UNIVERSITY OF CALIFORNIA, SAN DIEGO

Vehicle Classification and Identification of Salient Information in Images

A Thesis submitted in partial satisfaction of the requirements for the degree Master of Science

in

Electrical Engineering (Intelligent Systems, Robotics, and Control)

by

Dalar Vartanians

Committee in charge:

Professor Massimo Franceschetti, Chair
Professor Truong Nguyen
Professor Zhuowen Tu

2016

The Thesis of Dalar Vartanians is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2016

DEDICATION

To my parents, Vachik and Anahid and my fiancé, Arno.

EPIGRAPH

“To acquire knowledge, one must study; but to acquire wisdom, one must observe.”

- Marilyn vos Savant

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and for most, I would like to express my gratitude to my advisor Prof. Massimo Franceschetti for the continuous support of my studies and research, for believing in me, for his patience, and for giving me the freedom to explore different directions. I could have not imagined having a better advisor for my graduate studies.

I would also like to thank my fellow lab mates, J, Hamed and Mohammad Javad for their constructive comments, encouragement and for all the fun we had on 4th floor of Atkinson Hall which earned us the title of “happiest group in Atkinson Hall.” Meanwhile, I should thank Lorenzo Coviello for recommending me for the great opportunity of working with Prof. Franceschetti.

And, surely, I must thank my parents, Vachik and Anahid, for immigrating to the United States, leaving behind all they had worked very hard for to promise me a brighter future and better opportunities. I must specifically thank my mother who not only promoted the value of education ever since I was a child, but also set the best example of dedication and perseverance for achieving what holds value. Words can never express how grateful I am for having them. I can only promise to follow their legacy, dream big, work hard and contribute to my community.

Last but not least, I would like to thank my fiancé, Arno, who walked all the steps next to me in my educational journey. I thank him for always believing in me, for encouraging me to push the boundaries and never giving up on me even when I was not at my best. I could not ask for a better partner in life.

ABSTRACT OF THE THESIS

Vehicle Classification and Identification of Salient Information in Images

By

Dalar Vartanians

Master of Science in Electrical Engineering

(Intelligent Systems, Robotics, and Control)

University of California, San Diego, 2016

Professor Massimo Franceschetti, Chair

Vehicle classification is currently a widely implemented component in intelligent vehicles, surveillance systems, and traffic monitoring. The major component of vehicle classification is to learn what feature in the images of vehicle provides the most effective information, which distinguishes different models. In this work, the study of two different famous feature extraction mechanisms and three classifiers is carefully conducted to provide comparison and analysis. The next important component of this work is the investigation of the effect of viewing angle and lighting conditions on the performance of

the classifier. The latter is inspired by previous studies on face-recognition systems with different lighting conditions and poses [1].

Chapter 1

Introduction

1.1 Background

Vehicle recognition has become popular for many applications especially in traffic control and surveillance systems. While vehicle classification into general categories such as sedan versus truck may not be a very challenging task, recognizing more specific differences such as make and model of vehicles (jeep93 versus jeep99) which is known as fine-grained recognition is more difficult and interesting to pursue [2]. Having a rich dataset available which includes images of 10 vehicles with different make and models under different viewing angles and lighting conditions, the performance of three well-known learning methods, namely Softmax Regression, Linear Kernel Support Vector Machine and Adaptive Boosting, using features extracted from Convolutional Neural Networks (CNNs) and Histogram of Oriented Gradients (HOG) are explored. A comparison of the performance of the learning methods in terms of accuracy using CNNs and HOG features is presented.

Another important aspect of the work is the investigation of the role of lighting conditions and camera angles on the performance of the classifier. The latter is motivated by the results of edge detection of vehicles and achieved by conducting the occlusion and dome experiments. The occlusion experiment is inspired by studying previous work in the literature for the investigation of sensitivity to occlusion [3]. The Dome experiment is performed to illustrate the results of the occlusion experiment in a compact manner utilizing the information available in the dataset regarding the location of camera and light source for a given test image.

The main contributions of this work are:

- Investigation of different combinations of features and learning algorithms on the vehicle classification. This provides a comparison of the state of the art feature extraction and learning methods while achieving excellent classification performance in terms of accuracy for the vehicle classification task.
- Experiments on edge detection, occlusion and camera angle that measure the robustness of the network to diverse lighting and viewing angle changes. The outcome of the experiments illustrates the robustness of a given viewing angle and lighting condition for a camera set up. These experiment also provides validation in terms of robustness of the network to missing a portion of information within images which occurs frequently in practical and real-life situations. For instance, in traffic monitoring very frequently a vehicle is partially covered with another vehicle or object on the road. The occlusion experiment mimics this effect and enables the evaluation of the network in terms of robustness to such instances.

Chapter 2

Feature Extraction and Classification

2.1 Feature Extraction

In machine learning, feature extraction is referred to extraction of certain values from an initial set of measurements which in these work are images. The extracted values are called features. The features are supposed to be informative and non-repetitive. Features which have the mentioned qualities are capable of producing high-accuracy results with classifiers. In addition, in many cases, such features facilitate better human interpretation of data. While traditionally features were hand engineered, now there exists methods for automatic learning of the features. Convolutional Neural Nets is an example of such method that automatically learns the most informative features.

2.1.1 Convolutional Neural Nets

CNNs do not need substantial pre-processing of raw data relative to other mechanism for feature extraction. This means that the network learns the required filters that should be applied to the raw data for feature extraction that in traditional machine learning algorithms were hand-engineered [22]. The reduced human effort in identifying and extracting informative features make the CNNs a very powerful tool in machine learning tasks. CNN is also desirable as with CNN one has an end to end mechanism for both feature extraction and classification. Classification will be discussed in later chapters.

Another appealing characteristic of CNNs is the use of fixed weight in convolutional layers. This implies that a fixed filter is used for each pixel in the layer. This approach is efficient in terms of memory usage and it has pleasantly proved to result in high accuracy of classification [4] [22]. In this work, both CNNs feature extraction and classification are conducted in Caffe deep learning framework [5]. The CNN feature extraction steps are illustrated in Figure 2.1 for better visualization [21]. Figure 2.2 illustrates a simple CNN filter applied to an image. The 0 and 1 in red values are the values that should be learned by CNN. Note that these values do not have to be binary, 0 or 1 in general [21]. Figure 2.3 illustrates the pooling (sub-sampling) step [21]. This step again aims at deducing the dimension of features to speed up the learning without loss of information. The example in figure, illustrates "Max-pooling" which simply extracts the maximum value from a subset of pixels in the image. There exist "average-pooling" and other pooling formulations as well. However, in this work max-pooling was used.

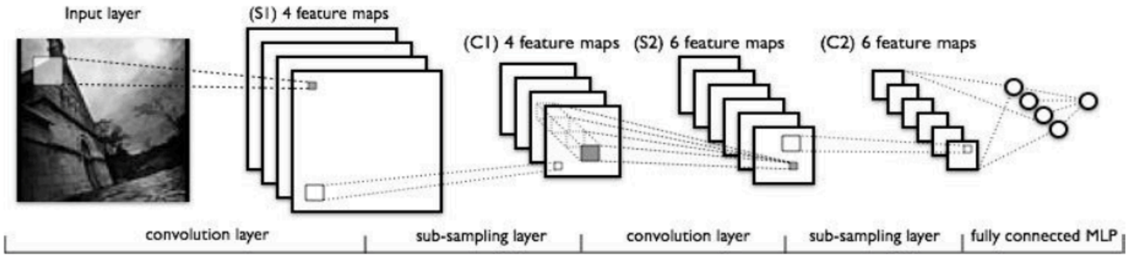
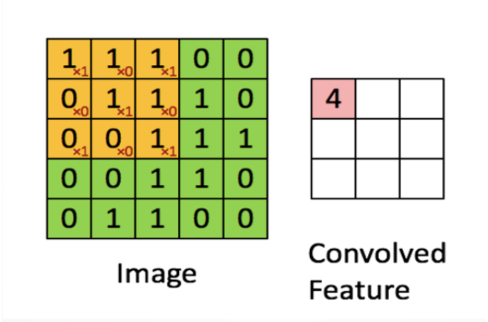


Figure 2.1: CNN Feature extraction pipeline



$1_{\times 1}$	$1_{\times 0}$	$1_{\times 1}$	0	0
$0_{\times 0}$	$1_{\times 1}$	$1_{\times 0}$	1	0
$0_{\times 1}$	$0_{\times 0}$	$1_{\times 1}$	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved Feature

Figure 2.2: A simple CNN filter illustration

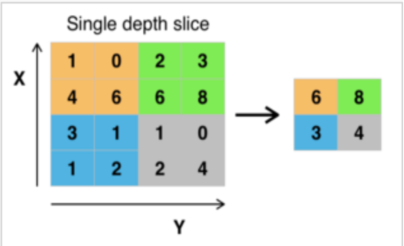


Figure 2.3: Pooling (sub-sampling)

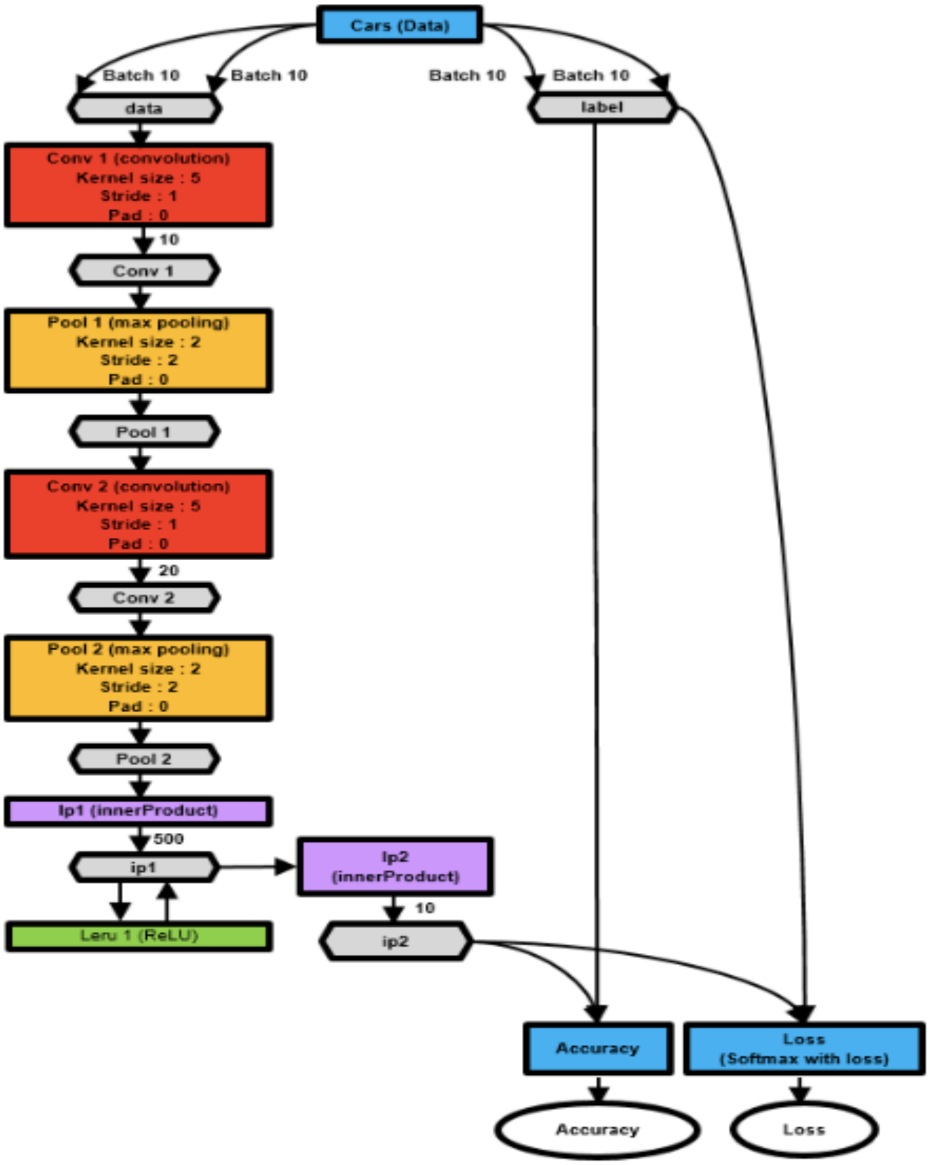


Figure 2.4: The network architecture

The network architecture is based on LeNet architecture as shown in Figure 2.4, which is famously used to classify MNIST digits. The network involves four layers with weights; the first two are convolutional and the remaining two are fully-connected. The input of the net is 256 × 256 gray-scale image; the output of the last fully-connected layer is fed to a 10-class softmax regression which produces a distribution over the 10 class labels. Like many other convolutional neural networks, each of the first two convolutional layers is followed by a subsampling (pooling) layer that expands the receptive fields of the next layer. This is one of many possible architectures. However, in practice it works well. One aspect of CNN is the fine-tuning of hyper parameters and experimenting with different architectures to arrive at an optimal model which results in highest possible accuracy of classification. Compared to LeNet architecture, a few parameters are adjusted such as batch size (reduced to 10) to enable the processing on GPU.

First, the network is fine-tuned on the vehicle dataset. Then, features are extracted from the last and the penultimate fully connected layers, respectively, for training different classifiers.

2.1.2 Histogram of Oriented Gradient (HOG)

Histogram of Oriented Gradient is a type of feature extraction method. It can be considered as the summary of local gradient in an image. HOG summarizes directions of gradient in a circular histogram. The strength is measured by the gradient magnitude. HOG was first formulated and described by Navneet Dalal and Bill Triggs in CVPR 2005 for human detection [6]. Its usage was later expanded for object detection, including vehicles detection in static images. One distinct characteristic of HOG is that it performs on local portions of images. Therefore, it can be promising in terms of invariance to geometric and lighting changes. Histogram of Oriented Gradient works as follows. Unlike many other descriptors, which require preprocessing such as color normalization, HOG directly aims at calculating the local gradients of sub-blocks of images. Local regions are referred to the equally divided sub-blocks in an image. A typical block size that is used for HOG features is 32 × 32, but can be varied depending on image size. First, the gradient calculation is performed. Then, orientation based histogram is formed. The histogram normally ranges from 0 to 180 degrees or 0 to 360 degrees. This corresponds to the choice of "signed" or "unsigned" gradient [6]. A demonstration of HOG features on a vehicle from the available dataset is illustrated in Figure 2.3.

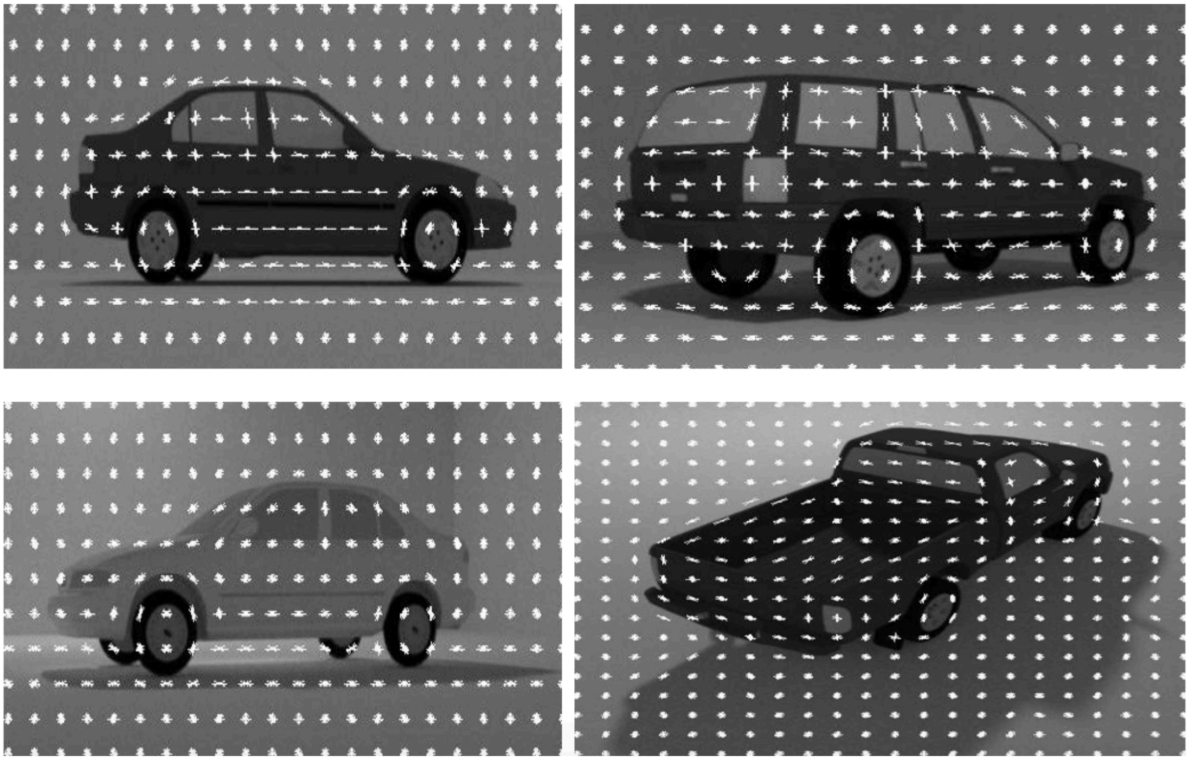


Figure 2.5: The visualization of HOG feature. The HOG feature descriptor operates on 24 × 24 blocks. It is clear to see the outline of the vehicles based on the shape of the local histogram.

2.2 Classification

In machine learning, classification aims at deciding what is the category of a newly observed data among certain list of classes. This is generally achieved utilizing a training set data containing observations or examples whose specific class is known during the training of the classifier [20]. To attain the best classification performance, it is crucial to use the most informative and distinguishing features for training the classifier. Otherwise, no classifier will achieve desirable accuracies. The following classifiers were explored with the features extracted by the methods described earlier.

2.2.1 Softmax Regression

Softmax regression is also referred to as multinomial logistic regression. It is a generalized version of logistic regression which enables the classification among multiple classes rather than the binary case only. CNNs widely use Softmax as the final classification layer. The following expression is the evaluation function for the Softmax Regression method.

$$h_\theta(x) = \begin{bmatrix} P(y=1|x;\theta) \\ P(y=2|x;\theta) \\ \vdots \\ P(y=k|x;\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{K} \exp(\theta^{(j)\top} x)} \begin{bmatrix} \exp(\theta^{(1)\top} x) \\ \exp(\theta^{(2)\top} x) \\ \vdots \\ \exp(\theta^{(K)\top} x) \end{bmatrix}$$

In this work vehicles of 10 classes are classified and thus K = 10. The goal is to minimize the following cost function to determine the optimal value of parameter θ.

$$J(\theta) = -\left[\sum_{i=1}^{m} \sum_{k=1}^{K} 1\{y^{(i)} = k\} \log \frac{\exp(\theta^{(k)\top} x^{(i)})}{\sum_{j=1}^{K} \exp(\theta^{(j)\top} x^{(i)})} \right]$$

, where 1{·} is an indicator function that does the following.

$$\mathbb{1}\{\text{statement}\} = \begin{cases} 1 & \text{if statement is true} \\ 0 & \text{if statement is false} \end{cases}$$

The gradient of the objective function, J(θ) is the following.

$$\nabla_{\theta^{(k)}} J(\theta) = -\sum_{i=1}^{m} \left[x^{(i)} \left(1\{y^{(i)} = k\} - P(y^{(i)} = k | x^{(i)}; \theta) \right) \right]$$

Softmax Regression (as part of Convolutional Neural Networks) outputs the class-label probabilities as:

$$P(y^{(i)} = k | x^{(i)}; \theta) = \frac{exp(\theta^{(k)\top} x^{(i)})}{\sum_{j=1}^{K} \exp(\theta^{(j)\top} x^{(i)})}$$

Finally, a sample is assigned with the label of highest probability [23].

2.2.2 Support Vector Machine (SVM)

A support vector machine (SVM) is a discriminative classifier which is defined by a separating hyperplane [7]. That is, utilizing a training set of labeled data (supervised learning), SVM outputs the optimal hyperplanes that maximize the margin between classes within the labeled data in the training set. Given a set of data x and corresponding labels y, SVM is aiming at

$$\min_{w,\, b} \|w\|^2 \text{ subject to } y_i(w^\top x_i + b) \geq 1 \ \forall i.$$

By solving the dual problem:

$$\max_{\alpha \geq 0} \frac{-1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^\top x_j + \sum_i \alpha_i \text{ subject to } \sum_i y_i \alpha_i$$

Then, compute,

$$w^* = \sum_{i \in SV} \alpha_i^* y_i x_i$$

$$b^* = \frac{-1}{2} \sum_{i \in SV} y_i \alpha_i^* (x_i^\top x^+ + x_i^\top x^-)$$

where x+ and x− are the examples on positive and negative side of the boundary respectively and SV is the set of resulting support vectors from learning. Support vectors are the examples on the margin. The decision function to categorize a new example is:

$$f(x) = sgn[\sum_{i \in SV} y_i \alpha_i y_j x_i^\top x_b^*]$$

Moreover, by introducing regularization constant (widely noted by C), the problem becomes:

$$\max_{0 \le \alpha_i \le C} \frac{-1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^\top x_j + \sum_i \alpha_i \text{ subject to } \sum_i y_i \alpha_i$$

with constraint of $0 \le \alpha_i \le C$. Then, SVM has a soft margin that allows outliers. Figure 2.4 illustrates this formulation geometrically.

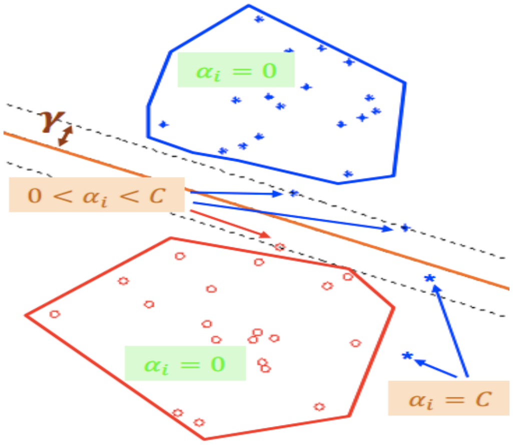


Figure 2.6: SVM with soft margin. γ is the distance from boundary to the closest point. α_i controls the regularization. And, C is the regularization constant.

This is important in the sense that by using a soft margin and allowing a few outliers in the training set, the trained network generalizes well with a test data which has not been seen during training and avoids the problem of overfitting. Overfitting is an important concept in machine learning and it refers to the case when a trained classifier is able to achieve excellent accuracy on training set while it performs poorly on a test set. This must be avoided in order to produce practical classifiers that perform well on unlabeled data that the classifier observes for the first time.

For the implementation of the training procedure, the LIBSVM library [2] with grid search and cross validation is used to find the optimal regularization constant C, which is $C = 2^{-8} \approx 0.00391$ for all features. In addition, multiclass SVM option is chosen to accomplish multiclass classification.

2.2.3 Adaptive Boosting (AdaBoost)

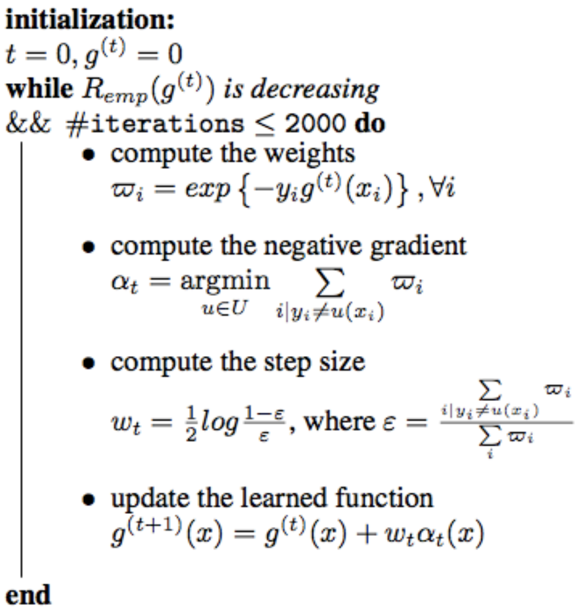
AdaBoost is the abrivation for “Adaptive Boosting” machine learning classification algorithm. The method was developed by Yoav Freund and Robert Schapire who won the G¨odel Prize for their contribution in 2003 [8]. The AdaBoost algorithm is used along with other types of learning algorithms to improve their performance. A series of classifiers ,which are referred to as weak learners by AdaBoost, output their results individually. The individual outputs are then united into a weighted sum that is essentially the final classifier by AdaBoost [8]. AdaBoost is adaptive since succeeding weak learners are fine-tuned in favor of those examples that were misclassified by previous classifiers. And the boost term implies that the algorithm iteratively boosts the performance of the week learners.

AdaBoost utilizes a given set of weak learning algorithm repeatedly in a series of iterations t = 1, …, T. In this work, the decision stumps with 20,000 threshold steps are used as weak learners. A decision stump is used as a machine learning model consisting of a single-level decision tree. That is, it is a decision tree with one main node, named the root. The root is directly connected to the terminal nodes, called its leaves. A decision stump predicts the class of an example based on the value of only a single feature. For this reason, they are also called 1-rules [19]. Mathematically, this is:

$$u(\mathbf{x}, j, h) = \begin{cases} 1 & x_j \geq h \\ -1 & x_j < h \end{cases}$$

where h = $\frac{i}{N}$, i ∈ {0,…,N}, and N = 20000.

The test accuracy does not increase after 2000 boosting rounds. Therefore, 2000 is chosen as the number of the weak learners. In addition, an early stopping criteria is set up which is simply letting the learning stop when training error is 0. The Adaboost algorithm works as simple as shown in Algorithm 1 in Figure bellow.



```
initialization:
$t = 0, g^{(t)} = 0$
while $R_{emp}(g^{(t)})$ is decreasing
&& #iterations ≤ 2000 do
    • compute the weights
      $\varpi_i = exp\left\{-y_i g^{(t)}(x_i)\right\}, \forall i$
    • compute the negative gradient
      $\alpha_t = \underset{u \in U}{\text{argmin}} \sum_{i|y_i \neq u(x_i)} \varpi_i$
    • compute the step size
      $w_t = \frac{1}{2} log \frac{1-\varepsilon}{\varepsilon}$, where $\varepsilon = \frac{\sum_{i|y_i \neq u(x_i)} \varpi_i}{\sum_i \varpi_i}$
    • update the learned function
      $g^{(t+1)}(x) = g^{(t)}(x) + w_t \alpha_t(x)$
end
```

Figure 2.7: Algorithm 1

The “one-vs-all” scheme is used to assign the image to the class of largest score after we learn a set of binary classifiers $g_k(x)$, k ∈ {1, …, 10} for each vehicle class.

Chapter 3

Viewing Angle and Lighting Conditions

3.1 Edge Detection

Edge detection consists of a set of mathematical techniques which aim at localizing pixels in a digital image where the image brightness changes sharply and exhibits discontinuities [11]. The pixels with sharp brightness discontinuities are organized into a set of line fragments. These fragments are called edges. Edge detection is a widely used technique in image recognition and for the task of feature extraction [11].

Edges in images are caused by a number of factors. The main factors are known to be "surface normal discontinuity, depth discontinuity, surface color discontinuity and illumination discontinuity." The figure bellow provides an illustration of these factors [11][12].

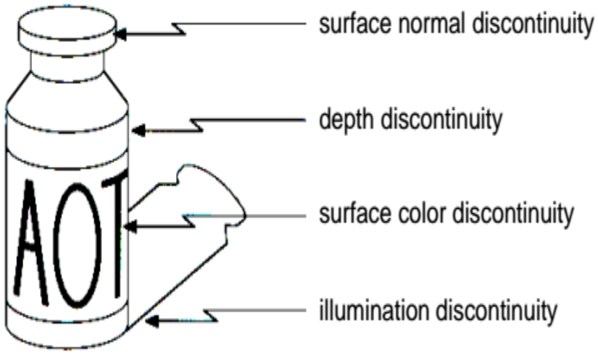
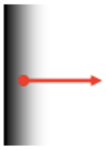

surface normal discontinuity
depth discontinuity
surface color discontinuity
illumination discontinuity
AOT

Figure 3.1 The main factors for existence of edges in images

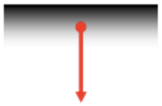
3.1.1 Image Processing Approach

In image processing applications, traditionally, image gradients have been employed for designing edge detection filters. Sober operator and Canny are famous examples of this approach. The main principle used is that the gradient of the image points in the direction of most notable change in intensity of the image. And the edge strength corresponds to the magnitude of the gradient. Below is a simple illustration of the main principles which are traditionally used for edge detection [12].

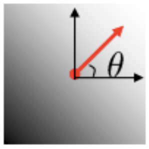
The gradient points in the direction of sharpest fluctuation in intensity:



$$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$$



$$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$$



$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

The gradient direction is given by:

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

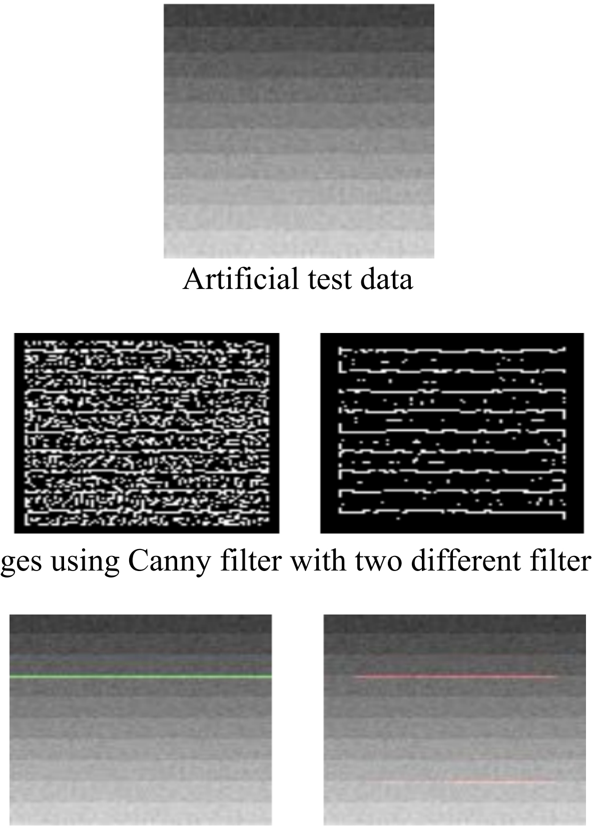
The magnitude of gradient defines the edge strength:

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

An optimal edge detector must perform well in regards to both detection and localization. Good detection corresponds to detecting the true edges rather than noise while good localization corresponds to the detection of edges near the location of true edges. The traditional approaches to edge detection in image processing with designing filters always face a trade-off challenge between detection and localization. A recent approach to edge detection is to handle the task of edge detection as an artificial intelligence/machine learning task rather than the traditional filtering approach. The next section, introduces a recent learning approach to edge detection which has proved to work well both in terms of accuracy of true edge detection and efficiency in terms of speed of performance.

3.1.2 Machine Learning Approach

the machine learning approach to edge detection requires no predefined rules. It utilizes human-labeled data as a training set and lets the classification algorithm learn the rules. The figure bellow illustrates a comparison of the traditional filter-design approach to the machine learning approach.



Artificial test data

Detected edges using Canny filter with two different filter parameters.

Ground truth

Learned edges

Figure 3.2 A comparison of filter-detected edges versus learned edges.

It is known that the edges in a local patch of an image are very codependent. They often contain recognized patterns, namely straight or parallel lines and junctions. In computer vision and machine learning fields, a class of learning methods called structured learning has been applied to problems with similar nature which has encouraged researchers to employ structured learning for the task of edge detection [13]. The "Structured Forests for Fast Edge Detection" developed by Piotr Dollar and Lawrence Zitnick at Microsoft Research has been employed in this work to provide a visualization and understanding of the effect of lighting conditions on the performance of the classifier and feature extractors explored for the task of vehicle classification. A brief summary of the principles involved in this edge detection method is followed by its application to the vehicle dataset bellow.

Structured learning has been developed and used for a wild range of classification and detection tasks. Structured learning main goal is to learn mapping between complex input and output spaces. The complex spaces include sequences, object postures, graphs and more [14, 15, 16]. The method of Structured Forests for Fast Edge Detection differs from other structured learning methods in regards to several aspects. First, in the edge detection case it is assumed that only the output space is structured. And, standard space is supposed for the input. In common approaches for structured prediction, parameters for a certain scoring function are learned. Moreover, in other structured prediction methods and applications, to obtain a prediction, an optimization over the output space is necessary [17, 18]. This necessitates the definition of an effective scoring function and an efficient optimization routine. In contrast, inference using structured forests for fast edge detection is broad-spectrum and efficient in regards to speed. The structured forests for fast edge detection provides a learning skeleton for structured output forests that can be used with an extensive class of output spaces with a high accuracy and much faster speed of performance compared to traditional filter approaches for edge detection [13].

3.1.3 Edge Detection for Visualization of Illumination effects

In order to visualize the effect of lighting changes on appearance of images and therefore on classification performance, edge detection of vehicles is performed. To perform this task Structured Forests method of Edge detection is employed [11]. The Structured Forrest method utilizes the structure present in local image pieces. It aims at learning edge detectors that are both computationally efficient and perform with high accuracy. The method is formulated as predicting local edge masks in a structured learning basis applied to random decision forests. It maps the structured labels to a discrete space to evaluate information-gain measures. This method of edge detection is used as it has proved to be superior compared to many other methods of edge-detection in regards to accuracy and efficiency [9].

To visualize the influence of lighting changes, the edges of the same vehicle under the same camera/viewing angle only at different lighting conditions are detected. This experiment very clearly illustrates the effect and importance of lighting changes. Across different lighting conditions diverse range of edges were detected for the same car under the same viewing angle. Mostly, this was due to presence of different shadows which are the result of change in lighting (location of the light source when the picture was taken). In figure 3.1, three extreme cases are depicted to illustrate this effect.

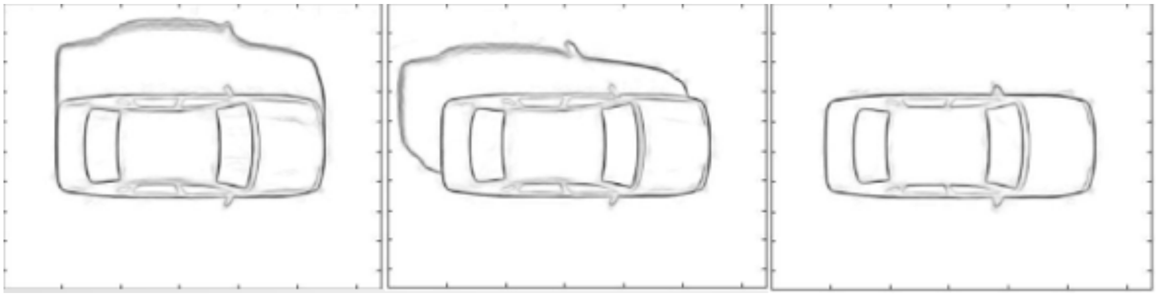


Figure 3.3: Edge-detected images of civic under 3 different lighting conditions and the same viewing angle.

Due to this diverse detection of edges under different lighting conditions, edge-detected images proved to be not a promising input to the network for learning as they would not eliminate the effects of lighting changes. However, it validated the importance of a more through investigation of lighting changes and view-angle studies. Two more experiments are followed to this end.

3.2 Training for Occlusion and Dome Experiments

To conduct a study of lighting conditions and Viewing angle effects, Convolutional Neural Nets (CNN) was used to train the model due to its ability of learning the most informative features which generalizes well in a broad range of applications. To better observe, the effects of lighting conditions and viewing angles, the features extracted from the last layer of the architecture illustrated earlier were used. This features are more sensitive to lighting and viewing angle changes and also facilitated the process in terms of end to end feature learning and classification.

In order to generate a fair result which does not employ too many images for training (network simply would have seen all possibilities during training otherwise), 50% of the dataset was randomly picked and used in the training process for the occlusion and dome experiments.

3.2.1 Occlusion

In order to assess the capability of the network to extract relevant features, occlusion experiment is performed, where every single image for each of the 10 car models is partially occluded by a square block (See figure 3.2 for visualization). Occlusion experiment is also important as in real-life traffic control application, portions of vehicle in the image may be covered by other objects or may be simply missing form the frame. Occlusion can create similar effect and one can observe the robustness of a given classification system to such scenarios.

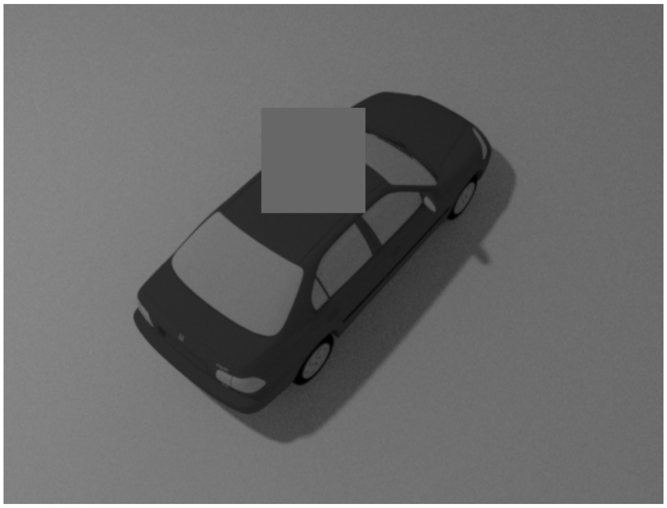


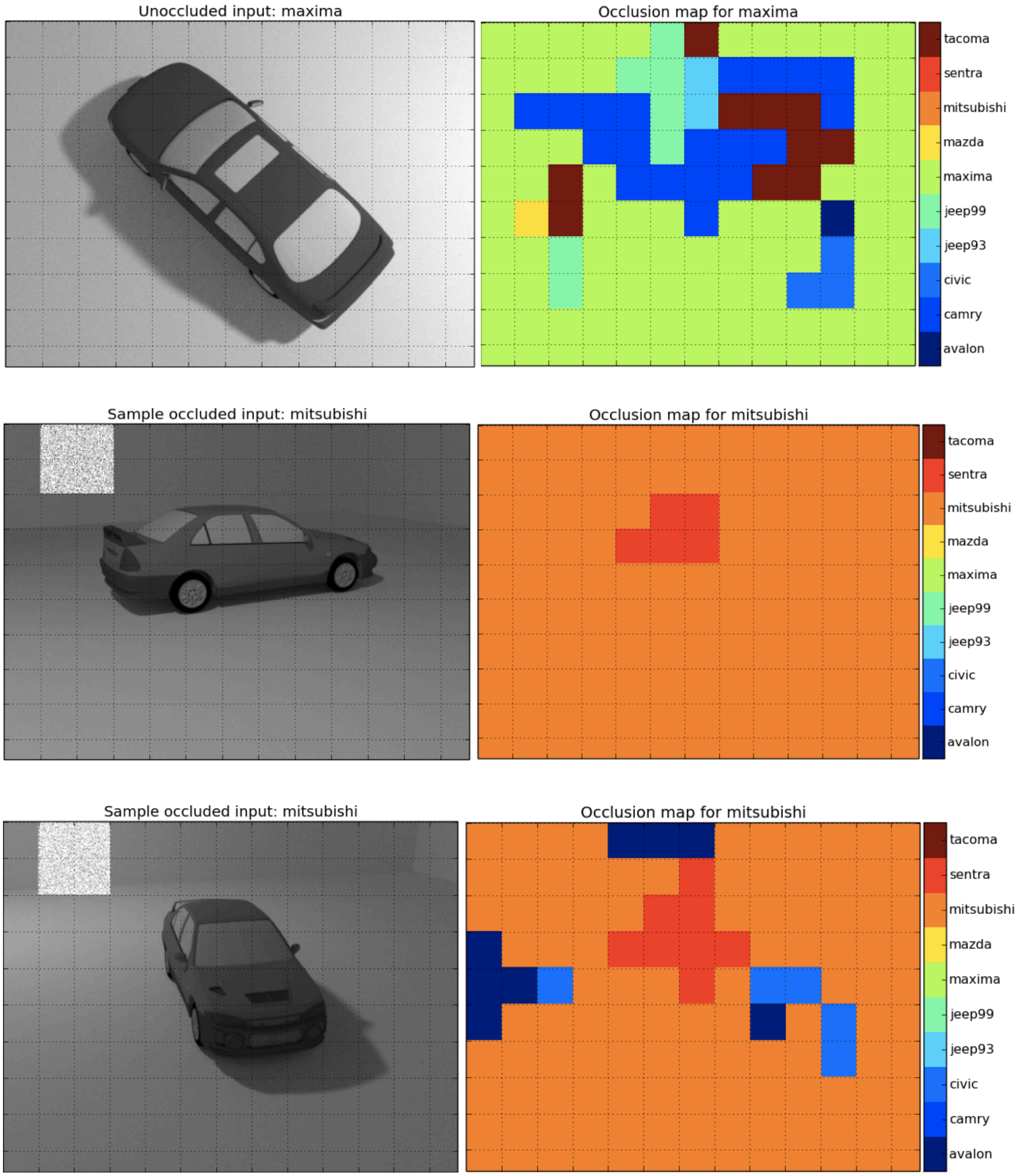
Figure 3.4: Occluded image of Honda Civic. The gray block is slid through out the image to create the effect of missing portions of the vehicle in the image being classified.

Every time a block in the image is occluded, the image is tested for classification by a pre-trained model to determine the importance of the features extracted from that portion of the image. When the occluded image is classified incorrectly it is an indication that the informative part of the image is lost and that block is an important part for feature extraction and classification use. This experiment also demonstrated the effectiveness of CNN in terms of learning effective features in terms of providing distinction among classes. The occlusion experiment verified that the most important portions of images are the ones that human would as well look at to identify the vehicle.

The occlusion experiment is also motivated by exploring previous work in the literature on part-detection for fine-grained recognition and acquiring linear subspaces under different lighting conditions [8] [9].

In addition to conclusion about the effectiveness of CNN in learning features, from the results, it is evident that certain lighting conditions and viewing angles are more robust to occlusion effects. As illustrated in Figure 3.3, occlusion results in different degree of classification error of a vehicle under different camera/viewing angle and different lighting conditions. This comparison illustrated the important role of lighting conditions since the camera angle is kept constant and only changing the lighting condition results in a very sensible change in classification results. This result is also constant with previous work done for face detection where lighting conditions are thoroughly explored and even reconstructed to attain a richer dataset for training and build more effective classifiers that generalize well to diverse lighting conditions [1].

Observing this interesting results in regards to lighting conditions, motivated another experiment to provide a compact visualization of these effects which is also more informative in terms of the effect of camera angle. This was promising to explore as one could observe the robust camera angles and conclude how the most distinguishing features were missing at certain camera angles.



Unoccluded input: maxima

Occlusion map for maxima

Sample occluded input: mitsubishi

Occlusion map for mitsubishi

Sample occluded input: mitsubishi

Occlusion map for mitsubishi

Figure 3.5: Occlusion map for Maxima and for Mitsubishi from two different camera angles.

3.2.2 Dome

As mentioned in the earlier section, the occlusion analysis is extended further to obtain a more comprehensive and compact visualization of the effect of occlusion on classification performance for each possible camera location, building a data dome of classification performance for each vehicle. Figure 3.4 provides a few examples of the domes of several vehicles.

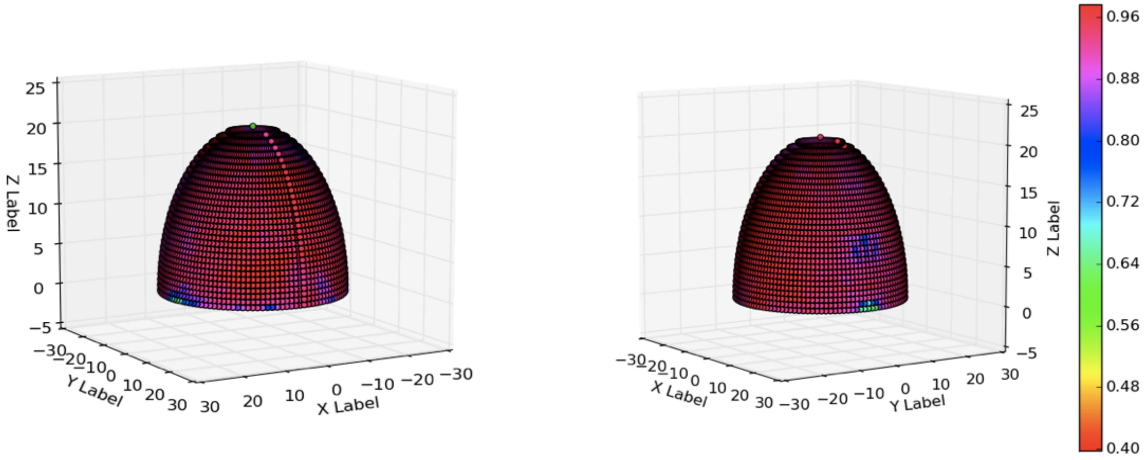


Figure 3.6: Data dome. The heat map refers to robustness to occlusion blocks in the image at different camera locations illuminating Avalon placed inside the dome under two different lighting.

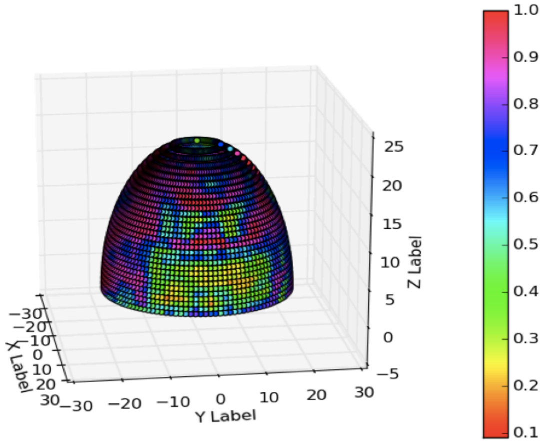


Figure 3.7: Heat map dome for Maxima.

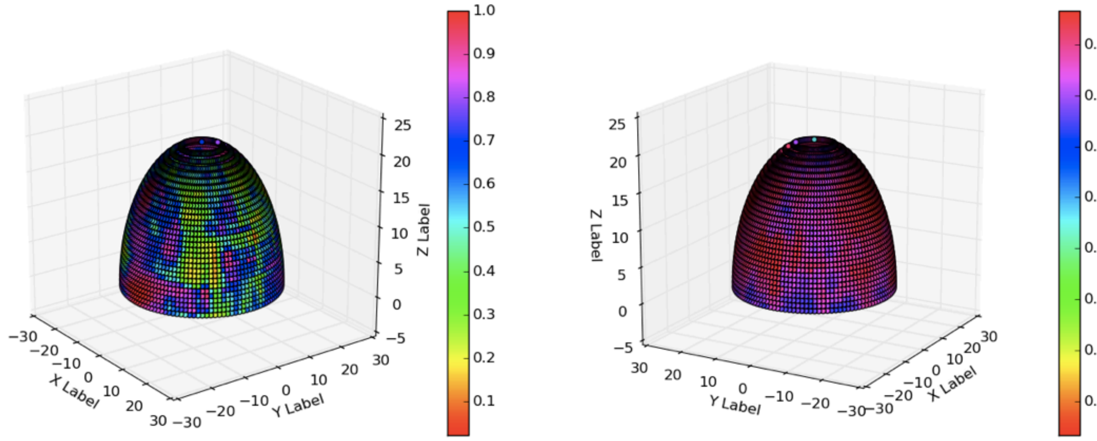


Figure 3.8: Heat map dome for Jeep93 and Jeep99 both under the same lighting condition.

In this case, images at every single viewing angle were occluded one block at a time and classified using a pre-trained model. Occluding every single image with a single block at a time results in 88 occluded versions of each image. When the occluded image is classified correctly it receives a score of 1 and when it is miss-classified it receives a score of 0. The average of the 88 scores is taken for every single view point to obtain a measure of robustness to occlusion at every camera angle (view point). The result is illustrated by a 3D heat-map. The colors indicate the accuracy of classification after occlusion at every camera angle.

Chapter 4

Experimental Evaluation

4.1 Data

The dataset used in this paper is provided by Matrix Research in Ohio. It consists of images of 10 different vehicle models (Avalon, Mazda, Jeep93, Jeep99, Civic, Mitsubishi, Camry, Maxima, Sentra and Tacoma). It contains synthetic images of each vehicle taken at 3601 different camera angles each under 17 different lighting conditions. The lighting condition refers to the location of the light source when the image is taken. Figure 4.1 bellow provides as visualization of the dataset.

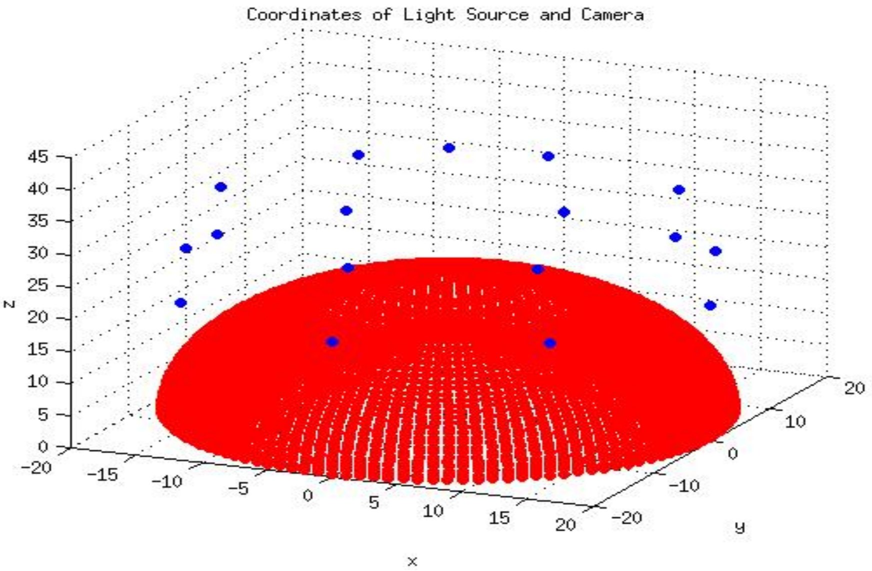


Figure 4.1: The vehicle is centered inside the dome. The red dots correspond to the locations of camera when the image was taken which numbers to 3601 different locations. The blue dots correspond to the location of the light source when the image was taken and there are 17 of them.

One benefit that is gained from using synthetic data is that there is no noise or other objects besides the vehicle, which reduces the effort in noise removal and object segmentation.

Some examples of the data are shown in Figure 4.2 bellow. Since real-world constraints may limit the number of samples available for learning, we explore the performance of the models using 10% of the data for training and 1% for testing. Both training and testing sets are picked randomly. As expected the performance of the network downgraded (see the evaluation section) which suggests some over-fitting by neural nets. This motivated exploring different feature extraction techniques as well as using different learning methods/algorithms for classification. (add more data examples)

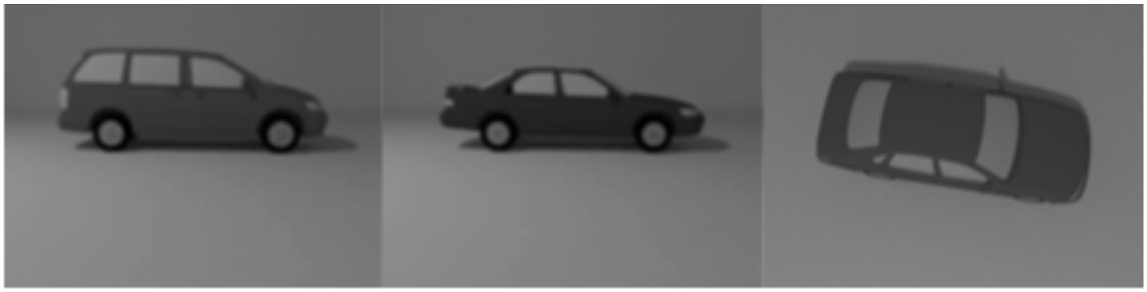


Figure Figure 4.2: Examples of synthetic vehicle dataset.

4.2 Classifier Performance

Table 1 shows classification performance with 9 different combinations of features and learning algorithms. From the feature extraction perspective, CNNs are known to be the current best feature descriptor due to its ability to automatically learn the best features rather than hand-designing them. The experiments are performed on CNN’s features of the 2 fully connected layers of the same network, respectively. HOG (Histogram of Oriented Gradients), another popular features in the area of visual recognition, are also investigated and its performance is compared with CNN features. The number of bins of HOG is 8 and the cell size is 8 × 8.

To train classifiers on these features, 3 different learning algorithms are experimented, Softmax Regression, Support Vector Machine with linear kernel, and Adaptive Boosting, respectively, and their performance is compared.

All of the combinations achieved very good classification accuracy, nearly all of them give less than 5% test error. SVM and AdaBoost with CNN-IP1 features (features extracted from the penultimate CNN fully connected layer) work best. Both achieving 100% train and near 100% test accuracies. CNN features extracted from the last fully connected layer (CNN-IP2) does not work as well as the other two, even worse than the hand-crafted HOG features. This is because the penultimate fully connected layer captures richer information from the images and less over-fits the data. Softmax classifier is generally worse than SVM and AdaBoost since it does not encourage large margins.

Table 1: Results of Classification

Feature	Method	Train (%)	Test (%)
HOG	Softmax	96.86	96.41
HOG	SVM	97.57	97.09
HOG	AdaBoost	100	99.17
CNN-IP1	Softmax	99.72	98.77
CNN-IP1	SVM	100	99.88
CNN-IP1	AdaBoost	100	99.64
CNN-IP2	Softmax	99.80	92.03
CNN-IP2	SVM	96.50	95.98
CNN-IP2	AdaBoost	98.07	96.96

4.3 Occlusion and Dome

The occlusion experiment validates that the network is effectively localizing important parts of the image and representing parts in a discriminative way as global cues, which is a key feature for fine-grained recognition [2]. This is by observing missclassification mostly happens when discriminative portions of the image are lost due to occlusion.

The dome experiment results show that while most viewing angles are robust to occlusion, there are some viewing angles where performance downgrades. This indicates that despite the networks learning relevant features, they are generally not invariant to nuisance conditions such as target pose angle and lighting conditions.

The dome experiment also provides a compact visualization of the robustness of CNN to occlusion with respect to camera angle and lighting conditions. Looking at the dome of each vehicle the color at each camera angle indicates how discriminative that viewing angle is and it also indicated the robustness of that angle to occlusion effect. The camera angles with high accuracies can be considered as containing more distinguishing features and more effective to be used for classification. One can also observe the effect of lighting conditions by comparing the domes of a single vehicle obtained for two different lighting condition. For instance, in figure 3.6. The dome of Avalon under two different lighting conditions looks different.

Chapter 5

Conclusion

Very high accuracies of classification were achieved on test set using all three learning methods, namely Softmax Regression, Linear-kernel SVM and AdaBoost applied to the fine-grained recognition of vehicles. The features extracted from the second to last layer of CNNs worked best while the CNNs features extracted from the last layer resulted in smallest accuracy with Softmax Regression. While the second-layer features resulted in relatively poor performance with Softmax Regression, Linear-kernel SVM and AdaBoost on second layer still performed well. This can be justified by recognizing the ability of the latter methods on enforcing a larger classification margin to avoid outliner and focusing on hard examples iteratively, respectively.

Moreover, occlusion and dome experiments performed with CNNs features and Softmax Regression indicate the importance of viewing angle and lighting conditions as certain lighting conditions are much more robust to occlusion compared to others under the same viewing angle. The viewing angles (camera locations) also show different degrees of tolerance to occlusion. Moreover, the results of occlusion maps confirm that the convolutional neural nets are effectively localizing the vehicles in images and use the informative features to enable the classifier best distinguish among classes. In addition, the dome experiment provides a compact visualization of the best camera set up to be used for vehicle recognition tasks.

Bibliography

[1] A. S. Georghiades, P. N. Belumeur and D. J. Keriegman, "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence Volume 23 Issue 6, June 2001, pp. 643-660, 2001.*

[2] J. Krause, T. Gebru, J. Deng, L. Li and L. Fei-Fei, " Learning Features and Parts for Fine-Grained Recognition," in *ICPR '14 Proceedings of the 2014 22nd International Conference on Pattern Recognition*, Washington, DC, 2014.

[3] M. D. Zeiler and R Fergus, "Visualizing and Understanding Convolutional Networks," in European Conference on Computer Vision, 2014.

[4] Y. LenCun, L. Bottou, Y. Bengio and P. Haffener, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE, col. 86, no. 11, pp2278-2324, 1988.*

[5] Jia, Y. a. Shelhamer, E. a. Donhue, J. a. Karayev, S. Long, J. Girshik, R. Guandarrama, S. Darrel and Trevor, "Caffe: Convolutional Architecture for Fast Feature Embedding," *arXiv preprint,* 2014.

[6] N. Dalal and B. Triggs, "Histogram of Oriented Gradients for Human Detection," in *Computer Vision and Pattern Recognition*, San Diego, 2005.

[7] B.E. Boser, I.M. Guyon and V.N. Vapnik, "A training algorithm for optimal margin classifiers," in *COLT '92 Proceedings of the fifth annual workshop on Computational learning theory*, New York, 1992.

[8] Y. Freund and R. E. Schapire, "A Short Introduction to Boosting," *Journal of Japanese Society for Artificial Intelligence,* vol. 14, no 5, pp. 771-780, 1999.

[9] L.C Zitnick and P. Dollar, "Structured Forests for Fast Edge Detection," in *European Conference on Computer Vision,* 2014.

[10] L. C. Lee, J. Ho and D. J. Kriegman, "Acquiring Linear Subspaces for Face Recognition under Variable Lighitng," IEEE Transaction on Pattern Analysis and Machine Intellignece, vol. 27, no. 5, pp. 684-698, 2005.

[11] Umbagh Scott E (2010). Digital Image Processing and Analysis: Human and omputer vision applications with CVIP Tools (2nd ed.). Boca Raton, FL: CRC Press. ISBN 9-7814-3980-2052.

[12] R. Gonzalez and R. Woods, *Digital Image Processing*. Upper Saddle River, N.J.: Prent Hall, 2008.

[13] Structured Forests: P. Dollr and C. Zitnick. "Structured Forests for Fast Edge Detect." ICCV 2013. Structured Edge Detection Toolbox.

[14] I. Tsochantaridis, T. Hofmann, T. Joachims, and T. Altun. Learning for Structured Output Spaces. In ICML, 2004.

[15] B. Taskar, V. Chatalbashev, D. Koller, and Guestrin. Learning Structured Prediction Model: A Large Margin Approach, *IN ICML, 2005*.

[16] M. Blaschko and C. Lampert. Learning to Localize Objects with Structured Output Regression. In *ECCV, 2008.*

[17] W. T. Freeman and E. H. Adelson. The Design and Use of Steerable Filters. *PAMI, 13:891-906, 1991.*

[18] I.T. Joliffe. Principle Components Analysis. Springer-Verlag, 1986.

[19] W. Iba and P. Langlay, "Introduction of One-Level Decision Trees," 1992, pp. 233-240.

[20] E. Alpaydin, Introduction to machine learning. Cambridge, Mass.:MIT Press, 2010.

[21] "Deep Learning in a Nutshell: Core Concepts", *Parallel Forall*, 2015. [Online]. Available: https://devblogs.nvidia.com/parallelforall/deep-learning-nutshell-core-concepts/. [Accessed: 01- May- 2016].

[22] LeCun, Yann. *"LeNet 5, Convolutional Neural Nets."* Retrieved 16 Nov. 2013.

[23] http://ufldl.stanford.edu/tutorial/supervised/SoftmaxRegression/.[Accessed:01-May-2016].