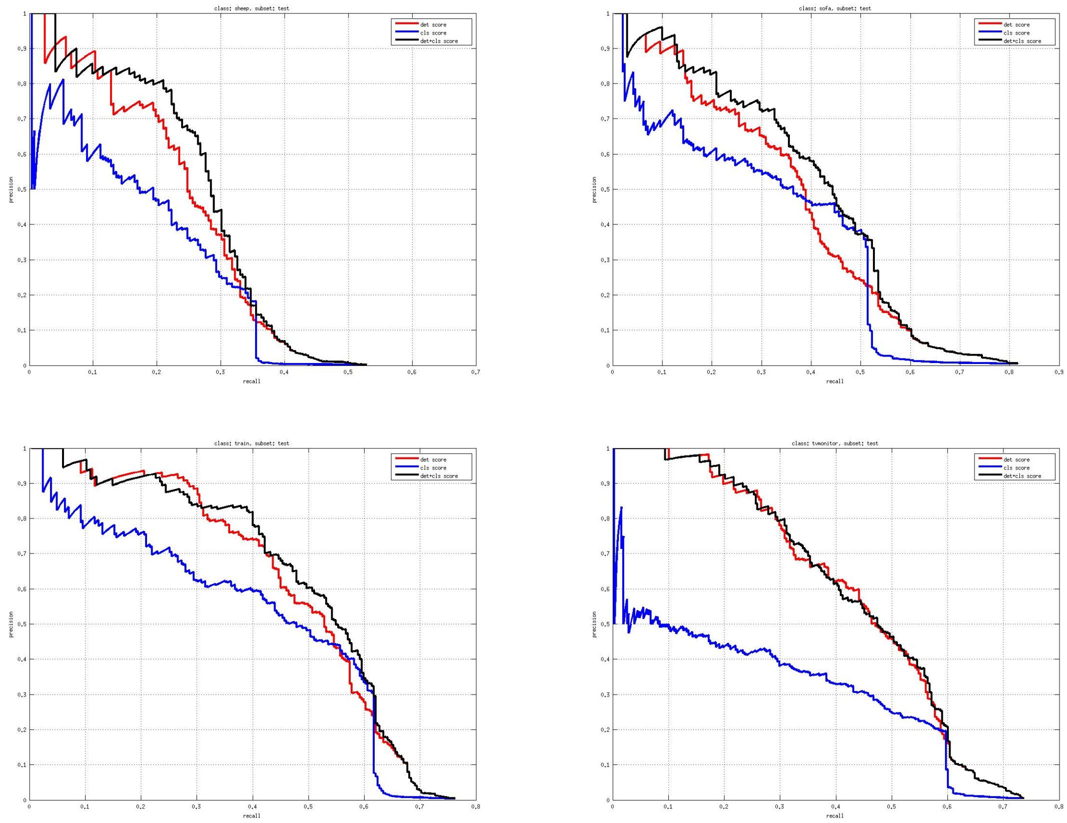

Figure 3.1: AP using detection scores, supporting region classification scores and their combination(PASCAL 2007 category 17-20).

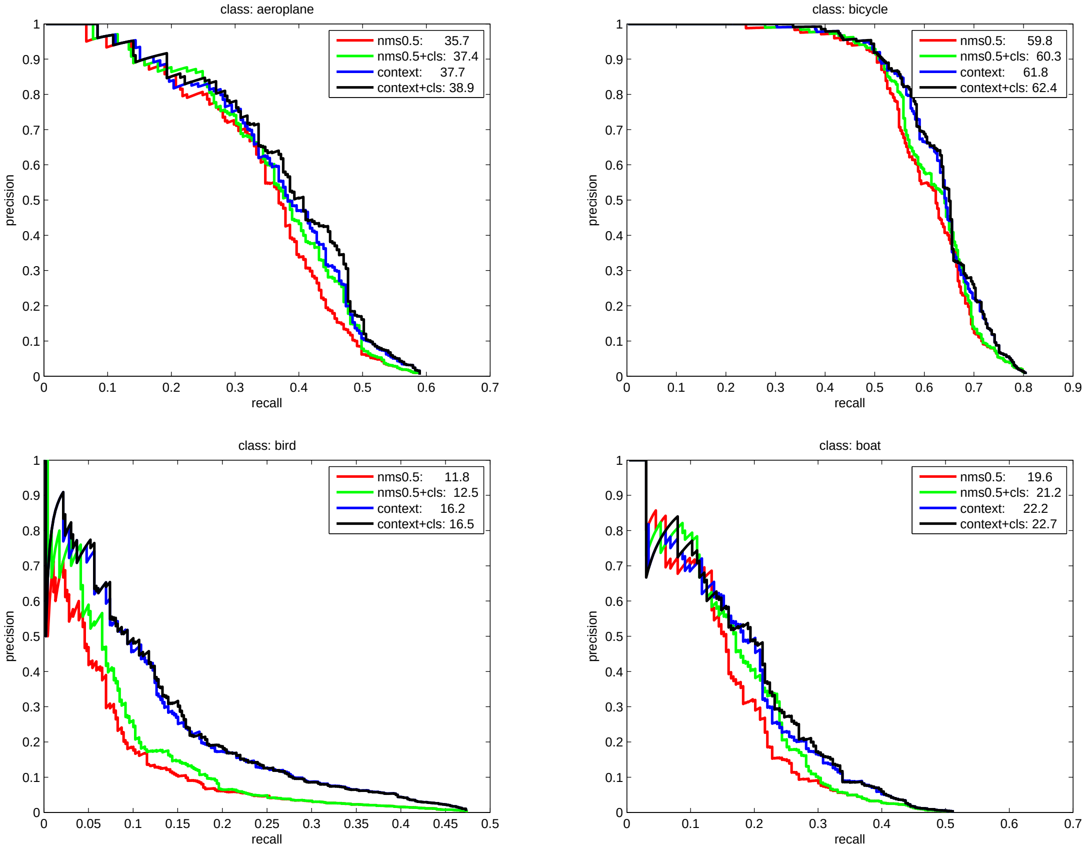

Figure 3.2: AP for context and CLOD (PASCAL 2007 category 1-4). nms05 means DPM detector performance; nms05+cls means CLOD on boxes from DPM detector; context means performance of apply context to DPM detector; context+cls means apply CLOD to context rescored bounding boxes.

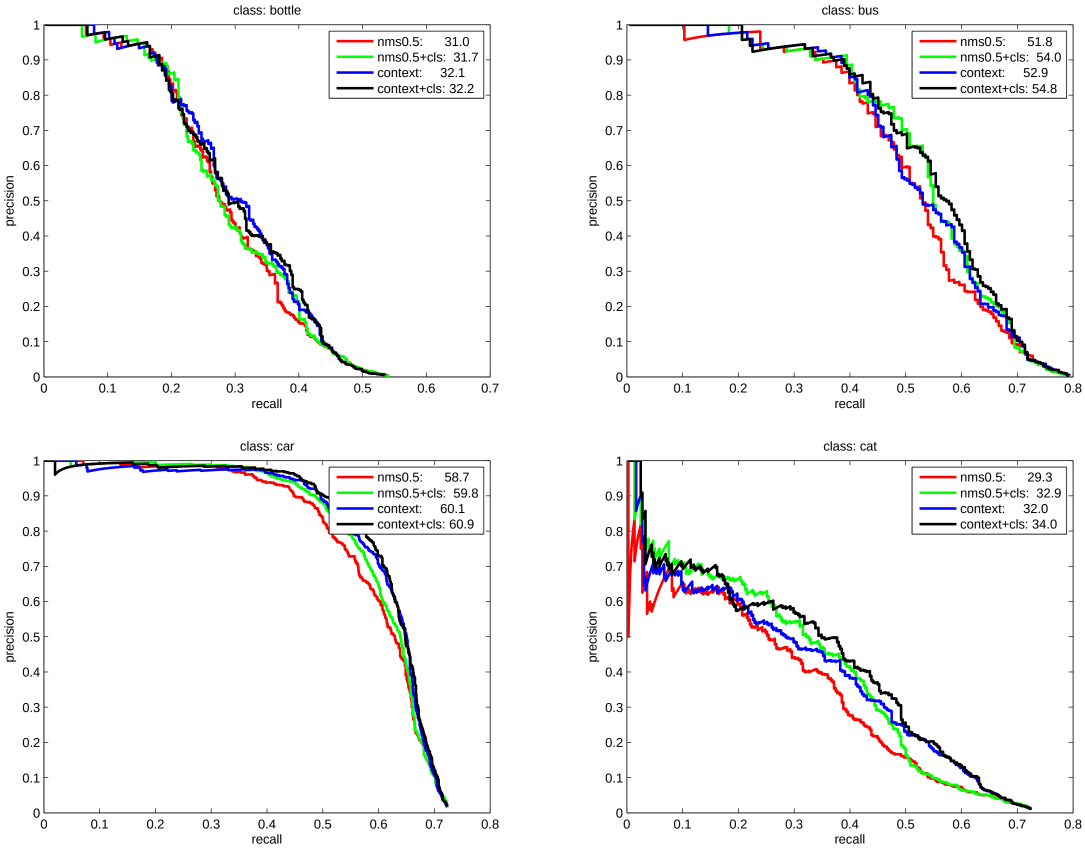

Figure 3.2: AP for context and CLOD (PASCAL 2007 category 5-8)

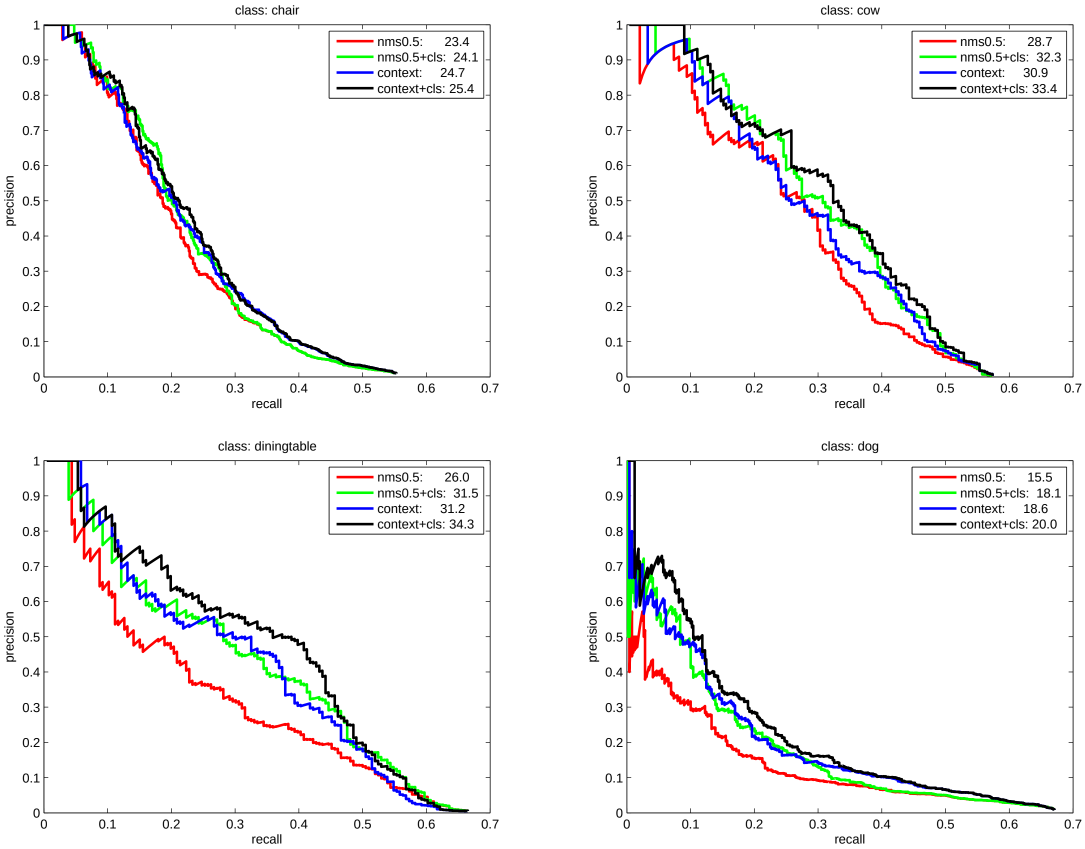

Figure 3.2: AP for context and CLOD (PASCAL 2007 category 9-12)

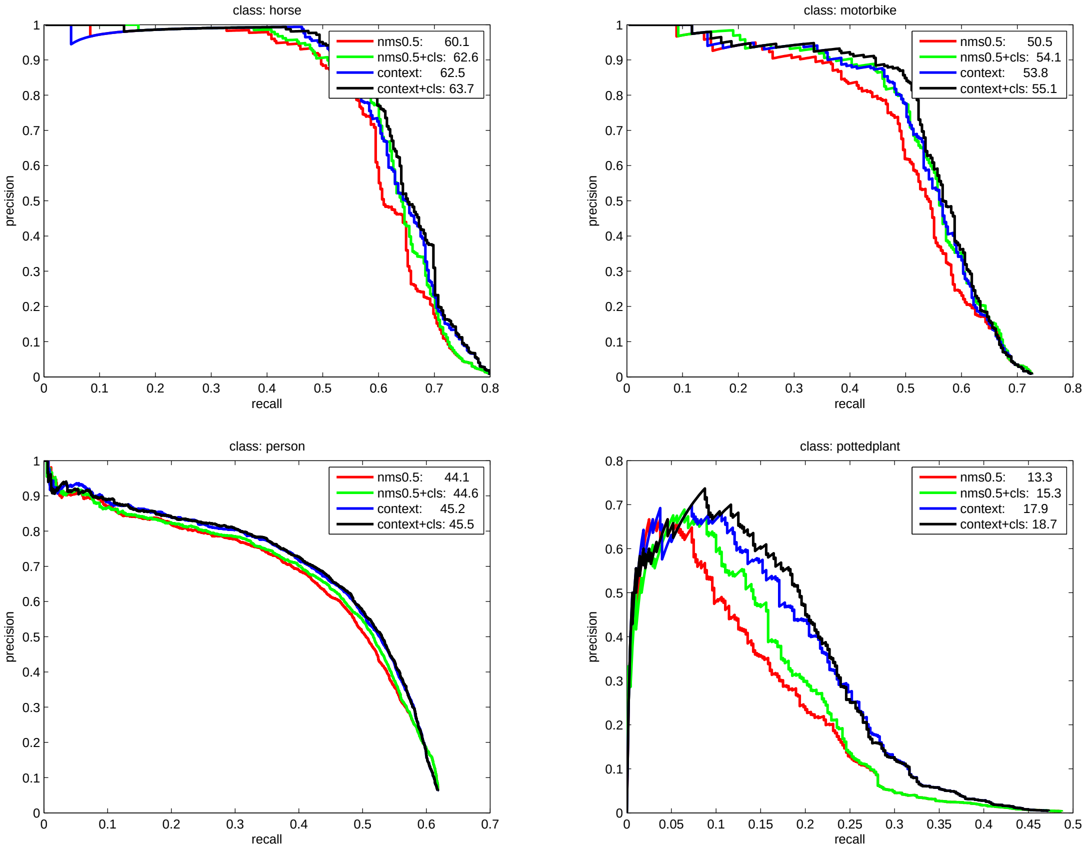

Figure 3.2: AP for context and CLOD (PASCAL 2007 category 13-16)

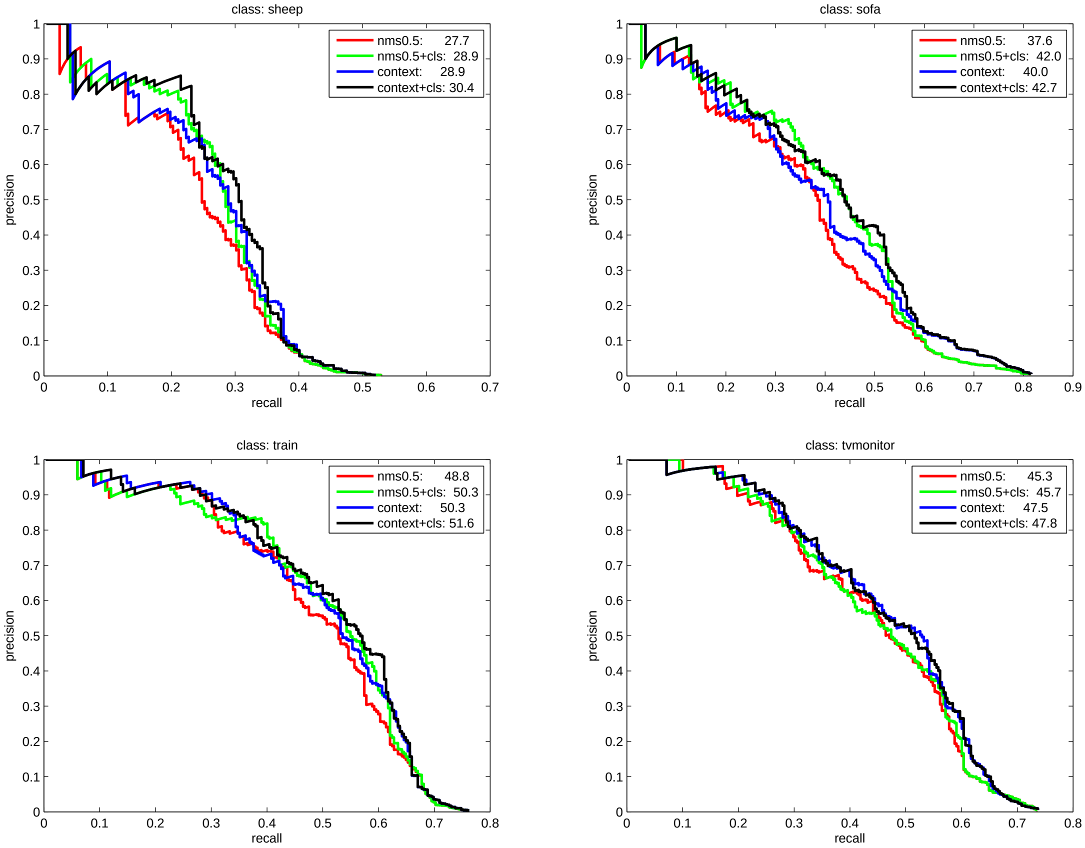

Figure 3.2: AP for context and CLOD (PASCAL 2007 category 17-20)

Chapter 4

Summary and Concluding Remarks

In this paper, we have proposed a simple but powerful object detector called Image-classification Leveraged Object Detector. This detector needs a detection model and a classification model for each class. Extensive experiments on PASCAL2007 has shown the advantage of our approach. we achieved rank 1st for 9 categores and the mean AP is 39.5%, which outperforms all other results.

Bibliography

[1] M. V. A. Vedaldi, V. Gulshan and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009.

[2] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *CVPR*, 2008.

[3] A. Bosch, A. Zisserman, and X. Muñoz. Scene classification using a hybrid generative/discriminative approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(4):712–727, 2008.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[5] David G. Lowe Distinctive image features from scale-invariant keypoints *IJCV*, 2004.

[6] P. Sabzmeydani and G. Mori Detecting pedestrians by learning shapelet features In *CVPR*, 2007.

[7] T. Ojala, M. Pietikinen, and D. Harwood A comparative study of texture measures with classification based on feature distributions *Pattern Recognition*, 29(1):5159, 1996.

[8] T. Ojala, M. Pietikinen, and T. Maenpaa Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns *IEEE Transactions On Pattern Analysis and Machine Intelligence*, 24(7):971987, 2002.

[9] T. Ahonen, A. Hadid, and M. Pietikinen Face description with local binary patterns: Application to face recognition *IEEE Trans. Pattern Anal. Mach. Intell*, 28(12):20372041, 2006.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[11] C. Desai, D. Ramanan, and C. C. Fowlkes. Discriminative models for multi-class object layout. *International Journal of Computer Vision*, 95(1):1–12, 2011.

[12] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge. *IJCV*, 2010.

[13] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2012 results. *http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2012/*, 2012.

[14] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010.

[15] P. F. Felzenszwalb, D. A. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[16] P. F. Felzenszwalb, R. Girshick, and D. McAllester. Cascade object detection with deformable part models. In *CVPR*, 2010.

[17] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 5. http://people.cs.uchicago.edu/ rbg/latent-release5/.

[18] M. Pedersoli, A. Vedaldi, and J. Gonzalez. A coarse-to-fine approach for fast deformable object detection. In *CVPR*, 2011.

[19] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *In ICCV*, pages 1458–1465, 2005.

[20] F. S. Khan, J. van de Weijer, and M. Vanrell. Modulating shape features by color attention for object recognition. *International Journal of Computer Vision*, 98(1):49–64, 2012.

[21] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, pages 2169–2178, Washington, DC, USA, 2006. IEEE Computer Society.

[22] C. Li, D. Parikh, and T. Chen. Extracting adaptive contextual cues from unlabeled regions. In *ICCV*, 2011.

[23] Z. Song, Q. Chen, Z. Huang, Y. Hua, and S. Yan. Contextualizing object detection and classification. In *CVPR*, 2011.

[24] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders. Segmentation as selective search for object recognition. In *IEEE International Conference on Computer Vision*, 2011.

[25] J. Wang, J. Yang, K. Yu, F. Lv, T. S. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, pages 3360–3367, 2010.

[26] Y. Yang and S. Newsam. Spatial pyramid co-occurrence for image classification. In *ICCV*, 2011.

[27] H. Zhang, A. C. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. CVPR '06, pages 2126–2136, Washington, DC, USA, 2006. IEEE Computer Society.

[28] J. Zhang, K. Huang, Y. Yu, and T. Tan. Boosted local structured hog-lbp for object localization. In *CVPR*, 2010.

[29] X. Zhou, N. Cui, Z. Li, F. Liang, and T. S. Huang. Hierarchical gaussianization for image classification. In *ICCV*, pages 1971–1977, 2009.

[30] L. Zhu, Y. Chen, A. L. Yuille, and W. T. Freeman. Latent hierarchical structural learning for object detection. In *CVPR*, 2010.

[31] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints In *ECCV*, 2004.

[32] S. Tong and D. Koller. Support vector machine active learning with applications to text classification In *ICML*, 2000.

[33] K. Grauman and T. Darrell. Pyramid match kernels: Discriminative classification with sets of image feature In *ICCV*, 2005.

[34] J. Willamowski, D. Arregui, G. Csurka, C. R. Dance, and L. Fan. Categorizing nine visual classes using local appearance descriptors In *ICPR Workshop on Learning for Adaptable Visual Systems*, 2004.

[35] F.-F. Li, R. Fergus, and P. Perona. Learning generative visual models from few training examples:an incremental bayesian approach tested on 101 object categories. In *CVPR Workshop on Generative-Model Based Vision*, 2004.

[36] K. Yu, T. Zhang, and Y. Gong. Nonlinear learning using local coordinate coding. In *Proc. of ICCV09* , 2009.

[37] Salton, G., McGill, M. J. Introduction to modern information retrieval. *McGraw-Hill.*, 1986.

[38] Guang Chen, Yuanyuan Ding, Jing Xiao, Tony X. Han Detection Evolution with Multi-Order Contextual Co-occurrence *CVPR*, 2013