OBJECT CATEGORIZATION USING COLLECTIONS OF PARTS AND SECOND ORDER POOLING FEATURES

BY

PRITAM PURUSHOTHAMA SUKUMAR

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2013

Urbana, Illinois

Adviser:

Assistant Professor Derek W. Hoiem

ABSTRACT

This thesis presents an investigation of the Collection of Parts Model for object categorization. Multiclass categorization is performed using the Collections of Parts model. Results using Support Vector Machines, L_1 Logistic Regression and Boosted Decision Trees are presented and discussed. Methods to analyze confusion in these results are developed and results are presented. The Collections of Parts model is augmented with features from features generated by Second Order Pooling resulting in a significant improvement in performance.

To my mother.

ACKNOWLEDGMENTS

I would like to first thank my advisor Derek Hoiem. I am grateful that he took a chance on me, gave me a very interesting project to work on and was patient with my learning curve. This thesis would not have been possible if he had not taken the time to discuss areas and answer my - sometimes silly - questions with clarity and an amazing knowledge of the field. I am in debt to my labmates Ian Endres and Kevin Shih for providing me with the part scores that set this thesis into motion. They were always being ready to help with code, discussions and answers to questions about their work (and mine!). It is on their shoulders and work that this thesis stands and looks (not too far) ahead.

I am grateful to my family who have been there to scold me, praise me and advise me at every turn of the road. Their unflinching support for everything I do, even when they disagree strongly, never ceases to amaze me.

I am thankful also to my former advisor, Michael Selig. It is under him that I learned how to do research and how to approach problems critically. He has taught me a great deal.

Special thanks go out to Sophie, who has been a partner in many memories over the last two years. And to Neha, who was always available to talk and provide a light when things were dark. I am lucky to have good friends, old and new, who have added noise and color to my grayscale life.

I am very thankful to the staff in The Department of Computer Science and Aerospace Engineering. Their efficiency and ability to deal with all my unique issues is amazing. Without their help, my ride would have been much, much rougher.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Object recognition refers to the identification of objects in images and videos. This area includes the challenging problems of localizing the objects, separating them from background, and identifying object properties including but not limited to category, pose, occlusion, size, shape and attributes.

In this thesis, we explore a subset of the above algorithms - those that deal with identifying categories in an image. The algorithms are prone to confusion between similar categories, such as animals and vehicles. Multiclass classification is performed using state-of-the-art object detection algorithms and performance with a focus on confusion is analyzed. Tools are developed to analyze confusion in the results of classification.

1.1 Background

The main computational difficulty in object recognition is the fact that to be useful, a detection algorithm needs to account for large intra-category variations shape, size, texture, occlusion and other properties [1]. Features need to be engineered very carefully to account for variations in these properties within categories, while maintaining enough specificity to distinguish between categories. For example, cats and dogs have large intra-class variations due to breed, size, shape and texture.

Ideally, features need to be chosen to ensure that all relevant information is captured from an image without any extraneous information. The size of the features also needs to be kept to a minimum to reduce computational expense. Object recognition research has focused on multiple approaches to feature engineering. Images are commonly represented in terms of color, position of objects, gradients and histograms of gradients (HOG), texture and shape.

Color is used a predictor of material and, to some extent, lighting as well. Texture is commonly used to differentiate between regions of different labeling in images. HOG features, initially used for pedestrian detection [2] are now used extensively for detection of other categories. HOG features are robust to changes in lighting and can be made to be robust to changes in position as well.

Based on the above descriptions of images, features are combined to generate an overall feature vector for the image. The features are also generally evaluated over a multi-level spatial pyramid in order to ensure that local variations in the features are not lost.

In recent years, parts-based models [3, 4] have been used for object recognition, where objects are represented as collections of parts, which are either deformable [5] or without constraints on structure [3]. Parts are generated by pooling over candidate object regions and choosing high scoring regions. Scores are based on overlap with ground truth and HOG features over spatial pyramids. Parts trained in this manner have the advantage that they can be used in a variety of applications, including sharing of parts across categories. Boosting can be used to improve performance of these algorithms as well.

There are many sources of error in object detection algorithms: occlusion, poor localization, variations in scale, shape and pose within a category, confusion with background and confusion with semantically similar categories [6]. In particular, the research by Hoiem et al. [6] concluded that sensitivity to scale, localization errors and confusion with similar objects are the leading reasons for error in object detectors. They suggested that further analysis is required in the analysis of these aspects of object recognition. In Chapter 3, results are presented for confusion with similar objects.

1.2 Contributions

This thesis explores the performance of object recognition algorithms with a focus on analysis of confusion resulting from similarities between different categories of objects in parts-based models. We use the results of the model developed by Endres et al. [3]. The parts generated by the model are used to generate features for a multi-class object detector with good accuracy on the Pascal 2010 data set [7]. The performance of different machine learning

algorithms including Support Vector Machines (SVMs), L_1 Logistic Regreession (L_1 LR) and Boosted Decision Trees (BDT) is compared.

A new method is developed to identify confused categories using the above algorithms. Further analysis is performed to investigate whether the reasons for confusion and results on the most confused categories is presented.

Second Order Pooling, a recent approach to feature generation and incorporation of region properties into the feature vector, is introduced briefly. Results using Collections of Parts, Second Order Pooling and a combined feature vector with part scores and Second Order Pooling scores are presented and compared.

1.3 Image Dataset

The PASCAL VOC 2010 [7] dataset was used for all analysis and to generate the results presented in this report. The dataset consists of a total of 10103 images, split into approximately two equal sets for training (**train** set with 4998 images) and validation (**val** set with 5105 images).

For the classification task, which is what is explored in this report, each image is provided with annotations marking bounding boxes for the objects. One image can contain multiple, overlapping object windows. In the dataset, there are 23374 annotated objects, split as 11577 images for training and 11797 for validation. These numbers are only for objects marked as non-difficult, which are the only object windows studied in this report.

1.4 Organization

The rest of this thesis is organized as follows:

- In Chapter 2, the generation of features for multi-class object detection from collections of parts is presented and discussed. Different algorithms are used to evaluate performance using the resulting features.

- In Chapter 3, the methods used to evaluate confusion are presented, along with results

on the categories that are most confused with each other. Further analysis is performed to dig deeper into the reasons for confusion, including identification of parts that responsible for confusion.

- In Chapter 4, a recent feature extraction method - second order pooling - is briefly introduced. A multiclass classifier is trained using second order pooling and results are presented. The collection of parts scores are augmented with second order pooling scores and the resulting performance is presented and discussed.

- Chapter 5 concludes the thesis with a brief summary.

CHAPTER 2

CLASSIFICATION BASED ON COLLECTIONS OF PARTS

In this thesis, the recently developed Collection of Boosted Parts model [3] is used. The model provides a collection of part detectors for a single category that can then be used in a variety of different contexts, including object detection.

Parts are generated for each image in the following fashion: For each category, candidate parts are generated from randomly chosen exemplar images. A subset of these parts is chosen that exhibit good coverage over the training set, following which each part is refined by searching through the training set and detecting matches in other training images. Parts are trained and scored based on a linear classifier over HOG features. Further details about the part detection scheme can be found in [3].

A set of 40 parts is then selected for each image category. For each object window in the training set, candidate locations for each part are stored in the form of bounding boxes and scores. Note that the above are trained on individual categories. For example, cat parts are scored based on similarity with cat images and dissimilarity with other categories.

Results from the work of Endres et al.[3] were provided by the author. Each category had independent part scores for every image. The part scores were scores of candidate part regions over the image. Bounding box information and information about the direction the part is facing were provided as well.

2.1 Generation of Features

Features were engineered from the part scores so as to include information about every part from every category for each object window. The following procedure was developed: for a given object window, all parts from all categories were considered in turn. For each part, all

the bounding boxes were compared with the ground truth and bounding boxes with good overlap were chosen as candidate parts. Good overlap is indicated by an overlap of greater than 80% with the ground truth bounding box, i.e. more than 80% of the part bounding box lies within the object window. If multiple bounding boxes had good overlap with the ground truth, the box with the highest score was chosen. In case no bounding box had good overlap with the ground truth, a low value (-10) was assigned to the score to indicate that no detections corresponding to the part for the current object window. When features were scaled, the missing detections were replaced with the mean of the feature. The high scoring parts with good overlap were then concatenated to form one feature vector with 800 elements, with each of the twenty categories in the PASCAL dataset contributing 40 parts. A summary of this algorithm is given in Algorithm 1.

```
Data: Part scores from Work of Endres [3], part scores per-image
for Each image do
    for Each object window in the image do
        for Each category do
            Load part score data
            for Each part do
                Find all parts with overlap > 80 % Store bounding box and score of
                highest scoring part
            end
        end
        Concatenate all part scores from all categories to form single feature vector
    end
end
```

Algorithm 1: Generation of features

Thus, the feature vector generated by Algorithm 1 for every object window contained information about how well the object window scored against each of the parts of each of the PASCAL categories. This was necessary in order to use these parts in a multi-class classifier capable of differentiating between the twenty object categories. Once the feature vector was generated, it was scaled to have zero mean.

With the above algorithm, the features were generated in about 40 minutes for 11577 object windows.

2.2 Classification

The features generated above were used to train a classifier using multiple learning algorithms. The algorithms and the packages used are discussed briefly and their performance in terms of speed and accuracy are compared in this section.

The algorithms chosen were:

1. L_1 Logistic Regression (L_1 LR): The *l1_logreg* package from Stanford was used to perform logistic regression with L_1 loss [8]. The features were normalized such that their mean was zero. The package is capable only of two-class classification. The One-vs-all method was implemented to enable multiclass classification using the package.

2. Support Vector Machines (SVMs): The package LIBSVM [9] was used to train and classify using SVMs. The default kernel (RBF) is used throughout this thesis. The RBF kernel is well suited to nonlinear classification problems and can handle large non-sparse features. The package uses one-vs-one classification by default for multiclass classification.

3. Boosted Decision Trees [10]: Code developed by Derek Hoiem was used to implement the boosted decision trees algorithm. The logistic regression version of AdaBoost (ABDT) is used. The code implements the one-vs-all method for multiclass classification.

The results from the algorithms are summarized in Table 2.1. The L_1 LR training time denoted includes the time for converting MATLAB data into a format readable by the package. The majority of the training time is thus disk access time. This could be shortened significantly by working more efficiently with the package. However, this was not explored in this thesis.

The mean ROC curve (averaged over the one-vs-all ROC curves for all the results of classification on the training set) for the case of ABDT classification is shown in Figure 2.1 for illustration. It can be seen from the shape of the curve and the area that the classifier performs very well. However, it can also be seen that the area under the curve suggests a

Algorithm	Training time	Testing time	Accuracy (Train/Val)
SVM (RBF kernel)	$\approx$ 16 min	153.6 s	55.3 % / 50.1 %
ABDT	$\approx$ 52 min	0.89 s	78.7 % / 57.6 %
L_1 LR	$\approx$ 130 min*	1.87 s	68.5 % / 59.3 %

Table 2.1: Summary of classification results using Collection of Parts model

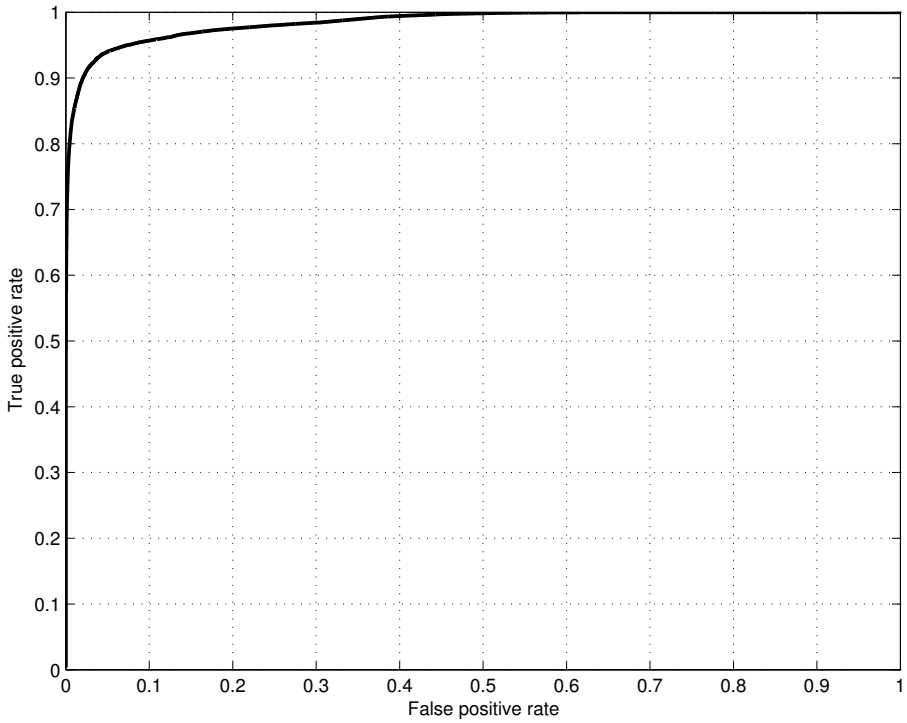

Figure 2.1: Mean ROC curve for classification using Boosted Decision Trees (Area under curve = 0.9850)

much higher performance than the results presented in Table 2.1. This can be caused by the fact that the recall of these algorithms is very high since for each example, the negative examples significantly outnumber the positive examples.

As can be seen from the table, SVM performs the worst out of the three algorithms. The performance of ABDT and L_1 LR is similar on the test, with ABDT outperforming L_1 LR on the test set.

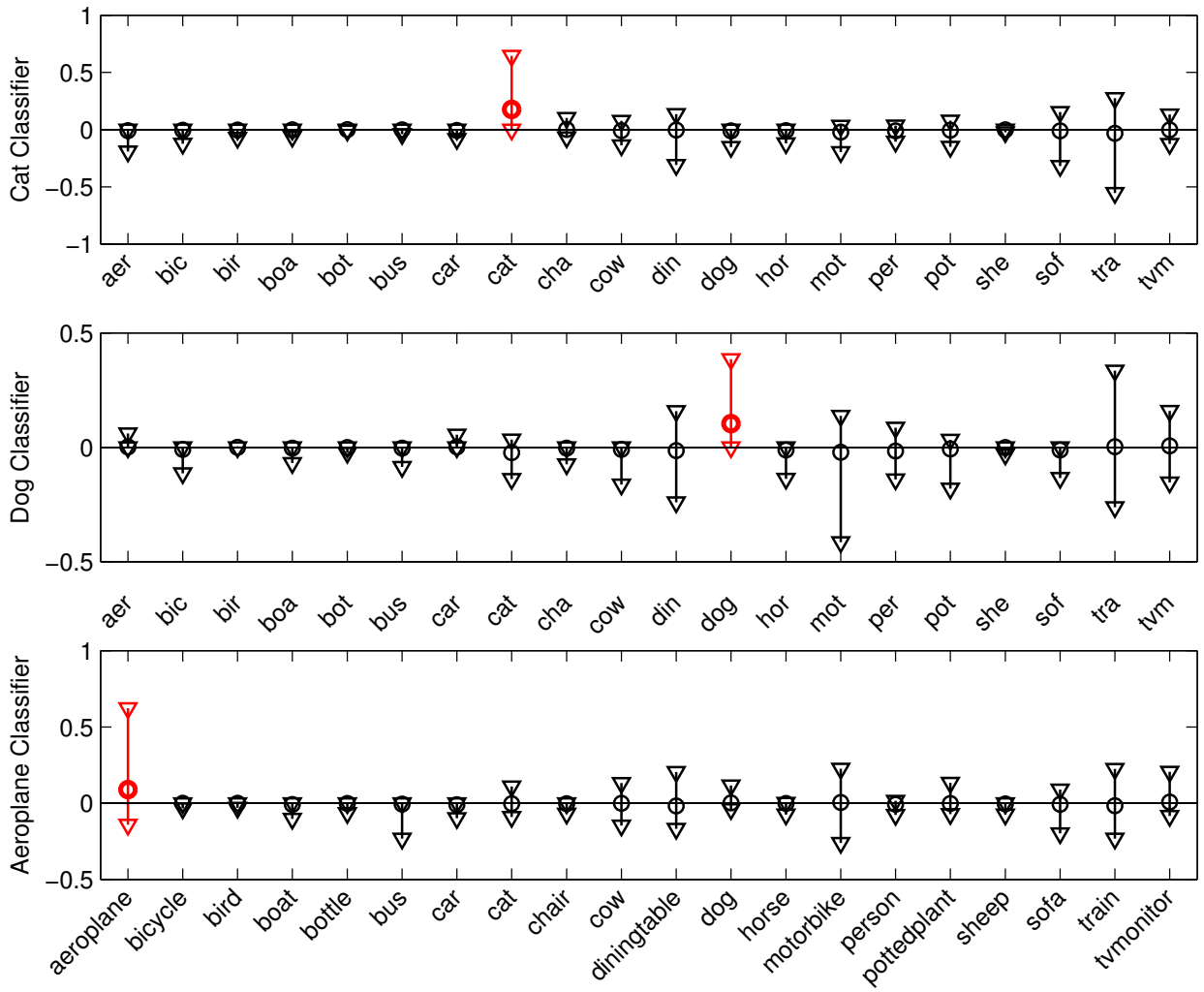

Figure 2.2: Weight vectors for L_1 classifier for Cat and Dog classes

2.3 Contribution of Parts

L_1 logistic regression provides as output, a weight vector with a weight corresponding to every element of the feature vector. This makes it especially useful to analyze the contribution of parts. As mentioned in Sec 2.2, the one-vs-all method of classification was used. A separate classifier was obtained for each class, Each containing a weight vector, with each weight corresponding to one of the 800 features.

Thus, each weight element corresponded to a specific part score. The influence of the parts was analyzed in order to investigate whether for each class, the correct parts were being chosen as important. It was found that the correct parts were being prioritized (i.e had a

positive contribution to the confidence scores) for the classifier. This shows that the parts are effective in classification. Figure 2.2 is an illustration of the part contributions, shown for three different classes. The triangles correspond to the minimum and maximum of the weights of the 40 parts corresponding to a class. The circles correspond to the mean of the weights. The parts corresponding to the classifier are shown in red.

It can be seen from Fig. 2.2 that each one-vs-all classifier assigns much higher weights to the parts of its own class than other class. The means of the weights also that the score of a particular one-vs-all classifier is based more on the object window having a high score corresponding to that category than negative scores in other categories. This shows that the parts are effective in distinguishing between categories.

CHAPTER 3

ANALYSIS OF CONFUSION

One of the main goals of this is thesis is to investigate confusion and identify commonly confused categories using the Collection of Parts features. In Section 3.1, a representative confusion matrix is presented and discussed. Section 3.3

3.1 Confusion Matrix

The confusion matrix for the multi-class Collection of Parts classifier is presented in Table 3.1. As can be seen, the diagonal elements are the strongest in each case. Values greater than 5 % are shown in bold. As can be seen, there is significant confusion between categories. For example, looking at Row 8 (Cat), it can be seen that there seem to be significant confusion with the classes Dog and Person.

The confusion matrix was not used to generate an ordered list of confused categories, since the confusion matrix does not account for imbalance in the dataset. For example, the PASCAL VOC 2010 dataset is heavily imbalanced towards the Person category with about half of the total number of objects belonging to that category. The effect of this imbalance can be seen in the fact that all the values in column corresponding to Person are very high. The top five categories account for over 60 % of the images in the dataset. To overcome this issue, Confusion Receiver Operating Characteristic (ROC) curves were used to rank confusion. The methodology followed to generate these curves and use them to rank confusion is discussed in the following section.

	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Diningtable	Dog	Horse	Motorbike	Person	Pottedplant	Sheep	Sofa	Train	Tvmonitor
Aeroplane	**63.7**	0.0	1.1	3.0	0.0	0.0	**5.4**	0.0	1.1	0.0	0.8	0.0	0.3	1.4	**18.2**	0.3	0.0	0.0	4.3	0.5
Bicycle	0.0	**50.8**	0.3	1.0	0.3	0.0	1.3	0.6	3.2	0.3	1.9	1.3	1.3	**5.5**	**28.5**	2.3	0.0	0.3	0.6	0.3
Bird	1.6	0.0	**20.2**	1.0	0.4	0.0	3.9	4.5	2.7	0.8	0.2	**7.6**	1.0	1.6	**50.3**	1.4	1.0	0.2	0.6	0.6
Boat	**7.6**	0.3	0.6	**29.2**	0.0	0.9	**17.0**	0.0	2.9	0.0	0.9	0.0	0.0	0.6	**33.0**	0.9	0.3	0.3	4.4	1.2
Bottle	0.0	0.4	0.2	0.6	**14.6**	0.4	2.2	0.4	**8.9**	0.0	1.0	1.2	0.0	1.0	**63.5**	1.4	0.2	0.6	1.2	2.4
Bus	1.2	0.0	0.0	0.4	0.4	**58.1**	**13.0**	0.0	**7.9**	0.0	1.2	0.0	0.0	0.0	**9.9**	0.4	0.0	0.0	3.6	4.0
Car	0.9	0.3	0.3	0.8	0.2	0.5	**56.9**	0.3	4.3	0.0	0.2	0.2	0.1	0.9	**30.6**	0.6	0.3	0.3	1.2	0.8
Cat	0.0	0.0	1.1	0.2	0.0	0.0	0.7	**65.2**	1.9	0.4	0.2	**11.1**	0.2	0.2	**15.3**	0.5	0.7	1.2	0.5	0.7
Chair	0.5	0.5	0.3	0.3	0.6	0.0	3.8	1.3	**46.3**	0.2	3.0	0.8	0.3	0.4	**32.0**	2.2	0.3	2.0	0.7	4.2
Cow	0.0	0.0	2.1	0.0	0.0	0.0	1.3	2.5	1.3	**26.3**	0.0	**13.1**	**6.4**	0.4	**36.9**	0.0	**8.5**	0.0	1.3	0.0
Diningtable	0.4	0.9	0.4	1.3	0.4	0.0	4.7	0.4	**17.1**	0.4	**41.5**	1.7	0.0	0.4	**21.8**	0.9	0.9	3.8	0.9	2.1
Dog	0.1	0.1	1.7	0.0	0.7	0.0	1.4	**13.4**	1.4	1.3	0.1	**45.0**	1.6	0.7	**29.3**	0.3	1.6	1.0	0.3	0.0
Horse	0.6	0.0	0.6	0.3	0.0	0.0	1.0	2.2	1.3	1.9	0.0	**8.9**	**46.3**	0.6	**31.1**	0.0	4.1	0.0	1.0	0.0
Motorbike	0.3	3.6	1.0	0.3	1.0	0.3	**7.2**	0.7	0.7	0.0	1.6	1.3	0.3	**51.1**	**27.2**	2.3	0.7	0.0	0.3	0.0
Person	0.0	0.2	0.5	0.0	0.2	0.0	0.8	0.8	2.7	0.1	0.1	1.1	0.3	0.6	**91.0**	0.7	0.1	0.3	0.2	0.2
Pottedplant	0.5	1.0	1.0	0.2	0.7	0.0	1.7	1.5	**5.3**	0.2	0.7	2.7	0.7	1.5	**50.4**	**28.6**	1.0	0.2	1.7	0.5
Sheep	0.0	0.0	0.3	0.3	0.0	0.0	1.7	3.4	0.8	**5.9**	0.0	**12.6**	2.8	0.3	**33.1**	1.7	**36.1**	0.0	0.6	0.6
Sofa	0.0	0.0	0.0	1.8	0.4	0.4	4.0	**6.2**	**11.5**	0.4	1.3	2.6	0.9	0.9	**27.8**	2.2	0.4	**36.1**	1.3	1.8
Train	0.4	0.0	1.1	2.7	0.0	3.4	3.4	0.4	3.8	0.0	0.0	0.0	0.8	1.5	**14.1**	1.1	0.0	0.8	**64.3**	2.3
Tvmonitor	0.6	0.0	0.0	0.3	0.3	0.9	3.5	0.3	**11.7**	0.0	0.6	0.6	0.0	0.3	**22.3**	0.9	0.0	0.0	2.1	**55.7**

Table 3.1: Confusion matrix for multi-class classifier (values shown in percentages)

3.2 Confusion ROC Curves

Receiver Operating Characteristic (ROC) curves are commonly used to evaluate performance of machine learning algorithms. The curve is a plot of the false positive rate versus the true positive rate and hence it is immune to dataset imbalance. The false positive rate and true positive rate are generated based on the probability of an example corresponding to a particular class.

For the purposes of evaluating, we are interested in relating the performance of the algorithm on confused categories. To do that, a modified formulation of the ROC curve (Confusion ROC curve) was defined and used.

A confusion ROC curve is defined for two classes (say Class 1 and Class 2) and measures the confusion between the two classes. Only images that misclassify Class 1 as Class 2 are considered to plot the confusion ROC curve. For each image, the confidence values used to generate the ROC curve are the differences between the confidence that the object is in Class 2 and the confidence that the object is in Class 1.

The area of the confusion ROC curve is thus a measure of confusion, and is invariant to dataset imbalance. Note that for 20 classes, there will be 190 curves, since the area under confusion ROC curve [Class 1, Class 2] is the same as the area under the confusion ROC curve for [Class 2, Class 1].

The areas of all the confusion ROC curves is shown in Table 3.2.

	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Diningtable	Dog	Horse	Motorbike	Person	Pottedplant	Sheep	Sofa	Train	Tvmonitor
Aeroplane	-	0.98	0.94	0.90	0.97	0.98	0.94	0.99	0.96	0.98	0.96	0.99	0.98	0.97	0.96	0.97	0.99	0.98	0.95	0.98
Bicycle	0.98	-	0.94	0.96	0.92	0.99	0.97	0.99	0.92	0.96	0.94	0.98	0.96	0.89	0.92	0.91	0.98	0.98	0.98	0.98
Bird	0.94	0.94	-	0.91	0.88	0.98	0.92	0.92	0.91	0.83	0.95	0.85	0.90	0.90	0.86	0.87	0.89	0.94	0.96	0.96
Boat	0.90	0.96	0.91	-	0.93	0.94	0.85	0.98	0.93	0.96	0.94	0.98	0.96	0.95	0.93	0.93	0.97	0.94	0.89	0.95
Bottle	0.97	0.92	0.88	0.93	-	0.95	0.94	0.97	0.84	0.93	0.93	0.95	0.95	0.93	0.84	0.86	0.94	0.96	0.94	0.91
Bus	0.98	0.99	0.98	0.94	0.95	-	0.91	0.99	0.92	0.99	0.97	0.99	0.99	0.98	0.97	0.97	0.99	0.98	0.93	0.95
Car	0.94	0.97	0.92	0.85	0.94	0.91	-	0.99	0.92	0.95	0.94	0.98	0.97	0.93	0.94	0.93	0.96	0.94	0.93	0.94
Cat	0.99	0.99	0.92	0.98	0.97	0.99	0.99	-	0.96	0.95	0.97	0.86	0.97	0.98	0.96	0.96	0.94	0.95	0.99	0.99
Chair	0.96	0.92	0.91	0.93	0.84	0.92	0.92	0.96	-	0.92	0.84	0.97	0.95	0.94	0.91	0.89	0.95	0.88	0.95	0.86
Cow	0.98	0.96	0.83	0.96	0.93	0.99	0.95	0.95	0.92	-	0.96	0.87	0.85	0.94	0.91	0.93	0.79	0.95	0.97	0.98
Diningtable	0.96	0.94	0.95	0.94	0.93	0.97	0.94	0.97	0.84	0.96	-	0.97	0.98	0.94	0.96	0.92	0.97	0.92	0.96	0.95
Dog	0.99	0.98	0.85	0.98	0.95	0.99	0.98	0.86	0.97	0.87	0.97	-	0.91	0.97	0.92	0.95	0.87	0.93	0.99	0.98
Horse	0.98	0.96	0.90	0.96	0.95	0.99	0.97	0.97	0.95	0.85	0.98	0.91	-	0.96	0.91	0.94	0.90	0.96	0.98	0.99
Motorbike	0.97	0.89	0.90	0.95	0.93	0.98	0.93	0.98	0.94	0.94	0.94	0.97	0.96	-	0.93	0.88	0.97	0.96	0.97	0.98
Person	0.96	0.92	0.86	0.94	0.84	0.97	0.94	0.96	0.91	0.91	0.96	0.92	0.91	0.93	-	0.88	0.93	0.90	0.97	0.94
Pottedplant	0.97	0.91	0.87	0.93	0.86	0.97	0.93	0.96	0.89	0.93	0.92	0.95	0.94	0.88	0.88	-	0.92	0.93	0.95	0.95
Sheep	0.99	0.98	0.89	0.97	0.94	0.99	0.96	0.94	0.95	0.79	0.97	0.87	0.90	0.97	0.93	0.92	-	0.97	0.98	0.98
Sofa	0.98	0.98	0.94	0.94	0.96	0.98	0.94	0.95	0.88	0.95	0.92	0.93	0.96	0.96	0.90	0.93	0.97	-	0.97	0.97
Train	0.95	0.98	0.96	0.89	0.94	0.93	0.93	0.99	0.95	0.97	0.96	0.99	0.98	0.97	0.97	0.95	0.98	0.97	-	0.96
Tvmonitor	0.98	0.98	0.96	0.95	0.91	0.95	0.94	0.99	0.86	0.98	0.95	0.98	0.99	0.98	0.94	0.95	0.98	0.97	0.96	-

Table 3.2: Confusion ROC areas

3.3 Confused Categories

Using the confusion ROC curves (generted based on L_1 LR results), pairs of categories were ranked in decreasing order of confusion, defined as the area of the confusion ROC curve. The most confused categories (top 5) are shown in Table 3.3. It is interesting to note that that the animals are mostly confused with each other, as are the vehicles.

The results of the confusion indicate that the algorithm is confusing categories that are similar to each other in terms of natural relation. For example, animals are commonly confused with each other (see for example, rows Cat, Dog and Sheep in Table 3.3), vehicles are commonly confused with each other (see for example, rows Aeroplane and Car) and furniture categories are commonly confused among themselves as well (see for example, rows Sofa, Chair and Diningtable).

However, there are some surprising results in Table 3.3. For example, Cat and Sofa have high confusion with each other. It is suspected that this confusion occurs from the fact that cats commonly occur on sofas and that their textures are similar. Also, Person and Bottle have very high confusion. This might stem from the similar shape (when scale isn't accounted for) between the two categories. Also, bottles commonly occur in images with people.

Category	Top 5 confused categories
Aeroplane	Boat, Train, Bus, Car, Motorbike
Bicycle	Motorbike, Pottedplant, Diningtable, Horse, Chair
Bird	Dog, Cat, Sheep, Cow, Aeroplane
Boat	Aeroplane, Train, Bus, Car, Diningtable
Bottle	Tvmonitor, Pottedplant, Chair, Bus, Person
Bus	Train, Tvmonitor, Car, Aeroplane, Boat
Car	Bus, Boat, Aeroplane, Motorbike, Train
Cat	Dog, Sofa, Sheep, Bird, Cow
Chair	Tvmonitor, Diningtable, Sofa, Bottle, Pottedplant
Cow	Sheep, Horse, Dog, Cat, Bird
Diningtable	Chair, Sofa, Bicycle, Train, Motorbike
Dog	Cat, Sheep, Cow, Horse, Bird
Horse	Cow, Sheep, Dog, Cat, Bicycle
Motorbike	Bicycle, Pottedplant, Dog, Diningtable, Bird
Person	Bottle, Bird, Pottedplant, Horse, Dog
Pottedplant	Bicycle, Motorbike, Bottle, Sheep, Cat
Sheep	Cow, Horse, Dog, Cat, Bird
Sofa	Diningtable, Cat, Chair, Aeroplane, Dog
Train	Bus, Boat, Aeroplane, Diningtable, Tvmonitor
Tvmonitor	Bus, Train, Chair, Bottle, Boat

Table 3.3: Top 5 confused categories each category of the PASCAL VOC 2010 dataset

3.4 Confused Object Windows

For each pair of categories, the analysis was further extended to investigate which images were most confused. For example, let us consider the two categories Cat and Dog. For each object window, the confusion is defined as the difference between the Dog confidence and the Cat confidence. Thus an object window is said to have a greater confusion if its Dog confidence is much greater than its Cat confidence.

The object windows are then ranked in order of decreasing confusion, thus giving us a list of the most confused images for a particular pair of categories. The part scores for the Cat and Dog parts in each of these windows is also extracted and sorted in descending order, thus providing us with the highest scoring parts in each of the confused categories. For illustrative purposes, the top six confused Cat-vs-Dog images are shown in Fig. 3.1. These results are based on probabilities resulting from classification using ABDT. The confidence

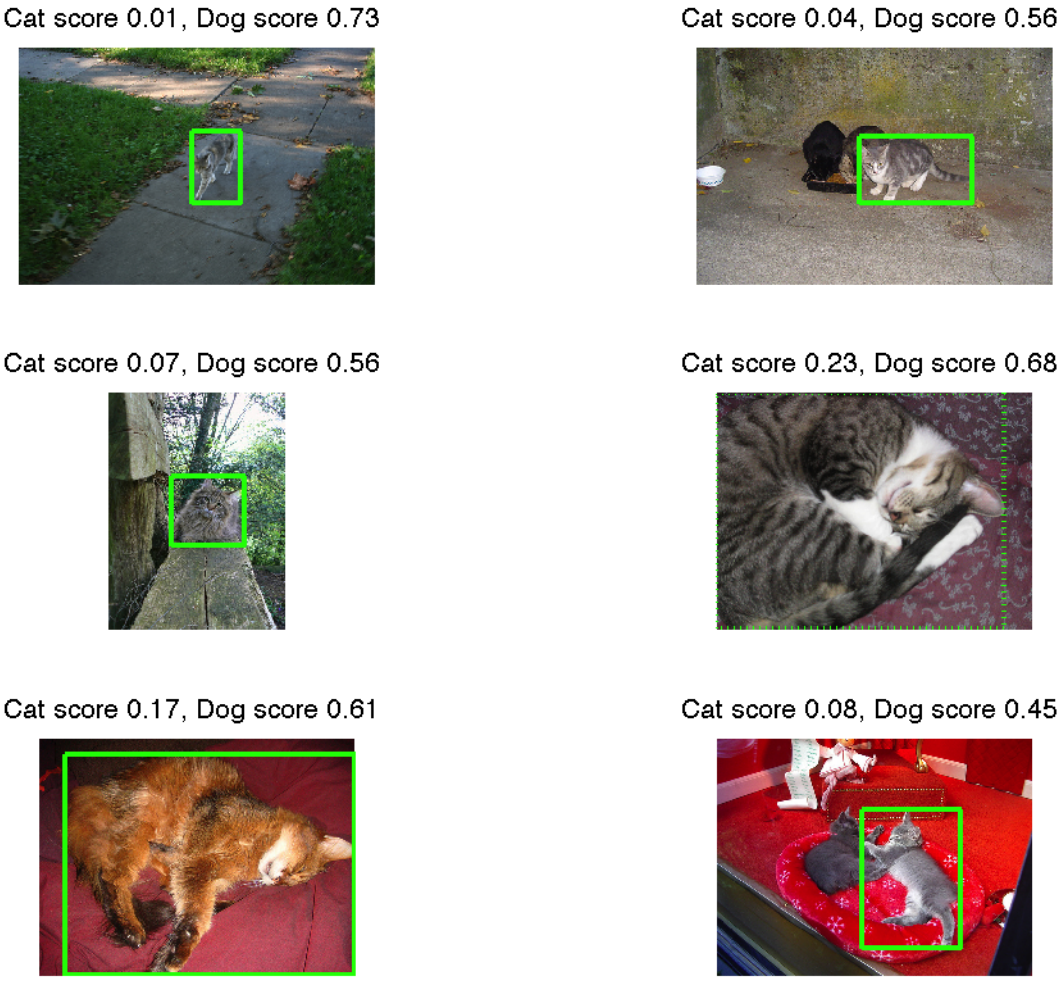

Figure 3.1: Top confused Cat-vs-Dog images

scores for the Cat and Dog classifiers are included as well for illustration.

From the images, it can be seen that the highest confusion results from images in which the Dog detector scores very highly while the cat detector scores very low, thus implying that the confusion in fact results from the cat detector not firing on these images.

CHAPTER 4

SECOND ORDER POOLING

Second order pooling (O2P) is used to augment a feature-based classifier with second order information collected over the image region [11]. Local features are extracted for the whole image, following which a pooling stage pools these features over regions (generally levels of a spatial pyramid) to generate descriptors for each region. Local features are extracted over the whole image and mapped into a higher dimensional space, which associates each feature with elements in a predefined codebook (for example, defined using Fisher encoding). Pooling is then used to produce a summary of the coded features inside each local region. The summary is a column vector that can then be used as a feature in classification algorithms.

Up to the work of Carreira et al.[11], pooling was done using first order statistics (Average or Max Pooling). However, in the aforementioned work, the authors introduced second order pooling, where they extended Average and Max Pooling. Instead of taking pooling over the coded features, the pooling is performed over the outer products of the raw features themselves ($\mathbf{xx^T}$ where $\mathbf{x}$ is the raw feature vector). This allows the pooling process to capture the correlations between every pair of local features, and the coding stage is completely bypassed.

The work of Carreira et al. [11] indicates that second order pooling is very effective for the tasks of segmentation and classification. In this thesis, O2P is used over SIFT features on the object window to generate feature vectors that contain information on the second order statistics of the features over the window. The features are pooled over a specified number of levels of a spatial pyramid and then concatenated to form a long feature vector.

In this chapter, the use of second order pooling to augment the Collection of Parts classifier for multi-class classification over the PASCAL VOC 2010 dataset is investigated. First, results using only second order pooling for the task of classification on the PASCAL VOC

Algorithm	Training time	Testing time	Accuracy (Train/Val)
SVM (RBF kernel)	$\approx$ 7 hours	$\approx$ 1 hour	65.0 % / 59.3 %
ABDT	$\approx$ 10 hours	$\approx$ 3 s	76.5 % / 52.3 %

Table 4.1: Summary of classification results with O2P features

2010 dataset are presented, following which the feature vector with part scores is augmented with second order pooling scores. Augmentation is done using a one-vs-all classifer for each of the twenty classes and also one-vs-one classifiers over the top confused classes. The effect of second order pooling on confusion is discussed.

4.1 Classification Using Second Order Pooling

Second order pooling features were generated for object windows only and a multiclass classifier was trained using ABDT and SVMs. Only one level of spatial pyramid was used to speed up training time. Each feature vector contained 8256 elements if one level of the pyramid was used. The results are summarized in Table 4.1. From the table, it can be seen that second order pooling provides accuracy comparable to the Collection of Parts model. Note that since the feature vectors are large and not sparse, L_1 logistic regression does not perform well. Hence, results of L_1 LR are not included here.

Feature generation is quick and takes about 0.1 s per image with one level of spatial pyramid. However, because of the size of the feature vectors (8256 for one level of spatial pyramid, $\approx$ 1.7 million for three levels), training and testing take an order of magnitude longer than for the collection of parts model.

4.2 Augmentation of Collection of Parts Model

The results in the previous section indicate that classification performance would be improved by augmenting the collection of parts scores with scores from second order pooling. That possibility is explored in this section.

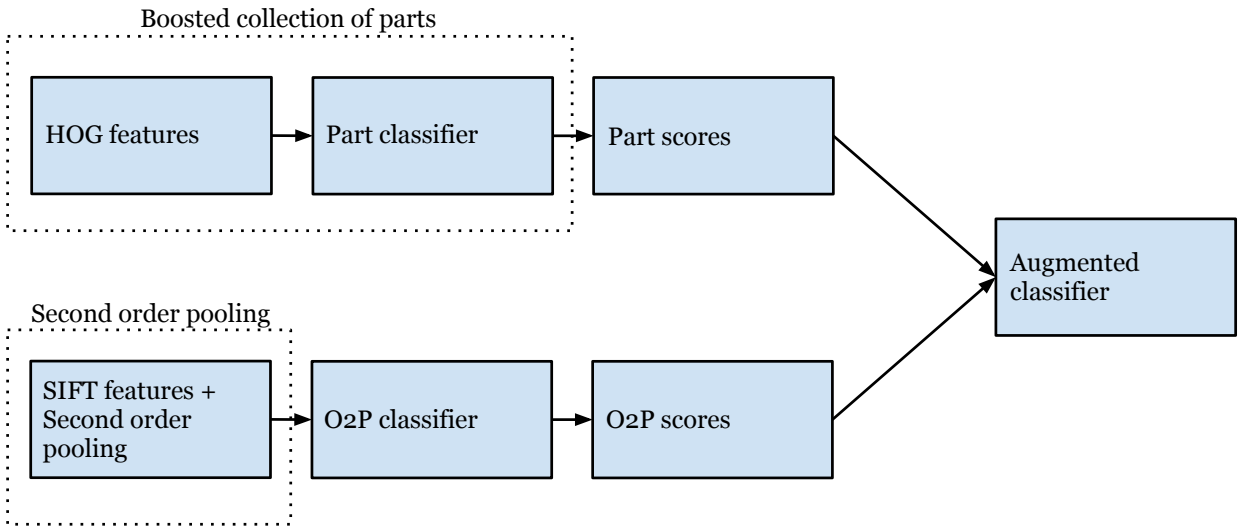

Figure 4.1: Flowchart illustrating augmentation of features

As discussed in Sec 2.2, the part scores were generated by linear classifiers over Histogram of Gradient (HOG) features. Similarly, second order pooling category (O2P) scores are generated by linear classifiers over pooled SIFT features. For each object window, O2P scores are generated for each of the twenty one-vs-all classifiers. ABDT was used to generate the second order pooling scores. Each feature vector consists of:

1. 800 part scores, each corresponding to the highest scoring parts with good overlap (based on linear classifiers over HOG features [3]) from each of the 40 parts for each of the 20 PASCAL VOC classes as described in Section 2.1.

2. 20 O2P scores, each corresponding the confidence (based on second order pooling over SIFT features [11]) that an object window belongs to one of the 20 PASCAL VOC classes.

A flowchart for the augmentation methodology is shown in Figure 4.1.

The O2P scores for the training set were generated using a three-way split in the training data. Scores for each three-way split of the data were generated by training a classifier on the other two-thirds. Thus, scores were generated for the whole of the training set. These scores were then used to train the overall classifier over the newly generated feature vectors with 820 elements. This process was used to avoid overfitting on the training data.

Algorithm	Training time	Testing time	Accuracy (Train/Val)
SVM	$\approx$ 17 min	$\approx$ 4 min	86.0 % / 63.9 %
ABDT	$\approx$ 140 min	$\approx$ 2 s	81.2 % / 61.9 %
L_1 LR	$\approx$ 130 min	$\approx$ 2 s	70.9 % / 64.0 %

Table 4.2: Summary of classification results with augmented features

Features	Best Performance (Train/Val)
Collections of Parts (CP)	78.7 % (ABDT) / 59.3 % (L_1 LR)
Second Order Pooling (O2P)	76.5 % (ABDT) / 59.3 % (SVM)
CP + O2P	86.0 % (SVM) / 64.0 % (L_1 LR)

Table 4.3: Summary of classification results with augmented features

Thus, each feature vector contains a total of 820 elements. Classification was performed using ADBT and SVMs and the results are summarized in Table 4.2. The change in performance is summarized in Table 4.3 and it can be seen that using augmented features results in a significant improvement in performance. This improvement comes from the fact that in addition to the part scores, which describe the probability of the window containing an object (as the sum of its parts), the algorithm now also has a a global statistical description of the region from the second order pooling process. Thus, the part scores provide part-by-part (local) information while second order pooling provides global information about the object window. It can be seen that augmenting scores leads to a significant increase in performance.

The confusion matrix for the results using ABDT is shown in Table 4.4. Comparing with Table 3.1, it can be seen that the major confused categories, for the most part, are the same, implying that second order pooling features result in an overall improvement in performance. The improvement does not result from a decrease in category-specific confusion.

	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Diningtable	Dog	Horse	Motorbike	Person	Pottedplant	Sheep	Sofa	Train	Tvmonitor
Aeroplane	**71.5**	0.3	3.5	**6.8**	0.5	0.5	**6.8**	0.5	1.6	0.5	0.5	0.5	0.5	0.5	1.4	0.3	0.0	1.1	1.9	0.5
Bicycle	0.3	**52.1**	1.0	1.0	1.3	0.0	1.9	0.3	3.6	0.3	2.3	0.6	3.2	**5.8**	**20.7**	2.6	1.3	0.6	1.0	0.0
Bird	**5.4**	0.8	**26.6**	3.5	0.8	0.0	**7.2**	3.3	4.1	1.2	0.0	**10.1**	1.9	0.6	**28.7**	2.9	1.9	0.2	0.2	0.6
Boat	**13.2**	0.6	1.8	**43.3**	0.0	1.8	**17.0**	0.6	5.0	0.3	1.2	0.6	0.9	0.0	**5.6**	3.8	0.0	1.2	3.2	0.3
Bottle	0.6	0.0	1.8	0.6	**37.1**	0.4	2.4	0.4	**5.1**	0.2	0.2	2.2	0.0	0.6	**45.0**	1.6	0.2	0.2	0.8	0.8
Bus	2.4	0.0	0.4	1.6	1.6	**61.7**	**12.6**	0.0	**9.1**	0.4	0.8	0.0	0.0	0.4	2.8	0.0	0.0	0.0	4.3	2.0
Car	2.3	0.3	0.5	2.6	0.6	1.5	**76.4**	0.2	4.6	0.1	0.2	0.5	0.1	0.8	**5.2**	0.9	0.3	0.9	1.4	0.6
Cat	0.9	0.0	4.9	0.2	0.4	0.0	0.4	**55.0**	2.3	0.7	0.7	**16.9**	1.6	0.2	**10.9**	1.2	1.9	1.4	0.4	0.2
Chair	0.6	1.0	1.2	1.2	1.9	0.5	3.4	1.4	**53.4**	0.2	2.8	1.6	0.7	0.8	**17.5**	1.8	0.7	3.0	1.1	5.3
Cow	0.0	0.0	**6.4**	0.4	0.4	0.0	2.5	2.5	1.7	**23.7**	0.0	**14.0**	**10.2**	0.0	**23.7**	1.3	**12.7**	0.4	0.0	0.0
Diningtable	2.6	2.1	1.7	3.8	1.3	0.4	**5.6**	2.1	**15.8**	0.4	**35.0**	1.7	1.7	1.3	**12.8**	3.0	1.7	3.0	2.6	1.3
Dog	0.6	0.1	**5.8**	0.1	0.3	0.0	1.0	**11.3**	1.1	2.7	0.0	**48.5**	3.7	0.8	**15.7**	1.3	**5.6**	1.1	0.0	0.3
Horse	0.6	0.3	2.9	0.6	0.0	0.0	1.3	2.5	2.2	**5.7**	0.0	**9.5**	**49.2**	1.9	**14.3**	1.6	**5.7**	1.0	0.3	0.3
Motorbike	1.6	4.6	1.6	0.0	1.0	0.0	**6.6**	0.3	1.3	0.3	2.6	0.7	1.6	**49.8**	**24.3**	2.0	0.7	0.3	0.7	0.0
Person	0.3	0.5	2.1	0.4	1.2	0.0	1.2	0.9	2.2	0.3	0.3	2.1	1.1	0.6	**84.7**	1.2	0.4	0.5	0.0	0.1
Pottedplant	0.7	1.9	2.2	1.5	3.6	0.2	3.6	1.0	**7.0**	0.0	0.7	4.6	1.2	1.2	**25.2**	**41.2**	1.5	1.0	0.5	1.2
Sheep	0.0	0.0	0.8	0.0	0.0	0.0	**5.3**	3.9	0.8	**7.3**	0.0	**11.2**	**5.6**	0.0	**16.0**	1.7	**46.8**	0.3	0.0	0.3
Sofa	0.9	0.9	2.6	3.5	0.4	0.4	4.4	**6.6**	**14.5**	0.4	3.5	3.5	1.8	1.8	**13.2**	2.2	3.1	**34.4**	0.4	1.3
Train	0.4	0.0	0.4	3.0	1.1	3.4	**7.6**	0.0	4.9	0.8	0.4	0.0	1.1	0.4	**7.2**	0.8	0.8	1.1	**64.6**	1.9
Tvmonitor	0.3	0.0	0.3	1.2	1.2	0.9	2.6	0.9	**15.8**	0.0	0.6	0.3	0.3	0.0	**8.5**	1.8	0.3	0.6	0.3	**64.2**

Table 4.4: Confusion matrix for Augmented Features (values shown in percentages)

CHAPTER 5

CONCLUSIONS

This thesis explored different aspects of multiclass classification using Collections of Parts.

Following a brief introduction and a discussion of relevant background in object recognition, results for multiclass classification using features generated by concatenating parts from parts of different categories were presented. Performance of three different learning algorithms (Support Vector Machines, Boosted Decision Trees and L_1 Logistic Regression) were presented and compared.

Confusion in the context of multiclass classification was analyzed. A confusion matrix was presented following which Confusion ROC curves were developed and used to sort pairs of categories in order of confusion. Results of confused categories on the PASCAL VOC 2010 dataset were presented and discussed. A procedure to obtain the top confused images for each pair of confused categories was presented and discussed.

The collection of Parts model was augmented using a recently developed method - Second Order Pooling. Second order pooling scores for images were generated and concatenated with part scores, which resulted in a significant improvement in object detection performance.

REFERENCES

[1] M. Riesenhuber and T. Poggio, "Models of Object Recognition," *Nature Neuroscience*, vol. 3, pp. 1199–1204, 2000.

[2] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005*, vol. 1, 2005, pp. 886–893.

[3] I. Endres, "Expanding the Breadth and Detail of Object Recognition," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2013.

[4] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial Structures for Object Recognition," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, Jan. 2005.

[5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-based Models." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–45, Sep. 2010.

[6] D. Hoiem, Y. Chodpathumwan, and Q. Dai, "Diagnosing error in object detectors," *European Conference on Computer Vision*, 2012.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results."

[8] K. Koh, S. Kim, and S. Boyd, "An Interior-Point Method for Large-Scale L_1 Logistic Regression," *Journal of Machine Learning Research*, vol. 8, pp. 1519–1555, 2007.

[9] R. Fan, P. Chen, and C. Lin, "Working Set Selection Using Second Order Information for Training Support Vector Machines," *The Journal of Machine Learning Research*, vol. 6, pp. 1889–1918, 2005.

[10] Y. Freund and R. Schapire, "Experiments with a New Boosting Algorithm," *International Conference on Machine Learning*, 1996.

[11] J. Carreira and R. Caseiro, "Semantic Segmentation with Second-order Pooling," *European Conference on Computer Vision*, no. 1, 2012.