Vehicle Identification and Classification System

A

Project Report

Submitted for the Partial Fulfilment

of B.Tech Degree

in

Information Technology

By

Vishal Polley (1505213053)

Under the Supervision of

Ms. Shipra Gautam

Department of Computer Science and Engineering

Institute of Engineering and Technology, Lucknow

Dr. A. P. J. Abdul Kalam Technical University, Uttar Pradesh, Lucknow

June 2019

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

List of Figures

List of Tables

CERTIFICATE

This is to Certify that the project entitled **"Vehicle Identification and Classification System"** submitted by **Vishal Polley** (1505213053), in the partial fulfilment for the award of degree of Bachelor of Technology in Information Technology is a record of work carried out by him under my supervision at the **Department of Computer Science and Engineering**, **Institute of Engineering and Technology, Lucknow.**

Ms. Shipra Gautam

Assistant Professor

Department of Computer Science and Engineering

Institute of Engineering and Technology

Dr. A.P.J. Abdul Kalam Technical University,

Uttar Pradesh, India

DECLARATION

I here by declare that the project entitled **"Vehicle Identification and Classification System"** submitted by me, in the partial fulfilment for the award of **Bachelor of Technology** in **Information Technology,** is a record of bona-fide work carried out by me under the supervision and guidance of **Ms. Shipra Gautam** at the **Department of Computer Science and Engineering**, **Institute of Engineering and Technology, Lucknow.** This project has not been submitted by me at any other institute for the requirement of any other degree.

Date: 04/06/2019 Signature of Student:

Vishal Polley (1505213053)

ACKNOWLEGEMENT

The completion of any inter-disciplinary project depends upon completion, co-ordination and combined efforts of several sources of knowledge. I am grateful to my project supervisor, **Ms. Shipra Gautam,** for her even willingnesss to give me valuable advice and direction whenever I approached her with a problem. I am thankful to her for providing immense guidence for this project.

I owe special debt of gratitude to my project coordinator, **Mrs. Tulika Narang,** for her constant support and guidance throughout the course of work. Her sincerity, thoroughness and perseverance have been a constant source of inspiration for me.

I deeply express my sincere thanks to our Head of Department **Dr. Y. N. Singh** for encouraging and allowing me to present the project on the topic **"Vehicle Identification and Classification System"** at department premises for the partial fulfilment of the requirements leading to the award of B. Tech degree.

I take this opportunity to thank all my lecturers who have directly or indirectly helped in the completion of the project.

Vishal Polley (1505213053)

ABSTRACT

Feature extraction and classification are two most important modules for any vision-based object recognition system. In the case of vehicles, most of the methods in these modules found to be less accurate in recognition even though they work well for other objects. We are interested in recognition of vehicles on Indian roads. There are number of challenges in implementing vehicle recognition in Indian scenario like bad road conditions, traffic rules violation and variance among vehicles, etc. In order to overcome these difficulties, we implemented feature extraction module using bag of features (combination of Harris-corner detector and SIFT features), and classiffication is performed using Support vector machines (SVM). To validate our proposed method, we have used Indian Vehicle Database. The images in this database are extracted from day-light Indian urban traffic scenes. Our proposed method achieves 40-45 percent improvement over the baseline methods.

In this report, we also summarize our approach to the tasks of detecting and localizing these classes of objects e.g. as cars, trucks, pedestrians, etc. in the MIO-TCD dataset. Specifically, we will go over our methodology and results for data preprocessing, localization and classification methods. Finally, we will conclude the report; tackling localization and classification on the dataset with our deep learning models.

1. Introduction

Computer vision is an important field of artificial intelligence where decision about real world scene having high dimensional data is taken. The general steps used in this process are acquiring, processing and analyzing the image and convert it into numerical or symbolic form. It is used to understand the scene electronically and the process is equivalent to the ability of human vision. The numerical or symbolic information of a scene is decided based on the appropriate model constructed with the support of object geometry, physics, statistic, and learning theory. The scene under consideration is converted into the image(s) or the video(s), comprising of many images, using camera(s) focused from different locations on a scene. The various vision related areas such as seen reconstruction, event detection, video tracking, object recognition, object pose estimation and image restoration are considered as subareas of computer vision. Similarly, various other fields such as image processing, image analysis and machine vision are also closely related to computer vision. The techniques and applications of various above said areas overlap with each other. Moreover, the techniques used in all these areas are more or less identical. The difference in names only lies on the applications where the techniques are applied.

Image processing and image analysis both deals with 2D images. In image processing an image is transformed into another by applying some operations such as contrast enhancement, edge detection, noise removal and geometrical transformations. The image contents are not interpreted in image processing whereas in computer vision the interpretation of images is made based on the properties of the contents they contain. Computer vision may include analysis of 3D images from 2D.

In recent years, Computer Vision based vehicle recognition (recognizing the vehicle in a digital image or video sequence) has become an active research area in Intelligent Transportation Systems (ITS). This is mainly due to impact in numerous applications like electronic toll collection management (collect tolls on highways electronically to eliminate delay), to identify unauthorized vehicles on roads as a part of vehicle surveillance and traffic data analysis (useful in decision making in terms of safety evaluation, pavement design, funding, forecasting, and modeling). The other applications of vehicle recognition are, license plate localization, computer assisted driving and methods for reducing road accidents.

Although vehicle recognition has been an area of recent interest to the Computer Vision community, no prior research study has been used to build an on-road vehicle recognition. The main reason is vehicle recognition is quite different from other object recognition scenarios. This is mainly due to the following two difficulty's. First, since vehicle recognition task deals with only images in outdoor or natural lighting it is known to cause noise related problems. The other is typical geometry of vehicles. Vehicles chassis are constructed in a significantly greater variety of geometries as compared to other objects. There are number of dissimilarities in vehicles like height, number of wheels, body shape and color.

Traffic management and information systems rely on a suite of sensors for estimating traffic parameters. Currently, magnetic loop detectors are often used to count vehicles passing over them. Vision-based video monitoring systems offer a number of advantages. In addition to vehicle counts, a much larger set of traffic parameters such as vehicle classifications, lane changes, etc. can be measured. Besides, cameras are much less disruptive to install than loop detectors.

Vehicle classification is important in the computation of the percentages of vehicle classes that use state-aid streets and highways. The current situation is described by outdated data and often, human operators manually count vehicles at a specific street. The use of an automated system can lead to accurate design of pavements (e.g., the decision about thickness) with obvious results in cost and quality. Even in metro areas, there is a need for data about vehicle classes that use a particular street.

1.1 Video Analysis / Video Interpretation

Video analysis is the process in which a video is automatically analyzed to detect and determine the temporal and spatial events. In video analysis some algorithms are implemented as software on general machine or as hardware in video processing units. In video analysis the video motion detection, video tracking, background abstraction, behaviour analysis and situation awareness are the main factors.

Video interpretation is a video telecommunication service that uses devices such as video cameras or videophones to provide sign language or spoken language interpreting language. This is done through the remote or offsite interpreter, in order to communicate with whom there is a communication barrier. Video interpreter facilitates communication

between the participants who are located together at the other site. They are communicating by using headphones or microphone.

Object identification is a process where identity of an object under consideration in a scene is made. It may include the identification of an individual, animal, bird, vehicle, tree, river etc. Thegiven scene is converted into an image using image capturing device and some pre-processing techniques are applied on it to convert it to desire form. In case of content based interpretation smaller regions of interests (ROI) are extracted from images using simple and fast computing techniques. These ROI are further analysed by more computationally demanding techniques to produce a correct interpretation.

1.2 Vehicle identification and classification system (VICS)

The VICS system for identification and classification of moving vehicles on the road side from the videos is a great importance today. In India the traffic related information are gathered manually. One of the easy way to exchange information related to traffic between different computers by using network which is helpful for making many kind of decision related to traffic management. A VICS system can identify and classify vehicles on the basis of road side videos. AVICS system is helpful for traffic control and collecting statistics data related to vehicles which are helpful for taking many decision. A number of vehicle identification and classification systems have been developed by various prominent authors but 100% accuracy is not available. In VICS system, vehicle identification and classification can be done in two ways online and offline. In offline system, the identification and classification of vehicles is done from the videos related to the traffic whereas in online system, the images are captured by CCTV camera installed on the road side and the system identifies and classifies the vehicles directly from that video(s).

Vehicle identification and classification system (VICS) is an intelligent vehicle recognition system used to manage traffic on roads. There is dire need of monitoring and controlling traffic on road using efficient and effective cost effective method. In VICS the images of the vehicles are captured using a video camera installed on road side. In VICS the decision may be made on the basis of a single camera on one side or multiple cameras installed at different locations at particular angle depending upon the requirement and level of sophistication of the system. The video are converted into shots and frames, then features extraction and classification method are applied to identify and

classify the vehicles. General steps in video based vehicle identification and classification system, and associated applications are given in Figure.

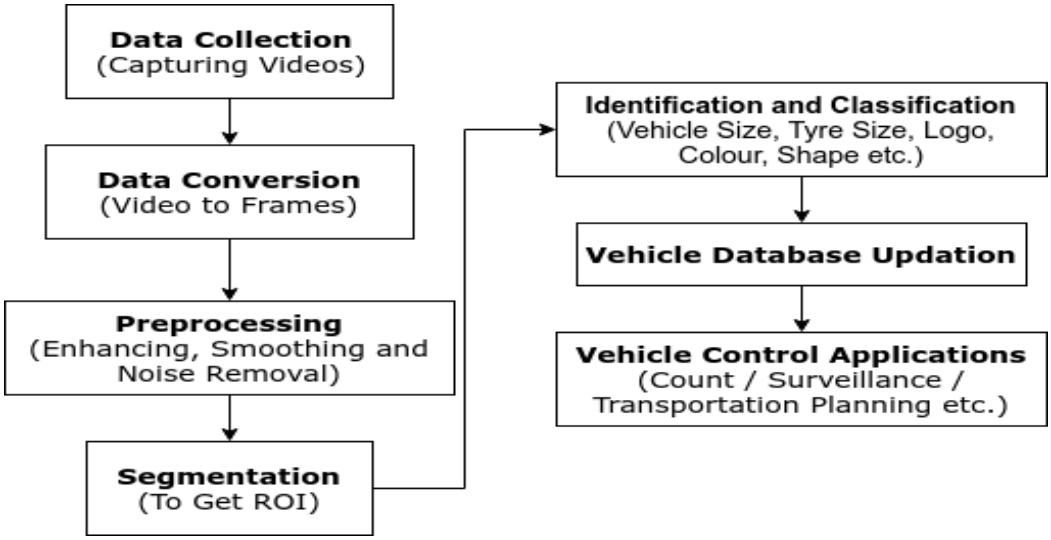

Figure 1: General steps in Video based identification and classification system (VICS).

The traffic on roads can be broadly classified into two categories i.e. homogenous and heterogeneous. Homogeneous traffic is a hypothetical synchronized flow of traffic of similar vehicles where all vehicle move with the same speed, irrespective of time and maintain the same space gap between them. The condition under which the traffic move is also called as homogenous traffic conditions. Heterogeneous traffic is unsynchronized and unregulated i.e. there is an irregular movement of all type of vehicles. In VICS, the traffic data extractor is used to detect and track each object moving through the scene to check its speed, path, class and counting. It analysis the video sequences in order to locate the object passed through a view of the cameras and to track them until they leave the obscured seen. The object parameter extractor analyse each object, as soon as it is available, in the object database list already stored in system.It identifies, classifies and tracks that object and all its parameters are saved the result list.

1.3 Applications of VICS

The various applications of image/video based image analysis, recognition and understanding are as

1.Video Surveillance.

2.Traffic Management System.

3.Vision based intelligent Transport System.

4.Intersection Control.

5.Incident Detection

6.Vehicle Classification.

7.Monitoring.

8.Revenue collection.

9.Historical Traffic Data.

10.Congestion Map and travel time estimate.

11.Public Transport information.

12.Individual Vehicle Management.

13.Accident Handling.

14.Conventional Driver Assistance system.

15.Traffic Surveillance.

1.4 Challenges

While designing system based on image/video based image analysis, recognition and side traffic management, a number of challenges are faced by the researcher and some of these are as:

• In western countries like USA, lane marking system is implemented. Vehicles are allowed to move in specific rows according to speed limit and vehicle type, etc. But in India in most of the cases traffic system is non lane based.

• Road conditions are more varied and traffic is unstructured, there is lack of discipline and overloaded vehicle movement is quite natural.

• In India, vehicles are parked frequently by the sides of the roads. There is no separate system for vehicle parking management.

• Roads are not only occupied with vehicles, so many obstacles on roads create disturbance to the traffic. Pedestrians do not have separate ways for their movement in most of the cases.

• Shapes of the vehicles have a key role in recognition; there is high intra-class variance among Indian vehicles. It creates the chances for miss recognition.

• Within same vehicle class there are large variety and models. These look different in size and appearance. It is generally observed in Indian vehicles like cars and Truck's.

• Vehicles detection using optical sensor is very challenging due to huge within class variability in vehicles appearance.

• Hundred of images and billion of classes.

2. Classification using Bag of Features

2.1 Description of Contents of the Dataset

In order to apply vehicle recognition in Indian scenario, we have introduced Indian vehicle database in this paper. It is build by capturing vehicle images on several Indian highways and traffic roads. Vehicle types considered are Auto, Car, Bus and Truck. Sample images from the entire database are shown in Figure-2. The complete description is presented in Table-1.

Figure 2: Sample Images from Indian Vehicle database includes four classes (Truck, Auto, Bus and Car respectively). Images were captured with variance in pose, view and lightning constraints.

Properties	Description
Name	Indian Vehicle database
Sources	Static vehicle pictures captured using camera on Indian roads, Pictures collected from Internet resources like Google etc.
Constraints	Pose, lightning and view
Number of classes	4
vehicle types	Truck, Auto, Bus and Car
Number of images per class	450
Total Images	1800

Table 1: Description of Indian Vehicle database

2.2 Literature Review

Vehicle identification and classification system, Lai et al [1] , used to calculate the speed of the vehicles where a number of loops are automatically assigned to each lane. To automatic calculate the speed of a vehicle the inductive loop detector method is applied. The merit of doing this is that a). It accommodates pan–tilt–zoom (PTZ) actions without the further requirement of human interaction. b). The size of the virtual loops is much smaller for estimation of accuracy. This enables the use of standard block-based motion estimation techniques that are well developed for video coding. c). The number of virtual loops per lane is large. The motion content of each block may be weighted and the collective result offers a more reliable and robust approach in motion estimation. There is no failure rate associated with the virtual loops or physical installation. As the loops are defined on the image sequence, changing the detection configuration or redeploying the loops to other locations on the same image sequence requires only a change of the assignment parameters. d). Virtual loops may be reallocated anywhere on the frame, giving flexibility in detecting different parameters.

Dhanya et al [2] developed a computer vision system for detecting and tracking the moving vehicle at day time and night time. First the videos are converted into frames and background and foreground of the image are detected. The headlight and the taillight of the vehicle is used for detecting and identifying the vehicle, after that image segmentation and pattern analysis techniques are applied. A fast bright object is identified and classified spatial clustering.

Mishra et al [3] develop an algorithm for detection and classification of vehicle in heterogeneous traffic. The entire process is divided into four steps i.e. camera calibration, vehicle detection, speed estimation, and classification. Vehicle detection is carried using background subtraction and blob tracking methods. Speed of the vehicle is estimated by using start and stop lane marker and calibration parameter. Classification of vehicles depends upon the various features of the detected vehicles. These features give the input to SVM for classification. A non-linear kernel is used as the classifier.

Chaoyang et al [4] recognizes logos in video stream in real-time. A new technique is developed that combines both coarse template matching approach and pair wise learning method together. The logo recognition becomes effective and efficient by eliminating the false alarms and further refines the recognition results. Image alignment

for template matching improves the stability of the coarse stage. Experimental results show that this approach outperform the DOT matching approach and traditional multiple classifiers combination.

Daigavane et al [5] developed an application based on neural network for vehicle detection and classification. This system identifies and classifies the vehicles with their success rate 90%. Vehicle are tracked by using blob tracking method and neural networks classify these vehicles on the basis of length and height There have been cases where the system is unable to do the classification correctly. When multiple vehicles move together, with approximately the same velocity, they tend to get grouped together as one vehicle. Also, the presence of shadows can cause the system to classify vehicles incorrectly.

Chen et al [6] investigate the effectiveness of state-of-the-art classification algorithms to categorise road vehicles for an urban traffic monitoring system using a multi-shape descriptor. The analysis is applied to monocular video acquired from a static pole-mounted road side CCTV camera on a busy street. These are used to classify the objects into four main vehicle categories i.e. car, van, bus and motorcycle. Image analysis for vehicle classification can be generally categorised into three principle approaches: model-based classification, Feature based classification and Measurement based classification. A number of experiments have been conducted to compare support vector machines (SVM) and random forests (RF) classifiers. 10-fold cross validation has been used to evaluate the performance of the classification methods. The results demonstrate that all methods achieve a recognition rate above 95% on the dataset, with SVM consistently outperforming RF. A combination of MBF and IPHOG features give the best performance of 99.78%.

Iwaski et al [7] studied road traffic flow surveillance under various environmental circumstances that cause poor visibility of vehicles on road. Authors used thermal images taken with infrared thermal cameras to detection vehicle. Two methods have been proposed. The first method uses pattern recognition for windshields and their surroundings to detect vehicles. The second method uses tires' thermal energy reflection areas on a road as the detection targets.

Messelodi et al [8] developed a system SCOCA, for counting and classifying vehicles automatically. The aim is to collect data for statistical purpose. The traffic data are

extractor by installing CCTV cameras on a pool. After detecting the scene, the second step is object parameter extractor. The methodology used for tracking an object is model based, region based, contour based and feature based. The object attributes determined are class, speed and path. The model based classification is used. The SCOCA system works in real time at 25 frames per second. A separate test has been conductor to measure the performance of second label cycle, motorcycle classifier based on SVM (Support Vector Machine) classifier techniques (189 vehicles, 45 bicycles, 144 motorcycles extracted from two video sequences. The classifier provides an average error rate 6.7%.

Deb [9] developed an automatic driver assistance system to alert a driver about driving environment. The most common approach used to vehicle detection is active sensor such as radar based system, laird ("Light Detection and Ranging") and acoustic based. Radar based system can see 150 meter ahead in fog or rain. To find the location of vehicles, the three approaches are used i.e. knowledge based, stereo based and motion based. The vehicle identification are done on the basis of symmetry, colour, shadow, corners, vertical and horizontal edges, texture, and vehicle light. In stereo vision system, the three methods used are disparity map, inverse perspective mapping and motion based and the location are find by using template based and appearance based.

Betke et al [10] described a real-time vision system that analyzes colour videos taken from a forward-looking video camera in a car driving on a highway. The system is a combination of colour, edge, and motion information to recognize and track the road boundaries, lane markings and other vehicles on the road. Cars are recognized by matching templates that are cropped from the input data online and by detecting highway scene features and evaluating the way they relate to each other. Cars are also detected by temporal differencing and by tracking motion parameters that are typical for cars. The system recognizes and tracks road boundaries and lane markings using a recursive least-squares filter. Experimental results demonstrate robust, real-time car detection and tracking over thousands of image frames. The data includes video taken under difficult visibility conditions.

Changlasetty et al [11] presented a system for identification and classification of vehicles where vehicles are tracked by using width, length and parameter area extracted using image techniques. Traffic data are recorded by stationary camera. Background of the image are extracted by using mixture model. For the classification LABVIEW and neural

networks is trained to classify vehicles using data monitoring WEKA tools It is used in intelligent transport system for Indian cites. A feed-forward neural network is trained to classify vehicles using data mining WEKA toolbox.

Chung et al [12] developed a real-time vision-based vehicle detection system that employs an online boosting algorithm. It is an online AdaBoost approach that cascades various strong classifiers instead of a single strong classifier. The cascade of strong classifiers for vehicle detection is online updated in response to traffic changing environment. The online algorithms efficiently catch the parameter based on the incoming image and their performance is up to date. This approach has been successfully validated in real traffic environments by performing experiments with an onboard charge-coupled-device camera in a roadway vehicle.

Shaoqing et al [13] real-time classification method is proposed for heavy traffic flow multi-lanes roads, which can classify vehicles into cars, trucks and buses. In this system three cameras are installed at 60 degree they focus on the lanes and vehicle features are extracted from it. The presence of car is determined by the colour of LPR and the absence of the car by segmented the image by combination of position mapping function. The feature extraction methods hybrid insensitive noise edge detection method based on Sobel operator and colors, the second is regions mergence according to colors and positions are applied to determine the car. Lastly noncars are classified into trucks and buses by a fuzzy rules classifier.

Suryatali et al [14] presents a system where a camera captures the images of the vehicles passing through a toll booth thus a vehicle is detected through camera. The classification of vehicles is done on the basis of area of vehicles. This information is further passed to the 'Raspberry Pi'which is having a web server set up on it. The Raspberry Pi is a credit card-sized single-board computer developed in the UK. The job of Raspberry pi in this system is processing large quantities of data and also it will keep detailed log of vehicles which are in the system.The Raspberry Pi is a good choice for a webserver that will not receive too much traffic and only uses around 5 Watts of power. This system can also counts moving vehicles from pre-recorded videos or stored videos by using the same algorithm. A new toll collection system is low cost alternative among all other system. This system is based on computer vision vehicle detection using open CV library embedded in Linux platform.

Xuehua et al [15] proposes an algorithm which can solve the problems effectively by the improved Gaussian mixture model and Support vector machine. First of all, the paper introduce an improved Gaussian mixture model which can effectively detect the moving objects and resolve the problems of Gaussian mixture model sensitive to light changes. Then the paper designs some classifiers to recognize the pedestrians and vehicles by the idea of the improved SVM. The experimental results show that the method has a high recognition rate and can also satisfy the real-time intelligent transportation surveillance.

Table 2: Summarization of studies on vehicle identification and classification vehicles.

Method	Identification / Classification	Foreign/Indian	Success Rate	References
SVM to identify PPlive, PPstream, UUsee, QQlive and Sopcast	Classification Online Video	China	93.5%	Liu et. al (2004)
Window based online learning algorithm	Identification Offline Videos	Canada	80%	V Nair et. al (2005)
Algorithm based on Width to Height Ratio (WHR) and Base to Abdomen Ratio (BAR)	Identification	Malaysia	93%	S Mohammed et. al(2005)
Bayesian Network	Classification and Identification Online	California, USA	88%	B. M et al (2006)
Online Boosting Algorithm	Classifier and Detection Offline	Foreign	97%	W Chung et. al(2010)
MBF and IPHOG	Classification Online Videos	London, UK	99.78%	Z Chen et. al(2010)
Virtual Detection Loops	Counting and Classification Online	USA	97.4%	S Li et. al(2013)
Thermal Energy in Traffic flow Surveillance	Thernal Vehicle Detection	Japan	91.2%	S Shantainya et. al(2014)
Fine – Grained Recognition Algorithms and	Classifier Offline Videos	China	97.38%	J Zhan et. al(2014)

SVM (Only for Cars)				
HMM	Identification and Classification	USA	86.6%	A. Jazayeri et. al(2015)
SVM	Classification Online	London	6.7%	Messelodi. et al
Algorithms k Gaussian distribution models	Both for Identification and Classification of Vehicle and Pedestrians	China	91.4% for Pedestration 93.75% for Vehicles	X Song et. al
Scale Invarient Feature Transform (SIFT)	Vehicle Recognition	India	89.5%	D Belongie, 2005
TRAZER	Classification Offline	Guwahati	88.25%	C. Mallikyarjuna et. al(2009)
Neural Network / Blob Tracking	Identification and Classification Online	Wardha, India	90%	P. M et al(2011)
Quantized Wavelet Features method SVM	Detection Online	New Delhi, India	93.94%	S Kumar et. al(2012)

2.3 Methodology

Vehicle recognition process in Indian scenario has several challenges that are discussed in Introduction. To address these challenges features of a individual vehicle from different directions are to be considered. For this reason, the present paper use Bag of features (BOF) and Support vector machines for vehicle recognition.

BOF works on the principle that every object can be represented by its parts. For example, a Truck contains parts like big-tyres, number plate, cabin etc. Also car contains wheels, number plate and windows, but the basic difference between an Truck and car is observed to be in size and tyres. So, in order to recognize an object it is necessary to first recognize the parts of it and based on the parts identify the object correctly.

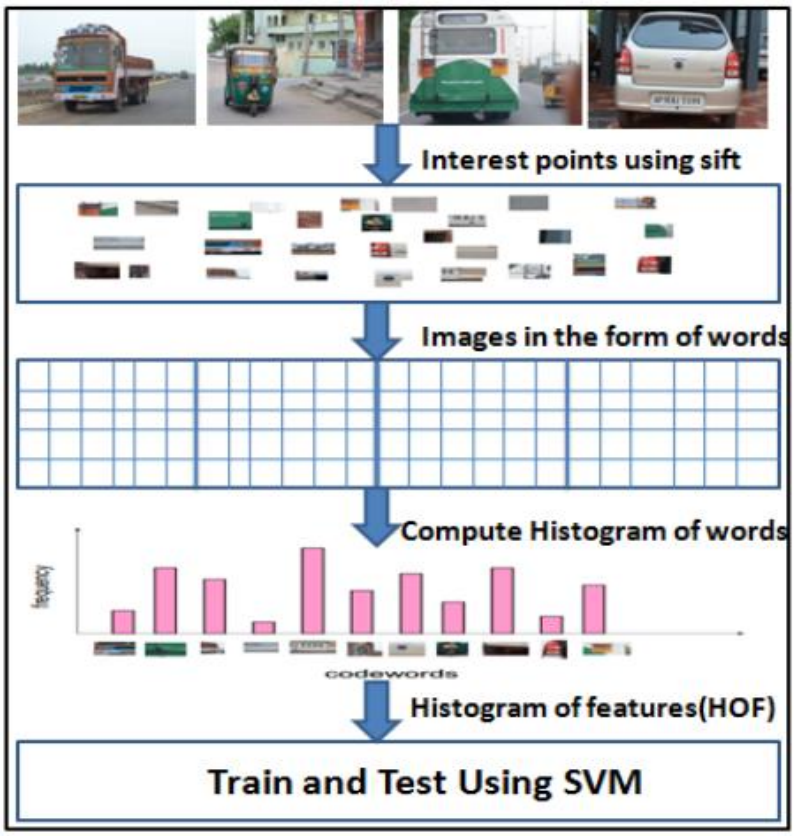

Figure 3: Vehicle recognition process in series of five stepsusing Histogram of features and Support vector machine

Basing on above principle vehicle recognition process in the Figure is initiated to recognize parts (step-1 and step-2) of the vehicle images by extracting image patches. This can be done by the combination of Harris corner and Sift points because these contain rich local information of the image.

For each patch, descriptors were calculated in the form of vectors. To address the cardinal and ordering problems, clusters are generated by apply K-means algorithmon these descriptors. Each cluster is refereed as word for an image. Thus image is represented as a bag of words or bag of visual words (step-3). These words are representedas a histogram (step-4) and referring as histogram of features. Features obtained in the above process are classified by train and test process using SVM (step-5).

The reason for the select of SVM as a classifier is that, it has some superiority over other approaches. The important points are global minimum solution, learning and generalization in huge dimensional input spaces, use of kernel function and classification is done by the separating hyperplane at a maximum distance to the closest points in the training set.

2.4 Experimental Results

Experimental results of vehicle recognition are presented in this section. Accuracy is evaluated for all the cases of recognition methods. Analysis is performed by varying the feature extraction and classification methods. Other procedures like SIFT descriptors and SVM are implemented using Vlfeat and Libsvm.

The two modules in recognition pipeline are feature extraction and classification. At first traditional methods like PCA, SIFT, LDA and LBP are applied as feature extraction and classified using KNN (K-nearest neighbor) and SRC (space representation classifier). Table 2 presents average performance of this base line result scheme.

The approximate accuracy is noticed between 50-74 percent only. But these methods have good accuracy in other object recognition methods like face etc. The main reason for less accuracy is due to use of local features. This is clearly addressed in the rest part of the experimental process and compensated by the virtue of visual features.

Experiments performed on 1800 vehicle image sequences. These image sequences are divided into 200 validation, 200 training and 50 testing samples randomly per each class.

Table 3: Accuracy (%) of vehicle recognition process using (PCA, SIFT, LDA and LBP over KNN and SRC) as a base line results

Feature Extraction	Classifier	Accuracy
PCA	SRC	50.00
SIFT	SRC	67.80
LDA	KNN	61.75
LBP	KNN	74.00

From validation samples shape parameters can be achieved through Harris corner and SIFT features.These are used for training the SVM. Testing is performed with total of 4*50=200 samples. Multi-class classification is done with a support vector machine (SVM) trained using the one-vs-rest rule.

Figure 4: Vehicle recognition result that shows the images which are correctly recognized

Figure 5: Vehicle recognition result that shows the images which are wrongly recognized

There is an increase in accuracy percent by the apply of BOF as feature extraction and SVM as a classifier. Accuracy percent for the present case is between 78-90. This has shown in Table 3. It also illustrates that the best results are yielded in the case of RBF kernel among the three types of kernels like linear, quadratic and RBF along with bag of features.

It is observed that the proposed method is performed correctly in recognition even for some complex conditions like overloaded vehicles and shape variations. The method also recognized different vehicle poses and views properly.

Table 4: The proposed method for vehicle recognition process deals with three types of kernels like linear, quadratic and RBF in SVM. Accuracy (%) along with Bag of features are tabulated here

Classifier	Kernel Type	Accuracy
SVM	Linear	78.54
SVM	Quadratic	81.00
SVM	RBF	90.00

The result of vehicle recognition is further analyzed using confusion matrix. It is a two dimensional array shows relationships between true and predicted classes. It is shown in table 4. Here we can observe a clear diagonal correlation between four types of vehicles.

Since Truck has some similar structure with other vehicles like bus, it is getting confused and viceversa. Generally car has similar features like shape and size with auto and vice versa. The 9% of confusion is noticed here. But these vehicles are not confused much in present case. This improvement is only due to implementation features from interest points. Confusion% in this case is 3.

We can assume two sets of vehicles from entire database. Heavy: bus & Truck, light: car & auto. These two sets of vehicles have dissimilar structure and features. Due to this reason a negligible confusion of 0.25% is noticed in this case.

Table 5: Performance evaluation of vehicle recognition using confusion matrix. Here 50 testing images from each of four classes are used. Images are considered from Indian Vehicle Database.

	Truck	Auto	Bus	Car
Truck	46	0	4	0
Auto	1	48	0	1
Bus	3	0	47	0
Car	0	2	0	48

3. Classification using Deep Learning

3.1 Description of Contents of Data Set

The classification task is performed on the MIO-TCD-Classification Dataset, which contains 648,959 images, with each image containing an object belonging exclusively to one of the 11 categories found in Table-6.

Category Name	Number of Images
Articulated truck	12,933
Bicycle	2,855
Bus	12,895
Car	325,649
Motorcycle	2,477
Non-motorized vehicle	2,189
Pedestrian	7,827
Pickup truck	63,633
Single unit truck	6,400
Work van	12,101
Background	200,000
Total	648,959

Table 6: MIO-TCD Classification challenge dataset category breakdown

In Figure-6, we show a sample of the training data. In terms of size, the imagesare all of different dimensions and were directly cropped out from the MIO-TCD-Localization Dataset. The largest single dimension for an image in the dataset isfound to be 720x720.

Figure 6: Sample training image - articulated truck

The model was trained with only 25% of the total classification data set which is equal to 129,787 images. All the images were preprocess in the same matter. They were first resized to a maximum shape of 96 pixels on both axis while making sure to keep the dimensions of the images. Then a black padding was applied to obtain a square image of 96 pixels squared.

These 129,787 images were divided into two sets a training set and a validation set. Considering the generous size, 90% of the original images were passed as training data and 10% to the validation data. This equals to 129,787 and 12,979 respectively in both sets. We then generated more training images by applying rotations and shifting the images width and height by 20%. We also applied horizontal flip to the images to generalize the predictions even more. We finally generated more data by zooming in and out by a factor of 0.1. The model was fed 256 images at a time.

3.2 Code and Description of Environment

The code was executed in a Google collaborator notebook server. This server had a total of 13GB of RAM and 2vCPU at 2.2GHz each. In addition, the server provide a 33GB of disk space. The python environment required the following python packages: os, cv2, matplotlib, pyplot, numpy, math, itertools, keras and random.

3.3 Literature Review

3.3.1 Convolutional Neural Networks (CNNs / ConvNets)

Convolutional Neural Networks are very similar to ordinary Neural Networks: they are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. The whole network still expresses a single differentiable score function: from the raw image pixels on one end to class scores at the other. And they still have a loss function (e.g. SVM/Softmax) on the last (fully-connected) layer.

ConvNet architectures make the explicit assumption that the inputs are images, which allows us to encode certain properties into the architecture. These then make the forward function more efficient to implement and vastly reduce the amount of parameters in the network.

3.3.2 Architecture Overview

Neural Networks receive an input (a single vector), and transform it through a series of hidden layers. Each hidden layer is made up of a set of neurons, where each neuron is fully connected to all neurons in the previous layer, and where neurons in a single layer function completely independently and do not share any connections. The last fully-connected layer is called the "output layer" and in classification settings it represents the class scores.

Convolutional Neural Networks take advantage of the fact that the input consists of images and they constrain the architecture in a more sensible way. In particular, unlike a regular Neural Network, the layers of a ConvNet have neurons arranged in 3 dimensions: width, height, depth. (Note that the word depth here refers to the third dimension of an activation volume, not to the depth of a full Neural Network, which can refer to the total number of layers in a network.) For example, the input images in CIFAR-10 are an input volume of activations, and the volume has dimensions 32x32x3 (width, height, depth respectively).

A ConvNet is made up of Layers. Every Layer has a simple API: It transforms an input 3D volume to an output 3D volume with some differentiable function that may or may not have parameters.

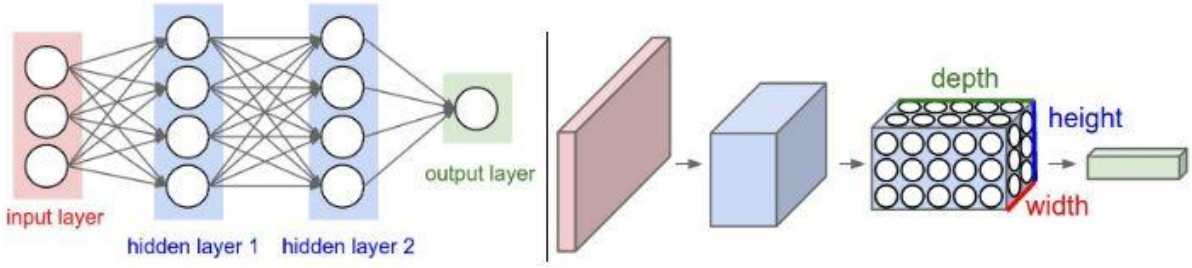

Figure 7: **Left:** A regular 3-layer Neural Network. **Right:** A ConvNet arranges its neurons in three dimensions

3.3.3 Layers used to build ConvNets

As we described above, a simple ConvNet is a sequence of layers, and every layer of a ConvNet transforms one volume of activations to another through a differentiable function. We use three main types of layers to build ConvNet architectures: Convolutional Layer, Pooling Layer, and Fully-Connected Layer (exactly as in regular Neural Networks). We will stack these layers to form a full ConvNet architecture.

A simple ConvNet for classification could have the architecture [INPUT - CONV - RELU - POOL - FC]. In more detail -

- INPUT [32x32x3] will hold the raw pixel values of the image, in this case an image of width 32, height 32, and with three color channels R,G,B.
- CONV layer will compute the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and a small region they are connected to in the input volume. This may result in volume such as [32x32x12] if we decided to use 12 filters.
- RELU layer will apply an elementwise activation function, such as the max(0,x) thresholding at zero. This leaves the size of the volume unchanged ([32x32x12]).
- POOL layer will perform a downsampling operation along the spatial dimensions (width, height), resulting in volume such as [16x16x12].
- FC (i.e. fully-connected) layer will compute the class scores, resulting in volume of size [1x1x10], where each of the 10 numbers correspond to a class score, such as among the 10

categories of CIFAR-10. As with ordinary Neural Networks and as the name implies, each neuron in this layer will be connected to all the numbers in the previous volume.

In this way, ConvNets transform the original image layer by layer from the original pixel values to the final class scores. Note that some layers contain parameters and other don't. In particular, the CONV/FC layers perform transformations that are a function of not only the activations in the input volume, but also of the parameters (the weights and biases of the neurons). On the other hand, the RELU/POOL layers will implement a fixed function. The parameters in the CONV/FC layers will be trained with gradient descent so that the class scores that the ConvNet computes are consistent with the labels in the training set for each image.

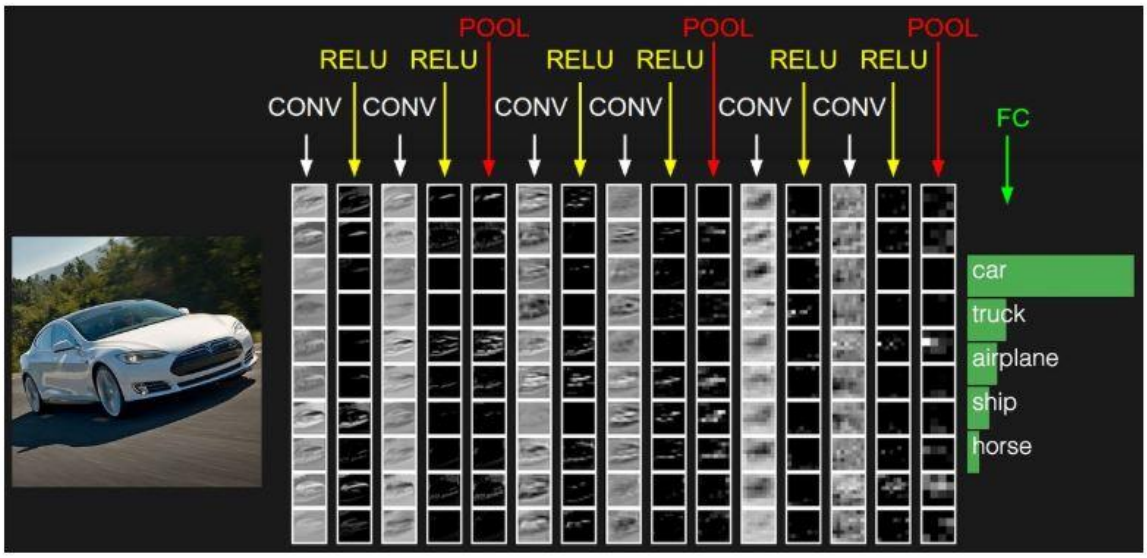

Figure 8: The activations of an example ConvNet architecture.

The initial volume stores the raw image pixels (left) and the last volume stores the class scores (right). Each volume of activations along the processing path is shown as a column. Since it's difficult to visualize 3D volumes, we lay out each volume's slices in rows. The last layer volume holds the scores for each class, but here we only visualize the sorted top 5 scores, and print the labels of each one.

3.3.4 Convolutional Layer

The Conv layer is the core building block of a Convolutional Network that does most of the computational heavy lifting.

Overview and intuition without brain stuff. Let's first discuss what the CONV layer computes without brain/neuron analogies. The CONV layer's parameters consist of a set of learnable filters. Every filter is small spatially (along width and height), but extends through the full depth of the input volume. For example, a typical filter on a first layer of a ConvNet might have size 5x5x3 (i.e. 5 pixels width and height, and 3 because images have depth 3, the color channels). During the forward pass, we slide (more precisely, convolve) each filter across the width and height of the input volume and compute dot products between the entries of the filter and the input at any position. As we slide the filter over the width and height of the input volume we will produce a 2-dimensional activation map that gives the responses of that filter at every spatial position. Intuitively, the network will learn filters that activate when they see some type of visual feature such as an edge of some orientation or a blotch of some color on the first layer, or eventually entire honeycomb or wheel-like patterns on higher layers of the network. Now, we will have an entire set of filters in each CONV layer (e.g. 12 filters), and each of them will produce a separate 2-dimensional activation map. We will stack these activation maps along the depth dimension and produce the output volume.

The brain view. If you're a fan of the brain/neuron analogies, every entry in the 3D output volume can also be interpreted as an output of a neuron that looks at only a small region in the input and shares parameters with all neurons to the left and right spatially (since these numbers all result from applying the same filter). We now discuss the details of the neuron connectivities, their arrangement in space, and their parameter sharing scheme.

Local Connectivity. When dealing with high-dimensional inputs such as images, as we saw above it is impractical to connect neurons to all neurons in the previous volume. Instead, we will connect each neuron to only a local region of the input volume. The spatial extent of this connectivity is a hyperparameter called the receptive field of the neuron (equivalently this is the filter size). The extent of the connectivity along the depth axis is always equal to the depth of the input volume. It is important to emphasize again this asymmetry in how we treat the spatial dimensions (width and height) and the depth dimension: The connections are local in space (along width and height), but always full along the entire depth of the input volume.

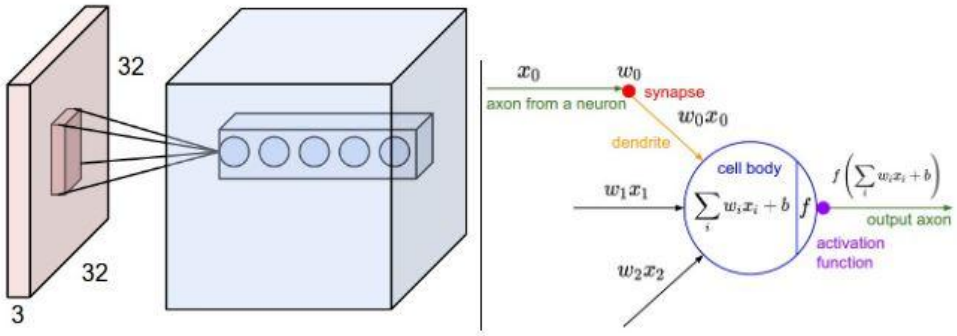

Figure 9: Example of Convolution Layer.

Left: An example input volume in red (e.g. a 32x32x3 CIFAR-10 image), and an example volume of neurons in the first Convolutional layer. Each neuron in the convolutional layer is connected only to a local region in the input volume spatially, but to the full depth (i.e. all color channels). Note, there are multiple neurons (5 in this example) along the depth, all looking at the same region in the input.

Right: The neurons from the Neural Network chapter remain unchanged: They still compute a dot product of their weights with the input followed by a non-linearity, but their connectivity is now restricted to be local spatially.

Spatial arrangement. We have explained the connectivity of each neuron in the Conv Layer to the input volume, but we haven't yet discussed how many neurons there are in the output volume or how they are arranged. Three hyperparameters control the size of the output volume: the **depth, stride** and **zero-padding**. We discuss these next:

1. First, the **depth** of the output volume is a hyperparameter: it corresponds to the number of filters we would like to use, each learning to look for something different in the input. For example, if the first Convolutional Layer takes as input the raw image, then different neurons along the depth dimension may activate in presence of various oriented edges, or blobs of color. We will refer to a set of neurons that are all looking at the same region of the input as a **depth column** (some people also prefer the term fibre).

2. Second, we must specify the **stride** with which we slide the filter. When the stride is 1 then we move the filters one pixel at a time. When the stride is 2 (or uncommonly 3 or

more, though this is rare in practice) then the filters jump 2 pixels at a time as we slide them around. This will produce smaller output volumes spatially.

3. As we will soon see, sometimes it will be convenient to pad the input volume with zeros around the border. The size of this **zero-padding** is a hyperparameter. The nice feature of zero padding is that it will allow us to control the spatial size of the output volumes (most commonly as we'll see soon we will use it to exactly preserve the spatial size of the input volume so the input and output width and height are the same).

3.3.5 Pooling Layer

It is common to periodically insert a Pooling layer in-between successive Conv layers in a ConvNet architecture. Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control overfitting. The Pooling Layer operates independently on every depth slice of the input and resizes it spatially, using the MAX operation. The most common form is a pooling layer with filters of size 2x2 applied with a stride of 2 downsamples every depth slice in the input by 2 along both width and height, discarding 75% of the activations. Every MAX operation would in this case be taking a max over 4 numbers (little 2x2 region in some depth slice). The depth dimension remains unchanged.

General pooling. In addition to max pooling, the pooling units can also perform other functions, such as average pooling or even L2-norm pooling. Average pooling was often used historically but has recently fallen out of favor compared to the max pooling operation, which has been shown to work better in practice.

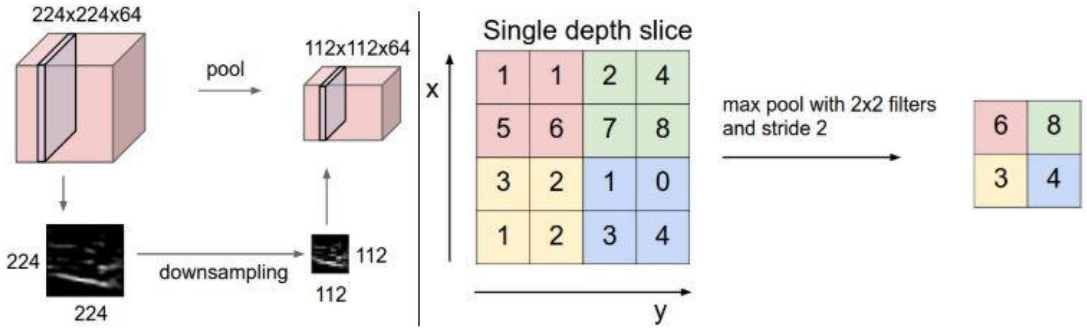

Figure 10: Pooling layer downsamples the volume spatially, independently in each depth slice of the input volume.

3.4 Methodology

3.4.1 Architecture

The architecture of the model is conventional. It is inspired from the MobileNet neural network which had impressive results on the ImageNet data set. This model is composed of a sequence of three convolutions blocks. One convolution block has two layers of 2 dimensional convolution with identical kernel size and number of filters. The activation layer on each convolution layer is relu for its good performances and faster computations than other activation layers such as tanh. All convolution layers are followed by a batch normalization. This technique improve the general performance of the model by denoising the image and also speed up the process. We add at the end of each convolutions block a max pooling with a squared pool size of two. To generalize the performance of the model and prevent over fitting.

The convolution blocks are implemented in keras. They have the following number of filters: 32, 64 and 128 pixels. The kernels all have the same squared dimension of 3. The output of this sequence is then flattened and fed into a densely-fully-connected layer with 512 perceptions.Dropout is applied to finally make a prediction over the eleven classes.

Nesterov Adam optimizer was chosen for this neural network for its fast convergence properties when combined with dropout. This learning rate was originally set to 0.001 but we used a learning rate decay when a plateau was reached on the validation categorical accuracy to speed upthe process and boost the performance.

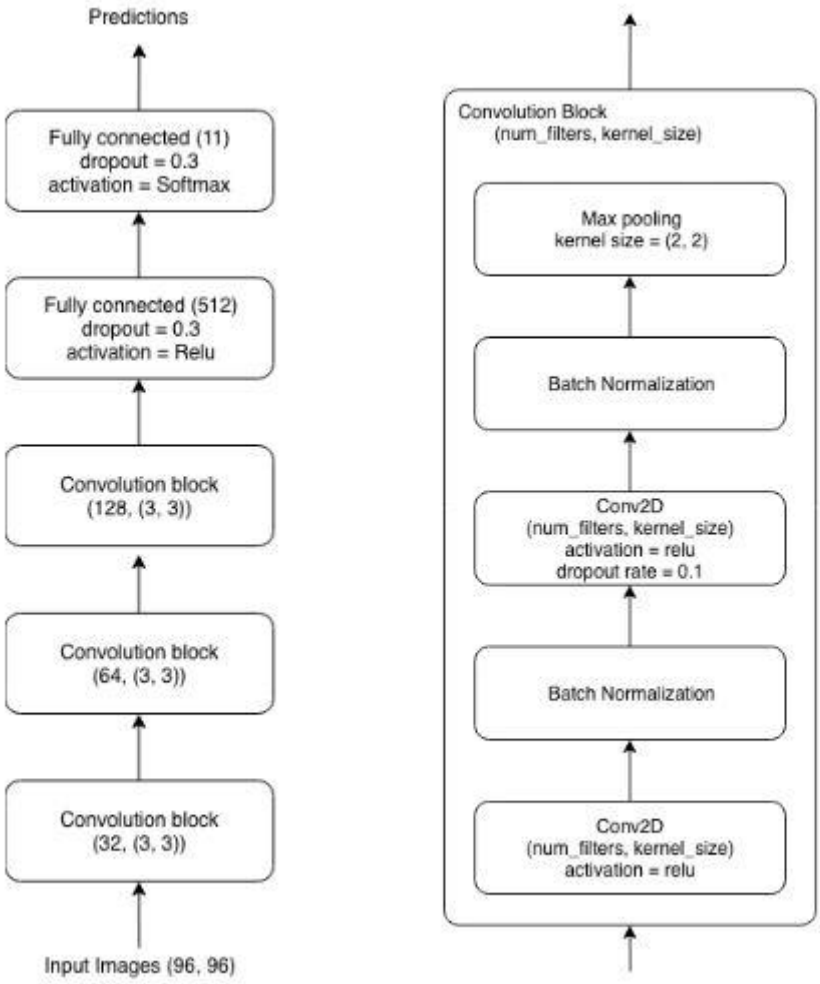

Figure 11: **Left:** Architecture of the Deep-Learning Model

Right: Modular Architecture of a Convolution Block

3.4.2 Training

The model was trained with only 25% of the total classification data set which is equal to 129,787 images. All the images were pre-process in the same matter. They were first re-sized to a maximum shape of 96 pixels on both axis while making sure to keep the dimensions of the images. Then a black padding was applied to obtain a square image of 96 pixels squared. These 129,787 images were divided into two sets a training set and a validation set. Considering the generous size, 90% of the original images were passed as training data and 10% to the validation data. This equals to 129,787 and 12,979 respectively in both sets. We then generated more training images by applying rotations and shifting the images width and height by 20%. We also applied horizontal flip to the images to generalize

the predictions even more. We finally generated more data by zooming in and out by a factor of 0.1. The model was fed 256 images at a time.

3.5 Experiemental Results

3.5.1 Evaluation of Performance

As the official website of the MIO-TCD data set does, we prioritized overall accuracy. When training the categorical accuracy was monitored to prevent under and over fitting. Other performance metrics will be used to evaluate the model on the outputs. For instance, the accuracy of the modelon each classes will be observed.

3.5.2 Validation

The model was trained over 40 epoch and was able to reach an overall accuracy of 93.8516%. This accuracy ranks the classifier in place 11 on the official website. It performs better than the AlexNet which is well known to achieve impressive results on image classification tasks. These results were made on the original unbalanced dataset and can be visualized here.

Figure 12: Confusion matrix with the Deep Learning Model

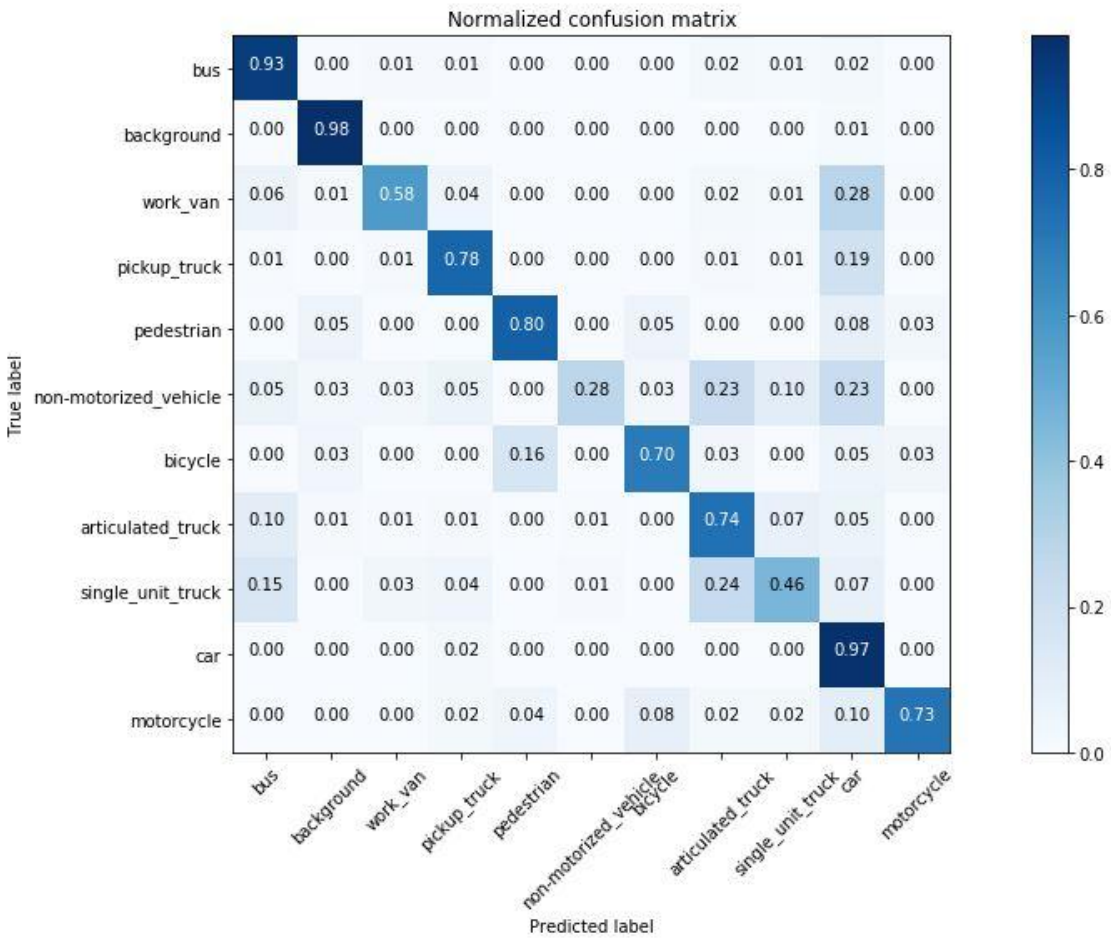

Figure 13: Normalized Confusion matrix with the Deep Learning Model

4. Discussion and Conclusion

4.1 Bag of Features Classifier

Figure-4 represents vehicle recognition results for the proposed method by correctly representing in to labels. As discussed in introduction, there are some challenges in vehicle recognition, due to this some failure cases are noticed. These are given in figure-5. Here Truck is recognized as bus, auto recognized as car, bus recognized as Truck and car recognized as auto.

The following four main reasons illustrates why failures are arises in our method. Figure-14 represents spectrum of images which are failed in recognition due to specified reasons.

- Objects in small size - If the objects of interest is very small in region it will be difficult to extract meaning full information and recognition accuracy decreases. Related example images as shown in Figure 14(a). This problem can be overcome by performing localization as a preprocessing step for recognition.
- Novel model - The database consists of images of some particular known number of classes in large variety's and shapes. It will be very difficult to build a model which captures all the intra-class variances. As a result some test images which has model variance from training images were recognized wrongly. This type of variances among car, Truck and auto are shown in figure 14(b) respectively.
- Clutter - As shown in Figure 14(c), some times unwanted information like hoardings and other obstacles on roads were also captured along with vehicle images. This introduces noise in features and makes confusion in recognition. Due to this reason testing becomes more complex.
- Poor quality images - Quality of the image has a key role in recognition. Important reason is it has a direct influence on performance of edge based features like sift which is implemented in this method. Figure 14(d) shows some images with poor quality. Recognition percent is observed to be very low in this case.

This report describes a robust method for recognizing the vehicles. A prototype system for vehicle recognition has been presented. This proposed method can be useful in several applications specifically suitable for Indian conditions. Objects of interest are recognized from images by the sift and Harris corner features, these are converted further in the form of

Histogram of words so called as bag of features. Classification is performed by using Support vector machine.

Figure 14. In the process of recognizing Indian vehicles the system has attained some failures.The important reasons noticed are due to (a) very small objects, (b) novel models (in car, truck and auto as an example), (c) clutter and (d) images with poor quality

With respect to the Indian conditions the recognition rates so far promised its usability within the context of a state-of-the-art. Experimental results show that, proposed vehicle recognition method is accurate up to moderate level as compared to base line result. Analysis for the failure cases in the system and their solutions also presented. Future work will concentrate on the optimization of the run time of the system and improvement of recognition robustness for the case of intra class variation.This work can also be extended by improving database size and number of vehicle classes.

4.2 Deep Learning Classifier

This classifier performed much better than the SVM classifier. That being said the SVM classifier did not use the same unbalanced class the original data set had, which help its results. Therefore, it is same to assume that the deep learning solution performs better than any other method.

5. References

[1] H. S. Lai and H. C. Yung, "Vehicle-Type Identification Through Automated Virtual Loop Assignment and Block-Based Direction-Biased Motion Estimation" ,IEEE Transactions on Intelligent Transportation System, Vol. 1, No. 2, June 2000.

[2] K. Dhanya, M. Manimekalai, B.Asmin and G. Vani, "Tracking and Identification of Multiple Vehicles ".

[3] P. Mishra, M. Athiq, A. Nandoriya and S. Chaudhuri,"Video based Vehicle Detection and Classification in Heterogeneous Traffic Conditions using a Novel Kernel Classifier" IETE journal of research vol 2013.

[4] C. Zhao, J. Wang, C. Xie and H. Lu," A COARSE-TO-FINE LOGO RECOGNITION METHOD IN VIDEO STREAMS" National Laboratory of Pattern Recognition, CASIA, Beijing China

[5] P.M. Daigavane and M.B. Daigavane," Vehicle Detection and Neural Network Application for Vehicle Classification" International Conference on Computational Intelligence and Communication Systems 2011.

[6] Z. Chen and T. Ellis," Multi-shape Descriptor Vehicle Classification for Urban Traffic" International Conference on Digital Image Computing: Techniques and Applications 2011.

[7] Y. Iwasaki, M. Misumi, and T Nakamiya, "Robust Vehicle Detection under Various Environments to Realize Road Traffic Flow Surveillance Using An Infrared Thermal Camera" The Scientific World Journal, Volume 2015, Article ID 947272, Hindawi Publishing Corporation.

[8] S. Messelodi, M. Modena, and M. Zanin ,"A computer vision system for the detection and classification of vehicles at urban road intersections" Springer-Verlag 2005 London Limited.

[9] K. Deb and K. Nathr "Vehicle Detection Based on Video for Traffic Surveillance on road". Int. J Comp Sci. Emerging Tech Vol-3 No 4 August, 2012.

[10] M. Betke, E. Haritaoglu and S. Davis, "Real-time multiple vehicle detection and tracking from a moving vehicle" Machine Vision and Applications pp 69–83Springer-Verlag 2000.

[11] B. Changalasetty , A. Badawy, W. Ghribi, H. Ashwi , A. Mohammed , A.Shehri , S. Thota and R. Medisetty, "Identification and Classification of Moving Vehicles on Road" Computer Engineering and Intelligent Systems pp 2222-2863 Vol.4, No.8, 2013 .

[12] W.Chang and W.Cho , "Online Boosting for Vehicle Detection" IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS VOL. 40, NO. 3, JUNE2010

[13] M.Shaoqing and L.Zhengguang, "Real-time Vehicle Classification Method for Multilane Roads "ICIEA 2009.

[14] A. Suryatali and V.B. Dharmadhikari, "Computer Vision Based Vehicle Detection for Toll CollectionSystem Using Embedded Linux". International Conference on Circuit, Power and Computing Technologies2015

[15] X. Song, L. Wang, H. Wang and Y. Zhang, "Detection and identification in the Intelligent Traffic Video Monitoring System for pedestrians and vehicles".

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and douard Duchesnay, "scikit-learn: Machine learning in python," 2011.

[17] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications", CoRR, vol. abs/1704.04861, 2017.

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "Imagenet large scale visual recognition challenge", CoRR, vol. abs/1409.0575, 2014.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems 25 (F. Pereira,C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates,Inc., 2012.

[20] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising", CoRR, vol. abs/1608.03981, 2016.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting”, J. Mach. Learn. Res., vol. 15,pp. 1929–1958, Jan. 2014.

[22] F. Cholletet al., “Keras” https://keras.io, 2015.

[23] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization”, CoRR,vol. abs/1412.6980, 2014.

[24] R. A. Jacobs, “Increased rates of convergence through learning rate adaptation”, tech. rep., Amherst, MA, USA, 1987.

[25] Z. Luo, F.B.Charron, C.Lemaire, J.Konrad, S.Li, A.Mishra, A. Achkar, J.Eichel, P-M Jodoin ”MIO-TCD: A new benchmark dataset for vehicle clas-sification and localization” in press atIEEE Transactions on Image Processing,2018

[26] R.S Vaddi, L.N.P Boggavarapu, K.R Anne, "Computer Vision based Vehicle Recognition on Indian Roads", International Journal Of Computer Vision And Signal Processing, 5(1), 8-13(2015)

[27] Baljit Singh Mokha and Satish Kumar, A Review Of Computer Vision System For The Vehicle Identification And Classification From Online And Offline Videos.