B.Tech. Project

First Stage Report on

Object Detection and Category Recognition

Submitted by

Arjun Malik

14075060

Under the supervision of

Prof. Rajeev Srivstava

Department of Computer Science And Engineering

Indian Institute of Technology (BHU) Varanasi

Acknowledgements

I have taken efforts in this project. However, it would not have been possible without the kind support and help of my professors .

I am highly indebted to **Prof. Rajeev Srivstava** for his guidance and constant supervision as well as for providing necessary information regarding the project and also for his support in completing the project.

Arjun Malik

14075060

CERTIFICATE

I hereby certify that the work which is being presented in the B.Tech. Project Report entitled **"Object Detection and Category Recognition",** in partial fulfillment of the requirements for the award of the **Bachelor of Technology in Computer Science and Engineering** and submitted to the Department of Computer Science and Engineering of ***Indian Institute of Technology (BHU) Varanasi*** is an authentic record of my own work carried out during a period from January 2017 to April 2017 under the supervision of **Prof. Rajeev Srivstava , CSE Department.**

The matter presented in this project has not been submitted by me for the award of any other degree elsewhere.

Signature of Candidate

Arjun Malik

14075060

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Signature of Supervisor

***Prof.* Rajeev Srivstava**

Department of Computer Science and Engineering

Indian Institute of Technology (BHU) Varanasi

TABLE OF CONTENTS

Section

PAGE

Sections

1 Abstract

The goal of this paper is to develop practical methods for detecting common object classes in real world images. An object detector that combines Selective Search technique for segmentation with Histogram of Oriented Gradient (HOG) features with efficient SVM classifiers and effective dimensionality reduction is presented and it's performance is tested on an important dataset, namely, PASCAL VOC2007.

The three main problems addressed are as follows.

Firstly, to generate locations which are precise and small in number such that they can be used for recognition of objects, Selective Search is used.

Secondly, to depict an object such that it's shape information is captured while being resistant to variations in lighting conditions,HOG features are used.

Lastly, in order to tackle the speed and memory usage problems associated with high-dimensional modern feature sets, we use an effective dimensionality

reduction techniques namely, Principal Component Analysis(PCA) , which allows detectors to be trained more rapidly with a slight increase in accuracy and no loss of run time speed for detectors.

2 Introduction

One of the main goals of computer vision is to equip computers with artificial visual systems having human-like image understanding capabilities. One fundamental task of such systems will be the interpretation and labeling of scene content. Such interpretation can occur at several levels within an image:

•Image classification is the task of annotating entire images according to the elements

present in them. It says what is present without necessarily saying where3 .

•Object detection/localization is the task of identifying the presence, location and

extent of any instances of a given object class that are present in the image. Extent can be indicated by, e.g., bounding boxes or pixel-level masks. Multiple categories can be also detected, however identifying all instances of all of the object classes that are present in everyday images is beyond the scope of this paper.

- Semantic segmentation is the task of labeling each image pixel with the object class that it was generated by.

In this paper, we will be concerned exclusively with the detection/localization task. The above recognition tasks will focus on generic class level detection (“person”, “car”, “horse”, ...).

Reliable practical object detectors would have many applications. Current image content management systems are based mainly on manually supplied meta-data provided either by the uploading user or by a specially employed workforce. Annotation is tedious, costly and error prone, and even at the best it seldom provides very complete coverage (many instances are missed). Multi-category object detectors would allow images to be labelled automatically based on their content, thus facilitating content-based browsing, search and retrieval. Object detection would be also useful for intelligent environments, for example surveillance systems could automatically identify intruders

or aged people in need of assistance, and smarts cars could use object/human detection coupled with other cues to avoid collisions. Further application domains include gaming, robotics, entertainment, advertising and manufacturing – indeed any application where intelligent systems need to observe or interact with humans, objects or animals.

2.1 Challenges

The detection of visual object classes remains a challenging problem that faces the following issues:

- Imaging factors: Real world objects appear under a wide range of illumination conditions in both indoor and outdoor settings. Object images are also highly viewpoint dependent: the depth dimension is lost; image scale varies due to the relative location of the sensor and object; and the appearance of complex object classes varies significantly with viewing angle.

- Intra-class Variance: Within a given class, objects can exhibit widely varying shape, color and texture. For example the car category includes convertibles, hatchbacks, limousines, etc. Natural classes such as person, cat, dog, etc., also have a wide range of articulated poses and non-rigid deformations, resulting in highly variable object layout and appearance.
- Background Clutter: Objects appear against a wide range of backgrounds, often in close proximity to or direct interaction with neighbouring objects.
- Occlusion/Truncation: Objects are often occluded by other objects that lie in front of them or truncated by the borders of the image, so that only a portion of the object is visible.

2.2 General Methodology

Given the above-mentioned challenges, the object detection problem appears to be too complex to model analytically, so we resort to a learning-based approach in which a diverse and representative set of training examples is used as a surrogate for a model.

Object detection is thus cast as a problem of classifying potential candidates proposed by an underlying object position hypothesis generator, and machine learning is used to learn a decision rule for these hypotheses from a representative training set. This formulation allows advances in machine learning to be leveraged.

A typical object detector must make choices in three areas: the object position hypothesis generator; the set of visual descriptors used to capture object shape, color and texture characteristics; and the object/non-object classifier based on these features.

Current systems can be divided into two main categories on the basis of object position hypothesis generator.

In the first, descriptors are computed only sparsely at locations given by some local feature detector, and these positions are used to generate possible object hypotheses. This provide a relatively sparse set of hypotheses and hence a computationally efficient detector.

In the second approach, a ‘sliding window’ detector is swept across the image at multiple positions and scales, robust visual features are extracted at each window position, and a window-level object/non-object classifier is evaluated on these, often followed by post processing to merge overlapping duplicate detections. It is computationally intensive and the final results are critically dependent on the quality of the underlying classifier.

Our selective search approach lies in the first category. Besides, it is able to capture all scales of objects as in the case of ‘sliding window’ detector, thus generating precise object hypotheses.

Both generative and discriminative approaches have been used to learn the underlying classifiers. Generative classifiers define prior and likelihood models for the appearance of class and non-class instances, deriving the output function indirectly from the likelihood ratio of these, while discriminative models directly model the class/non-class decision given the input features. Although generative approaches have considerable long-term potential for deeper image understanding, the best current object detectors are trained discriminatively.

We use a discriminative classifier, namely, Support ector Machines(SVM) to train our object detector.

The rest of this paper is organized as follows: a) Section 3.1.1 describes how the initial segmentation was generated b) Section 3.1.2

describes how the initial regions were grouped hierarchically

c)Section 3.2 describes HOG features and their dimensionality reduction

d)Section 3.3 describes the training procedure

e)Section 3.4 describes the testing procedure

f)Section 4 describes the results obtained.

3 Methods

The model for object recognition comprises of the following major techniques which are described as follows:

3.1 Selective Search

It is a hierarchical grouping algorithm divided into two stages.

3.1.1 Generating Oversegmenation

First, initial regions are generated using a graph based approach.

Let G= (V , E) be the graph generated from the image such that each pixel in the image is taken as vertex

vi. Weighted edges (forming set E) are taken between neighboring vertices in an 8-connected image such that the weight is difference in intensity between pixels.

Thus for each edge ei between vertices (v_i, v_j) we have it’s weight $w(e_i)$ such that

$$w(e_i) = |I(v_i) - I(v_j)|$$

where $I(v_i)$ is the intensity of pixel vi

A segmentation of this image is defined as a partition of the graph into components such that each component $C \subseteq V$ is connected. The quality of this segmentation depends upon the similarity of each component and dissimilarity between different components.

Let us define some variables which would be used to understand this method.

We define the internal difference of a component C to be the largest weight in the minimum spanning tree of the component $MST(C, E)$.

$$Id(C) = \max_{e \in MST(C, E)} w(e)$$

This measure is used as a given component C only remains connected when edges of weight at least Id(C) are considered.

We define the difference between two components C1 , C2 ⊆ V to be the minimum weight edge connecting the two components. That is,

$$\mathrm{Dif}\ (C1,C2) = \min_{vi \in C1,\ vj \in C2\ ,(vi\ ,\ v\ j) \in E} w(v\ i\ ,v\ j)$$

If there is no edge connecting the two components we let Dif (C 1, C 2)=∞.

This measure of difference could in principle be problematic, because it reflects only the smallest edge weight between two components. In practice, this the measure works quite well in spite of this apparent limitation.

We define the pairwise comparison predicate D(C1 , C2),as

$$D\ (C\ 1\ ,\ C\ 2)=$$

$$\{\text{true if } Dif\ (C\ 1\ ,\ C\ 2)> MInt\ (C\ 1\ ,\ C\ 2)\ \}$$

$$\{\ \text{false otherwise}\ \}$$

The region comparison predicate evaluates if there is evidence for a boundary between a pair or components by checking if the difference between the components, Dif (C 1, C 2), is large relative to the internal difference within at least one of the components, Id(C 1) and Id(C2).

A threshold function is used to control the degree to which the difference between components must be larger than minimum internal difference.

The minimum internal difference is defined as,

$MInt\ (C\ 1,C\ 2)=min(Id(\ C\ 1)+ \tau(C\ 1),Id\ (\ C\ 2)+ \tau(C2))$

The threshold function τ controls the degree to which the difference between two components must be greater than their internal differences in order for there to be evidence of a boundary between them (D to be true).

The threshold function is based on the size of the component,

$$\tau(C) = k/|C|$$

where |C| denotes the size of C, and k is some constant parameter.

Here, larger k causes a preference for larger components.

This algorithm is based on a comparison predicate which determines whether there exists a boundary between two regions. A greedy algorithm merges regions based on this principle to generate an over-segmentation.

3.1.2 Hierarchical Grouping

The algorithm given in section 3.1.1 is used to create initial regions.

Then the following greedy algorithm is used to iteratively group regions together: First the similarities between all neighbouring regions are calculated. The two most similar regions are grouped together, and new similarities are calculated between the resulting region and its neighbours. The process of grouping the most similar regions is repeated until the whole image becomes a single region.

For the similarity s(Ci , Cj) between region Ci and Cj we want a variety of complementary measures under the constraint that they are fast to compute.

The sum of following similarity criteria was used as the total similarity between two regions Ci and Cj:

1)**Scolour** measures colour similarity. Specifically, for each region we obtain one-dimensional colour histograms for each colour channel using 25 bins. This leads to a colour histogram

H(i) = {h(i1) , · · · , h(in) } for

each region ri with dimensionality n = 75 when three colour channels are used.

Similarity is measured using the histogram intersection.

2)**Ssize** measures size similarity. Ssize(Ci , Cj) is defined as the fraction of the image that CI and

Cj jointly occupy. It encourages small regions to merge early. This forces regions in S, i.e. regions which have not yet been merged, to be of similar sizes throughout the algorithm.

3)**Sfill** (Ci , Cj) measures how well regions Ci and Cj fit into each other.

3.2 Histograms of Oriented Gradients (HOG)

Histograms of Oriented Gradients (HOG) [Dalal and Triggs 2005] are one of the most successful recent feature sets for visual recognition. Like SIFT [Lowe 2004], HOG is based on the assumption that local image content can be effectively encoded by local distributions of edge directions or intensity gradients, even without recording the precise locations of these.

HOG uses a descriptor which is computed on a dense grid of uniformly spaced cells at a single scale, with overlapping local contrast normalization blocks for improved discrimination. HOG has proven to be particularly effective at capturing coarse object shape (contour) information, with strong resistance to illumination variations and some robustness to small spatial variations.

Different cell resolutions can be used to capture different levels of information, e.g. A large, coarse-resolution cell can be used to capture the overall object shape while smaller and finer-resolution ones capture details of object parts.

The computation of HOG involves three main steps: image gradients are computed; the image is divided into a dense grid of rectangular “cells” and a histogram of gradient orientations is computed for each cell; and finally the cells are grouped into small (and typically overlapping) “blocks” and a local contrast normalization is applied to the cell histogram within each block.

Our HOG features are typically based on 8 × 8 pixel cells arranged into 1 × 1 cell blocks, with image gradients quantized into 8 orientation bins (evenly spaced over $0 - 180^{\circ}$).

This results in a 2048 dimensional feature vector for 128*128 sized image. Images of other sizes are reduced to the same size.

3.2.1 Positives and Negatives Mining

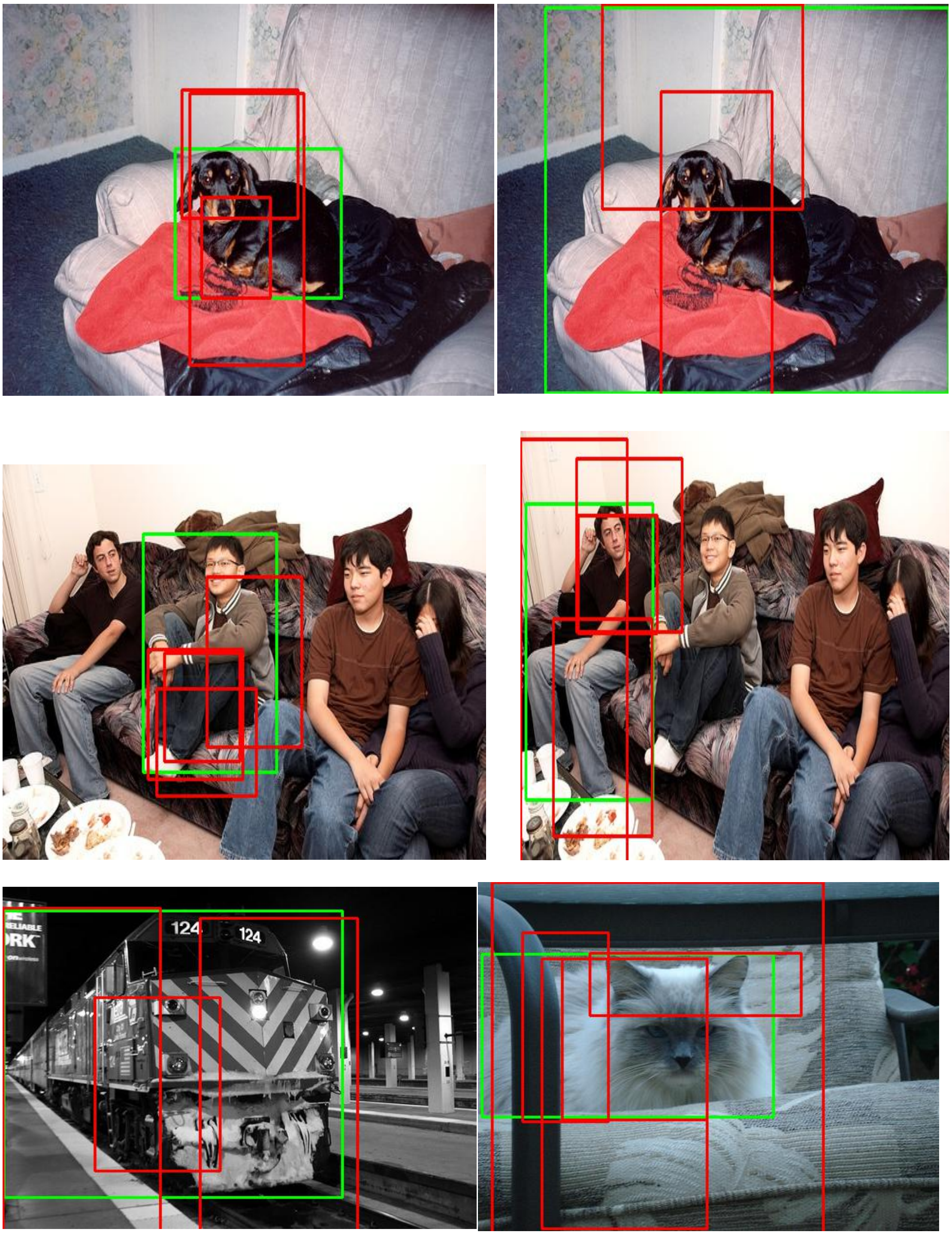
The possible locations at which objects can be detected are extracted from each image in the training database. Next, for each annotated image the ground truth of object locations are added to the positives of the corresponding object class. For each positive location, we take all those locations which have 20-50% overlap with it as negatives for that object class.

To avoid near-duplicate negative examples, a negative example is excluded if it has
more than 70 % overlap with another negative.

All negatives are clubbed into a separate class ‘Background’ which .To keep the number of initial negatives less, we randomly drop half of the negatives.

For each positive and negative locations, features are extracted.

Figure 1: Here are the object ground truths and corresponding negatives (show as red and green boxes respectively) for the training data



3.3 Dimensionality Reduction(HOG-PCA)

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components (or sometimes, principal modes of variation).

The number of principal components is less than or equal to the smaller of the number of original variables or the number of observations.

This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors are an uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variables.

If we collect HOG feature cells from a large corpus of images and use PCA to analyze them, it turns out that almost all of the energy lies in the first 100 PCA components. This suggests that HOG36 features could be mapped to an 100 dimensional feature space with little loss of discriminative power.

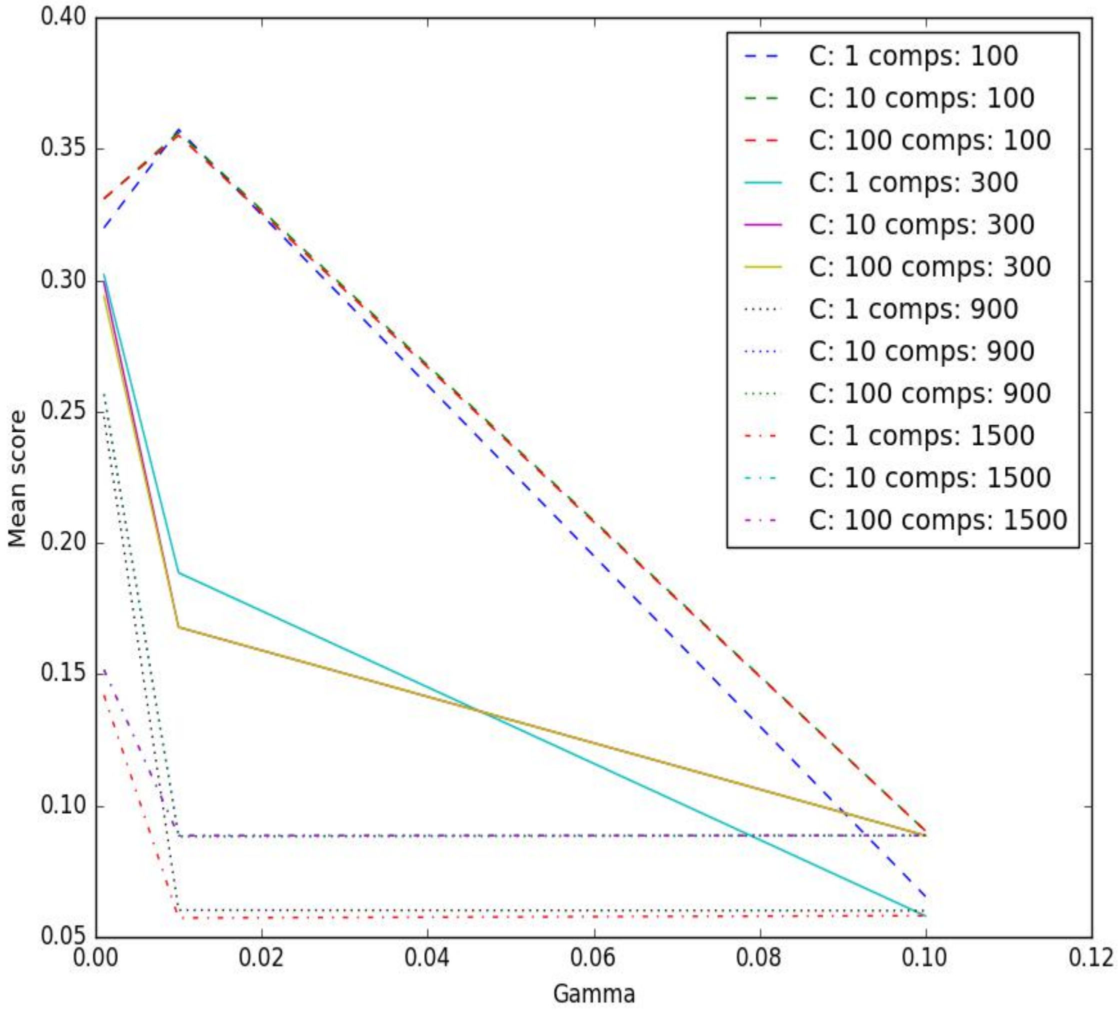


Figure 2: Tuning Plot

3.3 Training an Object Detector

A SVM classifier is trained with the HOG feature vectors as input. A suitable kernel is chosen and its parameters are tuned.

A RBF kernel was selected as feature dimension was small as compared to the training samples.

We used sklearn.svm to implement multi-class classification. The parameters gamma of radial basis function and the number of components were tuned with the help of 4-fold cross validation.

Gamma was plotted against accuracy as shown in **Figure 2** to find the optimal gamma, C and the number of components(comps) for dimensionality reduction (HOG-PCA). Using optimal gamma SVM model is trained.

3.4 Testing

For the test set, the final model is applied to all locations generated by our selective search on test images. The windows are sorted by classifier score while windows which have more than 30 % overlap with a higher scoring window are considered near-duplicates and are removed. Next, threasholding is done so that only those detections with a classifier score of more than 60% are taken into account.

4 Experiments and Results

The following dataset was used for experimentation purpose:

4.1 The PASCAL Visual Object Challenge Datasets

The PASCAL VOC datasets are important visual recognition benchmarks. The VOC has been run annually since 2005 with the aim of establishing the best performing methods and advancing the state of the art in image classification, object detection and semantic segmentation. There are twenty object classes:

- person
- bird, cat, cow, dog, horse, sheep
- aeroplane, bicycle, boat, bus, car, motorbike, train
- bottle, chair, dining table, potted plant, sofa, tv/monitor

Each dataset contains a training/validation subset ‘trainval’ and a test subset ‘test’. Only the provided trainval dataset is used for training.

4.2 Evaluation

The classification task is stated as follows:
Given bounding boxes of each object in a test image predict the class of the object(among one of the twenty classes). The classification task is judged by the precision recall and accuracy obtained using the bounding box annotation for the test images.

The detection task is stated as follows:
For each of the twenty classes predict the bounding boxes of each object of that class in a test image (if any). Each bounding box should be output with an associated real-valued confidence of the detection so that a precision/recall curve can be drawn.

The detection task is judged by the precision/recall curve. The principal quantitative measure used will be the average precision (AP). Detections are considered true or false positives based on the area of overlap with ground truth bounding boxes. To be considered a correct detection, the area of overlap ao between the predicted bounding box Bp and ground truth

bounding box Bgt must exceed 50% by the formula:

$$ao = area(Bp \cap Bgt)/area(Bp \cup Bgt)$$

4.3 Classification Results

An accuracy of 35.25% was achieved.

Table 1: The precision and recall are are stated in the following classificatin report:

Class	precision	recall	f1-score	support
Sheep	0.27	0.50	0.35	138
Horse	0.39	0.53	0.45	136
Bottle	0.37	0.49	0.42	99
Bicycle	0.12	0.27	0.17	64
Cow	0.22	0.41	0.29	160
Sofa	0.17	0.29	0.22	218
Dog	0.42	0.48	0.45	86
Bus	0.16	0.29	0.21	144
Cat	0.79	0.24	0.37	1479
Person	0.47	0.52	0.50	115
Train	0.32	0.37	0.34	86
Boat	0.56	0.50	0.53	88

Aeroplane	0.74	0.57	0.65	372
Car	0.20	0.24	0.22	162
Pottedplant	0.68	0.75	0.72	102
Tvmonitor	0.54	0.30	0.39	399
Chair	0.15	0.24	0.19	143
Bird	0.36	0.37	0.36	106
Diningtable	0.22	0.42	0.29	118
Motorbike	0.00	0.00	0.00	0
avg / total	0.53	0.35	0.38	4306

Table 2: The confusion matrix is shown as follows:

Predicted →

Actual ↓

Confusion Matrix

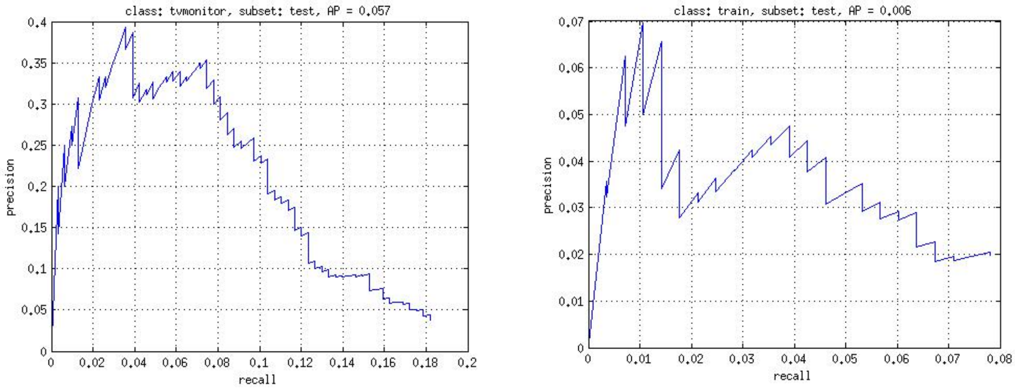
24	15	1	0	11	2	22	0	6	0	0	0	0	0	1	0	0	4	1	2	2
6	69	2	1	5	4	16	0	5	7	1	0	0	0	6	0	1	7	0	6	2
1	6	72	4	2	2	6	0	9	4	1	1	0	1	3	0	3	5	2	12	2
2	0	0	49	0	3	3	0	2	8	1	0	0	7	1	2	4	2	1	4	10
7	11	0	0	17	1	14	1	1	2	0	0	0	0	1	0	2	0	1	0	6
5	3	2	1	3	65	13	3	10	4	1	5	0	6	4	3	9	3	8	3	9
21	18	8	0	18	17	64	0	25	13	1	0	1	1	1	0	2	6	2	10	10
0	0	2	2	0	1	1	41	0	1	19	3	1	5	1	0	6	0	0	0	3
6	8	4	2	4	14	21	0	42	8	1	1	1	1	3	2	1	4	4	10	7
38	74	42	52	53	80	138	2	96	360	7	18	5	17	87	7	45	124	13	91	130
4	5	4	0	1	5	3	11	2	0	60	2	0	1	1	2	0	0	5	3	6
1	1	3	0	1	5	0	2	5	3	4	32	4	4	8	0	0	3	2	1	7
0	3	1	0	0	1	1	3	2	1	2	11	44	2	4	0	2	4	0	2	5
2	2	5	4	2	22	6	17	7	8	6	11	3	213	13	3	15	8	5	5	15
7	9	5	2	3	12	7	1	10	12	4	3	3	4	39	1	2	9	6	10	13
0	0	1	0	2	2	1	1	1	2	3	1	0	1	0	77	2	1	0	1	6
5	12	15	15	9	32	15	12	11	11	9	10	8	21	9	14	121	10	19	8	33
8	6	3	1	5	12	23	0	18	4	1	2	6	1	4	1	0	35	1	3	9

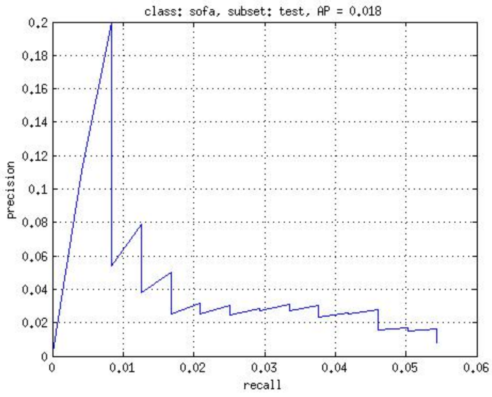
5	1	1	0	0	10	7	3	5	3	5	1	2	2	6	1	6	3	39	3	3
4	8	12	1	2	2	10	1	7	4	1	0	0	0	7	0	2	4	0	49	4
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

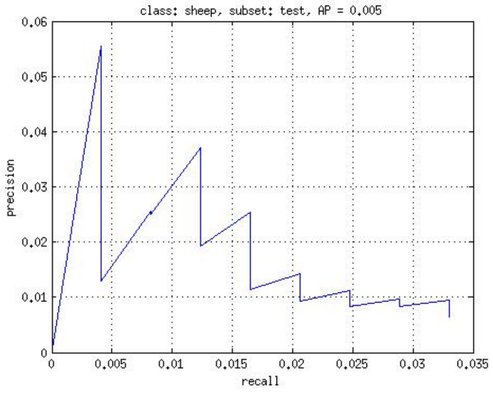
4.4 Delineation and Recognition

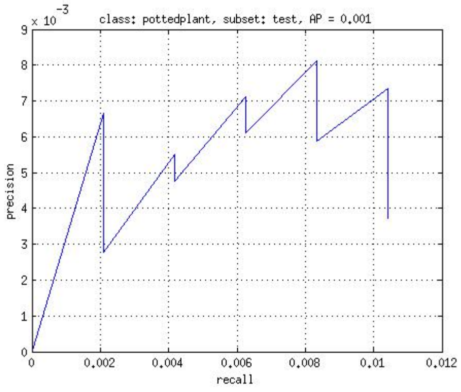
The average precision obtained on the PASCAL VOC 2007 dataset is about 0.02. As the database is highly complex, various other methods have average precision in the range of 0.05 to 0.2. Still the current system needs improvement which is stated in the section 5.

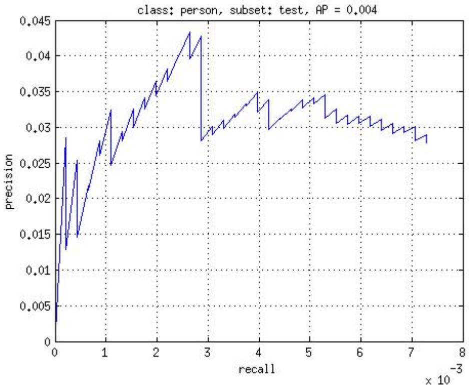
Figure 3: The precision recall graphs obtained are plotted as follows

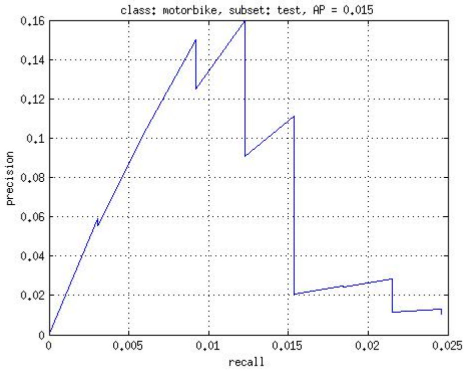


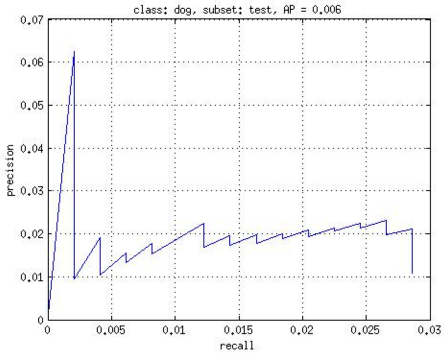

class: sofa, subset: test, AP = 0.018
precision
recall
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
0.01
0.02
0.03
0.04
0.05
0.06

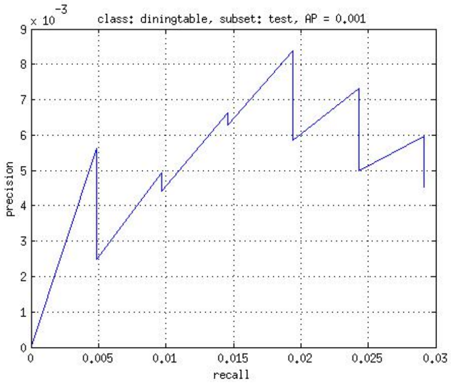

class: sheep, subset: test, AP = 0.005
precision
recall
0
0.01
0.02
0.03
0.04
0.05
0.06
0.005
0.01
0.015
0.02
0.025
0.03
0.035

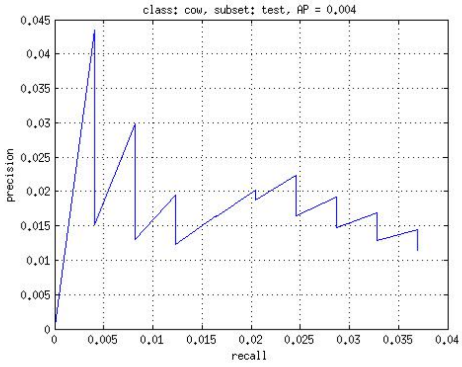

class: pottedplant, subset: test, AP = 0.001
x 10
-3
precision
recall
0
1
2
3
4
5
6
7
8
9
0.002
0.004
0.006
0.008
0.01
0.012

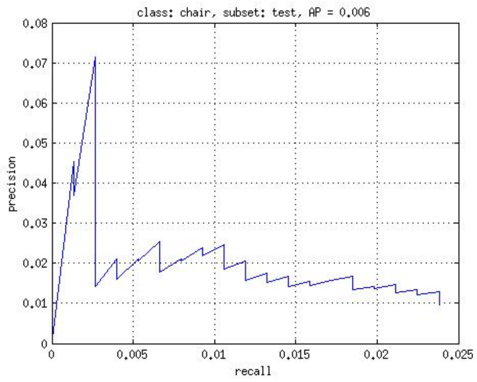

class: person, subset: test, AP = 0.004
precision
recall
x 10
-3
0
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0.04
0.045
1
2
3
4
5
6
7
8

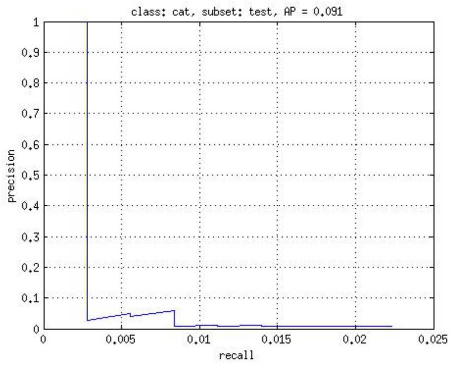

class: motorbike, subset: test, AP = 0.015
precision
recall
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.005
0.01
0.015
0.02
0.025

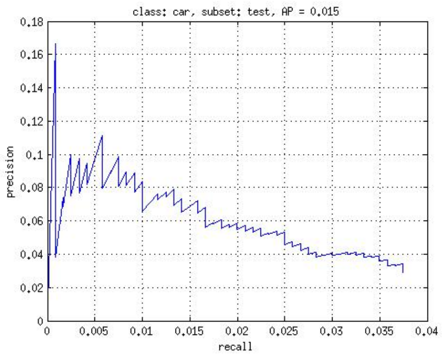

class: dog, subset: test, AP = 0.006
precision
recall
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.005
0.01
0.015
0.02
0.025
0.03

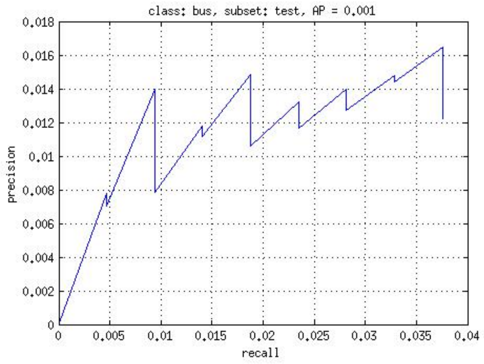

class: diningtable, subset: test, AP = 0.001
x 10
-3
precision
recall
0
1
2
3
4
5
6
7
8
9
0.005
0.01
0.015
0.02
0.025
0.03

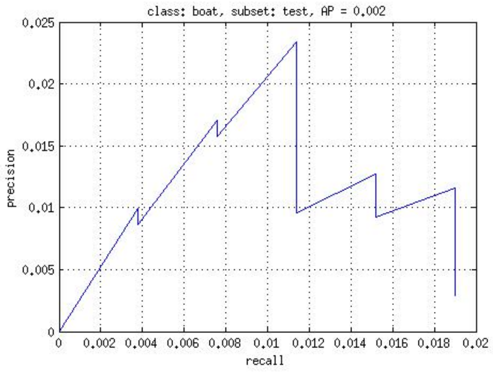

class: cow, subset: test, AP = 0.004
precision
recall
0
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0.04
0.045
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0.04

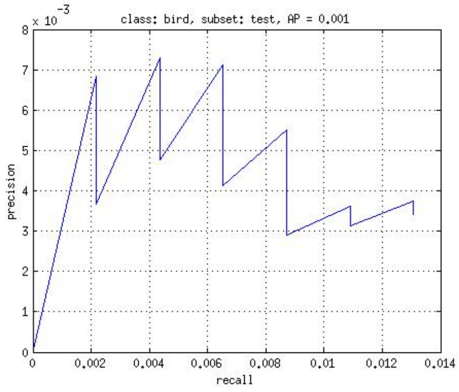

class: chair, subset: test, AP = 0.006
precision
recall
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.005
0.01
0.015
0.02
0.025

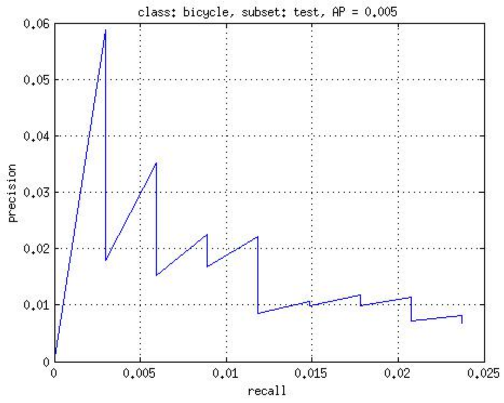

class: cat, subset: test, AP = 0.091
precision
recall
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.005
0.01
0.015
0.02
0.025


class: car, subset: test, AP = 0.015
precision
recall
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0.04


class: bus, subset: test, AP = 0.001
precision
recall
0
0.002
0.004
0.006
0.008
0.01
0.012
0.014
0.016
0.018
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0.04


class: boat, subset: test, AP = 0.002
precision
recall
0
0.005
0.01
0.015
0.02
0.025
0.002
0.004
0.006
0.008
0.01
0.012
0.014
0.016
0.018
0.02


class: bird, subset: test, AP = 0.001
x 10^-3
precision
recall
0
1
2
3
4
5
6
7
8
0.002
0.004
0.006
0.008
0.01
0.012
0.014


class: bicycle, subset: test, AP = 0.005
precision
recall
0
0.01
0.02
0.03
0.04
0.05
0.06
0.005
0.01
0.015
0.02
0.025

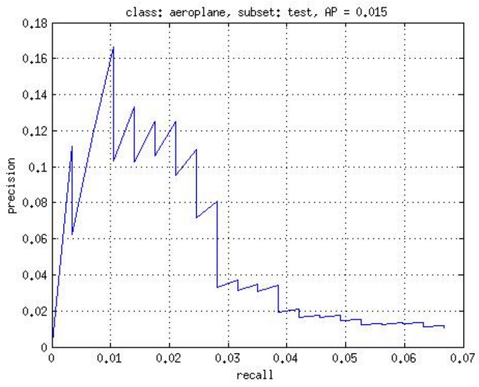

class: aeroplane, subset: test, AP = 0.015
precision
recall
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.01
0.02
0.03
0.04
0.05
0.06
0.07

Table: Detection on PASCAL VOC dataset

Class Name	Average Precision
Aeroplane	0.0152
Bicycle	0.0053
Bird	0.0007
Boat	0.0021
Bottle	0.0455
Bus	0.0015
Car	0.0152
Cat	0.0909
Chair	0.0065
Cow	0.0040
Diningtable	0.0080
Dog	0.0057
Horse	0.0202
Motorbike	0.0145
Person	0.0039
Pottedplant	0.0007
Sheep	0.0051
Sofa	0.0182
Train	0.0063
Tvmonitor	0.0572

5 Future Work

The current object detection system can be improved by using additional features like texture features (LBP) or colour features.

Latent Support Vector Machines with iterative addition hard negatives using a retraining phase Can also be used.

Post-Processing technique of Non-Maximum Suppression for best bounding box prediction can be used to improve accuracy.

The current state of the art approach of Deep Learning can also be explored to find relevant feature descriptors.

References

[1] Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. International Journal of Computer Vision, 59,167–181.

[2]J. R. R. Uijlings · K. E. A. van de Sande ·T. Gevers · A. W. M. Smeulders Selective Search for Object Recognition

[3]Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach

towardfeature space analysis. IEEE Transactions on Pattern Analysis andMachine Intelligence, 24, 603–619.

[4]Lampert, C. H., Blaschko, M. B., & Hofmann, T. (2009). Efficient subwindow search: A branch and bound framework for object

localization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31, 2129–2142.

[5]Everingham, M., Gool, L. V., Williams, C., Winn,

J., & Zisserman, A. (2011). The Pascal visual object classes challenge workshop: Overview and results of the detection challenge.

[6] Cormen, T.H., Leiserson, C.E., and Rivest, R.L. 1990. Introduction to Algorithms. The MIT Press: McGraw-Hill Book Company.

[7]P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, USA. 2008. pages 25, 29, 93, 94, 95, 103, 104

[8]S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In Proceedings of the Neural Information and Processing Systems, Vancouver, Canada, 15:561–568, 2002. pages 25