JUIVCDv1: Development of a still-image based Dataset for Indian Vehicle Classification

Thesis submitted in partial fulfillment of requirements
For the degree of
Master of Technology in Computer Technology
of
Computer Science and Engineering Department
of
Jadavpur University

by

Debam Saha
Registration No. - 154191 of 2020 - 2021
Examination Roll No. - M6TCT23028

under the supervision of

Prof. (Dr.) Ram Sarkar

Department of Computer Science and Engineering
Jadavpur University
Kolkata, West Bengal, India

2023

Certificate from the Supervisor

This is to certify that the work embodied in this thesis entitled **"JUIVCDv1: Development of a still-image based Dataset for Indian Vehicle Classification"** has been satisfactorily completed by **Debam Saha** (Registration Number 154191 of 2020 – 2021; Class Roll No. 002010504026; Examination Roll No. M6TCT23028. It is a bona-fide piece of work carried out under my supervision and guidance at Jadavpur University, Kolkata for partial fulfilment of the requirements for the awarding of the **Master of Technology in Computer Technology** degree of the Department of Computer Science and Engineering, Faculty of Engineering and Technology, Jadavpur University, during the academic year 2022 – 23.

Prof. (Dr.) Ram Sarkar,
Professor,
Department of Computer Science and Engineering,
Jadavpur University.
(Supervisor)

Forwarded By:

Prof. (Dr.) Nandini Mukherjee,
Head,
Department of Computer Science and Engineering,
Jadavpur University.

Prof. (Dr.) Ardhendu Ghoshal,
DEAN,
Faculty of Engineering & Technology,
Jadavpur University.

Department of Computer Science and Engineering
Faculty of Engineering And Technology
Jadavpur University, Kolkata - 700 032

Certificate of Approval

This is to certify that the thesis entitled **"JUIVCDv1: Development of a still-image based Dataset for Indian Vehicle Classification"** is a bona-fide record of work carried out by **Debam Saha** (Registration Number 154191 of 2020 – 2021; Class Roll No. 002010504026; Examination Roll No. M6TCT23028) in partial fulfilment of the requirements for the award of the degree of **Master of Technology in Computer Technology** in the **Department of Computer Science and Engineering, Jadavpur University**, during the period of September 2022 to June 2023. It is understood that by this approval, the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose of which it has been submitted.

Examiners:

(Signature of The Examiner)

(Signature of The Supervisor)

Department of Computer Science and Engineering
Faculty of Engineering And Technology
Jadavpur University, Kolkata - 700 032

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that the thesis entitled **"JUIVCDv1: Development of a still-image based Dataset for Indian Vehicle Classification"** contains literature survey and original research work by the undersigned candidate, as a part of his degree of **Master of Technology in Computer Technology** in the **Department of Computer Science and Engineering, Jadavpur University**. All information have been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Debam Saha

Examination Roll No.: M6TCT23028

Registration No.: 154191 of 2020 – 2021

Thesis Title: JUIVCDv1: Development of a still-image based Dataset for Indian Vehicle Classification

---

Signature of the Candidate

ACKNOWLEDGEMENT

I am pleased to express my gratitude and regards towards my Project Guide **Dr. Ram Sarkar**, Professor, Department of Computer Science and Engineering, Jadavpur University, without whose valuable guidance, inspiration and attention towards me, pursuing my project would have been impossible.

I am grateful towards all the members of **Center for Microprocessor Applications for Training Education and Research (*CMATER*) Lab, Department of Computer Science and Engineering, Jadavpur University,** for cooperating with me in all possible ways and providing me with a good working environment throughout the duration of the work.

I express my regards towards **parents** for bearing with me and for being a source of constant motivation during the entire term of the work.

I would also like to thank my friends Sourajit Maity, Satyajit Mahato, Subhomoy Saha, Laxman Das, Ankita Chowdhury, Amartya Roy, Mridul Biswas, Arup Sau, Sujan Sarkar, Satarupa Mal, Saba Nadim, Shahid Ahmed Siddique for constantly supporting and guiding me through the research work.

Debam Saha
MTCT Final Year
Exam Roll No. - M6TCT23028
Regn. No. - 154191 of 2020 – 2021
Department of Computer Science and Engineering,
Jadavpur University.

Abstract

Designing an automatic vehicle classification system from still images or videos would be highly beneficial for developing a traffic control system. On automatic vehicle classification, numerous articles have been published in the literature. Over the years, researchers in this subject have created and used a variety of databases, but most often, these databases are not found to be appropriate in Indian scenarios due to the specific peculiarities of the road conditions, nature of congestion, and vehicle types usually seen in India. This thesis primary goal is to create a new still image database called the JUIVCDv1 that contains 12 different vehicle classes that were gathered utilising mobile phone cameras in a variety of ways for developing an automated traffic management system. We have also mentioned the characteristics of the current databases and the various factors we took into account when creating the database for the Indian scenario. Apart from this, we have benchmarked the results on this database using a five-base model architecture. Five base models are used: EfficientNet, InceptionV3, DenseNet121, MobileNetV2, and VGG19. Among these five-base models, EfficientNet achieved the best accuracy, i.e., 93.82%.

Keywords: Automatic Vehicle classification, JUIVCDv1, Still image database, Deep Learning.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In recent times, the rising number of vehicles on the road has increased the demand for more efficient solutions to the traffic congestion problem. Automatic vehicle classification (AVC) systems and traffic information systems are needed for real-time traffic monitoring and management to deal with the ever-increasing volume of traffic. The rising number of vehicles on the road has been a concern for researchers, not only for the betterment of the road traffic scenario. Many studies on the traffic management system have been published in areas such as vehicle categorization, detection, make and model recognition, segmentation, lane detection, pedestrian detection, etc. Autonomous driving will play a huge role in vehicle-related research in the following years. For this reason, research on AVC systems is much needed in today's scenario. Such systems can also be used to collect information about vehicle makers and model numbers that are required for security purposes. Working on such problems using real-world traffic scenarios is difficult in terms of training, testing, and model validation. Huge amounts of realistic data are required for this purpose. During the study of this domain, it has been found that there are not many datasets available

for research, and many of them are based on speculative situations. Moreover, well-known datasets are mostly paid for, and the datasets with a decent number of images lack appropriate annotations, making it challenging to use them for research. On the other hand, sufficient samples are needed to create an effective supervised learning-based model that is accurate and capable of functioning in real-life scenarios. For vehicle localization, there are a few datasets available, but the number of datasets available for vehicle classification is limited. Also, all datasets do not capture real-life scenarios properly. For example, images taken in the Indian subcontinent frequently show multiple vehicles overlapping in a single frame due to traffic congestion. Moreover, this issue makes the classification, localization, detection, and segmentation processes extremely challenging. These challenges are relevant not only in India but also in Bangladesh, Pakistan, and many other South Asian nations. The available models, which were trained on well-managed traffic scenarios, might not be applicable to the datasets collected from these nations. Recently, Bhattacharya et al.[1] developed a dataset, called JUVDsi, for vehicle detection in Indian road scenarios.

1.1 Research Motivation

1. For vehicle localization, a large number of datasets are accessible; however, only a small number of datasets are valid for classification.

2. Not all datasets accurately reflect real-world situations. Similar to the Indian subcontinent, pictures taken in crowded traffic conditions sometimes show many automobiles overlapped in one image. This problem makes the classification, localization, detection, and segmentation processes exceptionally challenging.

3. These challenges are crucial not only in India but also in many other South Asian nations such as Bangladesh, Pakistan, and Sri Lanka.

4. There are no multi-view or multi-modal datasets available for classifying vehicles. Developing a viable solution for AVC requires a lot of research and information.

5. There is no more room for research on some datasets because researchers have already achieved 100% accuracy on those datasets.

1.2 Scope of the Thesis

Bearing the aforementioned information in mind, this study introduces JUIVCDv1, a new still image database with suitable annotation made up of frequently seen vehicles on Indian roads. We have collected the data from Kolkata, one of the largest metropolitan cities in Eastern India, which endures high traffic congestion because of the rising population and is the capital city of West Bengal. Also, the wide variety of vehicles seen in Kolkata portrays the traffic scenario of the Indian sub-continent as close to reality as possible. The main contributions to this work are as follows:

- The image database is appropriately labeled to enable testing of any algorithm created for the automatic classification of vehicles in an unrestricted environment.

- To make the visuals robust and realistic, we have added several complexities. In addition to being small, the cars in the database display required variations for ensuring the robustness of a developed algorithm, including several orientations, a single vehicle in one frame, multiple vehicles in one frame, changes in lighting conditions, specularities, or occlusions.

- In this database, we have considered 12 different types of vehicles.
- Each image has been offered in a variety of spectral bands and resolutions. Also, a precise experimental protocol has been provided, enabling accurate replication and comparison of the experimental data acquired by various algorithms.
- We have further benchmarked the results on this dataset using some state-of-the-art deep learning models. In doing so, we have considered five pre-trained deep learning models namely, EfficientNet[2], DenseNet[3], InceptionV3[4], MobileNetV2[5], and VGG19[6].

1.3 Organization of the Thesis

The thesis is organized into eight chapters. Chapter 1 provides an introduction to the thesis. The rest of the thesis is organized as follows:

- Chapter 2 provides a literature review of several classification-based approaches. In this chapter, we have reviewed the publications that have been conducted on the topic of categorization.
- Chapter 3 provides information about the dataset nomenclature. In this chapter, we have discussed how we have collected the raw data, processed the dataset, and how we have added annotation to the preprocessed data.
- Chapter 4 provides details about the training and testing sets of our dataset JUIVCDv1.
- Chapter 5 provides the basic architecture of the five-state-of-art deep learning models.

- Chapter 6 provides all the results that we have obtained after several testing. This chapter consists comparison of results that are obtained on five base models. The comparison of results includes a training accuracy and validation accuracy curves, training loss and validation loss curves, confusion matrix, and classification report.
- Chapter 7 concluding remarks of this study respectively and also discusses the limitations and the future scope of this thesis.

Chapter 2

Literature Review

AVC is often considered as one of the most challenging computer vision tasks, and it has a long research history in the literature. This section focuses on three aspects of this research problem: (i) AVC datasets publicly available to develop and validate them; (ii) different ways to measure the performance of an algorithm, and (iii) some recent state-of-the-art approaches available for AVC till date. During our study, we observed that there are some very costly and privately available datasets. The datasets that are freely available to the research community have been used so frequently over the years that researchers now get almost 100% accuracy. This success also demands a new database with more challenges portraying real-life scenarios. Also, our country, India, has a different scenario in this domain due to the road condition and multiple vehicles in a single frame. If we use other datasets that are not similar to the Indian road scenario, then the model may not be working properly in real-life conditions.

provides a summary of the AVC databases commonly used by researchers.

Name of database	Number of vehicle classes	Total number of images	Accuracy (%)
MIO TCD [7]	11	648959	97.95
FG3DCar [8]	30	300	95.3
Stanford Cars [9]	196	16185	96.8
BIT vehicle [10]	6	9850	96.1
BoxCars [11]	27	63750	86.57
CompCars [12]	163	136726	99
Poribohon BD [13]	15	9058	98.7
VMMR dataset [14]	9170	291752	92.9
Deshi BD [15]	13	10440	98
Frontal-103 [16]	103	65433	91
LSUN+Stanford [17]	196	2067710	99

Table 2.1: **List of freely image datasets available for AVC.**

There are many studies related to vehicle detection, but the number of classification methodology-based research is very. Maity et al.[18] surveyed this topic in 2021. Below, we have discussed some classification-based methods.

Sun et al.[19] proposed a novel vehicle-type classification using a lightweight convolutional neural network (CNN) with feature optimization and joint learning strategy. The first step was to create a lightweight convolutional network with feature optimization (LWCNN-FO). To minimize the parameters of the network, they employed depth-wise separable convolution. Additionally, the SENet module was included to automatically determine the significance of each feature channel using sample-based self-learning. This increased identification accuracy with little network parameter development. This research also utilized the joint learning technique, which combined softmax loss with contrastive-center loss to classify vehicle kinds, taking into account both between-class similarity and intra-class variation, enhancing the model's capacity to perform fine-grained classification.

Silva et al.[20] proposed an AVC system with a computer vision solution. A camera system was put up to verify the vehicle by taking note of its characteristics and

comparing them to those in the membership. They concentrate on constructing a fine-grained vehicle classification system that used the system's multicamera composition to fuel a CNN with many views of the vehicle in order to solve the problem of the vehicles' make and model classification. The authors suggest utilizing a multi-view network architecture to extract features from various perspectives and subsequently integrating them through late fusion for the purpose of classifying the make and model of a given vehicle. The authors also suggested a methodology for assigning weight to each autonomous perspective in order to enhance collective knowledge acquisition within the network. The evaluations presented indicate that incorporating data from multiple perspectives of a vehicle enhances the classification accuracy of its make and model, particularly in difficult tolling situations.

Ni et al.[21] proposed a vehicle attribute recognition system by appearance. This study presented a survey of current vehicle attribute recognition algorithms, covering both coarse-grained (vehicle type) and fine-grained (vehicle make and model) attributes. This paper aimed to perform vehicle type recognition by categorizing vehicles into broad classifications based on their sizes or intended usage, such as sedans, buses, and trucks. This study also conducted an analysis of Vehicle Make Recognition, which involves the classification of vehicles based on their respective manufacturers such as Ford, Toyota, and Chevrolet. The process of vehicle model recognition involves training a system to make predictions about the make and model of a vehicle, such as the Ford Puma, Toyota Corolla, or Chevrolet Volt. This study involved a higher level of complexity compared to the previously mentioned identification of vehicle makes. It served as a technique for analyzing consumer actions and attitudes towards a particular model of vehicle.

Silva et al.[22] proposed a computer vision solution for an automatic toll collection

(ATC) system that included a subscription/membership feature. This application system established a one-to-one correspondence between a distinct identifier (ID), a tangible automobile, and a membership. A camera-based system was implemented to authenticate that every transaction aligned with the factual membership data by cross-verifying the vehicle and ID information. The visual system employed various algorithms to extract distinct features of a vehicle such as a number plate number, make, model, color, number of axles, and so on. The system performed a comparison between the extracted characteristics and those present in the membership. The authors concentrated on addressing the vehicle's make classification problem and suggested a detailed vehicle classification approach that leverages the system's multi-camera configuration. This was achieved by utilizing a multibranch CNN that took in multiple vehicle perspectives. The network employed a cascade methodology in each of its branches to achieve vehicle localization and extraction of the most significant regions, while also obtaining multi-scale characteristics for each viewpoint. The features that had been extracted were fused in a late manner through a convolutional approach, which was then utilized to classify the make of the vehicle. The neural network utilized feature extraction techniques to differentiate between various perspectives and areas of significance and subsequently combines them optimally to enhance the accuracy of classification. The evaluations presented indicate that the multi-view network architecture proposed has a notable positive impact on the performance of vehicle make classification, as compared to single-view approaches.

Sahin et al.[23] proposed a primary objective of this research is to showcase the utilization of Light Detection and Ranging (LiDAR) sensor data to differentiate between distinct categories of truck trailers, surpassing the capabilities of conventional vehicle classification sensors such as piezoelectric sensors and inductive loop detec-

tors. Utilizing a multi-array LiDAR sensor, it is possible to produce 3D profiles of vehicles by gauging the distance to the object reflecting the emitted light. The present study demonstrates the processing of point-cloud data obtained from a 16-beam LiDAR sensor for the purpose of extracting valuable information and features. The extracted data is then utilized to classify semi-trailer trucks that are transporting ten distinct types of trailers, including a reefer and non-reefer dry van, 20 ft. and 40 ft. intermodal containers, 40 ft. refer intermodal containers, platforms, tanks, car transporter, open-top van/dump, and aggregated other types such as livestock and logging. The field data collected on a motorway segment, comprising of over seven-thousand trucks, is subjected to supervised machine learning algorithms such as K-Nearest Neighbours (KNN), Multilayer Perceptron (MLP), Adaptive Boosting (AdaBoost.M2), and Support Vector Machines (SVM). The outcomes indicate that the Support Vector Machine (SVM) model can effectively differentiate various caravan body types with a remarkably high degree of precision, ranging from 85% to 98%

Liu et al.[24] proposed a new end-to-end CNN architecture that can simultaneously detect and remove adversarial perturbations by utilizing denoising techniques. This approach is referred to as Denoising Detection and Denoising Adversarial Perturbations (DDAP). The DDAP denoiser utilizes the DDAP detector's adversarial examples to eliminate adversarial perturbations. The method being proposed can be considered a pre-processing measure. It does not necessitate any alterations to the configuration of the vehicle classification model and has minimal impact on the classification outcomes of clear images. To validate the capabilities of DDAP, they conducted testing. When training a model, it may be necessary to utilize public datasets such as BIT-Vehicle.

Butt et al.[25] have presented a vehicle classification system that utilizes a CNN to enhance the resilience of vehicle classification in real-time scenarios. The authors provide a dataset of vehicles consisting of 10,000 images that are classified into six distinct vehicle classes. The dataset is designed to account for challenging lighting conditions in order to enhance the reliability of vehicle classification systems in real time. The study involved fine-tuning pre-trained models such as AlexNet, GoogleNet, Inception-v3, VGG, and ResNet on a self-constructed vehicle dataset to assess their accuracy and convergence capabilities. The ResNet architecture has been enhanced by incorporating a novel classification block into the network, resulting in superior performance. To achieve generalization, the network was fine-tuned on the VeRi dataset, a publicly available collection of 50,000 images that have been classified into six distinct vehicle categories. A comparative analysis has been conducted to assess the efficacy of the suggested vehicle classification system in relation to the current methods.

Guo et al.[26] present a semi-supervised approach for vehicle type classification in Intelligent Transportation Systems (ITS) using broad ensemble learning. The methodology outlined comprises two primary components. The initial phase involves training a set of base Broad Learning System (BLS) classifiers using semi-supervised learning techniques to mitigate the growing burden of unlabeled samples and reduce the duration of the training process. In the second phase, a dynamic ensemble architecture is created using trained classifiers that possess distinct characteristics. This ensemble structure yields the highest probability of vehicle classification and determines the vehicle's category, resulting in better generalization performance compared to a solitary base classifier. The authors utilized the publicly available BIT-Vehicle dataset and MIOTCD dataset to conduct experiments and showcase that their pro-

posed method exhibits superior performance in terms of effectiveness and efficiency when compared to a single BLS classifier and other commonly used methods.

Mohine et al.[27] present a study that introduces a hybrid deep 1D CNN-bidirectional long short-term memory (CNN-BiLSTM) approach that utilizes acoustic modality for the purpose of moving vehicle categorization into two-wheeler, low, medium, and heavyweight groups, as well as noise analysis. The algorithm proposed utilizes a sequential process to extract high-level features from signals generated by experimental vehicles. These features are then stored in the network for the purpose of evaluating time-varying characteristics in order to facilitate classification. Furthermore, it underwent testing on the reference dataset SITEX02 to validate its performance, achieving an accuracy rate of 96%. The comparative analysis of the 1D CNN-BiLSTM model's performance has been conducted against traditional classifiers such as SVM, ANN, CNN, and CNN-LSTM models. Based on the empirical findings, it has been observed that CNN-BiLSTM has achieved a superior classification accuracy of 0.92 and a minimum misclassification rate of 0.08 in comparison to traditional classifiers.

Gholamalinejad et al.[28] have proposed a real-time CNN for vehicle type classification system. The proposed convolutional neural network (CNN) architecture incorporates traditional CNN layers, squeeze-and-excitation (SE) modules, and Haar wavelet pooling layers. The aforementioned architecture enhances the efficiency of the CNN classifier by accentuating significant feature maps and reducing the entropy of the network. A loss function based on cross-entropy has been suggested to enhance performance. A novel pooling technique utilizing the Haar transform has been proposed. The selection of network parameters and layer count is optimized for real-time performance. Empirical findings on two vehicular datasets demonstrate that this model outperforms others in terms of overall performance, encompassing recog-

nition time and accuracy. The utilization of Discrete Wavelet Transform (DWT) in lieu of Max-pooling resulted in an enhancement of the recognition accuracy on the Infrared Vehicle Dataset (IRVD) from 97.12% to 99.06%. The proposed model has approximately 5 million training parameters, which is significantly lower than other well-known networks such as VGG (128 million), ResNet152 (58 million), DarkNet (40 million), and Inception-V3 (24 million). The optimization resulted in a significant reduction in recognition time to 42 ms on the CPU, rendering it appropriate for real-time applications.

Chapter 3

Dataset Preparation

In this section the specifics of creating the JUIVCDv1 dataset have been discussed in detail. We have covered data collecting methods, dataset nomenclature, creating images from video data, and annotations in the following subsections.

3.1 Dataset Nomenclature

Our developed dataset has been named as JUIVCDv1, where JUIVCD stands for 'Jadavpur University Indian Vehicle Classification Dataset'. Currently, we have developed version 1 of the JUIVCDv1 dataset. It is to be noted that the dataset has 12 different vehicle classes namely, 'Car', 'Bus', 'Bicycle, 'Ambassador, 'Van, 'Autorickshaw', 'Rickshaw', 'Motorized Two Wheeler', 'Motorvan', 'Toto', 'Truck' and 'Minitruck'. Table Figure 3.1 provides an illustration of each of the vehicle classes, their class names, and class labels.

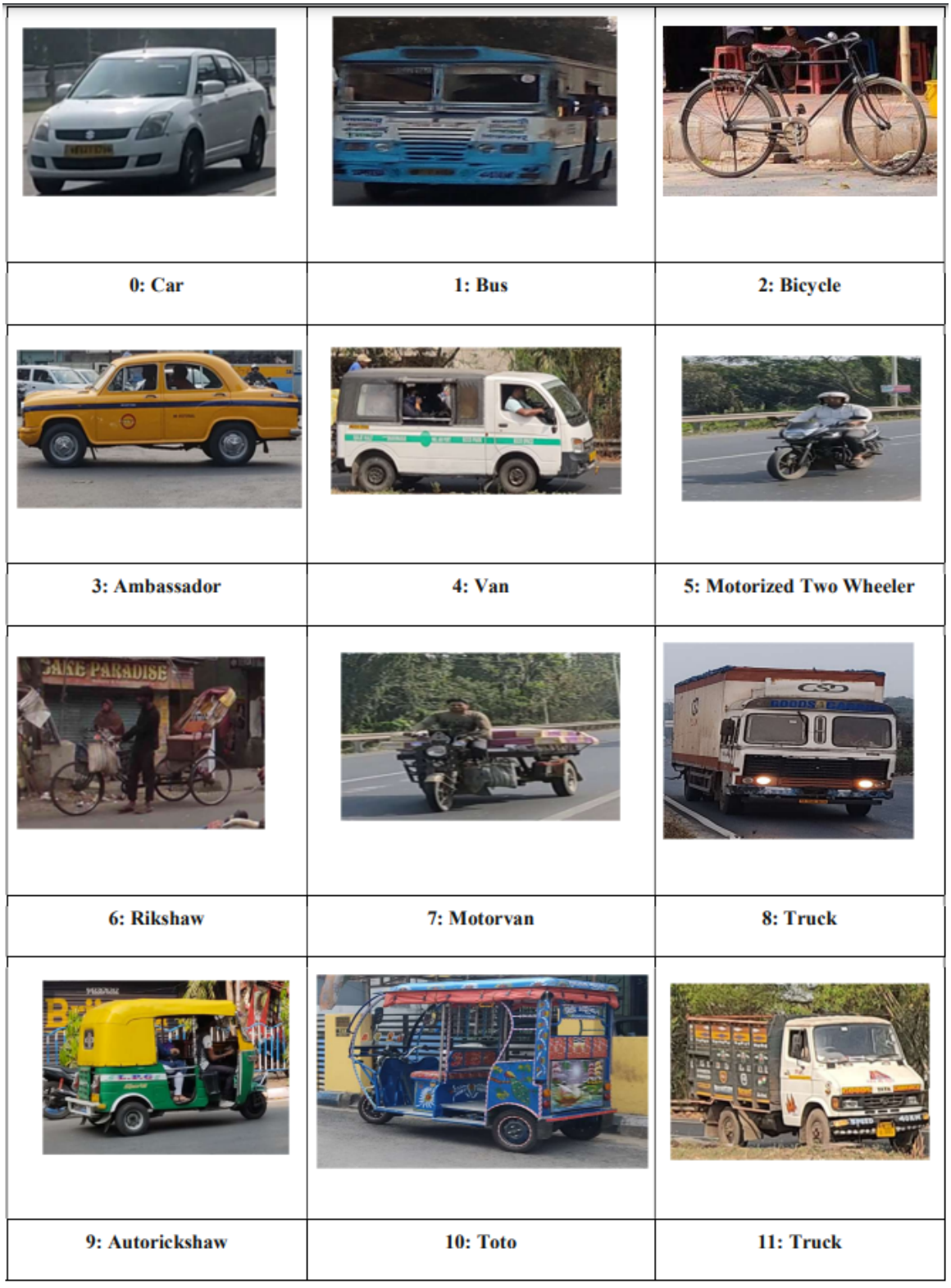

Figure 3.1: **Sample image of different vehicle classes considered in JUIVCDv1 dataset (digits in the caption indicate their class labels**

3.2 Collection of Raw Data Preparation

The information has been collected from highways in Kolkata, an Indian metropolis, and some rural locations around Kolkata. We made every effort to compile as many real-time traffic scenarios as possible. Videos are first taken, and then we used labelImg[29] to extract the frames and generate still images.JUIVCDv1 includes images from both fixed positions as well as from a moving vehicle during daytime and nighttime. We have provided bounding boxes of the vehicles in our dataset. On Indian urban streets, the most realistic traffic situations have been adopted.

- To take the videos and still images, we mostly used two different camera phones:

1. Redmi Note 9Pro (1280x720p)
2. Honor - HRY-AL00 (1080x2340p)

In order to make it easier for the image processing algorithms to process each video, we separated each video into image frames, taking one out of every 15 frames and saving the frame into the JPEG file format. The steps of this procedure are as follows:

1. Cropped frames have at least 30% and up to 70% of the image taken up by automobiles.
2. Now, specific image frames have been chosen such that they are visibly different from each other in the set of chosen frames and are not too fuzzy, i.e., intelligible.

All of the still images that were downsized from video frames, have been randomly divided into the training set (70%) and test set (30%). Depending on the needs, the programmer can divide the training data into a training set and a validation set.

3.3 Annotation of Processed Data

Accurate annotation is a crucial necessity for any developed dataset. Also, the researchers would benefit from having annotations in the test set photographs for evaluating performance when they design a new algorithm. Here format of annotation has been given in TXT and XML formats. Some models require the coordinates of the left top corner of the rectangle and the coordinates of the right bottom corner, along with the class ID. The selected images have been annotated using a standard tool called labelImg[29] tool. Table 3.1 shows the annotation format of JUIVCDv1. Figure 3.2 shows the annotation done on sample images using the said tool. The format of the annotations can be found in the GitHub link, where the dataset is provided. The bounding boxes of the objects are described as:-bx, by, the x and y coordinates representing the center of the box relative to the bounds of the grid cell. The bw, and bh are the predicted width and height of the bounding boxes relative to the entire image and c represents the class of the object. In TXT format '0' is defined as a class of the object and the next values are x_center, y_center, and width, and height respectively where, x_center, y_center are the normalized coordinates of the center of the bounding box and width, height is the normalized width and height of the image. In JSON format, annotation data is represented in the following order: the image name being the first, then the class of the vehicle, the x and y coordinate of the bounding box, and finally width and height of the bounding box.

Figure 3.2: **Annotation on a sample image from JUIVCDv1 dataset.**

Annotation Format in JSON	Annotation Format in JSON
[{"image": "3 (106).png", "annotations": [{"label": "ambasador_taxi", "coordinates": {"x": 204.5, "y": 167.45454545454544, "width": 395.0, "height": 228.0}}]}]	0 0.508706 0.491176 0.982587 0.670588

Table 3.1: **Annotation format of Figure 3.2**

Chapter 4

Details of Dataset

The dataset contains images that can be utilized to create a realistic AVC system that focuses on a typical Indian road scene. The images were taken at various times of the day and night conditions to accommodate every possible diversity of the typical Indian road scene. Images contain a single object or several objects from several vehicle classes in a single frame. In both the training and test sets of the dataset, there are a few extremely complex images with several items of different classifications, which is a regular sight in Indian traffic on a daily basis. This dataset covers every possible road and traffic scenario, including traffic congestion and all types of road conditions in Kolkata and rural locations in and around Kolkata, India. The videos are recorded from both the side of the road and while riding on a bus. This will strengthen the model and provide a diversity of images for the benefit of researchers. For the model to be incredibly robust and operate in any situation, the dataset is intentionally kept unbalanced.

4.1 Training Set

The training set of JUIVCDv1 consists of twelve folders namely '0_Car', '1_Bus', '2_Bicycle', '3_Ambassador', '4_Van', '5_Motorized2wheeler', '6_Rickshaw', '7_Motorvan', '8_Truck', '9_Autorickshaw', '10_Toto', '11_MiniTruck'.

'0_Car' folder has 560 number of images, '1_Bus' folder has 560 number of images, '2_Bicycle' folder has 120 images, '3_Ambassador' folder has 480 images, '4_Van', '5_Motorized2wheeler' and '6_Rickshaw'folders have 560 number of images, '7_Motorvan' has only 33 images, '8_Truck' has 140 images, '9_Autorickshaw' has 564 images, '10_Toto' has 36 images and '11_MiniTruck' has 181 images. A total 4300 number of images are given in the training set of the JUIVCDv1 dataset. Sample Images are shown in Figure 3.1. The distribution of the number of objects in each class in the training set has been shown in Figure 4.1 using bar graph. Here, the Y-axis represents the number of images, while the X-axis represents the number of items in a class.

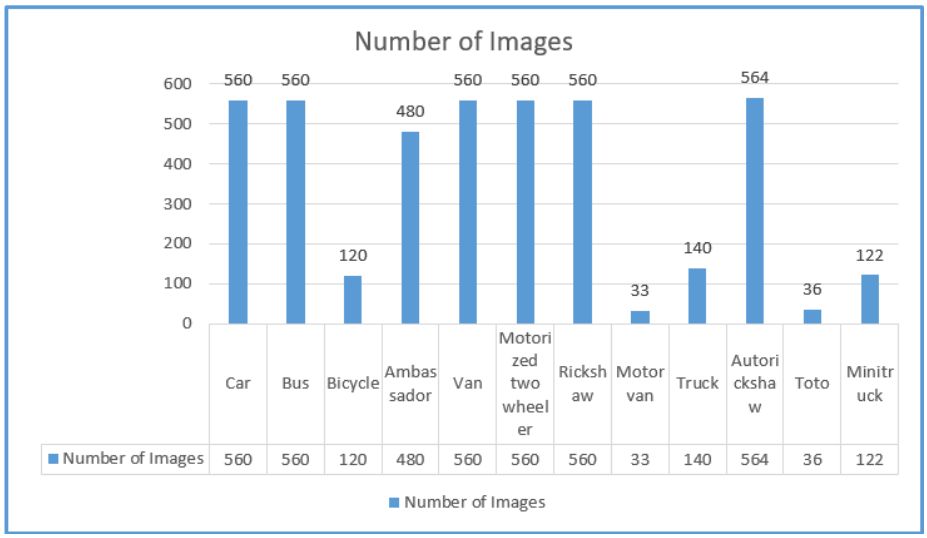

Figure 4.1: **Number of images in training set of JUIVCDv1**

4.2 Testing Set

The test set of JUIVCDv1 consists 12 folders namely '0_Car', '1_Bus', '2_Bicycle', '3_Ambassador', '4_Van', '5_Motorized2wheeler', '6_Rickshaw', '7_Motorvan', '8_Truck', '9_Autorickshaw', '10_Toto', '11_MiniTruck'.

In '0_Car' folder there are 240 number of images, '1_Bus' folder has 240 number of images, '2_Bicycle' folder has 80 images, '3_Ambassador' folder has 320 images, '4_Van', '5_Motorized2wheeler' and '6_Rickshaw'folders have 240 number of images, '7_Motorvan' has only 11 images, '8_Truck' have 59 images, '9_Autorickshaw' has 240 images, '10_Toto' has 23 images and '11_MiniTruck' has 122 images. A total number of 2035 images are given in the test set of the JUIVCDv1 dataset. Sample images are shown in Figure 3.1. The distribution of the number of objects in each class in the test set is shown in Figure 4.2 bar graph. Here, the Y-axis represents the number of images, while the X-axis represents the number of items in a class.

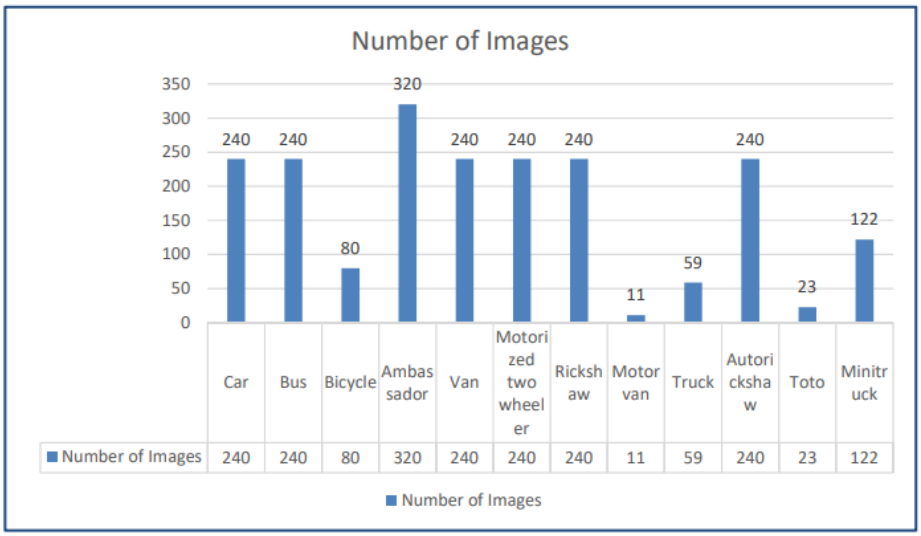

Figure 4.2: **Number of images in test set of JUIVCDv1**

Chapter 5

Benchmarking JUIVCDv1 Dataset

To categorize the automobiles in our database, we have focused on deep learning methods. All the models discussed in the following sections have used different deep-learning algorithms. In order to understand the qualities and the differences between the models, we need a lot of data. Deep learning algorithms have shown real-time competitive performance in comparison to other machine learning algorithms and conventional approaches in various applications, and deep learning models strive to outperform previous results in a given domain.

5.1 EfficientNet

EfficientNet is a CNN design and scaling technique that uses a compound coefficient to consistently scale all depth, breadth, and resolution dimensions. The EfficientNet scaling method uniformly increases network breadth, depth, and resolution using a set of preset scaling coefficients, in contrast to standard practice, which scales these variables arbitrarily. The authors have proposed a scaling technique that utilizes pre-existing Convolutional Neural Networks (ConvNets). To showcase the

efficacy of their scaling method, they have introduced a novel baseline model, named EfficientNet, which is optimized for mobile devices. The architecture of EfficientNet is shown in Figure 5.1.

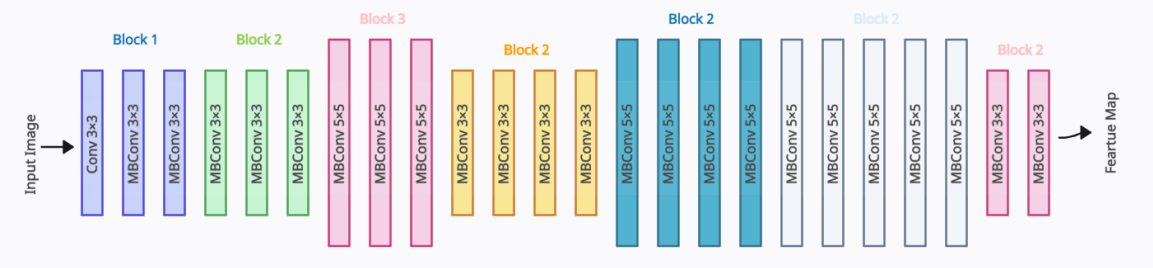

Figure 5.1: **EffientNet Architecture**

5.2 InceptionV3

In 2016, Szegedy et al. proposed a novel method for classification namely InceptionV3 in the study named Rethinking the Inception Architecture for Computer Vision. The Inception-v3 is a CNN model belonging to the Inception family. It incorporates various enhancements such as the utilization of Label Smoothing, Factorized 7 x 7 convolutions, and an auxiliary classifier to disseminate label information to lower network layers. Additionally, batch normalization is employed for layers in the side head. In Figure 5.2 architecture of Inception V3 is shown.

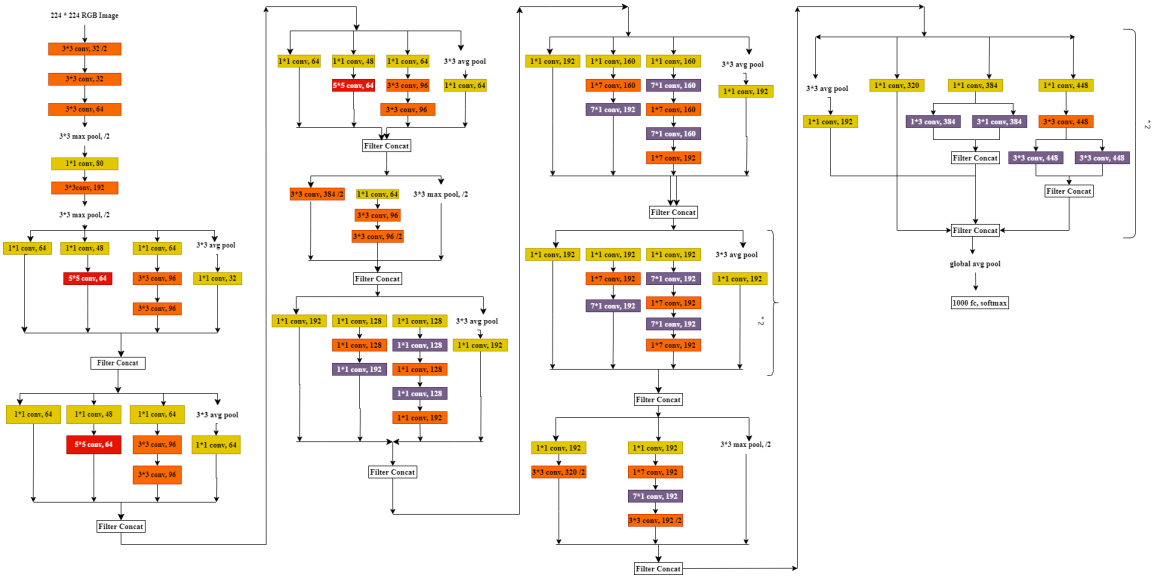

Figure 5.2: **EffientNet Architecture**

5.3 DenseNet121

DenseNet, is a CNN architecture that was recently presented, with an intriguing connection pattern. In this architecture, each layer is connected to all the others within a dense block. In this scenario, all the layers have access to the feature maps of their predecessors, which encourages significant feature reuse across all levels. As a direct result of this, the model is more condensed and less likely to suffer from overfitting. In addition to this, each individual layer is subject to direct supervision from the loss function via the shortcut pathways, which contributes to the provision of implicit deep supervision. Because it possesses so many useful characteristics, DenseNet is an obvious choice for solving issues involving per-pixel prediction. Each layer in DenseNet receives extra inputs from all of the layers that came before it and then pass on its own feature maps to all of the levels that come after it. There is a use for concatenation. Each layer is imparted with some of the "collective knowledge" of

all the layers that came before it. Because each layer is given feature maps from all of the layers that came before it, the network may be made more thin and compact; in other words, there can be fewer channels. The growth rate, shown by the symbol k, is the total number of extra channels added to each layer. As a result, both its computational and memory efficiencies are significantly improved. In Figure 5.3 architecture of DenseNet architecture is shown.

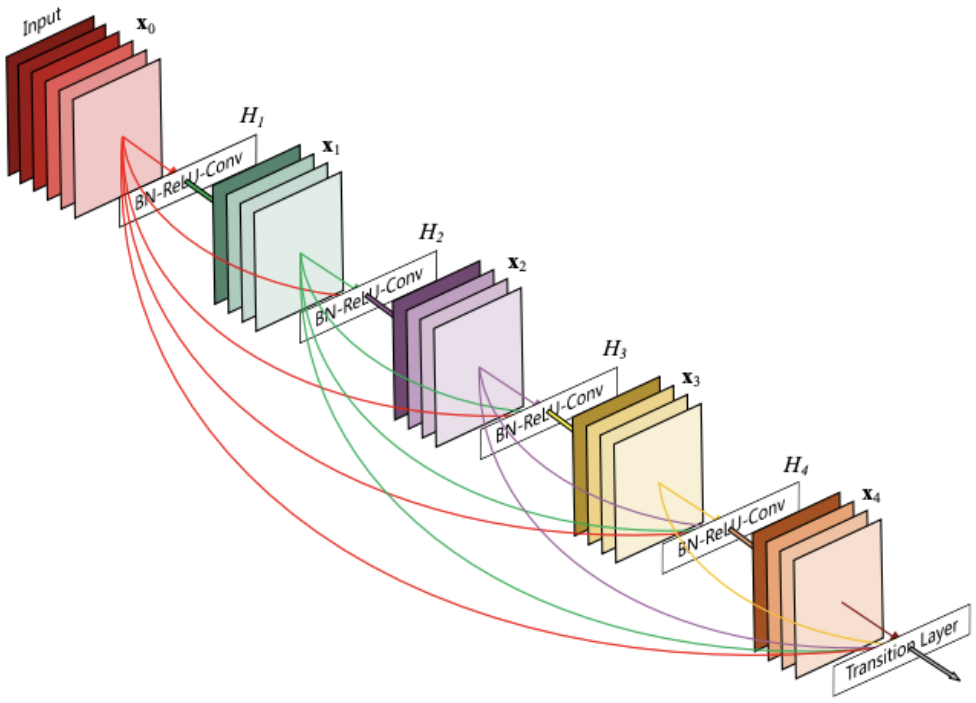

Figure 5.3: **DenseNet121 Architecture**[3]

5.4 MobileNet

An effective model for mobile and embedded vision applications is provided in MobileNet[30]. It is a simplified architecture that builds lightweight deep convolutional neural networks using depthwise separable convolutions. The architecture of the MobileNet model is depthwise separable convolutions, a type of factorized convolution that factors a conventional convolution into a depthwise convolution. Convolution and a pointwise convolution, which is a 1–1 convolution. Each input channel of MobileNet receives a single filter applies to depthwise convolution. The outputs of the depthwise convolution are combined using an 11 convolution after the pointwise convolution. In one step, a conventional convolution filters inputs and combines them into a new set of outputs. This is divided into two layers by the depthwise separable convolution: a layer for combining and a layer for filtering. The result of this factorization is a significant decrease in computation and model size. With the exception of the final fully connected layer, which has no nonlinearity and feeds into a softmax layer for classification, all layers are followed by a batchnorm and ReLU nonlinearity. Modern object detection systems can potentially use MobileNet as an efficient base network. The architecture of MobileNet is shown in Figure 5.4.

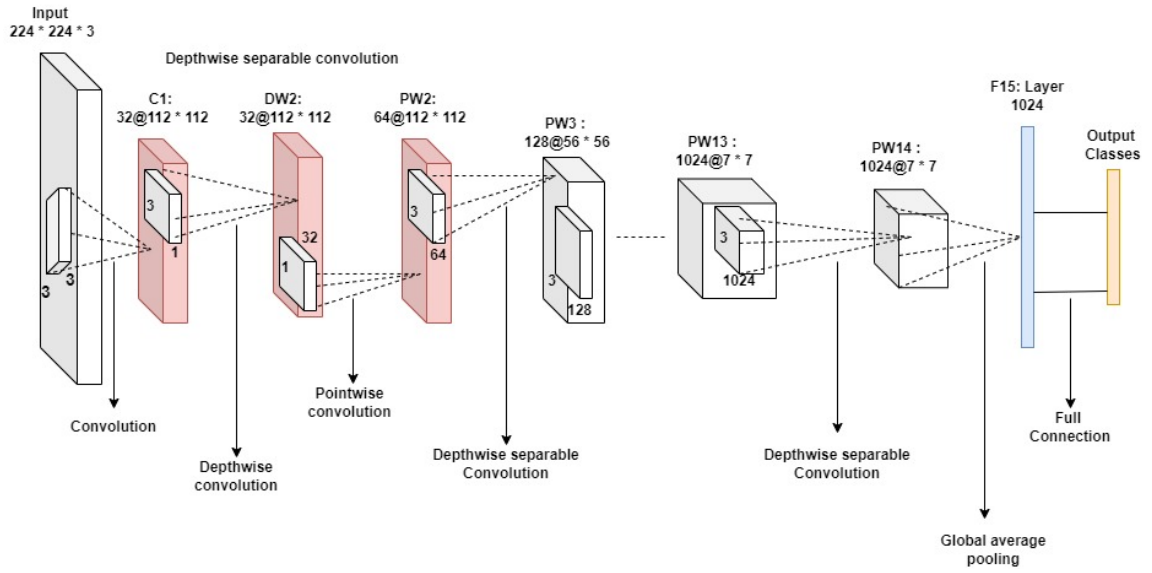

Figure 5.4: **MobileNetV2 Architecture**

5.5 VGG19

In 2014, the Visual Geometry Group—(VGG) at the University of Oxford developed VGG19, which is a CNN architecture. It consists of a total 19 layers which include sixteen layers of convolutional processing and three levels of fully linked processing. VGG19 has used an RGB image with dimensions 224*224 as the input. In the initial layers of the network, convolutional layers consisting of 3*3 filters have been used. The convolutional layers are then followed by max-pooling layers with 2*2 filters. These 2*2 filters can reduce the spatial size of the output of the convolutional layers in half. Of almost the 16 convolutional layers, the first 13 layers employ 3*3 filters and the remaining three use 1*1 filters. When smaller filters are used, a deeper network can be constructed using fewer parameters. The first convolutional layer starts with 64 filters and then works its way up to 512 filters in the final layer.

Each of the completely linked layers that make up the final stage of the network consists of 4096 neurons. A softmax layer serves as the last layer of the network and is responsible for producing the class probabilities The architecture of VGG19 is shown in Figure 5.5.

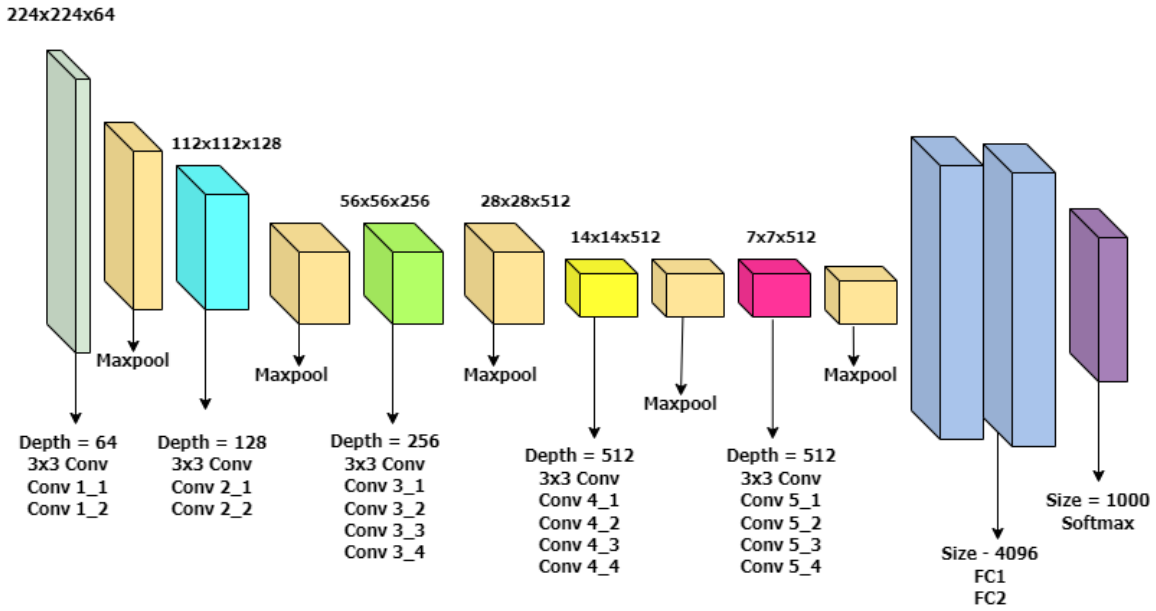

Figure 5.5: **VGG19 Architecture**

Chapter 6

Results and Discussion

The present effort includes the development of a still image dataset for vehicle classification. This dataset is called JUVCsi v1 and it is created keeping the Indian road environment in mind. Five different models are trained and evaluated on the dataset. These models are EfficientNet, InceptionV3, DenseNet121, MobileNetV2, and VGG19.

6.1 Model Evaluation

Evaluation metrics are used to measure the quality of the model. One of the most important topics in deep learning is how to evaluate a model. We have used different models to evaluate our dataset. To analyze these models, various metrics are used. Metrics, such as precision-recall, F1 scores can be applied to evaluate classification-based methods. Five models namely, EfficientNet, InceptionV3, DenseNet121, MobileNetV2, and VGG19 are trained and tested on this dataset. We have achieved 93.82% accuracy on the EfficientNet model, while InceptionV3 has achieved an accuracy of 93.33%, DenseNet121 has achieved an accuracy of 92.16%, MobileNetV2

has achieved an accuracy of 91.28% and VGG19 has achieved 84.48% accuracy. In the following subsections, we have defined the metrics, that are used for the determination of the classification accuracy of the model. Here, we have shown the results obtained by the models.

6.2 Evaluation Metrics for Classification

- **Accuracy**

Accuracy is only a measurement of how frequently the classifier makes accurate predictions. The ratio of the number of accurate forecasts to the total number of predictions is one way to quantify accuracy.

$$\textbf{Accuracy Score} = \textbf{(TP + TN) / (TP + TN + FP + FN)} \tag{6.1}$$

- **Precision**

The precision of a prediction is measured by calculating the number of positive observations that can be anticipated. A low percentage of false positives is indicative of high accuracy.

1. **True Positive (TP):** When an outcome is positive and it is also predicted as positive by the model, it is called True positive.

2. **True Negative (TN):** When an outcome is negative and it is also predicted as negative by the model, it is called True negative.

3. **False Positive (FP):** If the number of outcomes that is negative is predicted as positive, it is called false positive. These errors are also called Type 1 Errors.

4. **False Negative (FN):** When the number of outcomes that are positive is predicted as negative, it is called false negative. These errors are also called Type 2 Errors.

$$\textbf{Precision} = \textbf{TP} \ / \ (\textbf{TP} + \textbf{FP}) \tag{6.2}$$

- **Recall**

Recall calculates the ability of a classifier to find positive observations in the database. The greater the number of false negatives the model predicts, the lower the recall becomes.

$$\textbf{Recall} = \textbf{TP} \ / \ (\textbf{TP} + \textbf{FN}) \tag{6.3}$$

1. **Macro Averaged Recall:** The mean of the recalls for classes A, B, and C is known as the macro-average recall.

$$\textbf{Macro Averaged Recall} = \textbf{1 - Accuracy} \tag{6.4}$$

   It informs us how frequently the model may be incorrect.

2. **Micro Averaged Recall:** The micro average recall score is calculated by dividing the total number of true positives for each class by the total number of true positives across all the classes.

- **F1 score**

It is a single metric that combines both Precision and Recall. The performance of the model improves with increasing F1 scores. The range of F1-score is [0, 1]. The weighted average of recall and accuracy is the F1 score. Both the precision and recall must be high for the classifier to have a high F-score. This metric solely rewards classifiers with comparable recall and accuracy. The F1 score is a measurement that takes into account both accuracy and recall. The harmonic mean is defined as the simple weighted average of accuracy and recall. Using P for precision and R for recall, we can represent the F1 score as:

$$\mathbf{F1 = 2PR \ / \ (P + R)} \tag{6.5}$$

1. **Macro Averaged F1 score:** The macro-averaged F1 score is computed using the arithmetic mean of all the per-class F1 scores.

2. **Micro Averaged F1 score:** By adding the sums of the True Positives (TP), False Negatives (FN), and False Positives (FP), micro averaging determines the global average F1 score. In order to get the micro F1 score, the corresponding TP, FP, and FN values across all classes are added.

- **Confusion Matrix**

The performance of the classification models for a certain set of test data is evaluated using a matrix called the confusion matrix. It can be calculated only after the real values of the test data are known. Although the matrix itself is simple to understand, some of the terminology may be confusing. It is sometimes referred to as an error matrix as it displays the faults in the model performance in the form of a matrix.

6.3 Results obtained by deep learning models

The outcomes of the five deep learning models have been examined in this section. The outcomes of the foundational learners have been displayed graphically. When it comes to identifying certain vehicles, certain models have been observed to be more accurate than others. In addition, a report detailing the accuracy of the classifications has been provided as a classification report, and a confusion matrix is also given.

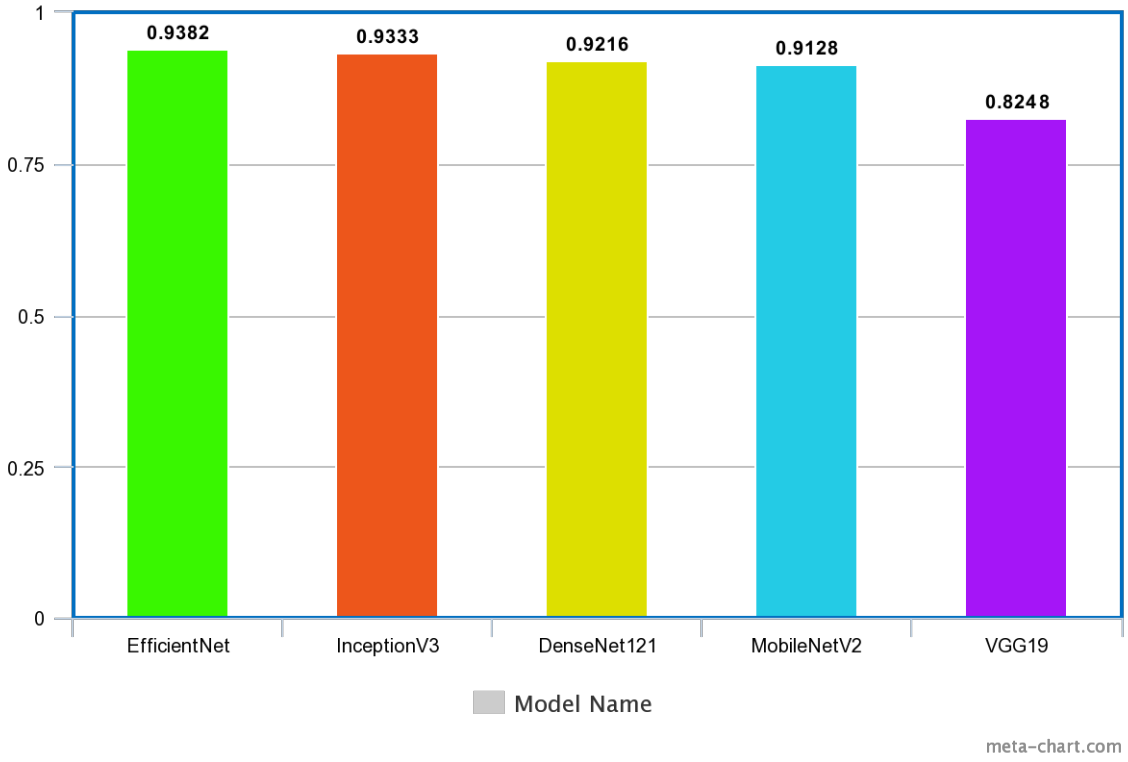

Figure 6.1: **Bar chart representation of test accuracy of five deep learning models used for AVC on JUIVCDv1 dataset**

6.4 Performance Comparison of deep learning models

6.4.1 Performance Comparison: Training Accuracy and Validation Accuracy vs Epoch

Figure 6.2 shows the training accuracy and validation accuracy curve with respect to the number of epochs for the EfficientNet model. We have trained and validated the model EfficientNet for 30 epochs on our proposed dataset JUIVCDv1.

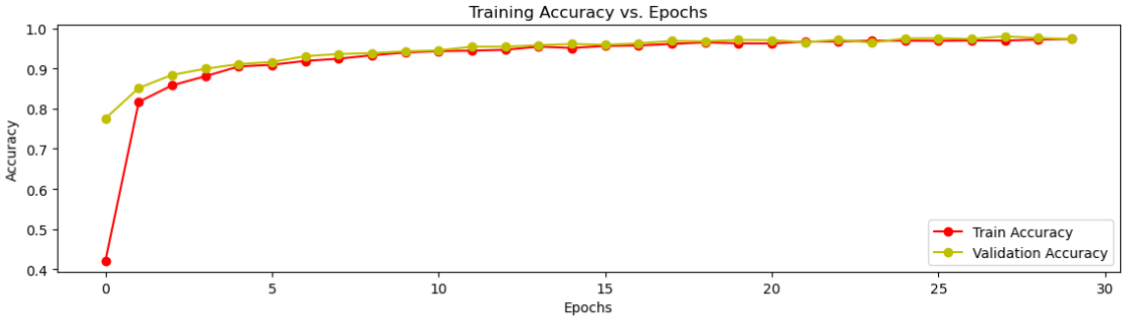

Figure 6.2: **Training and Validation accuracy curve of EfficientNet on JUIVCDv1 dataset**

Figure 6.3 shows the training accuracy and validation accuracy curve with respect to the number of epochs for the InceptionV3 model. We have trained and validated the model InceptionV3 for 25 epochs on our proposed dataset JUIVCDv1.

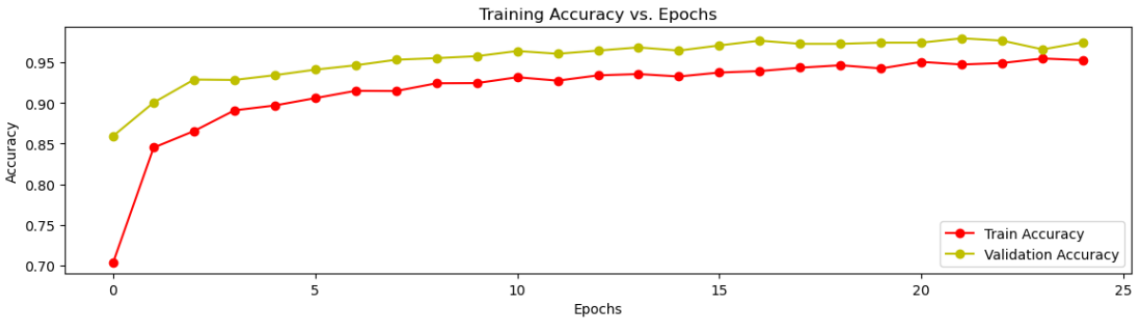

Figure 6.3: **Training and Validation accuracy curve of InceptionV3 on JUIVCDv1 dataset**

Figure 6.4 shows the training accuracy and validation accuracy curve with respect to the number of epochs for the DenseNet121 model. We have trained and validated the model DenseNet121 for 25 epochs on our proposed dataset JUIVCDv1.

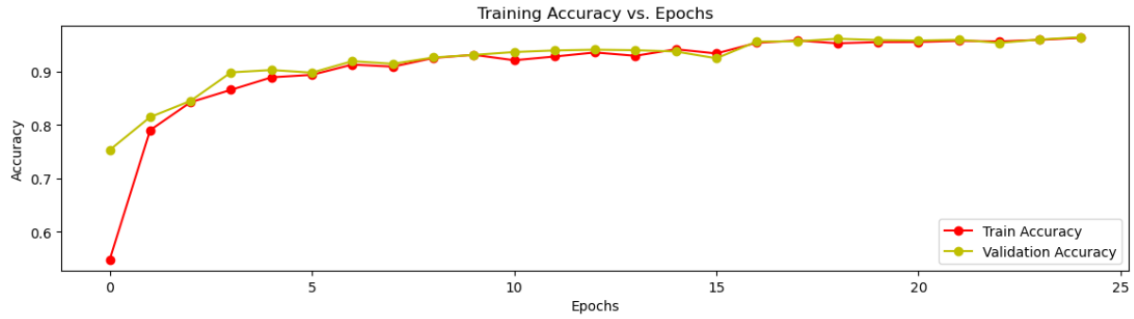

Figure 6.4: **Training and Validation accuracy curve of DenseNet121 on JUIVCDv1 dataset**

Figure 6.5 shows the training accuracy and validation accuracy curve with respect to the number of epochs for the MobileNetV2 model. We have trained and validated the model MobileNetV2 for 120 epochs on our proposed dataset JUIVCDv1.

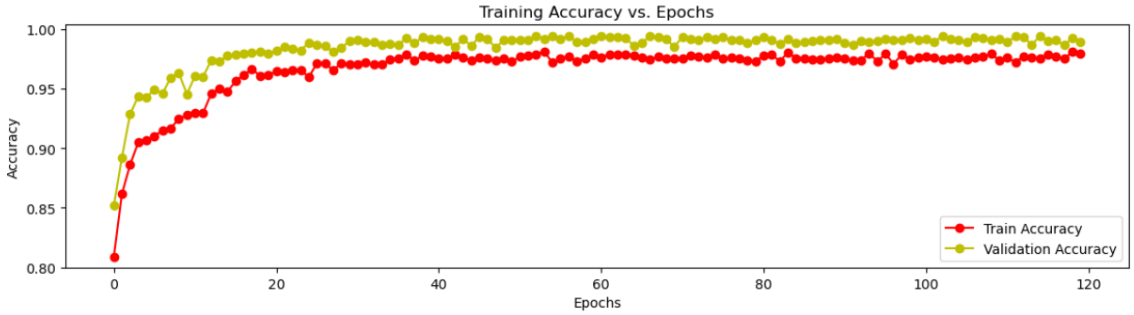

Figure 6.5: **Training and Validation accuracy curve of MobileNetV2 on JUIVCDv1 dataset**

Figure 6.6 shows the training accuracy and validation accuracy curve with respect to the number of epochs for the VGG19 model. The model has been trained and validated for 120 epochs on our proposed dataset JUIVCDv1.

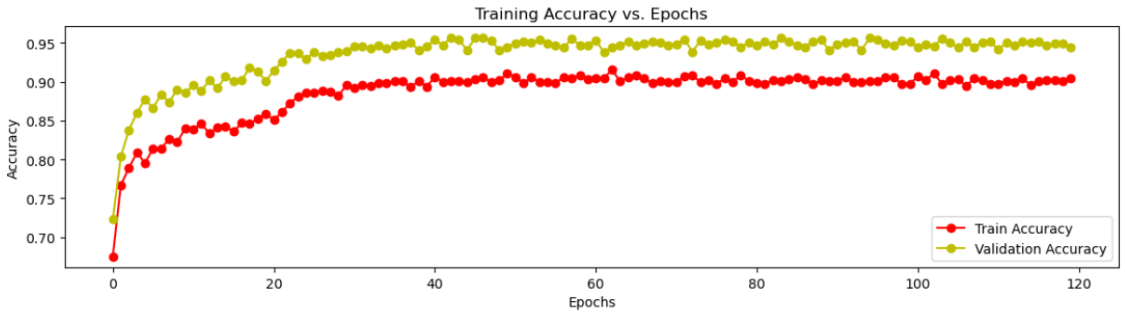

Figure 6.6: **Training and Validation accuracy curve of VGG19 on JUIVCDv1 dataset**

6.4.2 Performance Comparison: Train Loss and Validation Loss vs Epoch

Figure 6.7 shows the training loss and validation loss curves with respect to the number of epochs for the EfficientNet model. We have trained EfficientNet for 30 epochs to achieve this training loss and validation loss curve on our proposed dataset JUIVCDv1.

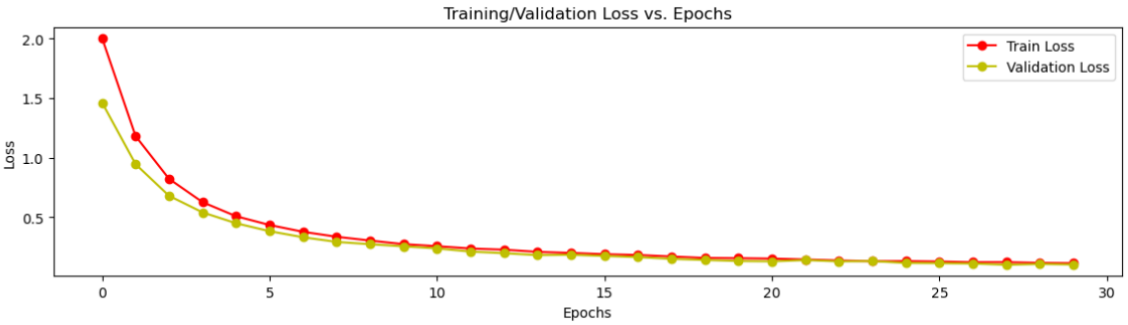

Figure 6.7: **Training Loss and Validation Loss curve of EfficientNet on JUIVCDv1 dataset**

Figure 6.8 shows the training loss and validation loss curves with respect to the number of epochs for the InceptionV3 model. We have trained InceptionV3 for 25 epochs to achieve this training loss and validation loss curve on our proposed dataset JUIVCDv1.

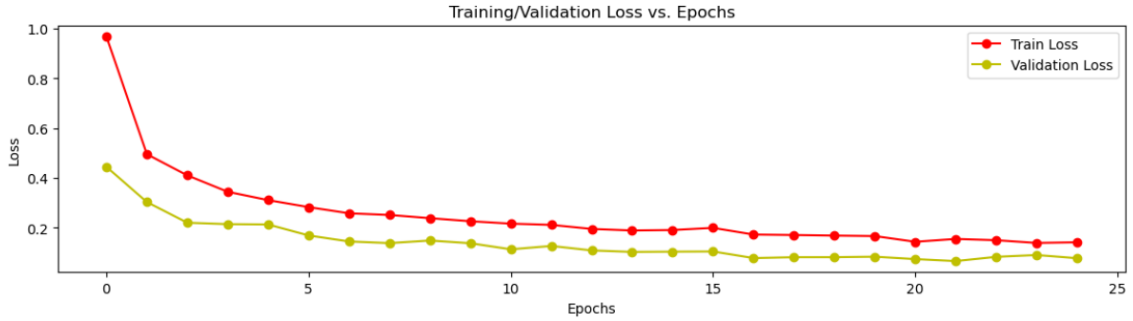

Figure 6.8: **Training Loss and Validation Loss curve of InceptionV3 on JUIVCDv1 dataset**

Figure 6.9 shows the training loss and validation loss curves with respect to the number of epochs for the DenseNet121 model. We have trained DenseNet121 for 25 epochs to achieve these training loss and validation loss curves on our proposed dataset JUIVCDv1.

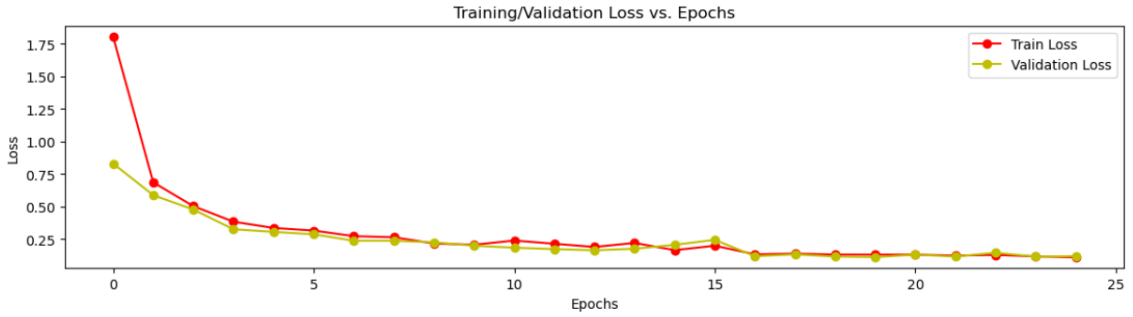

Figure 6.9: **Training Loss and Validation Loss curve of DenseNet121 on JUIVCDv1 dataset**

Figure 6.10 shows the training loss and validation loss curves with respect to the number of epochs for the MobileNetV2 model. We have trained MobileNetV2 for 120 epochs to achieve these training loss and validation loss curves on our proposed dataset JUIVCDv1.

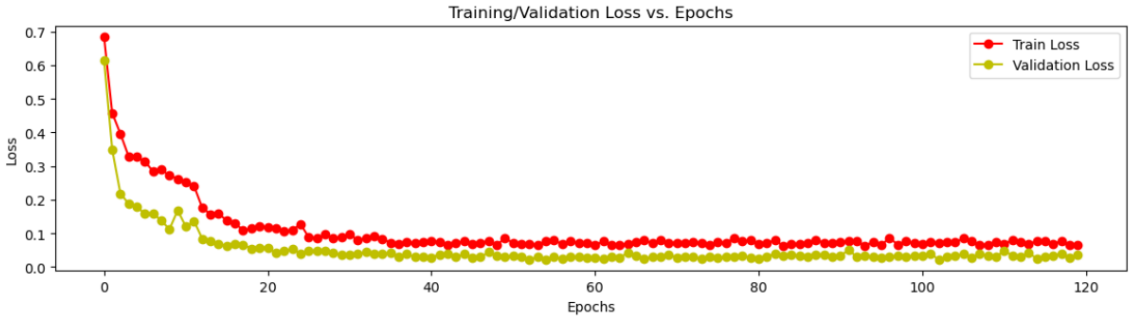

Figure 6.10: **Training Loss and Validation Loss curve of MobileNetV2 on JUIVCDv1 dataset**

Figure 6.11 shows the training loss and validation loss curves with respect to the number of epochs for the VGG19 model. We have trained VGG19 for 120 epochs to achieve this training loss and validation loss curve on our proposed dataset JUIV-CDv1.

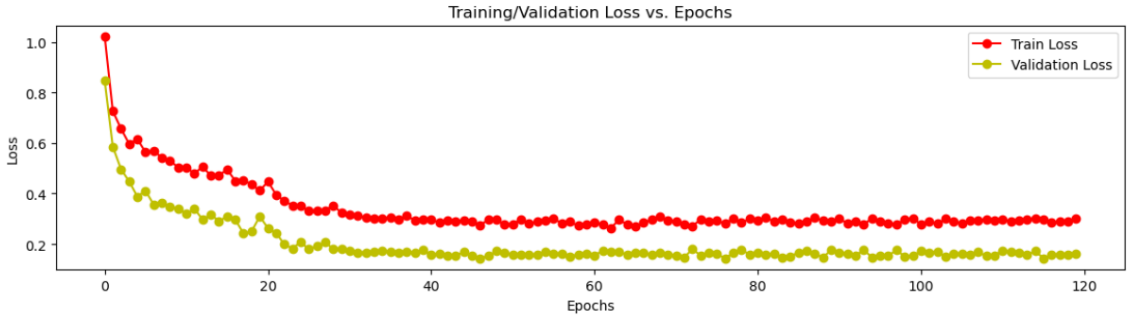

Figure 6.11: **Training Loss and Validation Loss curve of VGG19 on JUIVCDv1 dataset**

6.4.3 Performance Comparison: Confusion Matrix

Figure 6.12 shows the confusion matrix of the base model EfficientNet. From this figure, we have observed that the best true positive value is obtained for the class 0_Car but lowest true positive value is obtained for the class 10_Toto.

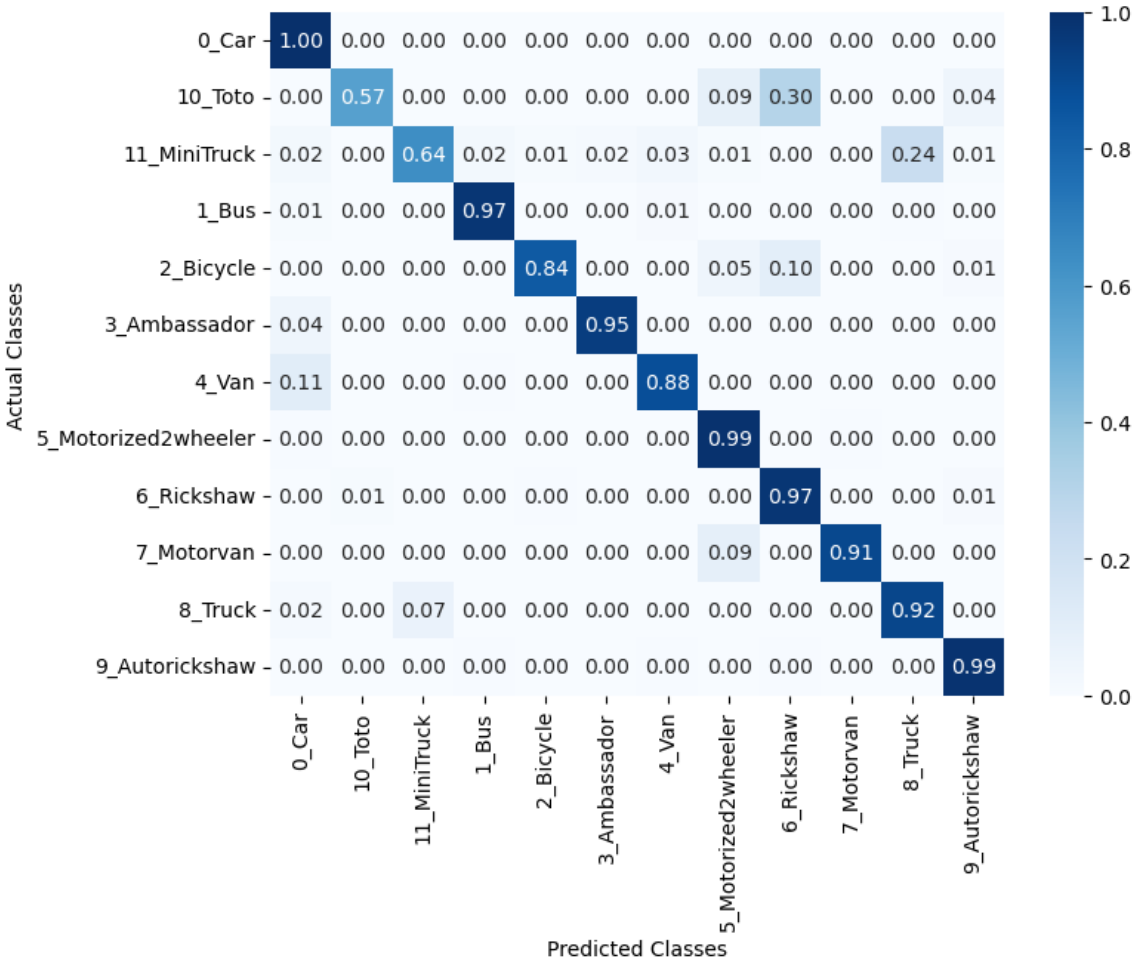

Figure 6.12: **Confusion matrix of EfficientNet on the test set of JUIVCDv1**

Figure 6.13 shows the confusion matrix of the base model InceptionV3. From this figure, we have found that the best true positive value is obtained for the class 0_Car but lowest true positive value is obtained for the class 10_Toto.

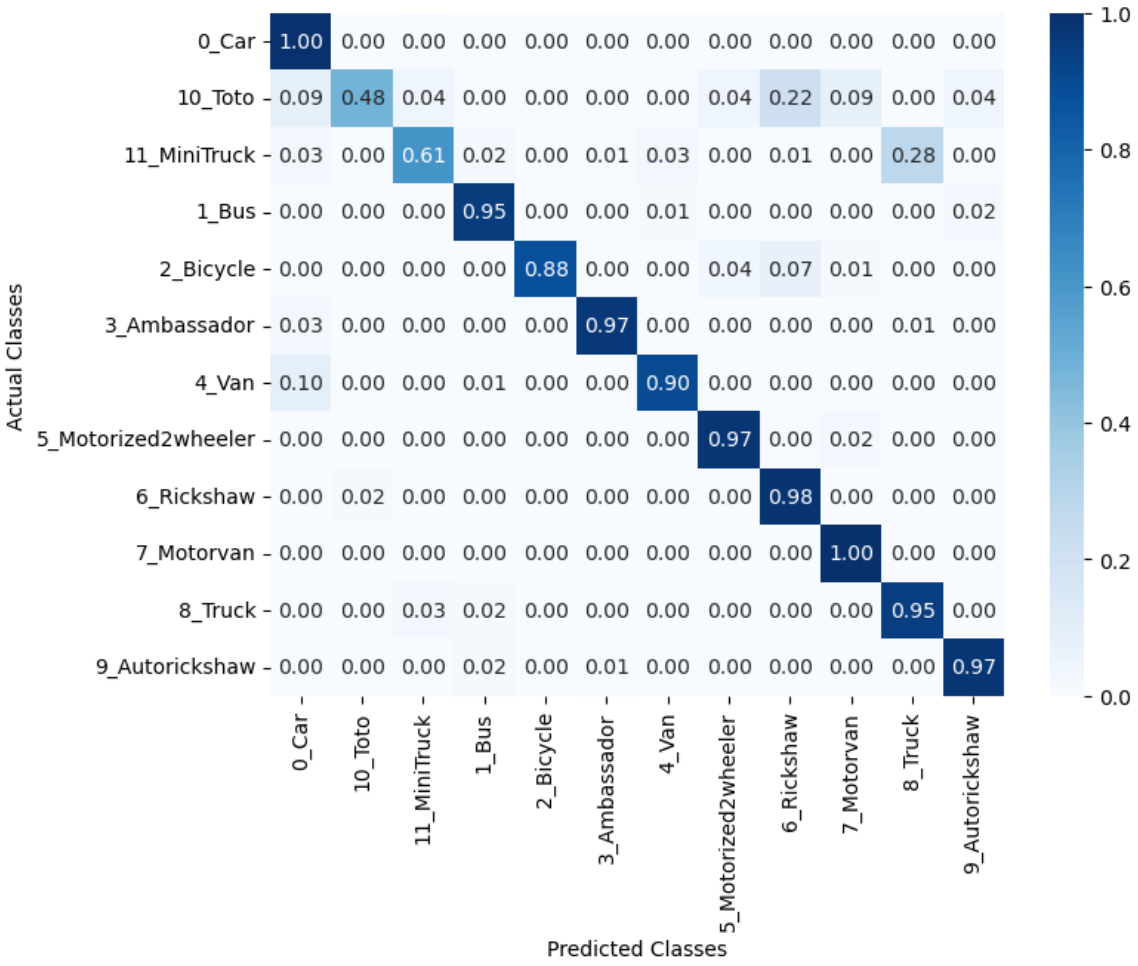

Figure 6.13: **Confusion matrix of InceptionV3**

In Figure 6.14 shows the Confusion Matrix of the base model DenseNet121.From this figure, we have observed that the best true positive value was obtained for the class 0_Car but lowest true positive value is obtained for the class 10_Toto.

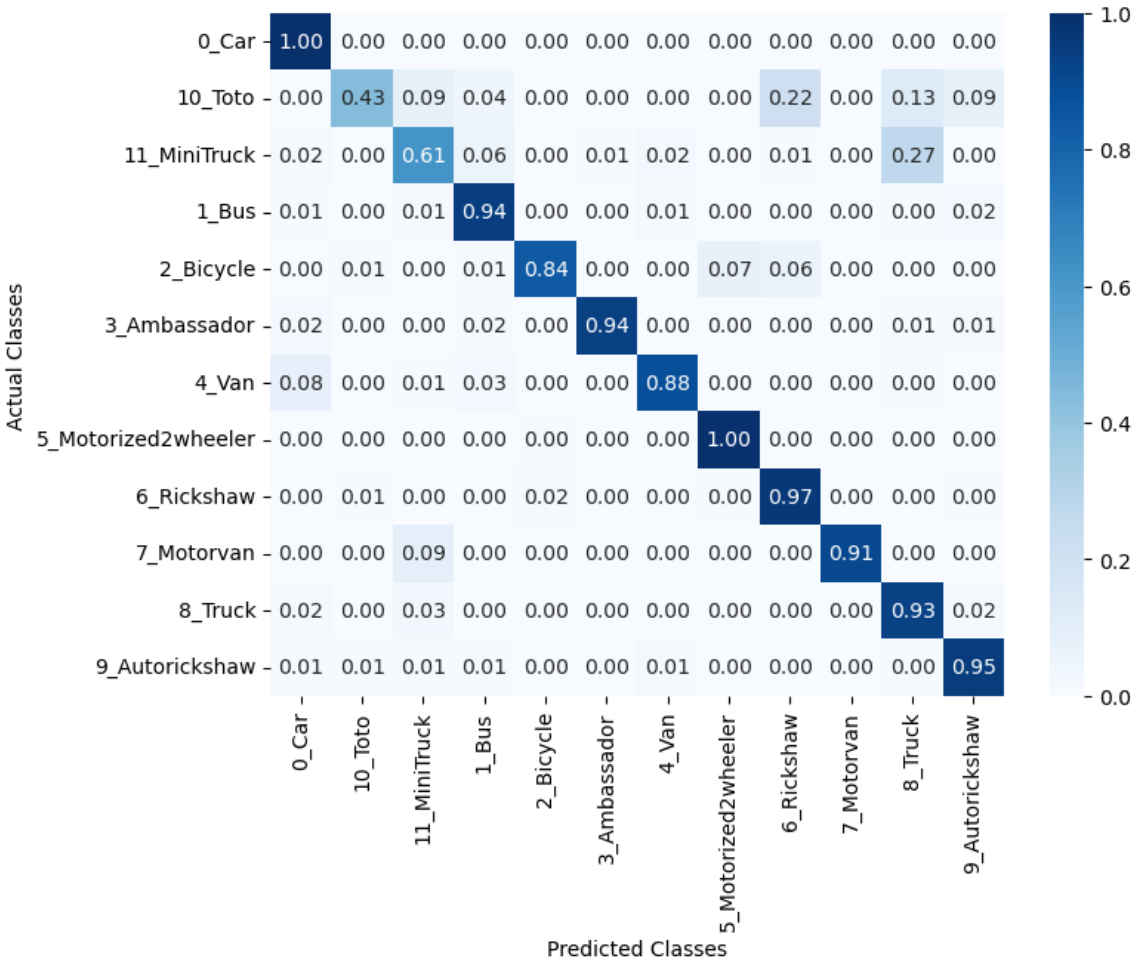

Figure 6.14: **Confusion matrix of DenseNet121 on the test set of JUIVCDv1**

Figure 6.15 shows the confusion matrix of the base model MobileNetV2. From this figure we have observed that, the best true positive value is obtained for the class 0_Car and the lowest true positive value is obtained for the class 10_Toto.

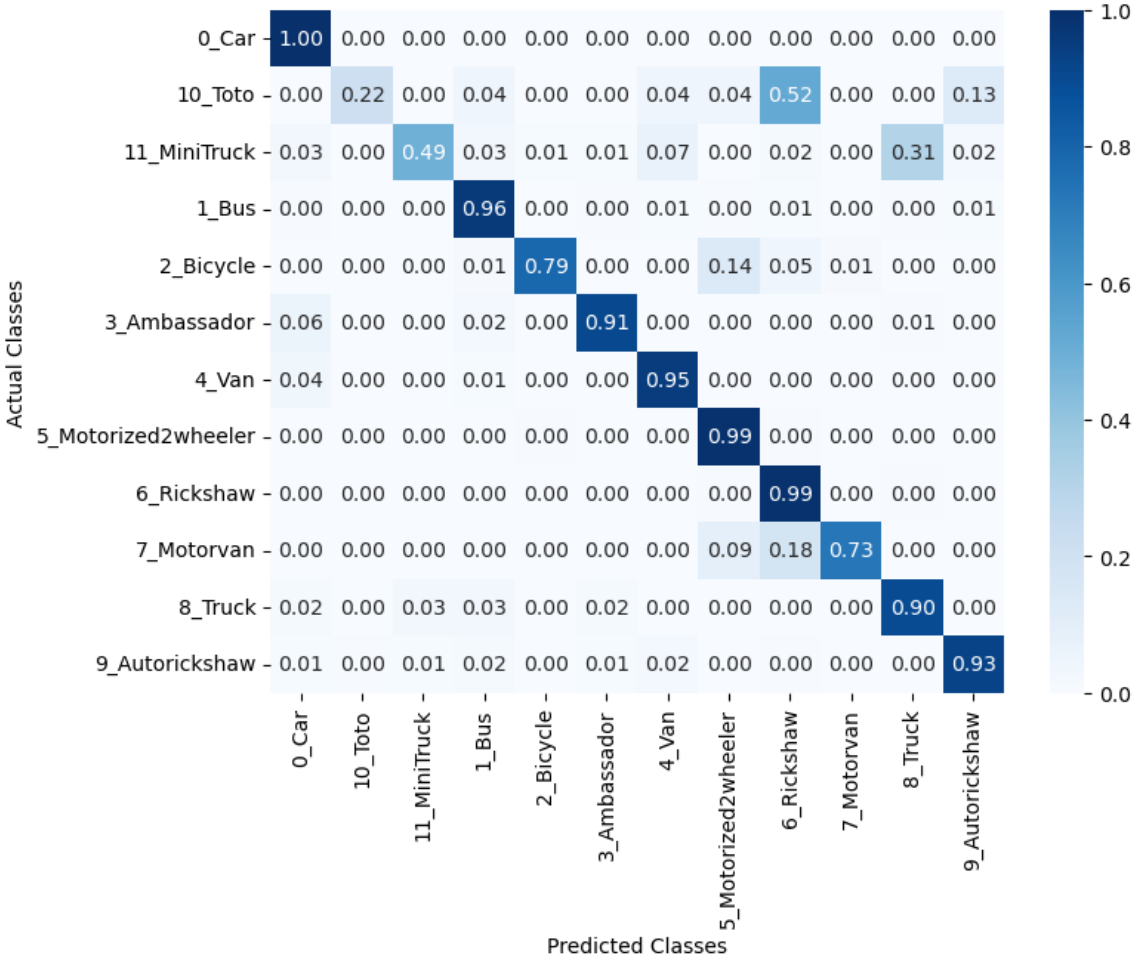

Figure 6.15: **Confusion matrix of MobileNetV2 on the test set of JUIVCDv1**

Figure 6.16 shows the confusion matrix for the base model VGG19. From this figure, we have observed that the best true positive value is obtained for the class 0_Car but lowest true positive value is obtained for the class 10_Toto.

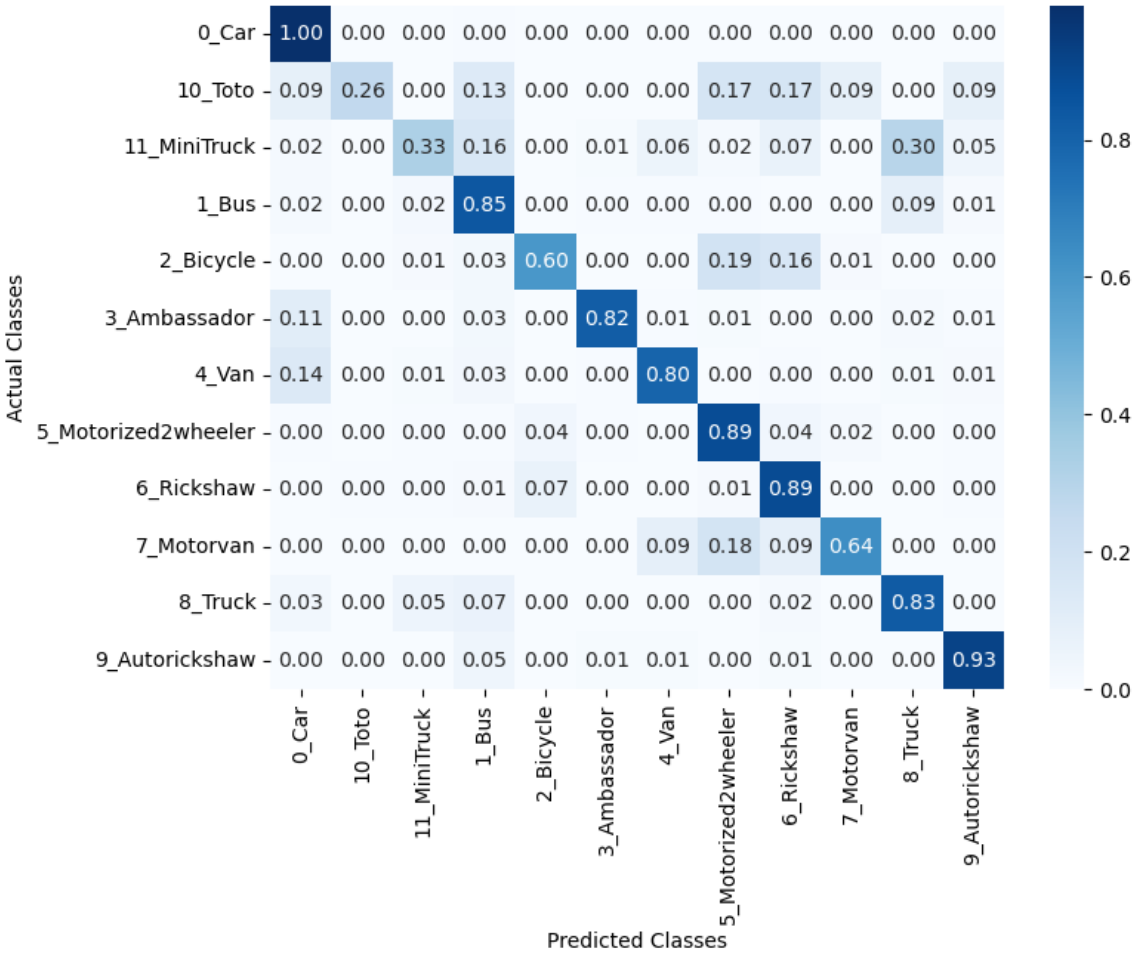

Figure 6.16: **Confusion matrix of VGG19 on the test set of JUIVCDv1**

6.4.4 Performance Comparison: Classification Report

Table 6.1 shows the performance of AVC on the test set of JUIVCDv1 dataset using the EfficientNet model. After evaluating EfficientNet on our dataset, we have observed that the highest precision value of 0.99 is achieved by the class 3_Ambassador and the class namely 8_Truck has the lowest precision value of 0.64. The class 0_Car has achieved highest recall value of 1.00 and the class 10_Toto has the lowest recall value of 0.57. Highest F1-Score has been achieved by the model for two classes, namely 5_Motorized2wheeler and 9_Autorickshaw i.e., 0.98. EfficientNet Model has achieved overall an accuracy of 0.93.

Vehicle Class	Precision	Recall	F1-Score
0_Car	0.83	1.00	0.91
10_Toto	0.87	0.57	0.68
11_Minitruck	0.95	0.64	0.76
1_Bus	0.97	0.97	0.97
2_Bicycle	0.97	0.84	0.90
3_Ambassador	0.99	0.95	0.97
4_Van	0.96	0.88	0.92
5_Motorized2wheeler	0.97	0.99	0.98
6_Rickshaw	0.94	0.97	0.96
7_Motorvan	0.91	0.91	0.91
8_Truck	0.64	0.92	0.75
9_Autorickshaw	0.98	0.99	0.98
OVERALL			
Accuracy	0.93		
Macro Avg.	0.91	0.89	0.89
Weighted Avg.	0.94	0.93	0.93

Table 6.1: **Classification report of the EfficientNet model.**

Table 6.2 shows the performance of AVC on the test set of JUIVCDv1 using the InceptionV3 model. After evaluating InceptionV3 on our dataset, we have observed that the model has achieved the highest precision value of 1.00 for the class 2_Bicycle and the class namely 7_Motorvan has the lowest precision of 0.58. Highest Recall value of 1.00 has been achieved by the model for two classes i.e., 0_Car and 7_Motorvan and 10_Toto has the lowest recall value of 0.48. Highest F1-Score value of 0.98 has been achieved by the model for two classes, namely 3_Ambassador and 5_Motorized2wheeler. InceptionV3 model has achieved an overall accuracy of 0.93.

Vehicle Class	Precision	Recall	F1-Score
0_Car	0.86	1.00	0.92
10_Toto	0.73	0.48	0.58
11_Minitruck	0.94	0.61	0.74
1_Bus	0.96	0.95	0.95
2_Bicycle	1.00	0.88	0.93
3_Ambassador	0.99	0.97	0.98
4_Van	0.97	0.90	0.93
5_Motorized2wheeler	0.98	0.97	0.98
6_Rickshaw	0.95	0.98	0.97
7_Motorvan	0.58	1.00	0.73
8_Truck	0.60	0.95	0.74
9_Autorickshaw	0.97	0.97	0.97
OVERALL			
Accuracy	0.93		
Macro Avg.	0.88	0.89	0.87
Weighted Avg.	0.94	0.93	0.93

Table 6.2: **Classification report of the InceptionV3 model.**

Table 6.3 shows the performance of AVC on the test set of JUIVCDv1 using DenseNet121 model. After evaluating DenseNet121 on our dataset, we have observed that for the classes 3_Ambassador and 7_Motorvan, the model has achieved the highest precision of 1.00 and for the class 8_Truck the model achieved the lowest precision of 0.57. The model has achieved the highest recall value of 1.00 for two classes 0_Car and 5_Motorized2wheeler and the lowest recall value of 0.43 for the class 10_Toto. The Highest F1-Score of 0.98 has been achieved by the model for the class 5_Motorized2wheeler. DenseNet121 model has achieved an overall accuracy of 0.92.

Vehicle Class	Precision	Recall	F1-Score
0_Car	0.88	1.00	0.93
10_Toto	0.67	0.43	0.53
11_Minitruck	0.86	0.61	0.72
1_Bus	0.91	0.94	0.93
2_Bicycle	0.93	0.84	0.88
3_Ambassador	1.00	0.94	0.97
4_Van	0.95	0.88	0.92
5_Motorized2wheeler	0.97	1.00	0.98
6_Rickshaw	0.95	0.97	0.96
7_Motorvan	1.00	0.91	0.95
8_Truck	0.57	0.93	0.71
9_Autorickshaw	0.95	0.95	0.95
OVERALL			
Accuracy	0.92		
Macro Avg.	0.89	0.87	0.87
Weighted Avg.	0.93	0.92	0.92

Table 6.3: **Classification report of the DenseNet121 model.**

Table 6.4 shows the performance of AVC on the test set of JUIVCDv1 using the MobileNetV2 model. After evaluating MobileNetV2 on our dataset, we have observed that the model has achieved the highest precision value of 0.98 for the class 3_Ambassador and the lowest precision of 0.56 for the class 8_Truck. Highest recall value of 1.00 has been achieved by the model for the class 0_Car and the class 10_Toto has the lowest recall value of 0.22. The Highest F1-Score of 0.97 has been achieved by the model for the class 5_Motorized2wheeler. MobileNetV2 model has achieved an overall accuracy of 0.91.

Vehicle Class	Precision	Recall	F1-Score
0_Car	0.87	1.00	0.93
10_Toto	0.83	0.22	0.34
11_Minitruck	0.94	0.49	0.65
1_Bus	0.92	0.96	0.94
2_Bicycle	0.97	0.79	0.87
3_Ambassador	0.98	0.91	0.94
4_Van	0.93	0.95	0.94
5_Motorized2wheeler	0.94	0.99	0.97
6_Rickshaw	0.90	0.99	0.95
7_Motorvan	0.89	0.73	0.80
8_Truck	0.56	0.90	0.69
9_Autorickshaw	0.96	0.93	0.94
OVERALL			
Accuracy	0.91		
Macro Avg.	0.89	0.82	0.83
Weighted Avg.	0.92	0.91	0.91

Table 6.4: **Classification Report of the MobileNetV2 model.**

Table 6.5 shows the performance of AVC on the test set of JUIVCDv1 using the VGG19 model. After evaluating on our dataset, we have observed that the model has achieved the highest precision value of 0.98 for the class 3_Ambassador and for the class 8_Truck the model has achieved the lowest precision of 0.56. The model has achieved the highest recall value of 1.00 for the class 0_Car and the class 10_Toto has the lowest recall value of 0.26. The highest F1-Score of 0.93 has been achieved by the model for the class 3_Ambassador. VGG19 model has achieved an overall accuracy of 0.82.

Vehicle Class	Precision	Recall	F1-Score
0_Car	0.75	1.00	0.86
10_Toto	0.86	0.26	0.40
11_Minitruck	0.75	0.33	0.46
1_Bus	0.77	0.85	0.81
2_Bicycle	0.65	0.60	0.62
3_Ambassador	0.98	0.82	0.89
4_Van	0.94	0.80	0.86
5_Motorized2wheeler	0.88	0.89	0.88
6_Rickshaw	0.84	0.89	0.86
7_Motorvan	0.50	0.64	0.56
8_Truck	0.43	0.83	0.56
9_Autorickshaw	0.93	0.93	0.93
OVER-ALL			
Accuracy	0.82		
Macro Avg.	0.77	0.73	0.72
Weighted Avg.	0.84	0.82	0.82

Table 6.5: **Classification Report of the VGG19 model.**

6.5 Discussion of the Obtained Outcomes

In comparison to the other five CNN models, EfficientNet has achieved the best accuracy of 93.82%, followed by InceptionV3 having 93.33%, DenseNet at 92.16%, MobileNetV2 having 91.28% accuracy, and EfficientNet with 82.48% accuracy and VGG19 with 82.48% accuracy.

Models Name	**Accuracy(%)**
EfficientNet	93.82
InceptionV3	93.33
DenseNet121	92.16
MobileNetV2	91.28
VGG19	82.48

Table 6.6: **Comparison AVC of accuracies obtained on the test set by the different CNN models used here for experimentation**.

- Class imbalance is a major problem in both the training and testing data sets. A large proportion of the still images in the training set include cars and motorcycles, whereas just a small fraction feature totos and bicycles.

- There are 560 cars in the training set, which is sufficient for the models to accurately classify vehicles in the test set. The 240 vehicles in the validation set allow for a more precise model and easier vehicle classification. In contrast, the test set only contains 80 vehicles from the class bicycle, whereas the training set has only 120 bicycles. Therefore, there is not enough data for the CNN models to properly learn bicycle class images. With less inequality across classes, we may have gotten a far better accuracy score.

- The total number of **'Rickshaw'** in this dataset is extremely close to that of a **'Toto'**. There are very few still images of **Toto** in our data collection. As a

result, several **'Toto'** have been incorrectly designated as **'Rickshaws'** in the models. Adding more **'Toto'** data to the base models will help them better distinguish **'Rickshaws'** and **'Toto'** and so to address the problem.

Chapter 7

Conclusion

Nowadays there is a large number of vehicles on the roads, and hence AVC has become more significant in the real-time traffic picture. A realistic image/video dataset is essential for this purpose. Researchers may utilize this dataset to assess the efficiencies of their approaches for both automated localization and classification. Though there are plenty of datasets available for vehicle localization, only a few of them can be used for classification and segmentation purposes. Additionally, not all datasets accurately reflect real-world situations. For example, images taken on the Indian subcontinent, frequently show two or more vehicles overlapping in a single frame owing to traffic congestion. Therefore, researchers have faced difficulties to use this information because of the distinctive features of Indian roads, such as the high volume of traffic, clogged highways, the poor state of the roads, and traffic congestion. In order to fill this research vacuum, we have created an image database suitable for Indian roads in the current study. This dataset can be utilized to classify vehicles. We have included the necessary annotation for the assessment of the AVC algorithms. This dataset is made publicly available for research purposes only. We have benchmarked the results

using five state-of-the-art deep learning models namely EfficientNet, DenseNet121, InceptionV3, MobileNetV2, and VGG19. Finally, we are able to reach an accuracy of 93.28% which is satisfactory given the complexity of the images.

7.1 Limitations and Future Scope

- Version 1 of the dataset has about 6000 images, which may not be sufficient to train deep learning models properly. Therefore, we would like to continue gathering images or videos for the up-gradation of the dataset.
- We will take steps in the future to alleviate the class imbalance found in the current edition of this dataset by gathering more images for certain classes like totos, vans, and rickshaws, which have been rarely included in any of the available datasets.
- We want to take pictures in a variety of weather situations, including foggy, nighttime, rainy, etc.
- We will attempt to include additional types of vehicles that are found on Indian roads.
- We also have a plan to create a video database for AVC.

References

[1] Avirup Bhattacharyya et al. "JUVDsi v1: developing and benchmarking a new still image database in Indian scenario for automatic vehicle detection". In: *Multimedia Tools and Applications* (2023), pp. 1–33.

[2] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.

[3] Gao Huang et al. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[4] Christian Szegedy et al. "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826.

[5] Mark Sandler et al. "Mobilenetv2: Inverted residuals and linear bottlenecks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4510–4520.

[6] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[7] Zhiming Luo et al. "MIO-TCD: A New Benchmark Dataset for Vehicle Classification and Localization". In: *IEEE Transactions on Image Processing* 27.10 (2018), pp. 5129–5141. DOI: 10.1109/TIP.2018.2848705.

[8] Yen-Liang Lin et al. "Jointly optimizing 3d model fitting and fine-grained classification". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*. Springer. 2014, pp. 466–480.

[9] Jonathan Krause et al. "3D Object Representations for Fine-Grained Categorization". In: *2013 IEEE International Conference on Computer Vision Workshops*. 2013, pp. 554–561. DOI: 10.1109/ICCVW.2013.77.

[10] Zhen Dong et al. "Vehicle Type Classification Using a Semisupervised Convolutional Neural Network". In: *IEEE Transactions on Intelligent Transportation Systems* 16.4 (2015), pp. 2247–2256. DOI: 10.1109/TITS.2015.2402438.

[11] Jakub Sochor, Adam Herout, and Jirí Havel. "BoxCars: 3D Boxes as CNN Input for Improved Fine-Grained Vehicle Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 3006–3015. DOI: 10.1109/CVPR.2016.328.

[12] Linjie Yang et al. "A large-scale car dataset for fine-grained categorization and verification". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3973–3981. DOI: 10.1109/CVPR.2015.7299023.

[13] Shaira Tabassum et al. "Poribohon-BD: Bangladeshi local vehicle image dataset with annotation for classification". In: *Data in Brief* 33 (2020), p. 106465. ISSN: 2352-3409. DOI: https://doi.org/10.1016/j.dib.2020.106465. URL: https://www.sciencedirect.com/science/article/pii/S2352340920313470.

[14] Faezeh Tafazzoli, Hichem Frigui, and Keishin Nishiyama. "A Large and Diverse Dataset for Improved Vehicle Make and Model Recognition". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2017, pp. 874–881. DOI: 10.1109/CVPRW.2017.121.

[15] Md Mahibul Hasan et al. "Bangladeshi native vehicle classification based on transfer learning with deep convolutional neural network". In: *Sensors* 21.22 (2021), p. 7545.

[16] Lei Lu, Ping Wang, and Hua Huang. "A large-scale frontal vehicle image dataset for fine-grained vehicle categorization". In: *IEEE Transactions on Intelligent Transportation Systems* 23.3 (2020), pp. 1818–1828.

[17] Tin Kramberger and Božidar Potočnik. "LSUN-stanford car dataset: Enhancing large-scale car image datasets using deep learning for usage in GAN training". In: *Applied Sciences* 10.14 (2020), p. 4913.

[18] Sourajit Maity et al. "Last Decade in Vehicle Detection and Classification: A Comprehensive Survey". In: *Archives of Computational Methods in Engineering* (2022), pp. 1–38.

[19] Wei Sun et al. "Fine-grained vehicle type classification using lightweight convolutional neural network with feature optimization and joint learning strategy". In: *Multimedia Tools and Applications* 80 (2021), pp. 30803–30816.

[20] Bruno Silva, Francisco Rodolfo Barbosa-Anda, and Jorge Batista. "Exploring Multi-Loss Learning for Multi-View Fine-Grained Vehicle Classification". In: *Journal of Intelligent & Robotic Systems* 105.2 (2022), p. 43.

[21] Sara Elkerdawy, Nilanjan Ray, and Hong Zhang. "Fine-grained vehicle classification with unsupervised parts co-occurrence learning". In: *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 2018, pp. 0–0.

[22] Bruno Silva et al. "Multi-View and Multi-Scale Fine-Grained Vehicle Classification with Channel Convolution Feature Fusion". In: *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE. 2021, pp. 3018–3025.

[23] Olcay Sahin, Reza Vatani Nezafat, and Mecit Cetin. "Methods for classification of truck trailers using side-fire light detection and ranging (LiDAR) Data". In: *Journal of Intelligent Transportation Systems* 26.1 (2021), pp. 1–13.

[24] Peng Liu, Huiyuan Fu, and Huadong Ma. "An end-to-end convolutional network for joint detecting and denoising adversarial perturbations in vehicle classification". In: *Computational Visual Media* 7 (2021), pp. 217–227.

[25] Muhammad Atif Butt et al. "Convolutional neural network based vehicle classification in adverse illuminous conditions for intelligent transportation systems". In: *Complexity* 2021 (2021), pp. 1–11.

[26] Li Guo, Runze Li, and Bin Jiang. "An ensemble broad learning scheme for semisupervised vehicle type classification". In: *IEEE transactions on neural networks and learning systems* 32.12 (2021), pp. 5287–5297.

[27] Shailesh Mohine et al. "Acoustic modality based hybrid deep 1D CNN-BiLSTM algorithm for moving vehicle classification". In: *IEEE Transactions on Intelligent Transportation Systems* 23.9 (2022), pp. 16206–16216.

[28] Hossein Gholamalinejad and Hossein Khosravi. "Vehicle classification using a real-time convolutional structure based on DWT pooling layer and SE blocks". In: *Expert Systems with Applications* 183 (2021), p. 115420.

[29] Darrenl Tzutalin. "LabelImg is a graphical image annotation tool and label object bounding boxes in images". In: *URL https://github. com/tzutalin/labelImg* (2022).

[30] Andrew G Howard et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861* (2017).

[31] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.