SHAPE BASED DETECTION AND CLASSIFICATION OF VEHICLES USING OMNIDIRECTIONAL VIDEOS

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Computer Engineering

by
Hakkı Can KARAİMER

June 2015
İZMİR

We approve the thesis of **Hakkı Can KARAİMER**

Examining Committee Members:

Assist. Prof. Dr. Yalın BAŞTANLAR
Department of Computer Engineering, İzmir Institute of Technology

---

Assist. Prof. Dr. Mustafa ÖZUYSAL
Department of Computer Engineering, İzmir Institute of Technology

---

Assoc. Prof. Dr. Yusuf Sinan AKGÜL
Department of Computer Engineering, Gebze Technical University

16 June 2015

Assist. Prof. Dr. Yalın BAŞTANLAR
Supervisor, Department of Computer Engineering
İzmir Institute of Technology

Prof. Dr. Halis PÜSKÜLCÜ
Head of the Department of
Computer Engineering

Prof. Dr. Bilge KARAÇALI
Dean of the Graduate School of
Engineering and Sciences

To My Family

ACKNOWLEDGMENTS

I would like to thank to Asst. Prof. Dr. Yalın Baştanlar and Asst. Prof. Dr. Mustafa Özuysal for their priceless guidance not only about computer science and engineering but also about whole life for previous two years. Their effect on me is highly similar how Essays of Michel de Montaigne affect a teenager. I wish to thank them to also setting up warm environments which are starting points of strong friendship between laboratory members: Ali, Ekincan, Eren, İbrahim, İpek, and me.

I would like to thank to Assoc. Prof. Dr. Yusuf Sinan Akgül, and Assoc. Prof. Dr. Aybars Uğur, professors who introduced me into Computer Vision, for their precious acceptance to evaluate this thesis. I wish to thank to Dr. Nesli Erdoğmuş for her advises, friendly attitude, and also for evaluating this work.

I thank to my family from the bottom of my heart for their endless love and supporting me through all my life. Especially my father who motivated me about science and engineering, my mother who gave her love and persistence to me, and my brother who gives his infinite energy to our family.

I also thank to Prof. Dr. Mehmet Polat, Asst. Prof. Dr. Ulaş Vural, Dr. Sema Candemir, and Asst. Prof. Dr. Ayşe Betül Oktay since they always motivate me. It was lucky for me that I met with them. I will always follow their footsteps on being hard-working person and gracious to everyone.

This thesis is supported by The Scientific and Technical Research Council of Turkey (TUBITAK) under the grant 113E107 (“Classification of Objects in Traffic Scenes using Omnidirectional and PTZ Cameras”).

ABSTRACT

SHAPE BASED DETECTION AND CLASSIFICATION OF VEHICLES USING OMNIDIRECTIONAL VIDEOS

To detect and classify vehicles in omnidirectional videos, an approach based on the shape (silhouette) of the moving object obtained by background subtraction is proposed. Different from other shape based classification techniques, the information available in multiple frames of the video is exploited. Two different approaches were investigated for this purpose. One is combining silhouettes extracted from a sequence of frames to create an average silhouette, the other is making individual decisions for all frames and use consensus of these decisions. Using multiple frames eliminates most of the wrong decisions which are caused by a poorly extracted silhouette from a single video frame. The vehicle types which are classified are motorcycle, car (sedan) and van (minibus). The features extracted from the silhouettes are convexity, elongation, rectangularity, and Hu moments. Three separate methods of classification is applied. The first one is a flowchart based (i.e. rule based) method, the second one is K nearest neighbor classification, and the third one is using a Deep Neural Network. 60% of the samples in the dataset are used for training. To ensure randomization, the procedure is repeated three times with the whole dataset split each time differently into training and testing samples (i.e. three-fold cross validation). The results indicate that using silhouettes in multiple frames performs better than using single frame silhouettes.

ÖZET

TÜMYÖNLÜ VİDEOLAR KULLANARAK ŞEKİL TABANLİ ARAÇ TESPİTİ VE SINIFLANDIRMASI

Tümyönlü videolarda araç tespiti ve sınıflandırması için, hareketli nesnenin arkaplan ayırması sonucu elde edilen şekline (siluetine) dayanan bir yöntem önerilmiştir. Diğer şekil tabanlı sınıflandırma yöntemlerden farklı olarak, ardışık video karelerinden elde edilen bilgiden yararlanılmıştır. Bu amaçla, iki farklı yaklaşım incelenmiştir. İlki, video karelerinden elde edilen siluetleri birleştirerek bir ortalama siluet oluşturmak iken, diğeri ise her bir kare için ayrı karar verilmesi ve bu kararların konsensusunun kullanılmasıdır. Çoklu çerçevelerin kullanılması, tek bir video karesinden elde edilen bozuk siluetten kaynaklanan yanlış kararların çoğunu elemektedir. Sınıflandırılan araç tipleri; motosiklet, binek araç ve dolmuştur. Siluetlerden çıkarılan öznitelikler; dışbükeylik, uzanım, dikdörtgensellik ve Hu momentleridir. Üç ayrı sınıflandırma yöntemi uygulanmıştır. İlki, akış şeması tabanlı yöntem, ikincisi K en yakın komşu ve üçüncüsü ise derin yapay sinir ağı sınıflandırmasıdır. Veri kümesindeki örneklerin %60'ı eğitim için kullanılmıştır. Rastsallaştırmayı sağlamak için, tüm veri seti, farklı eğitim ve test kümesi olmak üzere deney üç kere tekrarlanmıştır. Sonuçlar, çoklu çerçevelerdeki siluetleri kullanmanın, tek bir karedeki silueti kullanmaya göre daha başarılı olduğunu göstermektedir.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

LIST OF SYMBOLS

$O_{convexhull}$ ..... Perimeter of the convex hull
O ..... Perimeter of the original contour
$\{D_s\}$ ..... Set of detected silhouettes
$\{D_v\}$ ..... Set of valid detections
ρ ..... Convexity threshold
W ..... Short edge of the minimum bounding rectangle
L ..... Long edge of the minimum bounding rectangle
$\{D_m\}$ ..... Set of detected motorcycles
τ ..... Elongation threshold
A_s ..... Area of a shape
A_l ..... Area of the bounding rectangle
P_1 ..... Difference between average distance to cars and average distance to vans.
C_1 ..... Average distance to cars.
V_1 ..... Average distance to vans.
h_i^A ..... i_{th} Hu moment of shape A
C ..... C parameter of SVM.

LIST OF ABBREVIATIONS

LDA . . . . . Linear Discriminant Analysis
kNN . . . . . K Nearest Neighbor
SVM . . . . . Support Vector Machines
GEI . . . . . Gait Energy Image
DNN . . . . . Deep Neural Network
PCA . . . . . Principle Components Analysis
RBM . . . . . Restricted Boltzman Machine
FN . . . . . False-negatives
FN . . . . . False-negatives

CHAPTER 1

INTRODUCTION

Ability of omnidirectional cameras is providing 360 degree horizontal field of view in a single image (vertical field of view varies). When a convex mirror is placed in front of a conventional camera for this purpose, the imaging system is called a catadioptric omnidirectional camera. Two example images from such a camera are given in Figures 1.1a and 1.1b. This enlarged view is an important advantage in many application areas such as robot navigation (Goedeme et al., 2007), surveillance (Scotti et al., 2005), and 3D reconstruction (Bastanlar et al., 2012). However, so far omnidirectional cameras have not been widely used in object detection and also in traffic applications like vehicle classification. This is mainly because the objects are warped in omnidirectional images and most of the techniques developed for standard cameras cannot be applied directly.

(a) (b)

Figure 1.1. Two sample omnidirectional images from our dataset. (a) Image with a van (b) Image with a car.

Object detection and classification is an important research area in surveillance applications. A diverse range of approaches have been proposed for object detection. A

major group in these studies uses the sliding window approach in which the detection task is performed via a moving and gradually growing search window. Features based on gradient directions, gradient magnitudes, colors, etc. can be used for classification. A significant performance improvement was obtained with this approach by employing HOG (Histogram of Oriented Gradients) features (Dalal and Triggs (2005)). Later on, this technique was enhanced with part based models (Felzenszwalb et al. (2008)).

In some recent studies, the sliding window approach has been applied to omnidirectional cameras as well. For instance, HOG computation is modified for omnidirectional camera geometry to detect humans and vehicles (Cinaroglu and Bastanlar (2014), (Cinaroglu and Bastanlar, 2015)) Haar-like features are also used with omnidirectional cameras (Amine Iraqui et al. (2010); Dupuis et al. (2011); Karaimer and Bastanlar (2014)).

Another major group for object detection uses shape based features after background subtraction step. For instance, Morris and Trivedi (2006a) and Morris and Trivedi (2006b) created a feature vector consisting of area, breadth, compactness, elongation, perimeter, convex hull perimeter, length, axes of fitted ellipse, centroid and five image moments of the foreground blobs. Linear Discriminant Analysis (LDA) is used to project the data to lower dimensions. Objects are compared by weighted K nearest neighbor (kNN) classifier. Training set was made up by clustering prototype measurement vectors with fuzzy-C means algorithm.

When we compare the approaches that use image based features (HOG or Haar-like features) with approaches that use shape features extracted from silhouettes, extracting shape features is computationally cheaper. Extracting image based features for each position of sliding window requires a considerable amount of time. Regarding omnidirectional images, an extra load of converting original image to panoramic image (or conversion of features) is required. Also the storage requirement is much less with shape features. To decrease the computational load for image based features approach, one can extract features only for the region where the moving object exists. Even in that case, fitting a single window to the object is not possible. To give an example, in the study of Gandhi and Trivedi (2007), where HOG features are computed on virtual perspective views generated from omnidirectional images, the windows are located manually. These facts make the image based features unsuitable for real-time applications in most cases.

We are also able to compare the performances of the mentioned two approaches on standard images. The accuracy of the HOG based method, by Gandhi and Trivedi (2007),

is lower than the accuracy of shape based classification in their previous work (Morris and Trivedi (2006a)). The classification accuracy was reported as 64.3% for HOG based approach (accuracy is 34/36 for sedan, 17/34 for minivan and 5/17 for pickup) and 88.4% for shape based approach (accuracy is 94% for sedan, 87% for truck, 75% for SUV, 100% for semi, 90% for van, 0% for TSV and 85% for MT).

Motivated by the facts given above, we decided to develop a shape based method for omnidirectional cameras. The vehicle types that we worked on are motorcycle, car (sedan) and van (minibus). The extracted features are convexity, elongation, rectangularity, and Hu moments. We applied three different methods for vehicle classification. First one uses the features one after another in a flowchart (from now on will be referred as "flowchart method"). The second one is K nearest neighbor (kNN) classification, and the last one is Deep Neural Network (DNN) classification.

In the flowchart method, the convexity is used to eliminate poor silhouette extraction, the elongation is used to distinguish motorcycles from other vehicles, and two other features (rectangularity and a distance based on Hu moments) are used for labelling an object as a car or a van. The decision boundary is obtained by applying Support Vector Machines (SVM) on the training dataset. In our experiments, using the average silhouette rather than using a single frame (not averaging) improved the rate of correct classification from 80% to 95% for motorcycle, from 78% to 98% for car, and from 81% to 83% for van.

Vehicle classification with kNN was used many times before (e.g. Morris and Trivedi (2006a), Morris and Trivedi (2006b), (Luo et al., 2006), Rashid et al. (2010), Mithun et al. (2012)). Although they did not employ omnidirectional cameras, we can consider kNN with single silhouettes as the benchmark method and compare it with using multiple silhouettes for kNN classification. Using the average silhouette improved the rate of correct classification from 53% to 97% for motorcycle, from 53% to 98% for car, and from 72% to 99% for van.

In the DNN classification method which became popular with its powerful machine learning model. Correct classification rate is improved with the average silhouette method from 95% to 97% for motorcycle, from 85% to 95% for car.

Averaging silhouettes is not the only way to exploit the information available in multiple frames of a video. As an alternative, we investigated finding the "consensus" of silhouettes, where the silhouettes are not averaged but a separate decision is made for each video frame. When a predefined percentage of samples make the same decision,

that vehicle type is chosen. This can be considered as a decision-level fusion. The classification performance is not as good as average silhouettes; however it is faster than the average silhouettes for kNN classification since the operations in averaging step are more costly than extracting features from multiple silhouettes.

Our omnidirectional video dataset (Karaimer and Bastanlar, 2015), together with annotations and binary videos after background subtraction, can be downloaded from our website (`http://cvrg.iyte.edu.tr/`).

1.1. Other Related Work

Before giving the details of our method, let us briefly present more related work on shape based methods for vehicle classification. In one of the earliest studies on vehicle classification with shape based features, authors first apply adaptive background subtraction on the image to obtain foreground objects (Gupte et al. (2002)). Location, length, width and velocity of vehicle fragments are used to classify vehicles into two categories; cars and non-cars. Vehicle length is used to identify and count trucks on a highway with a 92% success rate in the study of Avery et al. (2004). Hasegawa and Kanade (2005) created a 11-dimensional vector of image features extracted from from the bounding box, width, height and area, target colour and different image moments. Their overall classification accuracy is 91% for 6 object categories using linear discriminant analysis. In another study, (Kumar et al. (2005)), authors use position and velocity in 2D, the major and minor axis of the ellipse modeling the target and the aspect ratio of the ellipse as features in a Bayesian Network. In a ship classification study, researchers use MPEG-7 region-based shape descriptor which applies a complex angular radial transform to a shape represented by a binary image and classified ships to 6 types with kNN (Luo et al. (2006)). Ji et al. (2007) record side views of vehicles, and Gabor filter and minimum distance classifier are used to classify vehicles into five categories, with classification rates of up to 95%. In a 3D vehicle detection and classification study which is based on shape based features, Buch et al. (2008) use the overlap of the object silhouette with region of interest mask which corresponds to the region occupied by the projection of the 3D object model on the image plane. In Chen et al. (2011), a similar 3D model based classification is compared with using 2D shape based features and SVM classifier. In Chen and Ellis (2011), shape based features and pyramid HOG features are merged prior to classification task. In

Chen et al. (2012), again silhouette and intensity based pyramid HOG features extracted following background subtraction. Then, foreground blobs are classified with majority voting. In a patent (Sethna et al. (2012)), researchers classified traffic scenes based on thresholding after computing shape based features obtained from foreground silhouettes.

Instead of standard video frames, some researchers employed time-spatial images, which are formed by using a virtual detection line in a video sequence. Rashid et al. (2010) construct a feature vector obtained from the foreground mask. Employed features are width, area, compactness, length-width ratio, major and minor axis ratio of fitted ellipse, rectangularity. The samples are classified by K nearest neighbor algorithm. Later, they improved their work using multiple time spatial images (Mithun et al., 2012).

Although not applied to vehicle classification, a radically different method that uses silhouettes was proposed by Dedeoglu et al. (2006). They define "silhouette distance signal" which is the sum of distances between center of a silhouette and contour points. They create a database of sample object silhouettes with manually labelling object types. An object is classified by comparing its silhouette distance signal with the ones in the template database.

In a human recognition study, Gait Energy Image (GEI) which is the result of averaging binary gait silhouette images, is proposed by Han and Bhanu (2006). They use GEI as a representation of gait properties to distinguish individuals from each other. They also show that GEI is less sensitive to silhouette noise in individual frames.

In a detailed review (Sivaraman and Trivedi, 2013), state of the art vehicle detection algorithms are summarized according to different aspects. For example, usage of stereo or monocular vision, usage of motion or appearance, and classification algorithms etc. In another comprehensive study of computer vision techniques for the analysis of urban traffic, Buch et al. (2011) reviewed previously proposed algorithms based on the usage of top-down and bottom-up classification techniques. In this work, algorithms stated as top-down use classification of features, and algorithms stated as bottom-up use interest point detector and descriptors. According to them, top-down methods can raise issues under challenging urban conditions. Bottom-up methods have promising result, but are also limited in some cases. As reported by the authors, clearer definition of scenarios and better fusion of top-down and bottom-up algorithms is required.

Regarding the shape based classification studies with omnidirectional cameras, the only work that we found in the literature (Khoshabeh et al., 2007) uses only the area of the blobs and classifies them into two classes; small and large vehicles.

1.2. Contributions

The main contribution in our study can be considered as exploiting the information available in multiple frames of the video for vehicle classification. The silhouettes extracted from a sequence of frames are combined to create an "average silhouette". This process is known as "temporal averaging of images" in image processing community and usually used to eliminate noise. We also investigated the use of decision-level fusion, where the classification is made for each video frame separately and the "consensus" of these decisions are determined. We experimentally show that both of these multi-frame approaches perform better than using a single frame.

Another contribution in this thesis is that the area of the silhouette is not a feature in our method which makes it suitable for portable image acquisition platforms. Previous work, that employ cameras fixed to buildings, use "area" as a feature to classify vehicles (Morris and Trivedi (2006a), Morris and Trivedi (2006b), Khoshabeh et al. (2007), Buch et al. (2008), Rashid et al. (2010), Mithun et al. (2012)). Since that feature becomes invalid when the distance between the camera and the scene objects change, those methods are not versatile.

1.3. Organization of Thesis

The organization of the thesis is as follows. In Chapter 2, we introduce the details of silhouette averaging and consensus of silhouettes approaches. Vehicle detector and classifier methods are described in Chapther 3. Experiment results are presented in Chapter 4 and finally conclusions are given in Chapter 5.

CHAPTER 2

USING MULTIPLE SILHOUETTES

The silhouettes are obtained after a background subtraction step and a morphological operation step. For background subtraction, the algorithm proposed by Yao and Odobez (2007) is used, which was one of the best performing algorithms in the review of Sobral and Vacavant (2014). The final binary mask is obtained by an opening operation with a disk, after which the largest blob is assigned as the silhouette belong to the moving object.

The rest of this chapter explains two different approaches of using multiple silhouettes. The first on is "average silhouette" which is the temporal average of silhouettes, and the second one is "consensus of silhouettes" which is based on majority of individual classifications of single silhouettes.

2.1. Average Silhouettes

This section presents details of "average silhouettes" (cf. Figure 2.1). To obtain an "average silhouette" we need to define how many frames are used and the silhouettes from these frames should coincide spatially. If a silhouette is in range of a previously specified angle (which we set as [30°, -30°], and 0° is assigned to the direction that camera is closest to the road), then the silhouette is rotated with respect to the center of omnidirectional image so that the center of the silhouette is at the level of the image center. This operation, also described in Figure 2.2, is repeated until the object leaves the angle range.

Silhouettes obtained in the previous step are added to each other so that the center of gravity of each blob coincides with others. The cumulative image is divided by the number of frames which results in "average silhouette" (Figure 2.3). We then apply an intensity threshold to convert average silhouette to a binary image and also to eliminate less significant parts which were supported by a lower number of frames. Thus we can work with more common part rather than taking into account every detail around a silhouette (Figure 2.3g). The threshold we select here eliminates the lowest 25% of grayscale levels.

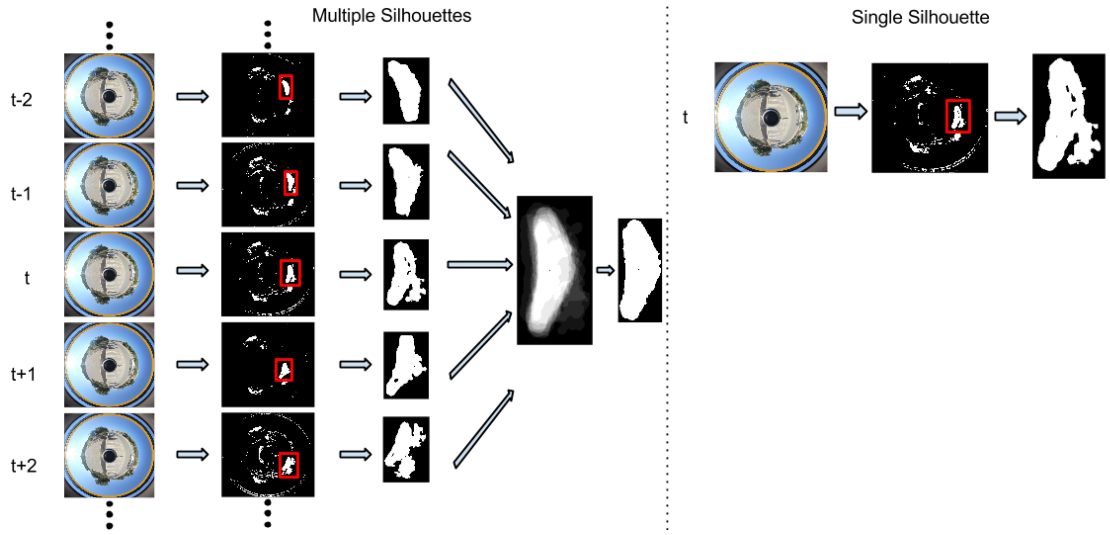


Figure 2.1. Summary of the use of average silhouettes, on the contrary of single silhouette.

2.2. Consensus of Silhouettes

In addition to silhouette averaging, we present a second way to merge information in multiple frames. The largest blob for each frame is considered as an input for the single frame classification method and a decision is made for each. When a predefined percentage, for instance 50%, of the samples make the same prediction, we consider that there is a "consensus" among the prediction of the frames and we call that prediction as the vehicle type.

In our analysis, we have seen that silhouette extraction for consensus of silhouettes is computationally cheaper than the average silhouette method. For consensus of silhouettes, morphological operations and rotation of silhouette with respect to omnidirectional image center takes 15 ms per frame, although for average silhouette, extra two operations, coinciding centers and addition to previous silhouettes takes 169 ms per frame.

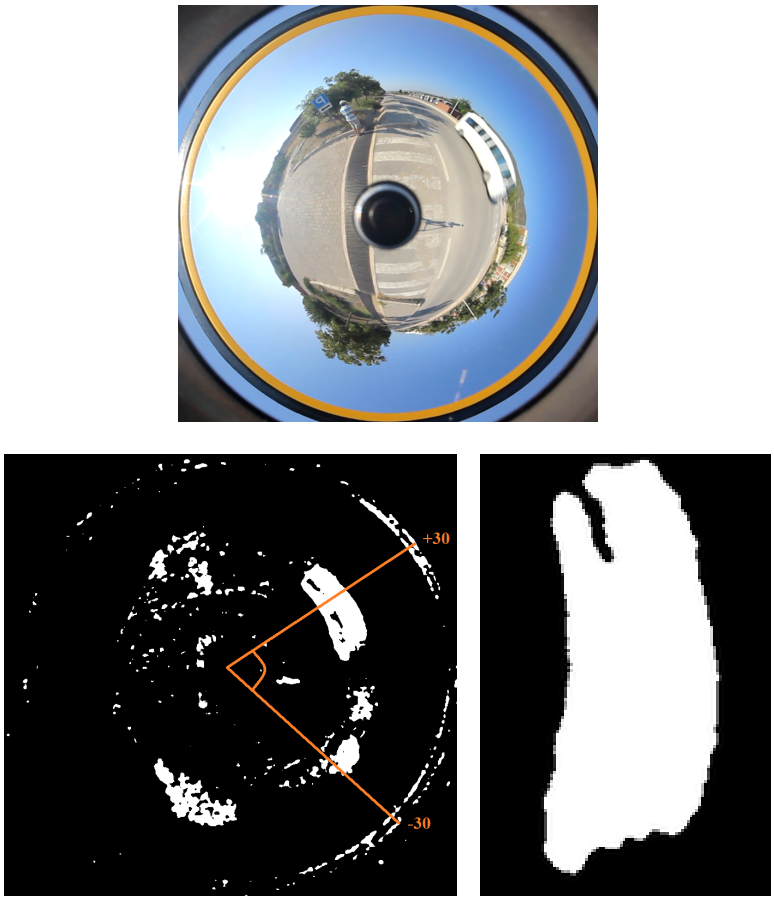


Figure 2.2. Top: An example omnidirectional video frame containing a van. Bottom-left: The same frame after background subtraction. Also the angle range that we used, namely [30°, -30°], is superimposed on the image. Centroid of the largest blob is at 29°. Bottom-right: Rotated blob after morphological operations.

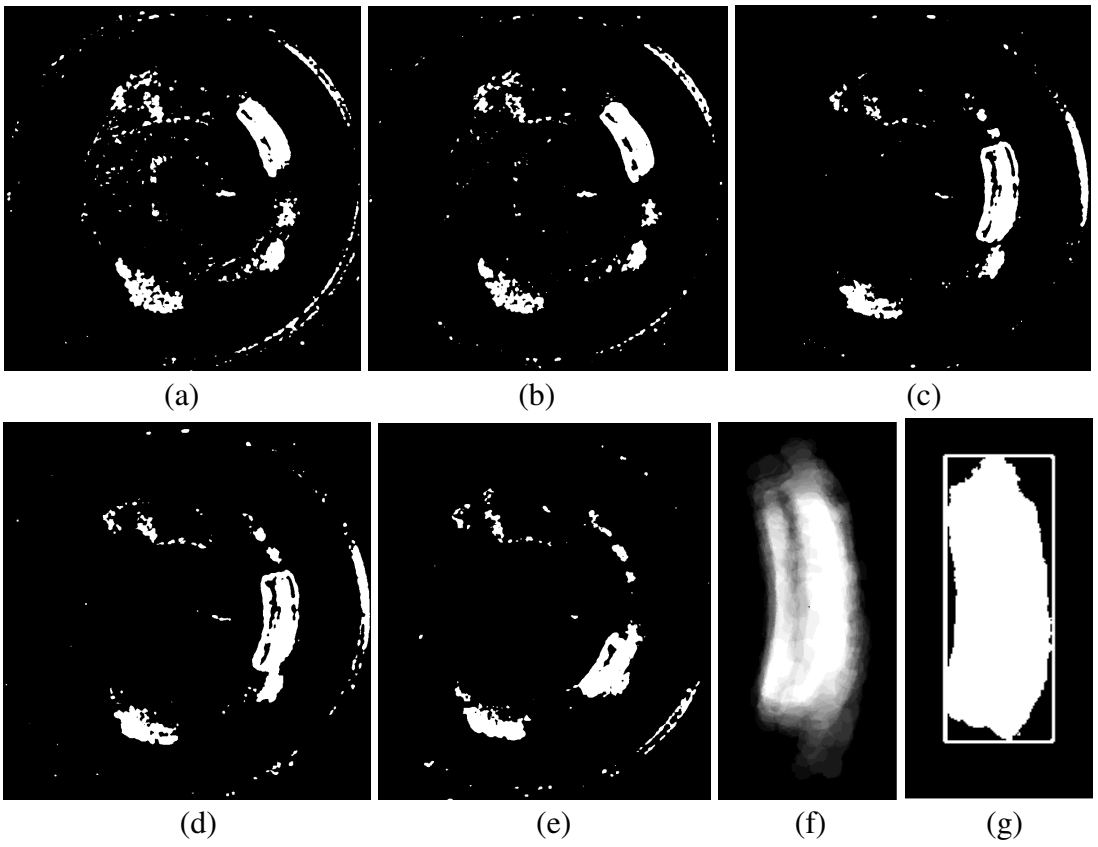


Figure 2.3. Example binary images when the centroid of the object is at (a) 29° (b) 26° (c) 0° (d) -11° (e) -29°. (f) Resultant "average silhouette" obtained by the largest blobs in the binary images. (g) Thresholded silhouette and the minimum bounding rectangle.

CHAPTER 3

DETECTION AND CLASSIFICATION

We compare three different approaches of using silhouettes, namely single silhouette that is closest to 0°, averaged silhouette and the consensus of multiple silhouettes. We applied three methods of classification for the mentioned three approaches. First one is the flowchart method that we developed, second one uses kNN classification and it was previously used for shape based vehicle classification, and the last one is DNN classification which became popular in recent years.

3.1. Flowchart Method

The steps of this method are summarized in the block diagram in Figure 3.1. Firstly, a convexity threshold is applied to a silhouette obtained after morphological operations. If the silhouette averaging approach is used, then the silhouette here is the one obtained by the procedure described in Section 2.1. Otherwise it is a single-frame silhouette.

The convexity (3.1) is used to eliminate detections that may not belong to a vehicle class or poorly extracted silhouettes from vehicles.

$$Convexity = \frac{O_{convexhull}}{O} \tag{3.1}$$

where $O_{convexhull}$ is the perimeter of the convex hull and O is the perimeter of the original contour (Yang et al. (2008)). Since we do not look for a jagged silhouette, the set of detected silhouettes $\{D_s\}$ is filtered to obtain a set of valid detections $\{D_v\}$ (3.2) using the convexity threshold ρ .

$$\{D_v\} = \{D_v | Convexity_{D_s} > \rho\} \tag{3.2}$$

We set $\rho = 0.75$ for our experiments. An example is shown for an eliminated silhouette using convexity threshold in Figure 3.2. The set of valid detections $\{D_v\}$ is passed to the classification step. The features we employ for classification are; elongation, rectangularity, and Hu moments. Elongation (3.3) is computed as follows

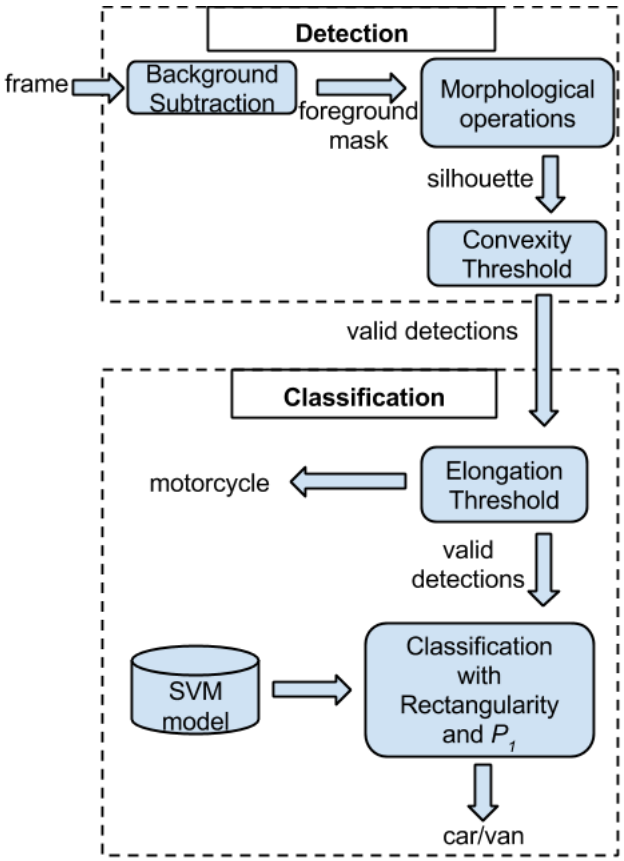


Figure 3.1. Block diagram of the detection and classification system. With the proposed method, multiple frames are processed and the extracted average silhouette is used instead of a silhouette from a single frame.

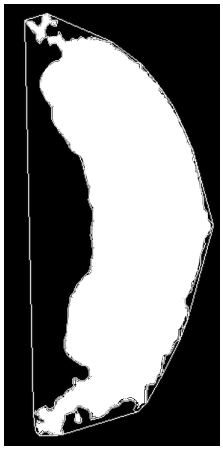


Figure 3.2. An example of an extracted silhouette and its convex hull. It is extracted from a van example using a single frame and its convexity is computed as 0.73 which is lower than the threshold. $\rho = 0.75$.

$$Elongation = 1 - W/L \tag{3.3}$$

where W is the short and L is the long edge of the minimum bounding rectangle (Figure 2.3g) which is the smallest rectangle that contains every point in the shape (Yang et al. (2008)).

We observed that the elongation is able to discriminate motorcycles from other vehicle types with a threshold. Then, the set of detected motorcycles $\{D_m\}$ (3.4) is given by

$$\{D_m\} = \{D_m | Elongation_{D_v} < \tau\} \tag{3.4}$$

where τ is the elongation threshold. τ is determined using the samples in the training set.

Rectangularity (3.5) measures how much a shape fills its minimum bounding rectangle (Yang et al. (2008)):

$$Rectangularity = A_s/A_l \tag{3.5}$$

where A_s represents area of a shape and A_l represents area of the bounding rectangle. Rectangularity is a meaningful feature to distinguish between sedan cars and vans since the silhouette of a van has a tendency to fill its minimum bounding box. In our trials, we observed that setting a threshold for rectangularity alone is not effective enough to discriminate cars from vans. To discriminate the cars and vans better, we defined an extra feature, named P_1 (3.8), which is based on Hu moments and measures if an extracted silhouette resembles the car silhouettes in the training set more than it resembles the van silhouettes. P_1 is an exemplar-based feature rather than a rule-based one and it is computed as follows:

$$C_1 = \frac{1}{\#cars} \sum_{i=0}^{\#cars} I_2(D_s, Car_i) \tag{3.6}$$

$$V_1 = \frac{1}{\#vans} \sum_{i=0}^{\#vans} I_2(D_s, Van_i) \tag{3.7}$$

$$P_1 = C_1 - V_1 \tag{3.8}$$

For a new sample, P_1 corresponds to the difference between the average I_2 (3.10) distance to the cars in the training set and the average I_2 distance to the vans in the training set.

The mentioned I_2 distance is one of the three possible distances, based on 7 Hu moments (Hu (1962)), used for computing the similarity of two silhouettes:

$$I_1(A,B) = \sum_{i=1...7} \left| \frac{1}{m_i^A} - \frac{1}{m_i^B} \right| \tag{3.9}$$

$$I_2(A,B) = \sum_{i=1...7} \left| m_i^A - m_i^B \right| \tag{3.10}$$

$$I_3(A,B) = \sum_{i=1...7} \left| \frac{m_i^A - m_i^B}{m_i^A} \right| \tag{3.11}$$

$$m_i^A = sign(h_i^A).log(h_i^A) \tag{3.12}$$

$$m_i^B = sign(h_i^B).log(h_i^B) \tag{3.13}$$

where h_i^A and h_i^B are the Hu moments of shapes A and B respectively (Bradski and Kaehler (2008)).

We select I_2 (3.10) since it achieved better discrimination in our experiments than I_1 (3.9) and I_3 (3.11).

If a detection is not classified as a motorcycle, in other words $Elongation > \tau$, then it can be either a car or a van. To determine the decision boundary between car and van classes we trained a SVM classifier with a linear kernel using the samples in the training set.

3.2. K Nearest Neighbors

Without using classification scheme in Figure 3.1 we applied kNN classification with using plausible features from flowchart method. Since vehicle classification with kNN using features extracted from a single silhouette can be considered as a benchmark method (e.g. Morris and Trivedi (2006a), Morris and Trivedi (2006b), Mithun et al. (2012)), this way we can investigate the improvement gained by using multiple frames.

Using kNN, in the simplest case, a new sample's label is determined according to its nearest neighbours label. Closest points label is assigned to new samples label in this case, i.e. K=1, or nearest neighbour rule. When K is a number greater than 1, majority of

nearest K neighbours' label in training set is assigned to new samples label. This is more powerful rather than using nearest neighbour rule. While exhaustive search takes $O(dn)$ time on d dimensional dataset, using KD-Tree (k-dimensional tree) which is a suitable, and efficient data structure, time complexity can be reduced (Duda et al., 2012).

In our case, each neighbours vote is counted 1. An extension to kNN is using a weighting scheme which is computed based on distance to sample. This, which is sensible, cause that a closer point to a new sample have a higher vote rather than farther points. The procedure which is mentioned here is called $weighted\ kNN$. (Duda et al., 2012).

kNN method is applied on average silhouette, consensus of silhouettes, and single frame silhouette approaches. On our dataset we used the features of elongation, rectangularity, convexity. We also computed solidity and ellipse axes ratio features. However, increasing the number of features did not improve the results. Therefore in Section 4 we present the results with three features we mentioned first.

3.3. Deep Neural Networks

As a third alternative classification algorithm, we employed Deep Neural Networks (DNN) to analyse improvement on classification performance using multiple silhouettes. The reason why we chose DNN's is their proved, powerful machine learning model (Hinton and Salakhutdinov, 2006). They described a nonlinear generalization of PCA (Principle Components Analysis) that uses an adaptive, multilayer "encoder" network to transform the high-dimensional data into a low-dimensional code and a similar "decoder" network to recover the data from the code. They also present experimental results on both document and image datasets.

In recent years, Deep Neural Networks have reached remarkable performance on image classification (Krizhevsky et al., 2012). Rather than simple neural network architectures, their complex architecture is able to model complex features. Later on, Szegedy et al. (2013) improved DNN's not only for object classification but also precise localization problems which is led to scalable object detection (Erhan et al., 2014).

Hinton and Salakhutdinov (2006) introduced "pretraining" procedure for binary data. The "pretraining" procedure learns one layer of features at a time. This way the procedure helps to gradient descent to work well by finding good initial weights. With

"pretraining" a stack of restricted Boltzman machines (RBM's) are learned. Each have only one layer of feature detectors. One RBM's learned feature activations are used as data to train the next RBM in stack. After the "pretraining", the RBM's are "unrolled", which means obtaining required gradients by chain rule to propogate first through the decoder network and then through the encoder network. Their whole system is called "autoencoder", which is later fine-tuned using backpropagation of error derivatives. An example to this "pretraining" procedure is given in Figures 3.3, 3.4, and 3.5. It should be noted that, while the last layer of "pretraining", cf. Figure 3.5, is trained in a supervised method, and the others are trained in an unsupervised method.

Similar to Flowchart, and kNN methods we applied Deep Neural Network on average silhouette, consensus of silhouettes and single frame silhouette approaches. The results are given in Section 4.3

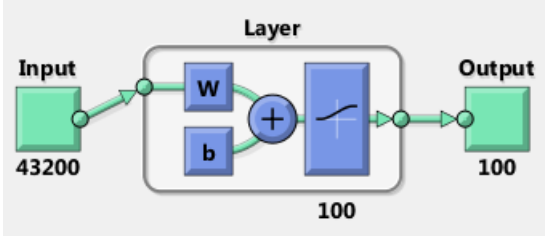


Figure 3.3. First layer of "pretraining" procedure. Input is the same size with the image, and output is the same size with the input of next layer.

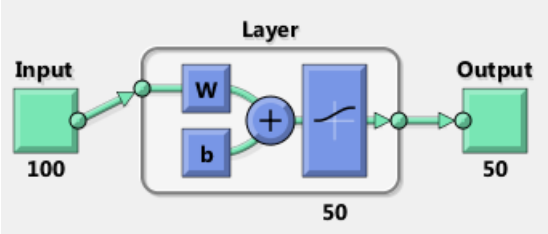


Figure 3.4. Second layer of "pretraining" procedure. Input is the same size with the previous layers output layer, and output is the same size with the input of next layer.

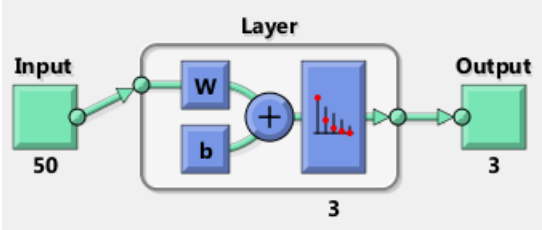


Figure 3.5. Third layer of "pretraining" procedure. Input is the same size with the previous layers output layer, and output is the same size with the number of classes.

CHAPTER 4

EXPERIMENTAL RESULTS

Using a Canon 600D SLR camera and a mirror apparatus (`www.gopano.com`) we obtained a catadioptric omnidirectional camera. Figure 4.1 shows a sample catadioptric omnidirectional camera. We constructed a dataset of 49 motorcycles, 124 cars and 104 vans totaling 277 vehicle instances. Dataset is divided into training and test sets. Training set contains approximately 60% percent of the total dataset corresponding to 29 motorcycles, 74 cars and 62 vans. The rest is used as test set. To ensure the randomization of data samples, the procedure is repeated three times with the dataset split randomly into training and testing samples. We summarize our experiment results under three subsections belonging to flowchart method, kNN, and DNN classification each.

Figure 4.1. A mirror apparatus is placed in front of a conventional camera to obtain a catadioptric omnidirectional camera.

4.1. Flowchart Method

We set $\rho = 0.75$ and SVM's parameter $C = 0.2$ for our training set. The elongation threshold is determined by choosing the maximum elongation value of motorcycles in the training set since this value discriminates motorcycles from other vehicles.

Regarding the training of car-van classifier, Figures 4.2 and 4.4 show the SVM's linear decision boundary, trained with the average silhouette and single frame silhouette respectively. Training the single frame method with the extracted single frame silhouettes would not be fair since they contain poorly extracted silhouettes. Therefore, the boundaries of the vehicles are manually annotated and used for the training of single frame method. Test results with and without averaging silhouettes are shown in Figures 4.3 and 4.5 respectively.

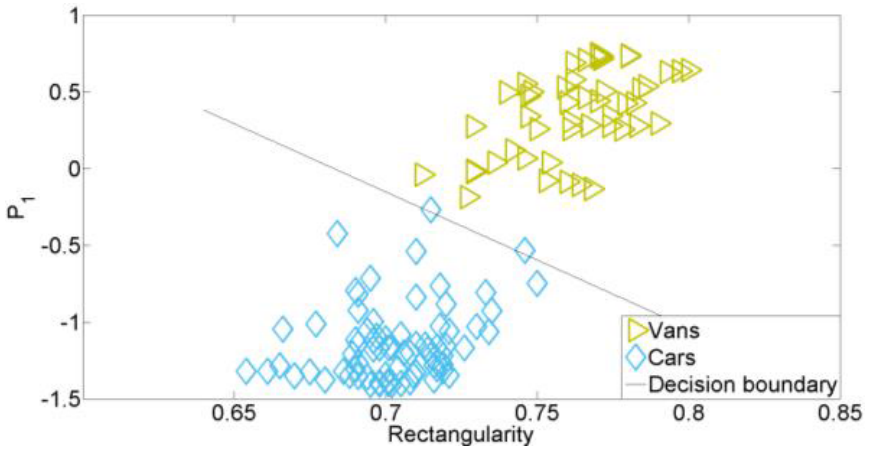


Figure 4.2. Training result of SVM using the average silhouette method.

We report the average results of the two compared methods in Table 4.1. Values in the table correspond to what percentage of the instances of a vehicle type is classified correctly. Not surprisingly, exploiting the information from multiple frames by averaging the silhouettes has a greater performance than using the silhouette in a single frame.

Tables 4.2 and 4.3 depict the total number of correctly classified and misclassified samples for three folds for each class with the average silhouette and single frame silhouette methods respectively. False negatives (FN) are the missed samples which are eliminated by convexity threshold.

Figure 4.6 shows an example where a car is correctly classified with using average

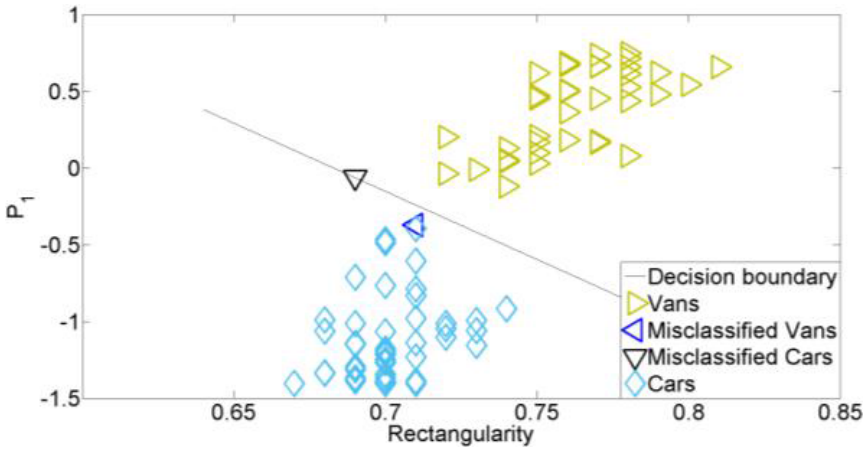


Figure 4.3. Test result with the average silhouette method.

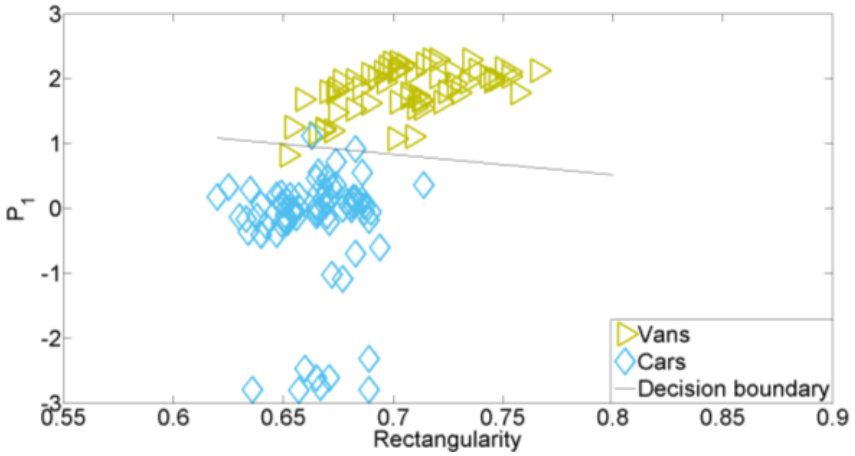


Figure 4.4. Training result of SVM without averaging silhouettes (single frame method.

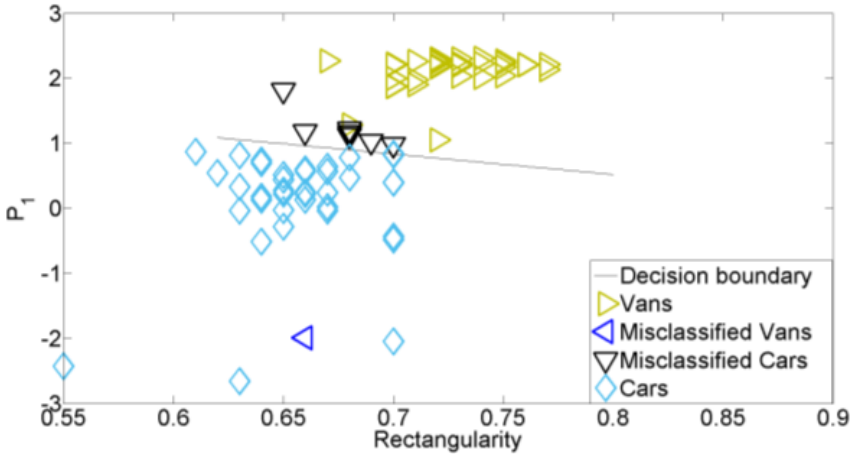


Figure 4.5. Test result without averaging silhouettes, i.e. using single frame silhouettes.

Table 4.1. Average classification accuracies for each class when $\rho = 0.75$ and $C = 0.2$ for the average silhouette method and for the single frame method.

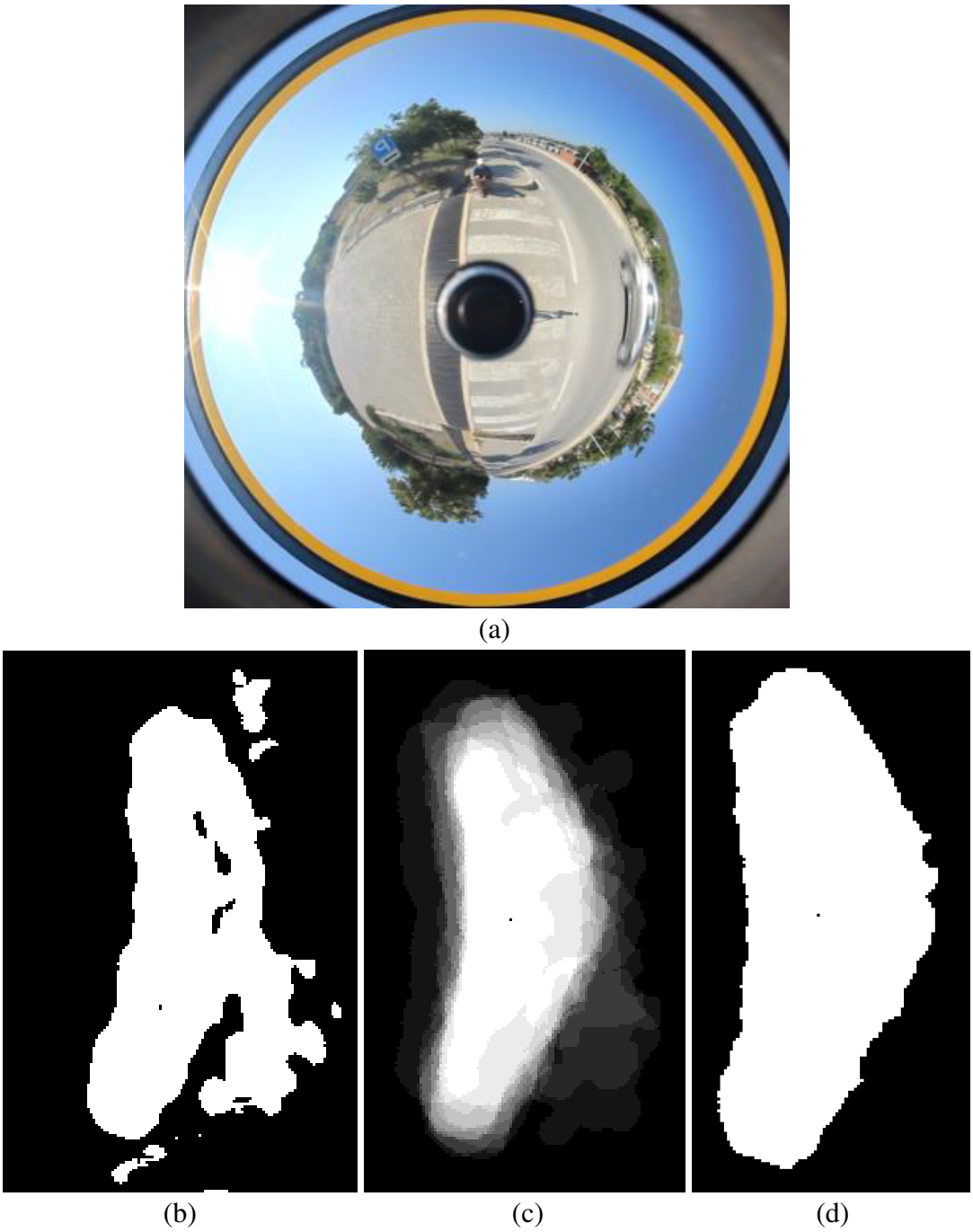
	Motorcycle	Car	Van	Overall
Average silhouette method	95%	98%	83%	92%
Single frame method	80%	78%	81%	79%

Table 4.2. Confusion matrix for the approach of using average silhouettes as sum of three folds (For each fold there are 20 motorcycles, 50 cars, and 42 vans in test set).

	Ground truth	Motorcycle	Car	Van
Detection	Motorcycle	57	0	1
	Car	2	146	2
	Van	1	4	104
	FN	0	0	20

Table 4.3. Confusion matrix for single frame method as sum of three folds (For each fold there are 20 motorcycles, 50 cars, and 42 vans in test set).

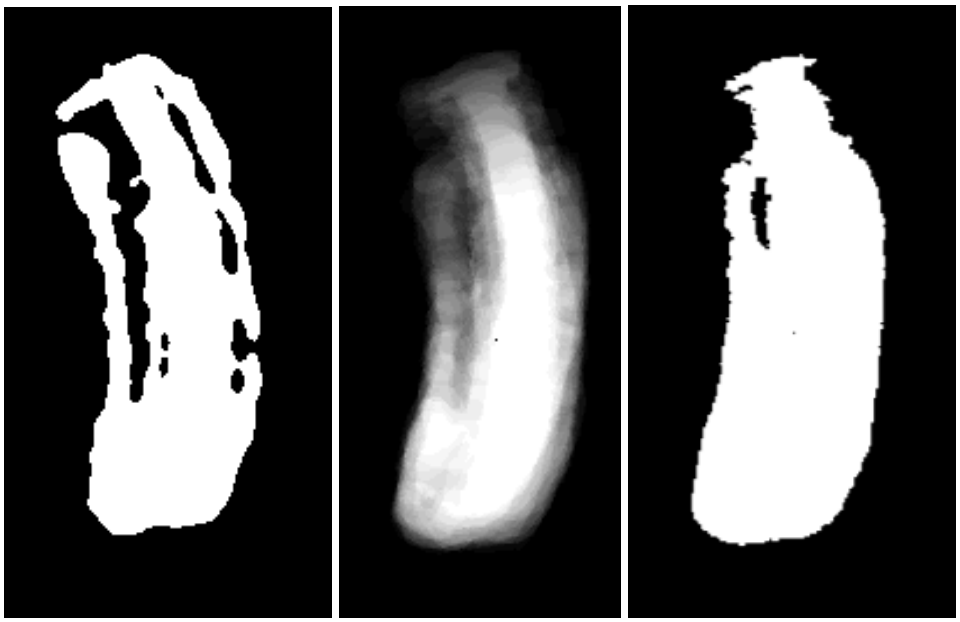
	Ground truth	Motorcycle	Car	Van
Detection	Motorcycle	48	8	13
	Car	0	118	2
	Van	2	20	101
	FN	10	4	10



(a)

(b) (c) (d)

Figure 4.6. Example car silhouettes (a) Original frame (b) Result of using a single silhouette which is misclassified with $rectangularity = 0.56$ and $P_1 = 3.381$, (c) Average silhouette, (d) Thresholded average silhouette classified as car $rectangularity = 0.68$ and $P_1 = -1.602$.



(a) (b) (c)

Figure 4.7. Example van silhouettes (a) Silhouette from a single frame which is eliminated since $\rho = 0.548$, (b) Average silhouette, (c) Thresholded average silhouette which is not eliminated since.$\rho = 0.823$.

silhouette, whereas it is misclassified with using a single silhouette. Figure 4.7 shows an example where a van has passed the detection phase with average silhouette method, but failed with the single silhouette method. Such cases constitute the main performance difference between the two compared methods.

We also examined the performance of "consensus of silhouettes" with flowchart method. Training set consists of annotated silhouettes. Thresholds, and SVM model used in scheme (cf. Figure 3.1) are obtained from the training set. As mentioned before, in consensus approach we require a predefined percentage of the samples make the same prediction. Table 4.4 shows classification accuracies when required consensus percentage changes from 70% to 34% (winner takes all). 34% is the lowest possible consensus percentage since after this value, the chosen class is no longer the largest group. Values in the table correspond to what percentage of the instances of a vehicle type is classified correctly. When we increase the percentage, samples which their consensus value is less than this number are assumed to be misclassified (i.e. false negative). Decreasing percentage more does not increase the accuracy because it no longer becomes the largest group. Number of frames used is not fixed for every sample, instead we fixed the angle range in the omnidirectional image as depicted in Figure 2.2. According to these results, for the

flowchart method, average silhouette approach has the highest performance. The overall performance of consensus (34%) approach is slightly below the single frame silhouette approach.

An important point is the required time to compute the features in the flowchart method. In our analysis we saw that computing P_1 takes 5.46 seconds, while the rest of the features take only 7 milliseconds.

Table 4.4. Classification accuracies for each class for consensus approach. Required consensus percentage changes from 70% to 34%.

Threshold	Motorcycle	Car	Van	Overall
70%	83%	50%	40%	52%
60%	87%	63%	55%	64%
50%	90%	71%	67%	73%
40%	93%	73%	67%	74%
34%	95%	73%	67%	75%

4.2. K Nearest Neighbor Experiments

As mentioned before, we also examined the classification performance of kNN. Figure 4.8a shows the features of the annotated silhouettes of all samples (using Euclidean distance) in 3D where dimensions are rectangularity elongation and convexity. Actual labels are indicated with different colors. We see that looking at a certain number of nearest neighbors of new sample can help us to determine the sample's label. Top-view of Figure 4.8a is shown in Figure 4.8b, where x and y axes refer to rectangularity and elongation respectively. It can be observed that elongation plays a dominant role to discriminate motorcycle class from others. Figure 4.8c shows the 2D space with dimensions convexity and rectangularity. Rectangularity is not adequate to discriminate cars from vans. With the help of convexity and elongation, car/van classification becomes more accurate.

By dividing the dataset as train and test parts randomly and repeating the experiments three times, we computed average accuracies for different K values. As stated in (Duda et al., 2012), selection of K is crucial. When K is small, noise can affect classification. Otherwise, when K is large, computation costs increase. In our experiments, K is selected 5, 10, and 15, and the results are quite similar to each other. Table 4.5, 4.8,

and 4.11 depicts averaged silhouettes, single frame silhouettes, and consensus of silhouettes kNN classification results when K is selected 5, 10, and 15 respectively. We again observe that average silhouette is the best performing approach. Performance of consensus approach is not as good as average silhouette, but it is considerably better than using single frame silhouettes. When K is selected 5, Tables 4.6, and 4.7 depict the number of correctly classified and misclassified samples for each class with the average silhouette and single frame silhouette methods respectively. When K is selected 10, Tables 4.9, and 4.10 depict the number of correctly classified and misclassified samples for each class with the average silhouette and single frame silhouette methods respectively. When K is selected 15, Tables 4.12, and 4.13 depict the number of correctly classified and misclassified samples for each class with the average silhouette and single frame silhouette methods respectively. Numbers in the Tables 4.6, 4.7, 4.9, 4.10, 4.12, and 4.13 are total correctly classified and misclassified samples of the three folds. False negatives (FN) are the missed samples.

Since P_1 feature is not used in kNN classification, calculation of features is much faster than the flowchart method. Regarding the two multi-frame approaches, although the performance of consensus approach is lower than average silhouette approach, it is more time efficient. Total time for consensus approach with kNN classification is 250 ms including silhouette and feature extraction (assuming 10 frames are used), whereas average silhouette with kNN takes 1850 ms. We used KD-tree implementation, as we mentioned in Section 3.2, rather than exhaustive search considering time complexity to classify a new sample. The required time for classification a new sample with kNN classification is only 3 milliseconds using the KD-tree implementation when K is selected 5.

Table 4.5. Classification accuracies with kNN ($K = 5$) for the average silhouette, consensus of silhouettes and single frame silhouette approaches.

	Motorcycle	Car	Van	Overall
Average silhouette	97%	98%	99%	98%
Consensus of silhouettes	95%	58%	100%	80%
Single frame silhouette	53%	53%	72%	60%

Table 4.6. Confusion matrix for the approach of using average silhouette classified with kNN ($K = 5$) as sum of three folds (For each fold there are 20 motorcycles, 50 cars, and 42 vans in test set).

	Ground truth	Motorcycle	Car	Van
Detection	Motorcycle	58	0	0
	Car	0	147	1
	Van	2	3	125
	FN	0	0	0

Table 4.7. Confusion matrix for the approach of using single silhouette classified with kNN ($K = 5$) as sum of three folds (For each fold there are 20 motorcycles, 50 cars, and 42 vans in test set).

	Ground truth	Motorcycle	Car	Van
Detection	Motorcycle	32	63	5
	Car	5	79	30
	Van	23	7	91
	FN	0	0	0

Table 4.8. Classification accuracies with kNN ($K = 10$) for the average silhouette, consensus of silhouettes and single frame silhouette approaches.

	Motorcycle	Car	Van	Overall
Average silhouette	97%	98%	100%	99%
Consensus of silhouettes	95%	57%	100%	80%
Single frame silhouette	53%	53%	73%	61%

Table 4.9. Confusion matrix for the approach of using average silhouette classified with kNN ($K = 10$) as sum of three folds (For each fold there are 20 motorcycles, 50 cars, and 42 vans in test set).

	Ground truth	Motorcycle	Car	Van
Detection	Motorcycle	58	0	0
	Car	0	147	0
	Van	2	3	126
	FN	0	0	0

Table 4.10. Confusion matrix for the approach of using single silhouette classified with kNN ($K = 10$) as sum of three folds (For each fold there are 20 motorcycles, 50 cars, and 42 vans in test set).

	Ground truth	Motorcycle	Car	Van
Detection	Motorcycle	32	55	4
	Car	5	80	30
	Van	23	15	92
	FN	0	0	0

Table 4.11. Classification accuracies with kNN ($K = 15$) for the average silhouette, consensus of silhouettes and single frame silhouette approaches.

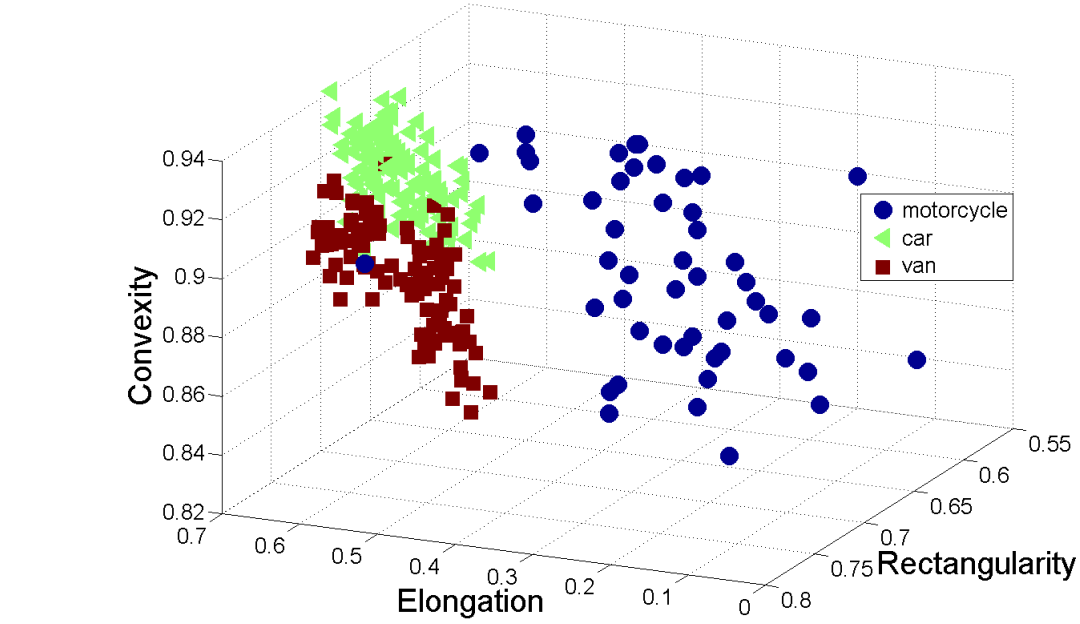
	Motorcycle	Car	Van	Overall
Average silhouette	97%	99%	99%	99%
Consensus of silhouettes	95%	56%	100%	79%
Single frame silhouette	43%	53%	74%	59%

Table 4.12. Confusion matrix for the approach of using average silhouette classified with kNN ($K = 15$) as sum of three folds (For each fold there are 20 motorcycles, 50 cars, and 42 vans in test set).

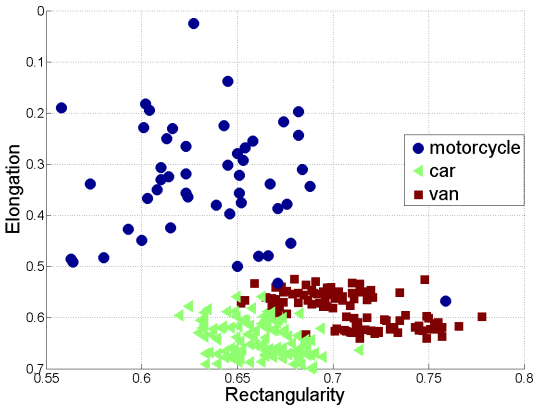
	Ground truth	Motorcycle	Car	Van
Detection	Motorcycle	58	0	0
	Car	0	148	1
	Van	2	2	125
	FN	0	0	0

Table 4.13. Confusion matrix for the approach of using single silhouette classified with kNN ($K = 15$) as sum of three folds (For each fold there are 20 motorcycles, 50 cars, and 42 vans in test set).

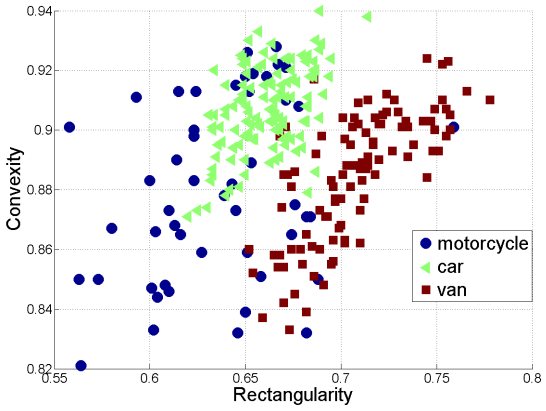
	Ground truth	Motorcycle	Car	Van
Detection	Motorcycle	26	50	3
	Car	6	79	30
	Van	28	21	93
	FN	0	0	0



(a)



(b)



(c)

Figure 4.8. Extracted features of the annotated silhouettes. (a) All dimensions. (b) First two dimensions. (c) Last two dimensions.

4.3. Deep Neural Network Experiments

As we mentioned earlier, we also examined the classification performance of DNN. As stated in (Hinton and Salakhutdinov, 2006), pretraining helps to generalize DNN's. Gradient descent is also used for fine-tuning of the weights of our DNN. Fine-tuning should be applied to all 165 training samples which corresponds to approximately 60% of the total dataset, the resultant system should be tested on the remaining 112 test samples, approximately 40% of the total dataset.

We trained two different DNN's using the Matlab example in MathWorks (2015). First one is for average silhouettes, and the other is for consensus of silhouettes and single silhouette. We can summarize the one trained for average silhouettes as follows: After layer-by-layer pretraining, and fine tuning in a (240 x 180)-100-50-10-3 network, backpropagation using steepest descent and a small learning rate achieves 95% overall classification accuracy for average silhouette. The whole system is shown in Figure 4.9, which is also the combination of the layers (cf. Figures 3.3, 3.4, and 3.5) mentioned in Section 3.3. It should be noted that similar to flowchart, and kNN experiments average silhouettes are trained using average silhouettes. Some of the learned features from the first layer is shown in Figure 4.10. It can be seen that shapes of vehicles can be discriminated from each other. If there is an accumulation on the center of blob, it is tend to be classified as motorcycle. Otherwise, it is tend to be classified as van.

Our second DNN is trained for consensus of silhouettes and single silhouette approaches using hand labelled groundtruth silhouettes. Again we trained (240 x 180)-100-50-10-3 network, and we obtained 92% overall classification accuracy for consensus of silhouettes, and 90% overall classification accuracy for single frame silhouette. The scheme of whole system is shown in Figure 4.9. Some of the learned features from the first layer is shown in Figure 4.11. Similar to Figure 4.10 accumulation on the center of blobs makes them to be classified more likely to motorcycle, accumulation on the border of blobs makes them to be classified more likely to van.

Again, similar to Flowchart and kNN experiments, by dividing the dataset as train and test parts randomly and repeating the experiments three times, we computed average accuracies for averaged silhouettes, single frame silhouettes, and consensus of silhouettes. Table 4.14 depicts average of three DNN classification experiments' results. Values in the table correspond to what percentage of the instances of a vehicle type is classified correctly. Tables 4.15 and 4.16 depict the number of correctly classified and misclassified

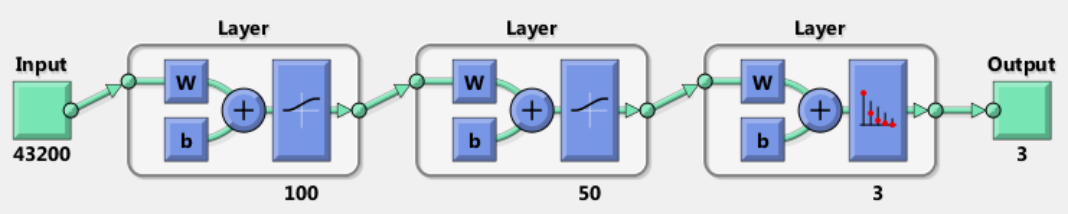


Figure 4.9. Whole Deep Neural Network system used to train and test of average silhouette, consensus of silhouettes and single silhouette.

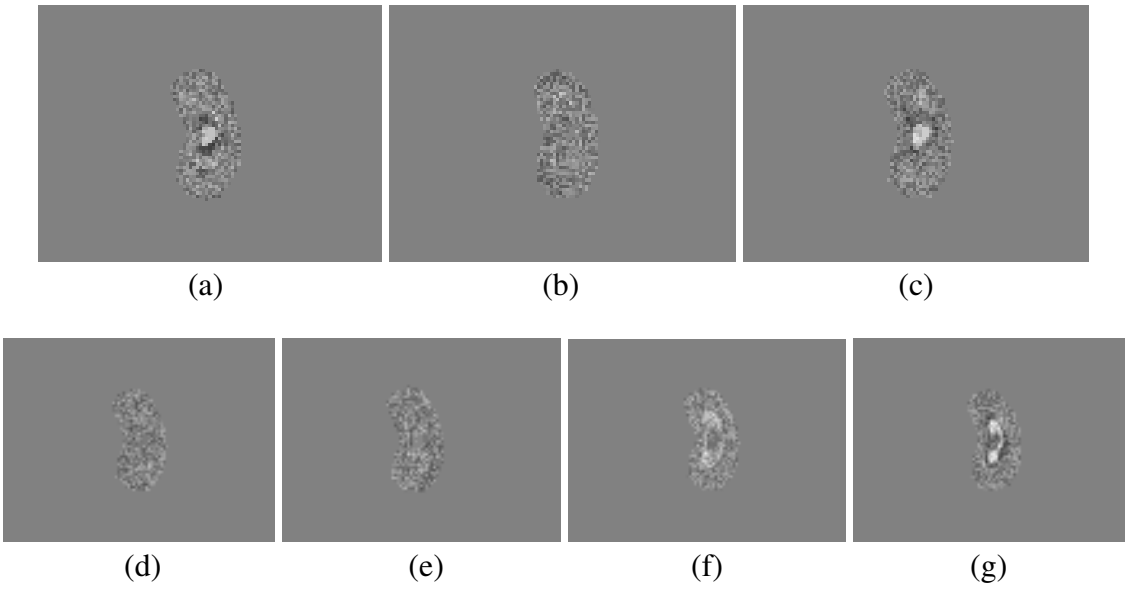


Figure 4.10. Example features obtained from the training of DNN using average silhouettes.

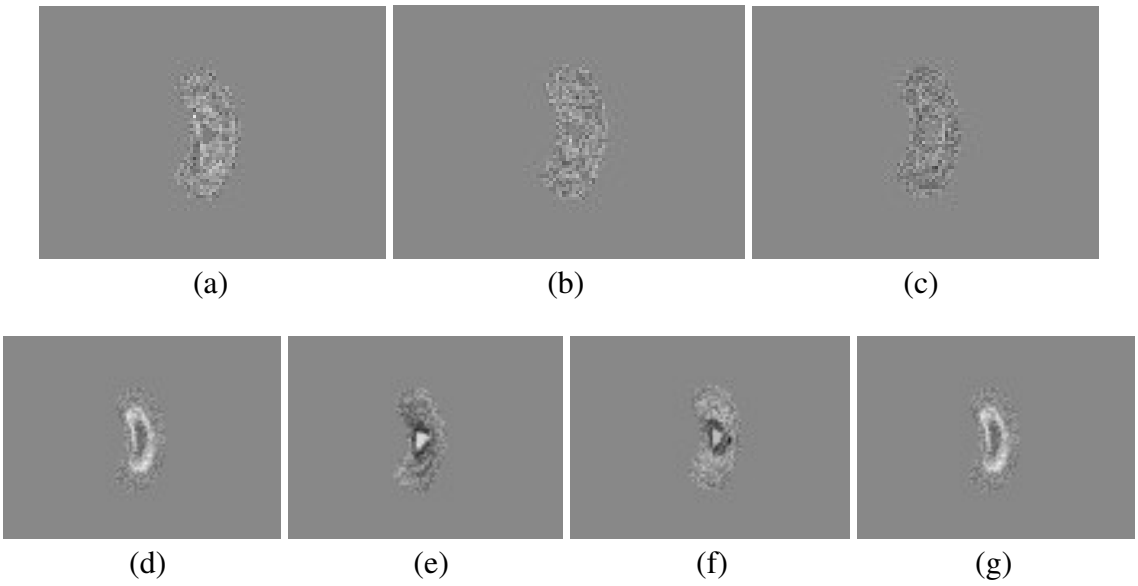


Figure 4.11. Example features obtained from the training of DNN using groundtruth silhouettes.

samples for each class with the average silhouette and single frame silhouette methods respectively. Numbers in the Tables 4.15 and 4.16 are total correctly classified and misclassified samples of the three folds. False negatives (FN) are the missed samples.

In spite of DNN's long training intervals, total time required to classify a sample is quite short. While total time required to classify an averaged silhouette and single silhouette is 3.6 milliseconds, and for the consensus approach (assuming 10 frames are classified) it is 36 milliseconds.

Table 4.14. Average classification accuracies with DNN for the average silhouette, consensus of silhouettes and single frame silhouette approaches.

	Motorcycle	Car	Van	Overall
Average silhouette	97%	95%	94%	95%
Consensus of silhouettes	100%	95%	84%	92%
Single frame silhouette	95%	85%	94%	90%

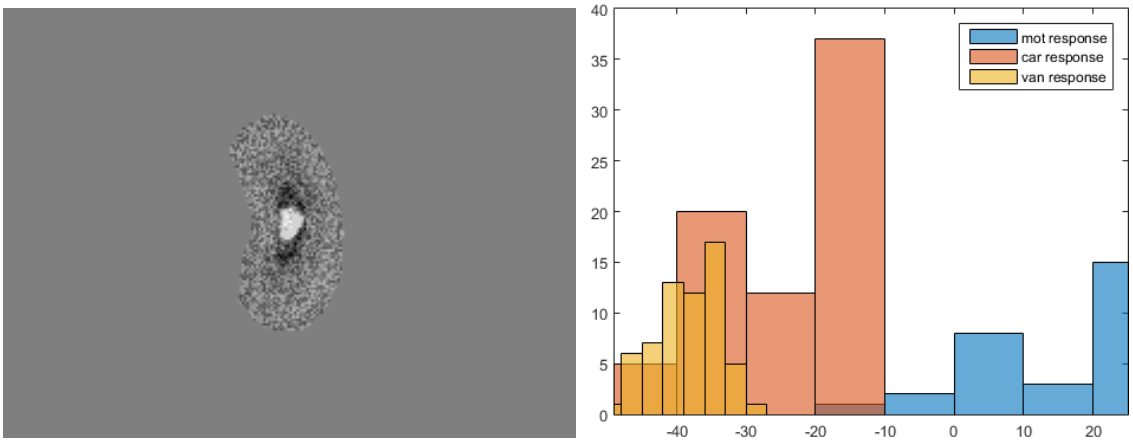
Table 4.15. Confusion matrix for the approach of using average silhouettes classified with DNN as sum of three folds (For each fold there are 20 motorcycles, 50 cars, and 42 vans in test set).

	Ground truth	Motorcycle	Car	Van
Detection	Motorcycle	58	0	0
	Car	1	143	7
	Van	1	7	119
	FN	0	0	0

Table 4.16. Confusion matrix for the approach of using single silhouette classified with DNN as sum of three folds (For each fold there are 20 motorcycles, 50 cars, and 42 vans in test set).

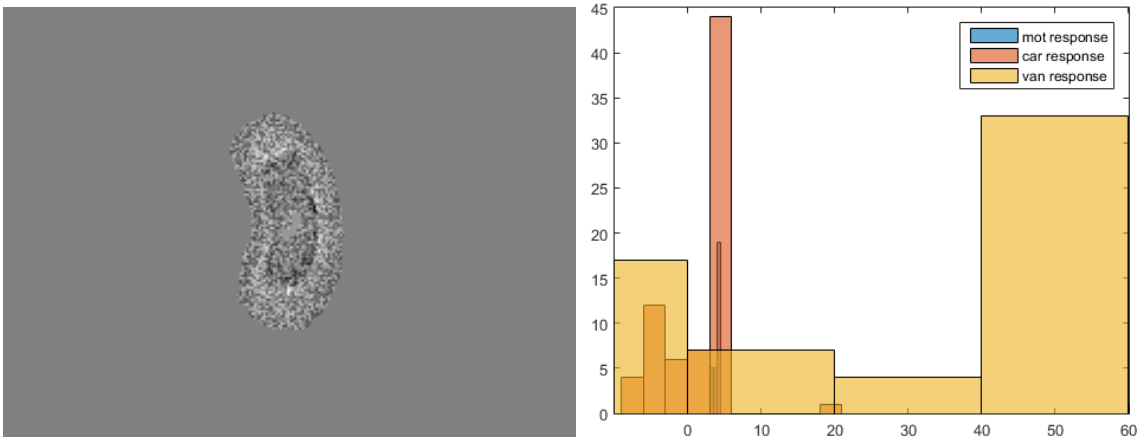
	Ground truth	Motorcycle	Car	Van
Detection	Motorcycle	57	5	1
	Car	3	127	7
	Van	0	18	118
	FN	0	0	0

We also made an experiment to see how classes response changes when we use different kind of features. Using five features, we calculated response in training set between three major vehicle types. Figure 4.12a, Figure 4.13a, Figure 4.14a, Figure 4.15a, and Figure 4.16a shows the five sample features we used. Figure 4.12b, Figure 4.13b, Figure 4.14b, Figure 4.15b, and Figure 4.16b shows motorcycle, car, and van response for each feature. Feature shown in Figure 4.12, shows positive response only to motorcycle class, since positive weights are accumulated at the center of blob. Feature shown in Figure 4.13, shows positive dominant response to van class, since weights are arranged mostly negative close to center of blob. Feature shown in Figure 4.14, shows positive dominant response to car class, since weights are arranged mostly positive close to center of blob. Feature shown in Figure 4.15, shows positive dominant response to van class, and also to motorcycle class. This feature presents also a negative dominant response to car class, since positive weights are positioned at the center and boundary of blob. Feature shown in Figure 4.16, shows positive dominant response to car class, and also negative responses to van and motorcycle classes, since negative weights are positioned at the center and boundary of blob.



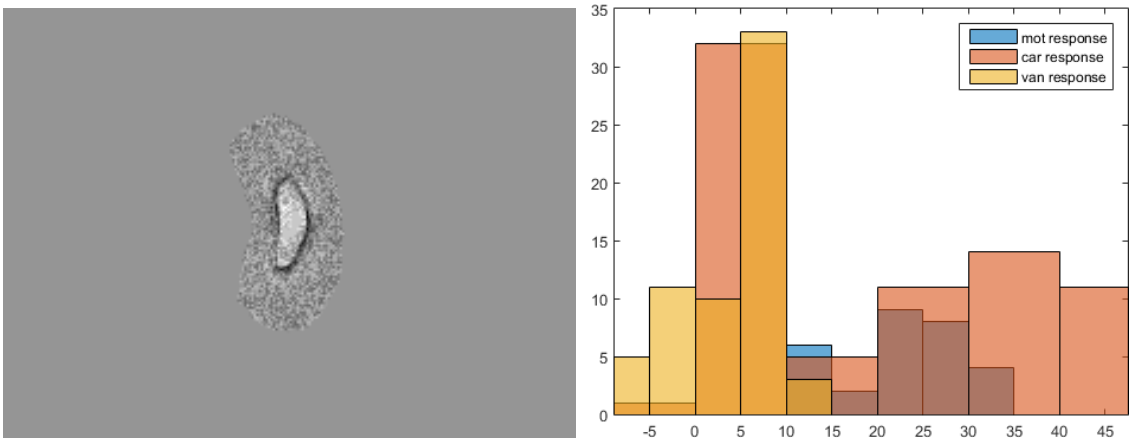
(a) (b)

Figure 4.12. Class response change by changing sample features. (a) Selected feature. (b) Response of three major vehicle type.



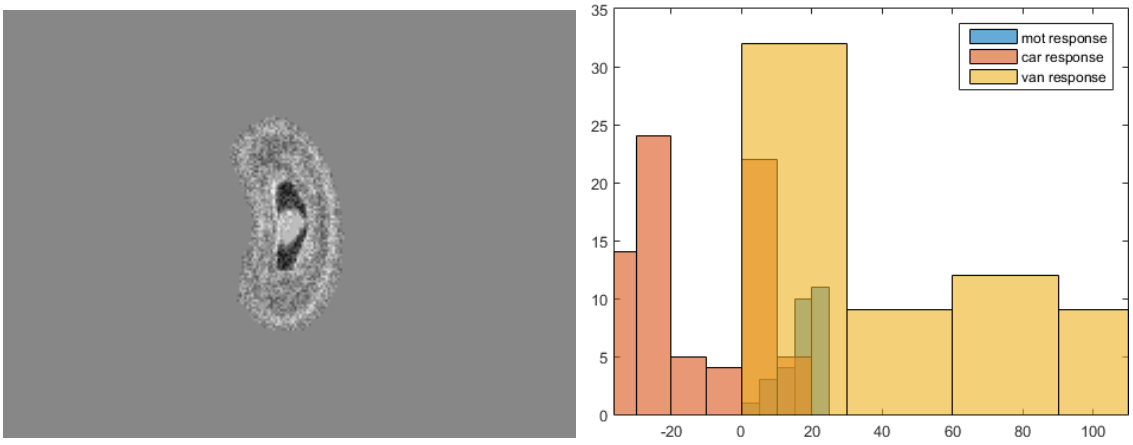
(a) (b)

Figure 4.13. Class response change by changing sample features. (a) Selected feature. (b) Response of three major vehicle type.



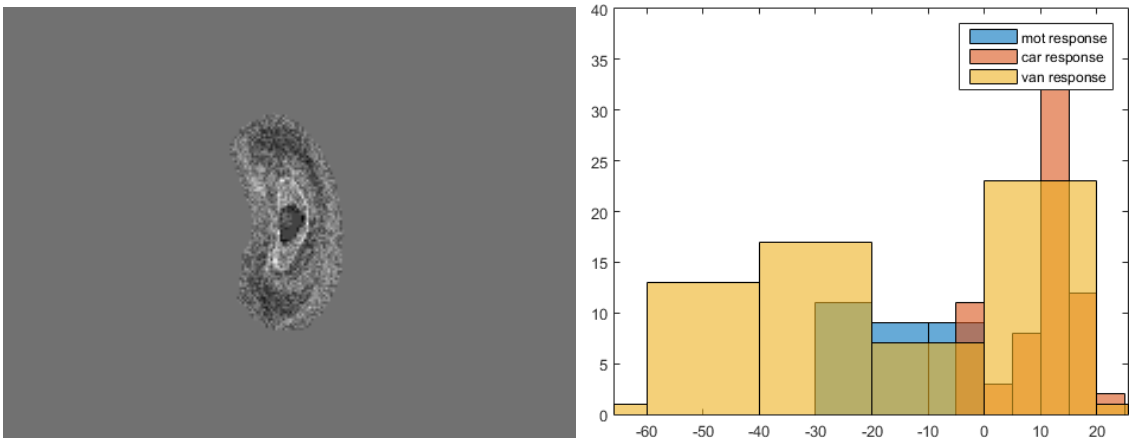
(a) (b)

Figure 4.14. Class response change by changing sample features. (a) Selected feature. (b) Response of three major vehicle type.



(a) (b)

Figure 4.15. Class response change by changing sample features. (a) Selected feature. (b) Response of three major vehicle type.



(a) (b)

Figure 4.16. Class response change by changing sample features. (a) Selected feature. (b) Response of three major vehicle type.

4.4. Comparison of Computation Times

In this section, we summarize, and compare computation times of the methods which are explained in details in Sections 4.1, 4.2, and 4.3. Table 4.20 summarizes total computation times to classify a new sample in milliseconds unit. This is the total time after background subtraction, including silhouette extraction, feature extraction, and classification. For multiple silhouettes cases, average silhouettes and consensus of silhouettes, we assume 10 frames are used in calculation of computation times.

We can summarize our inference from Table 4.20, which is combination of silhouette extraction times (cf. Table 4.17), feature extraction times (cf. Table 4.18), and classification times (cf. Table 4.19) as follows. Since flowchart method contains calculation of P_1 feature, total computation time to classify a new sample is too much with this method. When we compare kNN and DNN classification methods, computation time of consensus of silhouettes are much less rather than average silhouettes'. However, as we present in Sections 4.2, and 4.3 the average silhouettes' classification accuracy is superior than the consensus of silhouettes' classification accuracy. Table 4.20 makes clear that there is a trade-off between total computation times and classification accuracy of the two compared methods, average silhouette and consensus of silhouettes. According to requirements of the application area, one of them can be chosen.

In this study, we analyzed the results of five background subtraction algorithms,

Table 4.17. Computation time to extract silhouettes of a new sample for the average silhouette, consensus of silhouettes and single frame silhouette approaches (in milliseconds).

	Average silhouette	Consensus of silhouettes	Single frame silhouette
Required time	1840	150	4.3

Table 4.18. Computation time to extract features of a new sample for the average silhouette, consensus of silhouettes and single frame silhouette approaches (in milliseconds).

	Flowchart Method	kNN	DNN
Average silhouette	5467	7	0
Consensus of silhouettes	54670	70	0
Single frame silhouette	5467	7	0

namely Adaptive Background Subtraction, GMG (Godbehere et al., 2012), LB Adaptive SOM (Maddalena and Petrosino, 2008), Mixture Of Gaussian V1BGS (KaewTraKulPong and Bowden, 2002), and Multi Layer BGS (Yao and Odobez, 2007) reviewed in (Sobral and Vacavant, 2014). After reviewing the quality of these algorithms, we decided to work on the resultant foreground masks of our videos produced by (Yao and Odobez, 2007), since its background subtraction is more successful. However, required time to obtain those masks is almost 2 seconds for a video frame. According to (Sobral and Vacavant, 2014), there is a trade-off between algorithms accuracy and time complexities. For our videos (resolution is 810x1440 pixels) required time to obtain foreground masks for the reviewed algorithms are presented in Table 4.21.

Table 4.19. Computation time to classify a new sample for the average silhouette, consensus of silhouettes and single frame silhouette approaches (in milliseconds).

	Flowchart Method	kNN	DNN
Average silhouette	3.28	3	3.6
Consensus of silhouettes	32.8	30	36
Single frame silhouette	3.28	3	3.6

Table 4.20. Total computation time to classify a new sample for the average silhouette, consensus of silhouettes and single frame silhouette approaches (in milliseconds).

	Flowchart Method	kNN	DNN
Average silhouette	7310.28	1850	1843.6
Consensus of silhouettes	54852.8	250	186
Single frame silhouette	5470.8	14.3	8

Table 4.21. Total computation time to obtain foreground masks (in milliseconds/frame).

	ABS	GMG	LBAdaptiveSOM	MOG V1BGS	MultiLayerBGS
Required time	411	167	430	1390	2093

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

We proposed to use multiple frames of a video for shape based classification of vehicles. We applied three different methods of classification and compared the performance of using a silhouette from a single frame with the performance of using multiple frames. The first one is using features one after another in a flowchart to classify the vehicles. The second one is kNN classification. We decided to include this method in our experiments because using single frame silhouette with kNN classification can be considered as the benchmark method in shape based vehicle classification. Lastly we compared single and multiple frame approaches with a DNN classifier. We included this experiment in our study because DNN became very popular in recent years. Results of the experiments indicate a significant improvement in classification accuracy by using multiple frames.

When two alternative approaches of using multiple frames are compared, average silhouette has a higher performance than using consensus of decisions of multiple frames. However, consensus approach has the advantage of being computationally cheaper.

In essence, the advantage of the proposed approach is utilizing the information available in a longer time interval rather than a single frame. Therefore the improvement can be expected for other objects types and domains other than traffic applications.

We use a portable image acquisition platform and our method is independent of the distance between the camera and the objects which is more practical than the previously proposed methods that fix the cameras to buildings and use the object's area as a feature since the distance to objects stays same.

For shape based object detection, template based algorithms also exist, for example, Chamfer Matching. We also made experiments using Chamfer Matching. For different training sets, we obtained different templates. According to our experiments for similar templates, detection results were quite inconsistent, and unreliable. We observed that the method is pretty sensitive to little changes in the template. Due to this reason, we did not include the results of Chamfer Matching algorithm to this thesis.

Currently, we can detect and classify three different vehicle types namely, motor-

cycle, car, and van (minibus) in omnidirectional videos. In the future, we may increase number of classes detected and classified by our method. For example, we can record human, truck, bus, SUV videos with current setup and add them to the dataset.

We did not cover scenes that contain occlusion. In the future, we may focus to this issue. Using object tracking algorithms we can find a solution to classification of occluded silhouettes.

There may be other use of multiple silhouettes other than we discussed. Current use of multiple silhouettes gives equal weight to each silhouette. For instance, one can change this scheme by giving more importance to the silhouettes positioned closer to center angle. The performance of this scheme can be investigated in the future.

REFERENCES

Amine Iraqui, H., Y. Dupuis, R. Boutteau, J. Ertaud, and X. Savatier (2010, Sept). Fusion of omnidirectional and ptz cameras for face detection and tracking. In *Emerging Security Technologies (EST), 2010 International Conference on*, pp. 18–23.

Avery, R., Y. Wang, and G. Scott Rutherford (2004, Oct). Length-based vehicle classification using images from uncalibrated video cameras. In *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, pp. 737–742.

Bastanlar, Y., A. Temizel, Y. Yardimci, and P. Sturm (2012). Multi-view structure-from-motion for hybrid camera scenarios. *Image and Vision Computing 30*(8), 557 – 572.

Bradski, G. and A. Kaehler (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media.

Buch, N., J. Orwell, and S. Velastin (2008, July). Detection and classification of vehicles for urban traffic scenes. In *Visual Information Engineering, 2008. VIE 2008. 5th International Conference on*, pp. 182–187.

Buch, N., S. Velastin, and J. Orwell (2011, Sept). A review of computer vision techniques for the analysis of urban traffic. *Intelligent Transportation Systems, IEEE Transactions on 12*(3), 920–939.

Chen, Z. and T. Ellis (2011, Dec). Multi-shape descriptor vehicle classification for urban traffic. In *Digital Image Computing Techniques and Applications (DICTA), 2011 International Conference on*, pp. 456–461.

Chen, Z., T. Ellis, and S. Velastin (2011, Oct). Vehicle type categorization: A comparison of classification schemes. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pp. 74–79.

Chen, Z., T. Ellis, and S. Velastin (2012, Sept). Vehicle detection, tracking and classification in urban traffic. In *Intelligent Transportation Systems (ITSC), 2012 15th*

International IEEE Conference on, pp. 951–956.

Cinaroglu, I. and Y. Bastanlar (2014, April). A direct approach for human detection with catadioptric omnidirectional cameras. In *Signal Processing and Communications Applications Conference (SIU), 2014 22nd*, pp. 2275–2279.

Cinaroglu, I. and Y. Bastanlar (2015). A direct approach for object detection with catadioptric omnidirectional cameras. *Signal, Image and Video Processing*, 1–8.

Dalal, N. and B. Triggs (2005, June). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, Volume 1, pp. 886–893 vol. 1.

Dedeoglu, Y., B. Toreyin, U. Gudukbay, and A. Cetin (2006). Silhouette-based method for object classification and human action recognition in video. In *Computer Vision in Human-Computer Interaction*, Volume 3979 of *Lecture Notes in Computer Science*, pp. 64–77. Springer Berlin Heidelberg.

Duda, R. O., P. E. Hart, and D. G. Stork (2012). *Pattern classification*. John Wiley & Sons.

Dupuis, Y., X. Savatier, J. Ertaud, and P. Vasseur (2011, Sept). A direct approach for face detection on omnidirectional images. In *Robotic and Sensors Environments (ROSE), 2011 IEEE International Symposium on*, pp. 243–248.

Erhan, D., C. Szegedy, A. Toshev, and D. Anguelov (2014). Scalable object detection using deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 2155–2162. IEEE.

Felzenszwalb, P., D. McAllester, and D. Ramanan (2008, June). A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8.

Gandhi, T. and M. Trivedi (2007, June). Video based surround vehicle detection, classification and logging from moving platforms: Issues and approaches. In *Intelligent*

Vehicles Symposium, 2007 IEEE, pp. 1067–1071.

Godbehere, A., A. Matsukawa, and K. Goldberg (2012, June). Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation. In *American Control Conference (ACC), 2012*, pp. 4305–4312.

Goedeme, T., M. Nuttin, T. Tuytelaars, and L. Van Gool (2007). Omnidirectional vision based topological navigation. *Int Journal of Computer Vision 74*(3), 219–236.

Gupte, S., O. Masoud, R. Martin, and N. Papanikolopoulos (2002, Mar). Detection and classification of vehicles. *Intelligent Transportation Systems, IEEE Transactions on 3*(1), 37–47.

Han, J. and B. Bhanu (2006, Feb). Individual recognition using gait energy image. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 28*(2), 316–322.

Hasegawa, O. and T. Kanade (2005). Type classification, color estimation, and specific target detection of moving targets on public streets. *Machine Vision and Applications 16*(2), 116–121.

Hinton, G. E. and R. R. Salakhutdinov (2006). Reducing the dimensionality of data with neural networks. *Science 313*(5786), 504–507.

Hu, M.-K. (1962, February). Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on 8*(2), 179–187.

Ji, P., L. Jin, and X. Li (2007, Aug). Vision-based vehicle type classification using partial gabor filter bank. In *Automation and Logistics, 2007 IEEE International Conference on*, pp. 1037–1040.

KaewTraKulPong, P. and R. Bowden (2002). An improved adaptive background mixture model for real-time tracking with shadow detection. In *Video-based surveillance systems*, pp. 135–144. Springer.

Karaimer, H. C. and Y. Bastanlar (2014, April). Car detection with omnidirectional cam-

eras using haar-like features and cascaded boosting. In *Signal Processing and Communications Applications Conference (SIU), 2014 22nd*, pp. 301–304.

Karaimer, H. C. and Y. Bastanlar (2015). Detection and classification of vehicles from omnidirectional videos using temporal average of silhouettes. In *Proceedings of the Int. Conference on Computer Vision Theory and Applications*, pp. 197–204.

Khoshabeh, R., T. Gandhi, and M. Trivedi (2007, Sept). Multi-camera based traffic flow characterization and classification. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pp. 259–264.

Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105.

Kumar, P., S. Ranganath, H. Weimin, and K. Sengupta (2005, March). Framework for real-time behavior interpretation from traffic video. *Intelligent Transportation Systems, IEEE Transactions on 6*(1), 43–53.

Luo, Q., T. Khoshgoftaar, and A. Folleco (2006, Sept). Classification of ships in surveillance video. In *Information Reuse and Integration, 2006 IEEE International Conference on*, pp. 432–437.

Maddalena, L. and A. Petrosino (2008, July). A self-organizing approach to background subtraction for visual surveillance applications. *Image Processing, IEEE Transactions on 17*(7), 1168–1177.

MathWorks (2015). Training a deep neural network for digit classification. `http://www.mathworks.com/help/nnet/examples/training-a-deep-neural-network-for-digit-classification.html`. Accessed: 2015-04-21.

Mithun, N., N. Rashid, and S. Rahman (2012, Sept). Detection and classification of vehicles from video using multiple time-spatial images. *Intelligent Transportation Systems, IEEE Transactions on 13*(3), 1215–1225.

Morris, B. and M. Trivedi (2006a, Nov). Improved vehicle classification in long traffic video by cooperating tracker and classifier modules. In *Video and Signal Based Surveillance, 2006. AVSS '06. IEEE International Conference on*, pp. 9–9.

Morris, B. and M. Trivedi (2006b, Sept). Robust classification and tracking of vehicles in traffic video streams. In *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pp. 1078–1083.

Rashid, N., N. Mithun, B. Joy, and S. Rahman (2010, Dec). Detection and classification of vehicles from a video using time-spatial image. In *Electrical and Computer Engineering (ICECE), 2010 International Conference on*, pp. 502–505.

Scotti, G., L. Marcenaro, C. Coelho, F. Selvaggi, and C. Regazzoni (2005, April). Dual camera intelligent sensor for high definition 360 degrees surveillance. *Vision, Image and Signal Processing, IEE Proceedings - 152*(2), 250–257.

Sethna, B., M. John, P. Palaniappan, and S. Ganapathi (2012, February 2). System and method for classification of moving object during video surveillance. US Patent App. 13/194,706.

Sivaraman, S. and M. Trivedi (2013, June). A review of recent developments in vision-based vehicle detection. In *Intelligent Vehicles Symposium, 2013 IEEE*, pp. 310–315.

Sobral, A. and A. Vacavant (2014). A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding 122*(0), 4 – 21.

Szegedy, C., A. Toshev, and D. Erhan (2013). Deep neural networks for object detection. In *Advances in Neural Information Processing Systems*, pp. 2553–2561.

Yang, M., K. Kpalma, J. Ronsin, et al. (2008). A survey of shape feature extraction techniques. *Pattern recognition*, 43–90.

Yao, J. and J. Odobez (2007). Multi-layer background subtraction based on color and texture. In *IEEE Conf. on Computer Vision and Pattern Recognition, 2007.*, pp. 1–8.