Vehicle Detection Using Deep Learning:A Comparative Study

A project report submitted in fulfilment of the requirements for the degree of

MASTER OF SCIENCE

IN

INFORMATION TECHNOLOGY

Submitted by

CHIDANGSHA SEKHAR BEZBARUAH (PS-211-811-0004)

Supervised by

Dr. Hasin A. Ahmed
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE
GAUHATI UNIVERSITY, ASSAM
2023

DEPARTMENT OF COMPUTER SCIENCE
GAUHATI UNIVERSITY
GUWAHATI - 781014
ASSAM

CERTIFICATE

This is to certify that the project report entitled **Vehicle Detection Using Deep Learning:A Comparative Study** submitted by **Chidangsha Sekhar Bezbaruah**, for partial fulfilment for the requirement of award of the degree of Master of Science in **Information Technology**, Gauhati University is a work carried out by him under the supervision and guidance of **Dr. Hasin A. Ahmed** .

Date:
Place: Gauhati University

(Dr Anjana Kakoti Mahanta)
Head of the Department
Department of Computer Science

DEPARTMENT OF COMPUTER SCIENCE
GAUHATI UNIVERSITY
GUWAHATI - 781014
ASSAM

CERTIFICATE

This is to certify that the project report entitled **Vehicle Detection Using Deep Learning:A Comparative Study** submitted by **Chidangsha Sekhar Bezbaruah**, for partial fulfilment for the requirement of award of the degree of Master of Science in **Information Technology**, Gauhati University is a work carried out by him under my supervision and guidance.

To the best of my knowledge, the work has not been submitted to any other institute for the award of any other degree or diploma.

Date:
Place: Gauhati University

(Dr. Hasin A. Ahmed)
Supervisor
Assistant Professor
Department of Computer Science

DEPARTMENT OF COMPUTER SCIENCE
GAUHATI UNIVERSITY
GUWAHATI - 781014
ASSAM

CERTIFICATE

The project report entitled **Vehicle Detection Using Deep Learning:A Comparative Study** submitted by **Chidangsha Sekhar Bezbaruah**, for partial fulfilment for the requirement of award of the degree of Master of Science in **Information Technology**, Gauhati University has been examined.

Internal Examiner	External Examiner
Date:	Date:
Place:	Place:

DECLARATION

I hereby declare that the seminar report entitled **Vehicle Detection Using Deep Learning:A Comparative Study** has been compiled by me and submitted in partial fulfilment for the requirement of award of the degree of **Master of Science in Information Technology**, Gauhati University. I also declare that any or all contents incorporated in the report has not been submitted in any form for the award of any other degree of any other institute or university.

Date:
Place: Gauhati University

Chidangsha Sekhar Bezbaruah
Roll No.: PS-211-811-0004
Programme: Information Technology
Semester: Fourth Semester

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who have contributed to the completion of this project. First and foremost, I extend my deepest appreciation to my supervisor **Dr. Hasin A. Ahmed** for his invaluable guidance, support, and encouragement throughout the entire research process.

Chidangsha Sekhar Bezbaruah

ABSTRACT

Vehicle Detection Using Deep Learning:A Comparative Study

A comprehensive study is presented in this project on vehicle detection using Convolutional Neural Network (CNN) models. The aim is to develop an accurate and efficient system capable of distinguishing between images containing vehicles and non-vehicle objects.

The dataset for the project comprises labelled images, consisting of both vehicle and non-vehicle samples. Initially, the images undergo preprocessing to enhance their features and reduce noise. Various image processing techniques, such as resizing, normalization, and augmentation, are applied to improve the quality and diversity of the dataset.

Next, multiple CNN architectures are designed and trained to identify and classify vehicles. Relevant features are automatically learned and extracted from the input images using multiple convolutional layers in the models. The architectures are further augmented with pooling, dropout, and fully connected layers to enhance their discriminative power and mitigate overfitting.

To evaluate the performance of the CNN models, extensive experiments are conducted on the dataset. Evaluation metrics such as training accuracy, validation accuracy, precision, recall, and F1-score are utilized.

Moreover, a comparative analysis of different CNN architectures is performed, including a self customized CNN model and popular variants such as VGG16 and EfficientNet is taken as base models and customized accordingly. The performance of each model is evaluated based on their accuracy, training time, and computational complexity.

The experimental results demonstrate the effectiveness of the proposed CNN models in accurately detecting vehicles from input images. The comparative study reveals the trade-offs between model complexity, training time, and accuracy, enabling the identification of the most suitable architecture for vehicle detection in real-world scenarios.

In conclusion, insights are provided into the application of CNN models for vehicle detection tasks through this project. The system developed showcases the potential of deep learning techniques in achieving accurate and robust vehicle detection, contributing to advancements in various domains, including transportation systems, public safety, and autonomous vehicles.

Keywords: Machine Learning, Convolutional Neural Network (CNN),Basic CNN, VGG16, EfficientNet

ABBREVIATION

The list of Abbreviation used in the dissertation.

ANN: Artificial Neural Network.
CNN: Convolutional Neural Network.
ITS: Intelligent transportation systems.
HOG: Histogram of Oriented Gradients.
SVM: Support vector machines.
FFT: Fast Fourier Transform.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Vehicle detection is critical in a wide range of applications, from traffic surveillance to autonomous driving systems. Vehicle detection must be accurate and efficient in order to ensure road safety, optimise traffic flow, and enable advanced driver assistance systems. Significant progress has been achieved in the field of computer vision in recent years, particularly in the application of Convolutional Neural Networks (CNNs) for object detection tasks. The goal of this research is to investigate and construct a robust vehicle identification system based on CNN models. The major goal of this research is to create a CNN-based vehicle recognition system that can effectively identify vehicles in a variety of real-world circumstances. A vast dataset of labelled images of vehicles and non-vehicle items are used to train the system. The CNN models will learn to automatically extract high-level characteristics and spatial representations from input images by utilising the power of deep learning techniques, allowing them to distinguish vehicles from varied backdrops, scales, and orientations.

1.1 Problem Statement

The goal of this research is to use machine learning techniques, notably Convolutional Neural Networks (CNNs), to recognise vehicles accurately and efficiently in real-world circumstances. The aim is to create a robust vehicle recognition system capable of identifying and classifying cars in a variety of contexts with varied scales, orientations, and backgrounds.Existing vehicle identification algorithms frequently have issues in accurately detecting vehicles in difficult settings such as occlusion, changing lighting, and cluttered backdrops. Traditional computer vision algorithms may have difficulty generalising well across varied circumstances, resulting in false positives or missing detections. As a result, a dependable and efficient vehicle detecting system that can perform well under a variety of real-world scenarios is required.

The project aims to address the following key challenges:

1.Accurate Detection: Creating a vehicle detection system with high precision and recall that can properly identify and localise vehicles in complex scenarios. The system should be able to handle occlusion, partial vehicle visibility, and vehicles of varied scales.

2.Environmental Robustness: Creating a system that can manage fluctuations in lighting, weather, and crowded backgrounds that are regularly seen in real-world circumstances. The system should be adaptable and generalizable across settings.

3.Real-Time Performance: Creating an efficient vehicle identification system capable of processing incoming photos or video streams in real time. To be practical in real-world applications such as enhanced driver assistance systems or surveillance systems, the system should be able to achieve quick inference times.

4.Generalization:Development of a vehicle identification system that can generalise successfully across varied datasets and contexts. To ensure the system's ability to recognise vehicles

accurately in new, unknown surroundings, it should be trained on a broad and representative dataset.

1.2 Machine Learning

Machine Learning refers to a system's ability to self-train itself, so that it learns naturally from a data set and improves with experience without being explicitly programmed [2]. Machine Learning has been used in a variety of industries to enable automated and intelligent systems to produce accurate predictions and classifications.

Machine Learning approaches combined with Computer Vision and Image Processing algorithms provide a powerful solution for detecting vehicles in a precise and effective manner. It makes use of algorithms and statistical models to recognise patterns and make predictions based on the information supplied. Machine Learning algorithms can be trained on a dataset including labelled photos of vehicles and non-vehicles. The algorithms learn to discern between vehicles and non-vehicles and reliably anticipate the presence of vehicles in unseen photos by analysing the patterns and features within the images.

1.2.1 Types of Machine Learning

i.Supervised Machine Learning: Supervised learning is a popular method for detecting Vehicles. It entails training a model on a labelled dataset, with each image associated with a vehicle or non-vehicle label. The programme learns patterns and features from labelled data and then uses the learnt patterns to reliably categorise new, unseen vehicle and non-vehicle images. Convolutional Neural Networks (CNNs) and Support Vector Machines (SVMs) are common supervised learning methods used in object detection.

ii.Unsupervised Machine Learning: When a dataset lacks cases with labels, unsupervised learning is used. Without any predetermined classes, it focuses on identifying structures, patterns, or commonalities in the data. In order to group similar objects together based on their attributes and identify probable clusters or patterns within the dataset, unsupervised learning methods like Clustering techniques (e.g., K-means, DBSCAN) can be used.

iii.Semi-supervised Machine Learning: Obtaining a sizable labelled dataset for training can occasionally be difficult or timeconsuming. To create a powerful model, semi-supervised learning mixes labelled and unlabeled data. A model could be trained using a small batch of labelled photos and a larger set of unlabeled images in order to detect vehicles. The performance of the model may be enhanced by using this strategy to more effectively utilise the available labelled data.

Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN) are commonly used in the task of object detection due to their ability to distinguish complicated patterns in image data. An ANN may be trained to recognise automobiles from images using tagged photos as input and matched object classes as output. The network learns to distinguish patterns and features in photos by using hidden layers of interconnected neurons, allowing it to categorise new, previously viewed images based on its category classification. CNNs perform exceptionally well in image processing applications such as object detection. They are composed of convolutional layers that employ filters to extract relevant information from input images and collect spatial data. These features are subsequently fed into fully connected layers by the categorization process.

1.3 Convolutional Neural Network

Convolutional Neural Network(CNN) is a deep learning method that is built specifically for processing and analysing structured grid-like data, such as photos and videos. CNNs have revolutionised the field of computer vision and are widely used for image classification, object detection, and picture segmentation, among other tasks.CNNs are inspired by the structure and operation of the human visual system. They are made up of numerous interconnected layers that learn hierarchical representations of input data collectively. Convolutional layers, pooling layers, and fully linked layers are crucial components of a conventional CNN design.

There are numerous essential steps in the vehicle detection process using CNNs. To begin, a big dataset of labelled images of vehicles and non-vehicles is created. This dataset is used to train the CNN model, which uses internal parameters to learn to recognise and distinguish vehicle attributes.Metrics such as accuracy, precision, recall, and F1-score are used to assess the effectiveness of a CNN-based vehicle identification system. These metrics evaluate the system's capacity to detect vehicles correctly while minimising false positives and false negatives. The robustness, efficiency, and generalisation of the system across multiple settings are other essential considerations.

1.3.1 Variations of Convolutional Neural Network

i. VGGNet: VGGNet is a well-known CNN architecture that is well-known for its simplicity and efficacy. It is made up of several convolutional layers, each with a small 3x3 filter and a pooling layer for downsampling [6]. Because of its deep architecture, VGGNet can learn detailed patterns and features from images, making it suited for object detection tasks.

ii. ResNet: Residual Network (ResNet), is commonly used in computer vision tasks such as detecting object. Its skip connections allow the network to skip specific layers, allowing for improved gradient flow and quicker deep network training. The capacity of ResNet to handle deep architectures allows it to capture fine-grained details and subtle patterns in images [3].

iii. NASNet: NASNet (Neural Architecture Search Network) is a neural architecture search network architecture. It searches for and identifies optimal network designs for a specific task. By designing effective topologies that maximise detection accuracy, NASNet has demonstrated promising results in a variety of computer vision tasks, including object detection [10].

iv. EfficientNet: EfficientNet is a scalable CNN architecture that achieves cutting-edge image categorization performance. To balance model depth, width, and resolution, it employs a compound scaling mechanism. EfficientNet models, such as EfficientNet-B0 to EfficientNet-B7, provide a variety of architectures with varying complexities, allowing practitioners to select the best model based on computational resources and dataset size [7].

1.4 Aim of the Work

The project aims to create a system that can recognise and classify automobiles in images automatically. To achieve precise and efficient vehicle recognition, the project employs Convolutional Neural Networks (CNNs), a sort of deep learning technique.Throughout the project, the emphasis is on obtaining precise identification, real-time performance, and environmental robustness. The ultimate goal is to create a dependable and efficient vehicle detection system that can be integrated into a variety of applications such as traffic monitoring, autonomous driving, and enhanced driver assistance systems.The project intends to increase vehicle identification technologies by harnessing the capabilities of CNN models and training them on labelled datasets. This will improve safety, efficiency, and decision-making in transportation and automotive systems.

1.5 Oraganization of the report

This section describes the problem statement,machine learning techniques,a short description of convolutional neural network and aim of the work.In section 2,I have discussed about traditional vehicle detection approaches and CNN based vehicle detection approaches.Section 3 in the report contains the dataset description with data preprocessing and data augmentation.In the section 4, I have mentioned about proposed model of CNN based architecture,training and testing approach and hyperparameter tunining. In this section I also experiment with pretrained model such as VGG16 and EfficientNet as transfer learning techniques. Section 5 describes the results of the different models used.Lastly Section 6 describe the conclusion and future work of the study.

Chapter 2

Literature Review

2.1 Traditional Vehicle Detection approaches

In recent years, there has been a lot of interest in developing an intelligent transportation system. Furthermore, with an increasing number of cars on the road, most countries are implementing intelligent transportation systems (ITS) to handle issues like as traffic flow density, queue length, average traffic speed, total vehicles passing at a site in a given time interval, and so on. ITS assists traffic control centres in monitoring and regulating traffic by recording traffic photos and videos with cameras. Vehicle detection must be accurate and reliable for the ITS to function properly.Based on video processing systems, this paper[4] examines various methodologies and applications utilised around the world for vehicle detection under various environmental situations. This study also addresses the many types of cameras used for vehicle detection, as well as vehicle classification for traffic monitoring and control. Finally, this study shows the difficulties encountered during surveillance during harsh weather conditions.Traditional computer vision techniques were commonly utilised for vehicle recognition prior to the emergence of deep learning and convolutional neural networks. Here are some traditional methods for detecting vehicles:

i. Haar Cascade Classifiers: Haar cascades are prominent machine learning-based object detection classifiers. To detect objects, these classifiers use features collected from the Haar-like wavelet transform. Haar cascades can be trained to recognise vehicles based on specific characteristics such as edges, corners, and textures.

ii. Histogram of Oriented Gradients (HOG): The HOG technique reflects the shape and appearance of an item in an image by computing histograms of gradient orientations. It detects and recognises things by capturing local picture gradient patterns. For vehicle detection, HOG features can be combined with machine learning algorithms such as support vector machines (SVM).

iii. Template Matching: Template matching is the process of comparing a template image of a vehicle to distinct sections of an input image. Vehicles can be recognised by locating the best match between the template and picture regions. Template matching is based on pixel-level similarity measurements such as cross-correlation or sum of squared differences.

iv. Background Subtraction: Background subtraction is a video surveillance technique used to detect moving objects. To isolate the foreground objects, the background scene is modelled and subtracted from the current frame. Moving cars can be spotted by thresholding the difference image.

v. Edge-based Methods: To identify vehicle edges or borders, edge detection methods such as the Canny edge detector can be used. Edges are distinguished by sudden variations in pixel intensities, which can aid in distinguishing cars from the backdrop.

vi. Methods Based on Motion: Motion-based approaches locate locations with considerable

motion in consecutive frames by using optical flow or frame differencing. Moving cars generate significant motion, allowing them to be detected and tracked.

Traditional techniques frequently necessitate handcrafted features, explicit modelling, and meticulous parameter adjustment. While they may not attain the same level of accuracy as deep learning-based methods, they can nevertheless be useful in some contexts or as complementing techniques when used in conjunction with deep learning approaches.

2.2 Different CNN based vehicle detection approaches

Object detection, tracking, and categorization have a wide range of applications. In the Intelligent Transportation Systems (ITS) business, object detection is utilised for vehicle and pedestrian detection, traffic sign and lane detection, and vehicle make detection. The ability to detect or classify traffic-related things allows for greater road and traffic flow improvement, the prevention of serious traffic accidents, and even the reporting of traffic offences and crimes such as stolen vehicles or speeding [5]. This is especially important considering the increasing popularity of passenger vehicles. Furthermore, self-driving cars have recently acquired popularity.Humans can easily recognise vehicles in photos or movies and distinguish between different car kinds. The difficulty of vehicle detection and classification for a computer programme is heavily dependent on the type of data. The lighting and weather conditions are one of the most difficult problems, not to mention the overall quality of the photographs or video. Vehicles come in a variety of shapes and colours, although some models may be very identical. Furthermore, detecting a large number of moving objects in real time is unquestionably more difficult.

The goal of this thesis is to develop a convolutional neural network (CNN) to perform vehicle detection and classification on vehicle and background images. More precisely the objectives are as follows:

1.Implement a classifier that is able to predict correct image classes: vehicles or non-vehicles.

2. Implement a vehicle detector that has to predict bounding box coordinates of vehicles.

3.Use Fast Fourier Transform (FFT) during data preprocessing.

4.Investigate whether FFT improves or reduces the accuracy of the developed solution.

In this thesis, the detection is limited to finding one vehicle per each input imag

Vehicle detection is critical in a variety of applications, including autonomous driving, traffic monitoring, and surveillance systems[8]. Deep learning-based algorithms have gained amazing success in a variety of computer vision applications in recent years. This paper provides a framework for vehicle detection based on the VGG16 convolutional neural network (CNN) architecture.Vehicle detection is critical in a variety of applications, including autonomous driving, traffic monitoring, and surveillance systems. Deep learning-based algorithms have gained amazing success in a variety of computer vision applications in recent years. This paper provides a framework for vehicle detection based on the VGG16 convolutional neural network (CNN) architecture.The VGG16 model is well-known for its excellent performance in image classification tasks. We modify the VGG16 model for vehicle detection in this paper by fine-tuning the network using a large-scale dataset selected exclusively for vehicle recognition. The dataset comprises of annotated photos of several vehicle examples in various environmental situations.On a benchmark dataset of photos acquired from diverse traffic settings, we train and test our proposed vehicle detection system. The experimental results show that the VGG16-based methodology outperforms existing vehicle detection algorithms in terms of detection accuracy. Furthermore, our method is resistant to changes in vehicle appearance, occlusion, and complex backgrounds.To summarise, this paper proposes a VGG16-based vehicle detection system that uses deep learning to reliably detect automobiles in photos. The suggested method outperforms established methods and provides a solid foundation for future developments in vehicle detecting systems.

EfficientNet models have gained popularity due to their improved performance and efficient resource utilisation. In this paper[1], we modify the EfficientNet model for vehicle detection by fine-tuning the network using a large-scale dataset specifically curated for vehicle recognition. The dataset consists of annotated photos of several vehicle examples collected under various environmental situations.To extract discriminative features from input photos, the proposed method leverages the EfficientNet model. We construct image patches at several scales using a sliding window technique, which are subsequently processed through the EfficientNet network for feature extraction. Following that, the retrieved features are passed into a classifier to determine the presence of a vehicle within a specified region.Extensive trials are carried out using a benchmark dataset that includes photos obtained from real-world traffic conditions. The results show that the EfficientNet-based technique outperforms existing state-of-the-art methods in terms of vehicle detection accuracy.Furthermore, the EfficientNet architecture's efficiency enables real-time processing of high-resolution photos, making it suited for real-world applications with strict time limitations. The computational efficiency also enables deployment on devices with limited resources, such as embedded systems and edge devices.

Chapter 3

Dataset Description and Experimental work

3.1 Data Collection

The dataset "Vehicle Detection Image Set" was collected from Uruguay for Vehicle Detection.Which contains images of vehicles and non-vehicles.

Table 3.1: Dataset Description

Classes	**Number of Samples**
Vehicle	8792
Non-vehicle	8968

Figure 3.1: Non-Vehicle image

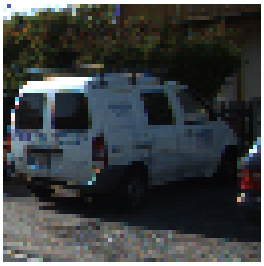


Figure 3.2: Vehicle Image

3.2 Data Preprocessing

3.2.1 Reshaping images

The change image dimension function reshapes an input array of data, which is expected to contain image data, into a new format appropriate for further processing. The input data's original shape is believed to be a list of image data, with each image stored as a flattened array. The function reshapes the data into a 4D array with the dimensions (number of images, 64, 64, 3), with each image having a width and height of 64 pixels and three RGB channels. When working with image data in deep learning models that require a specified input shape, this transformation is widely utilised.

3.2.2 Normalization

Normalisation is widely used to scale pixel values between 0 and 1, as many deep learning models operate best with inputs in this range. Assuming the pixel values were originally in the range of 0 to 255, dividing them by 255.0 reduces them to the range of 0 to 1.The code reshapes the supplied image data into a format that will most likely be required by following actions. It then normalises the picture data's pixel values by dividing them by 255.0, ensuring that the values are scaled to the 0 to 1 range often utilised in deep learning models.

3.2.3 Data Augmentation

The ImageDataGenerator class in Keras is a powerful tool for performing data augmentation during training of deep learning models on image data. Data augmentation is a technique used to artificially increase the size and diversity of the training dataset by applying various random transformations to the input images. It helps to improve the model's ability to generalize and perform well on unseen data.The ImageDataGenerator increases the diversity and variability of the training dataset by applying random transformations to the input images, such as rotation, shifting, shearing, zooming, and flipping. As a result, the deep learning model learns more robust features and patterns, leading to enhanced performance and generalisation on previously unseen test data.Overall, the ImageDataGenerator in Keras provides an easy and adaptable method for incorporating data augmentation into the training process, hence improving the effectiveness of deep learning models for image-related tasks.

3.3 Basic CNN model architecture-

A basic CNN model architecture for vehicle detection was constructed using images, with convolutional layers for feature extraction followed by fully connected layers for classification.The layers consist of-

i. Input Layer: The input layer receives image data with dimensions that match the input image size, such as 64x64 pixels.

ii. Convolutional Layers: Convolutional layers extract features from input images by applying a set of learnable filters. Each filter scans the image, convolutionally capturing local patterns and characteristics. The depth or number of output feature maps is determined by the number of filters. To induce non-linearity, these layers are generally followed by activation functions (e.g., ReLU).

iii. Pooling Layers: By lowering the spatial dimensions, pooling layers downsample the feature maps. Max pooling and average pooling are two common pooling strategies that help reduce computational complexity while capturing the most important information.

iv. Flatenning Layer: The flattening layer reshapes the output of the previous pooling layer into a one-dimensional vector. This gets the data ready for the fully connected layers.

v. Fully Connected Layers: Fully connected layers, also known as dense layers, take the flattened feature vector as input and perform classification. These layers connect every neuron from the previous layer to every neuron in the current layer, enabling high-level feature representation and classification. Activation functions (e.g., ReLU) are commonly applied to the output of each fully connected layer.

vi. Dropout Layer: The dropout layer is a regularisation technique often used to prevent overfitting in Convolutional Neural Networks (CNNs). Overfitting occurs when a model gets overly specialised to the training data and is unable to generalise well to new data.During training, the dropout layer randomly sets a fraction of input units to 0, removing those units from the network. Dropout prevents neurons from co-adapting and enables the network to learn more robust and generalised characteristics.

vii. Output Layer: Depending on the number of classes to be predicted, the output layer may include one or more neurons. Vehicle detection is divided into two categories: vehicle and non-vehicle. To provide the final class probabilities or predictions, the output layer applies a suitable activation function called sigmoid.

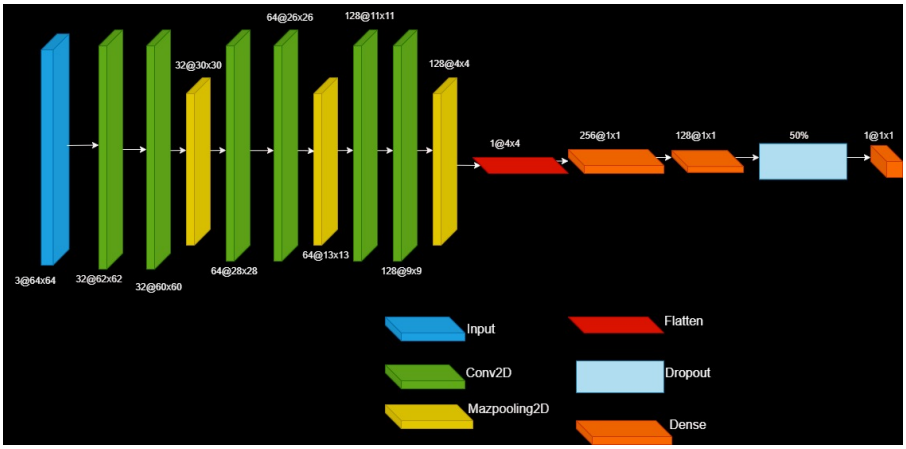


Figure 3.3: The Structure of the baseline Convolutional Neural Network model

3.3.1 Model Summary-

There are 846,577 trainable parameters in this model. For feature extraction, it has many convolutional layers followed by max pooling layers. The convolutional layers use the ReLU activation function and have varied filter sizes (32, 64, and 128).Following the convolutional layers, a flattening layer reshapes the output into a 1-dimensional vector. This is followed by three dense (completely linked) layers, each containing 256, 128, and 1 neuron. Except for the last layer, which uses the sigmoid activation function for binary classification (vehicle or non-vehicle), the dense layers use the ReLU activation function.In addition, a dropout layer with a dropout rate of 0.5 is introduced after the second dense layer to help reduce overfitting.Overall, the architecture of this model is composed of alternating convolutional and max pooling layers for feature extraction, followed by fully connected layers for classification.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 62, 62, 32)	896
conv2d_1 (Conv2D)	(None, 60, 60, 32)	9248
max_pooling2d (MaxPooling2D)	(None, 30, 30, 32)	0
conv2d_2 (Conv2D)	(None, 28, 28, 64)	18496
conv2d_3 (Conv2D)	(None, 26, 26, 64)	36928
max_pooling2d_1 (MaxPooling 2D)	(None, 13, 13, 64)	0
conv2d_4 (Conv2D)	(None, 11, 11, 128)	73856
conv2d_5 (Conv2D)	(None, 9, 9, 128)	147584
max_pooling2d_2 (MaxPooling 2D)	(None, 4, 4, 128)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 256)	524544
dense_1 (Dense)	(None, 128)	32896
dropout (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 1)	129

Figure 3.4: Model Summary of Basic CNN

3.3.2 Training and Testing

In this model the StratifiedShuffleSplit from the sklearn.model selection module is used to split a dataset into training and test sets while maintaining the class distribution.Further training and testing can be explained as-

i. Importing the required modules: The code begins by importing the StratifiedShuffleSplit and train-test-split modules from the sklearn.model selection module. These modules include functions for dividing datasets and cross-validation.

ii. Creating a Stratified Shuffle Split object: The StratifiedShuffleSplit class is instantiated with the required arguments. In this model, n-splits is set to 1, suggesting that just one split would be formed, and test-size is set to 0.15, indicating that the test set should account for 15% of the data. For reproducibility, the random-state parameter has been set to 42.

iii. Executing the split: The split.split(df, df['label']) line is responsible for splitting the dataset. The split() method requires two arguments: the input data (df) and the target variable (df['label']). It returns an iterator with indices for both the training and test sets.

iv. Iterating over the split: The for loop is used to iterate across the indices returned by split.split(). There is just one iteration in this model because n-splits is set to 1.

v. Creating the test and training datasets: The variables train-index and test-index are used within the loop to store the indices for the training and test sets, respectively. These indexes are then used by the iloc indexer to select the matching data rows from the original dataset (df). The chosen rows are allocated to the variables train-data and test-data, which represent the training and test datasets, respectively.

3.3.3 Hyper parameter tuning

The hyperparameters that were tuned are:

- Batch Size: [16,32,64], Activation Function: ['relu', 'sigmoid'], Number of Epochs: [10,50,100]

3.4 Transfer Learning Approaches

Deep learning techniques, particularly Convolutional Neural Networks (CNNs), have exhibited exceptional performance in numerous computer vision applications, including object identification and recognition, in recent years. Vehicle detection, a critical problem in intelligent transportation systems and autonomous driving, might considerably benefit from deep learning advances.In this study[9], we use the power of two prominent CNN architectures, VGG16 and EfficientNet, as base models for vehicle detection using pictures. VGG16 is known for its deep design with stacked convolutional layers, whereas EfficientNet is noted for its efficiency and scalability across multiple network sizes.We can utilise the pre-trained weights of VGG16 and EfficientNet as base models and exploit the learned feature representations from big picture datasets like ImageNet by employing them as base models. This transfer learning strategy allows us to take advantage of these models' strong feature extraction capabilities while saving significant computational resources and shortening training time.

3.4.1 VGG16 model

VGG16 is a deep convolutional neural network design that performs exceptionally well in image classification tasks. It is distinguished by a dense stack of convolutional layers that allows it to learn hierarchical characteristics from images. To extract generalizable features from a wide range of objects and scenarios, the architecture has been intensively trained on large-scale

datasets such as ImageNet.Transfer learning, a technique that exploits the acquired representations of pre-trained models for specific tasks, is used to adapt VGG16 for vehicle identification. We may benefit from the comprehensive feature extraction capabilities acquired by VGG16 during its training on big image datasets by using its pre-trained weights as a starting point.Further this model have been customized with layers such as Conv2D,MaxPooling,Dense,Flatten and Dropout Layers to achieve the best result possible.By training this customised VGG16 architecture on a huge dataset of vehicle and non-vehicle photos, we hope to construct a highly accurate and efficient vehicle detection model. This model has real-world applications such as traffic monitoring, parking lot surveillance, and sophisticated driver assistance systems.

4.2.1.1 Model Summary

The model has 2,867,745 trainable parameters that are optimised during training to increase the model's performance in vehicle detection tasks.

Here is a brief explanation of the layers:

i.The first convolutional layer uses the ReLU activation function to apply 32 filters of size (3, 3) to the input picture. It extracts the image's essential elements.

ii.Using the ReLU activation function, the second convolutional layer applies 32 filters of size (3, 3) to the output of the preceding layer. It extracts even more intricate properties.

iii.By picking the maximum value inside a 2x2 window, the max pooling layer decreases the spatial dimensions of the feature maps. This aids in downsampling and preserving the most significant characteristics.

iv.The third convolutional layer uses the ReLU activation function to apply 64 filters of size (3, 3) to the output of the preceding layer. It continues to extract higher-level features.

v.The fourth convolutional layer employs the ReLU activation function to apply 64 filters of size (3, 3) to the output of the preceding layer. It improves on the previously learnt characteristics.

vi.Another max pooling layer is applied to downsample the feature maps.

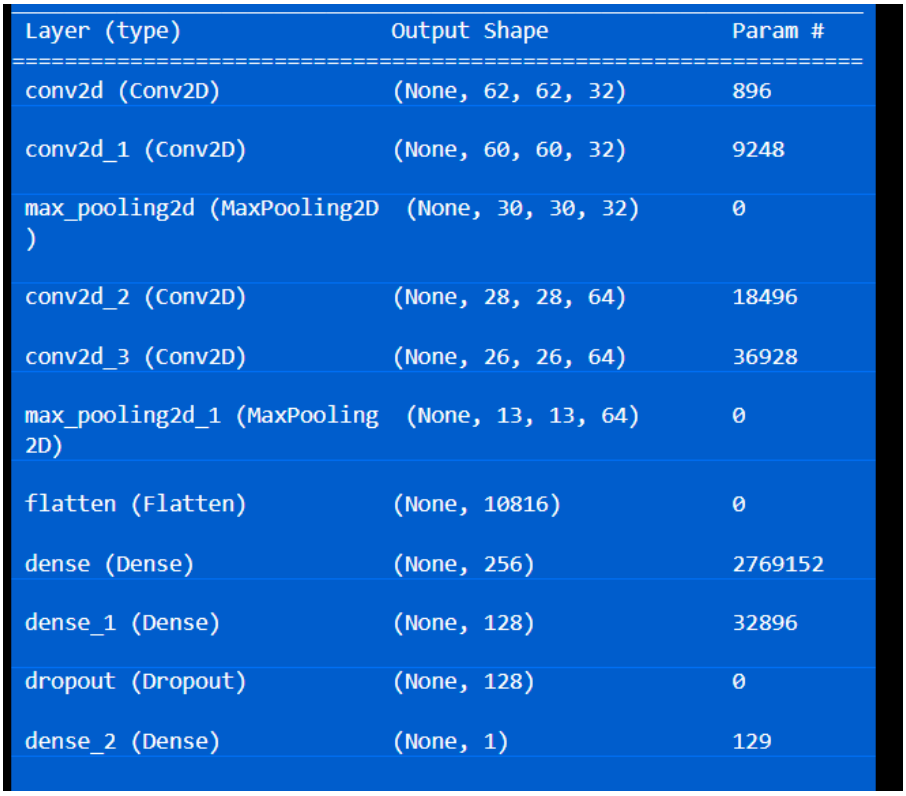
vii.The flatten layer turns the 2D feature maps into a 1D vector before passing them to the fully connected layers.

viii.The first dense layer is made up of 256 units and employs the ReLU activation function. From the flattened features, it learns complicated patterns and representations.

ix.The ReLU activation function is used in the second dense layer, which comprises 128 units. It also includes high-level representations.

x.During training, the dropout layer randomly sets 50% of the inputs to 0. It prevents overfitting by increasing regularisation and enhancing model generalisation.

xi.The final output layer contains a single unit with sigmoid activation. It returns the chance of a vehicle being in the input image, with values ranging from 0 to 1.



Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 62, 62, 32)	896
conv2d_1 (Conv2D)	(None, 60, 60, 32)	9248
max_pooling2d (MaxPooling2D)	(None, 30, 30, 32)	0
conv2d_2 (Conv2D)	(None, 28, 28, 64)	18496
conv2d_3 (Conv2D)	(None, 26, 26, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 13, 13, 64)	0
flatten (Flatten)	(None, 10816)	0
dense (Dense)	(None, 256)	2769152
dense_1 (Dense)	(None, 128)	32896
dropout (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 1)	129

Figure 3.5: Model Summary of VGG16 model

4.2.1.2 Hyper parameter tuning

The hyperparameters that were tuned are:

- Batch Size: [16,32,64], Activation Function: ['relu', 'sigmoid'], Number of Epochs: [10,20,50]

3.4.2 EffiecientNet Model

In this research, we used the EfficientNet model as the foundation and added customised layers for vehicle detection using photos. The EfficientNet model is well-known for its efficiency and efficacy in dealing with numerous computer vision problems, such as object detection. We hope to construct a robust and accurate vehicle detection system by adopting and fine-tuning this model specifically for vehicle detection.The architecture employs a compound scaling method that uniformly scales the depth, width, and resolution of the network to achieve an optimal balance between accuracy and computational efficiency. This characteristic makes EfficientNet a suitable candidate for our vehicle detection task, as it can handle large-scale datasets while minimizing computational resources.Further this model have been customized with layers such as Conv2D,MaxPooling,Dense,Flatten and Dropout Layers to achieve the best result possible.By training this customised EfficientNet architecture on a huge dataset of vehicle and non-vehicle photos, we hope to construct a highly accurate and efficient vehicle detection model.

4.2.2.1 Model Summary

The model has 844,577 trainable parameters that are optimised during training to increase the model's performance in vehicle detection tasks.

Here is a brief explanation of the layers:

i. The first convolutional layer uses the ReLU activation function to apply 32 filters of size (3, 3) to the input picture. It extracts the image's essential elements.

ii. Using the ReLU activation function, the second convolutional layer applies 32 filters of size (3, 3) to the output of the preceding layer. It extracts even more intricate properties.

iii. The max pooling layer decreases the spatial dimensions of the feature maps by taking the maximum value within a 2x2 rectangle. This aids in downsampling and keeping the most significant elements.

iv. Using the ReLU activation function, the third convolutional layer applies 64 filters of size (3, 3) to the output of the preceding layer. It keeps extracting higher-level features.

v. The fourth convolutional layer employs the ReLU activation function to apply 64 filters of size (3, 3) to the output of the preceding layer. It improves on the previously learnt characteristics.

vi. To downsample the feature maps, another max pooling layer is used.

vii. The fifth convolutional layer employs the ReLU activation function to apply 128 filters of size (3, 3) to the output of the previous layer. It retrieves more complicated and abstract characteristics.

viii. The sixth convolutional layer employs the ReLU activation function to apply 128 filters of size (3, 3) to the output of the previous layer. It improves on the previously learnt characteristics.

ix. Another max pooling layer is applied to downsample the feature maps.

x. The flatten layer turns the 2D feature maps into a 1D vector before passing them to the fully connected layers.

xi. The first dense layer is made up of 256 units and employs the ReLU activation function. From the flattened features, it learns complicated patterns and representations.

xii. The ReLU activation function is used in the second dense layer, which comprises 128 units. It also includes high-level representations.

xiii. During training, the dropout layer randomly sets 50% of the inputs to 0. It prevents overfitting by increasing regularisation and enhancing model generalisation.

xiv. The final output layer contains a single unit with sigmoid activation. It returns the chance of a vehicle being in the input image, with values ranging from 0 to 1.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 62, 62, 32)	896
conv2d_1 (Conv2D)	(None, 60, 60, 32)	9248
max_pooling2d (MaxPooling2D)	(None, 30, 30, 32)	0
conv2d_2 (Conv2D)	(None, 28, 28, 64)	18496
conv2d_3 (Conv2D)	(None, 26, 26, 64)	36928
max_pooling2d_1 (MaxPooling 2D)	(None, 13, 13, 64)	0
conv2d_4 (Conv2D)	(None, 11, 11, 128)	73856
conv2d_5 (Conv2D)	(None, 9, 9, 128)	147584
max_pooling2d_2 (MaxPooling 2D)	(None, 4, 4, 128)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 256)	524544
dense_1 (Dense)	(None, 128)	32896
dropout (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 1)	129

Figure 3.6: Model Summary of EfficientNet model

4.2.2.2 Hyper parameter tuning

The hyperparameters that were tuned are:

- Batch Size: [16,32,64], Activation Function: ['relu', 'sigmoid'], Number of Epochs: [10,20,30]

Chapter 4

Results and Discussion

4.1 Performance Analysis

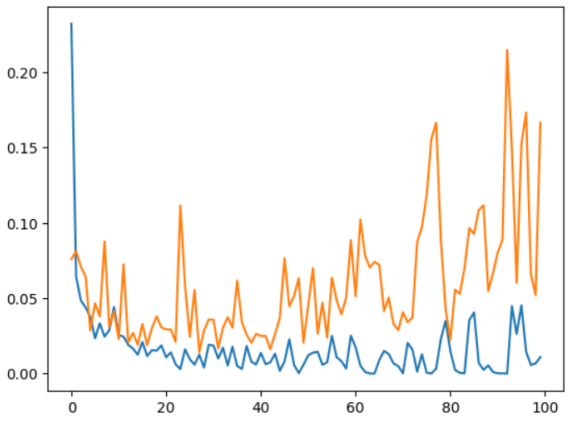


Figure 4.1: Trainging & Validation loss of basic CNN model with batch size:[32],activation function:[relu,sigmoid'] and number of epochs:[100]

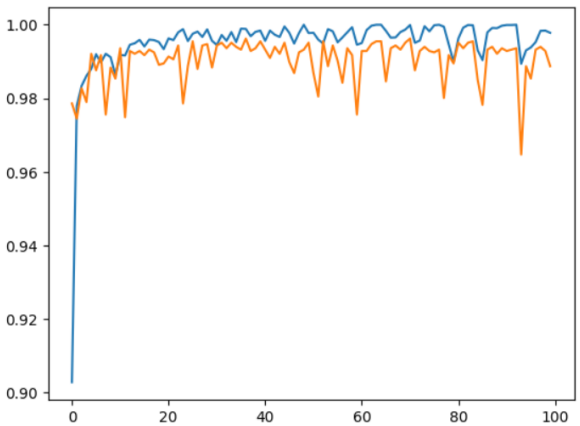


Figure 4.2: Trainging & Validation Accuracy of basic CNN model with batch size:[32],activation function:[relu,sigmoid'] and number of epochs:[100]

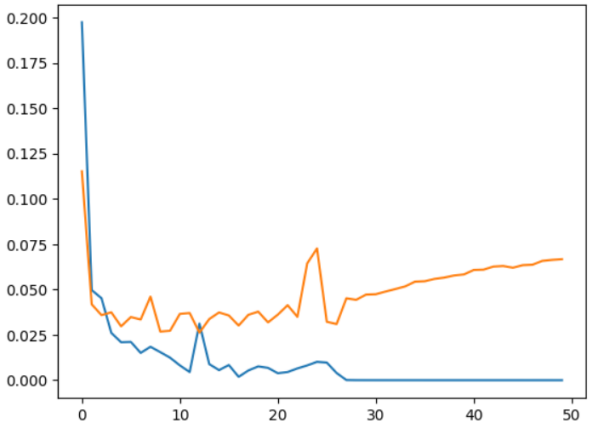


Figure 4.3: Training & Validation loss of VGG16 model with batch size:[64],activation function:[relu,sigmoid'] and number of epochs:[50]

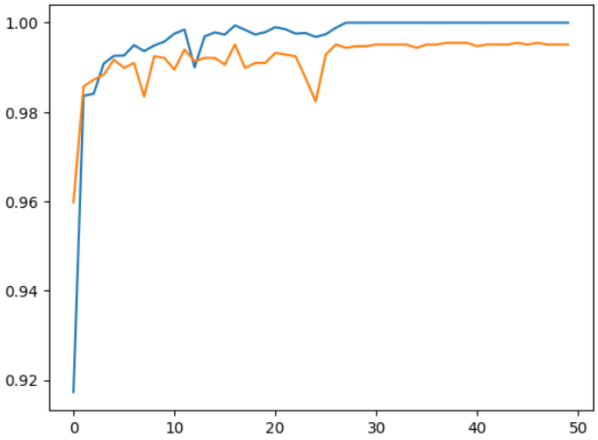


Figure 4.4: Training & Validation Accuracy of VGG16 model with batch size:[64],activation function:[relu,sigmoid'] and number of epochs:[50]

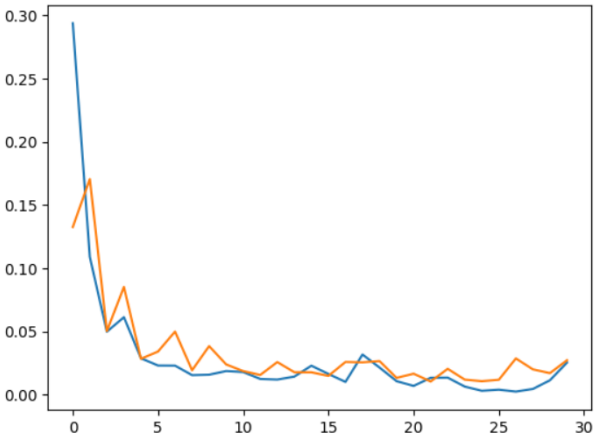


Figure 4.5: Training & Validation loss of EfficienNet model with batch size:[128],activation function:[relu,sigmoid'] and number of epochs:[30]

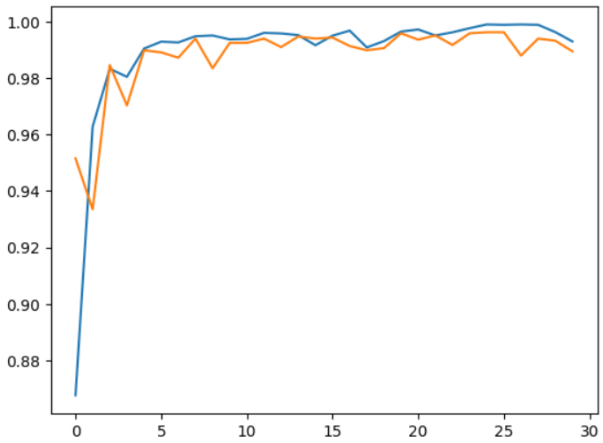


Figure 4.6: Training & Validation Accuracy of EfficientNet model with batch size:[128],activation function:[relu,sigmoid'] and number of epochs:[30]

4.2 Output Analysis

The CNN model was trained and evaluated using the validation set.The accuracy achieved by this model on the test set is 98.87%. I tested the predicted Vehicles and Non-vehicles in the dataset shown in fig5.7.

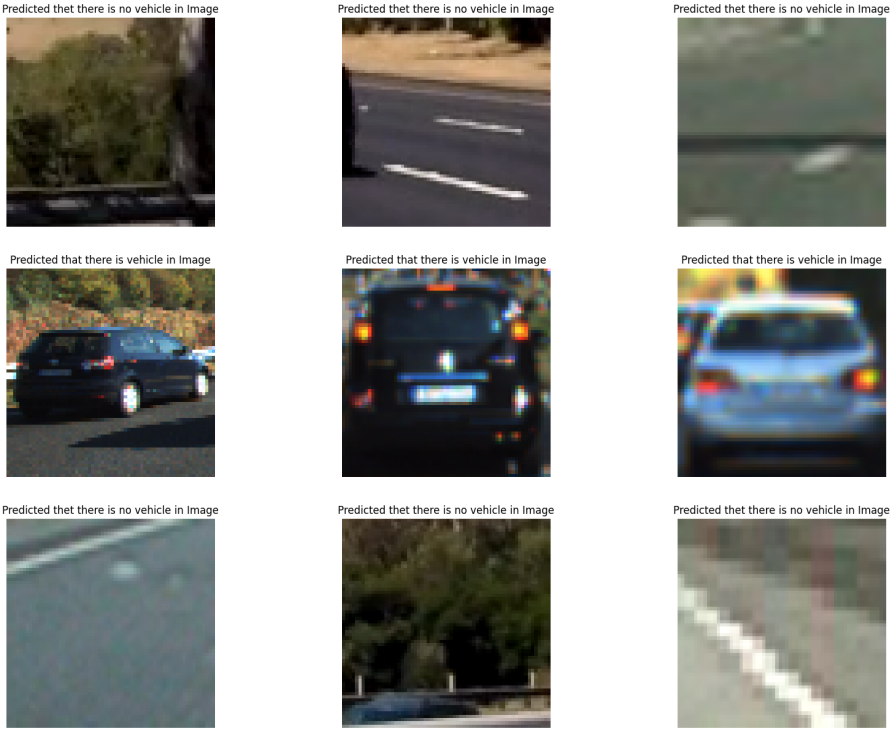


Figure 4.7: Predicted Vehicles & Non-vehicles by Basic CNN

The VGG16 model was trained and evaluated using the validation set.The accuracy achieved by this model on the test set is 99.51%. I tested the predicted Vehicles and Non-vehicles in the dataset shown in fig5.8.

Figure 4.8: Predicted Vehicles & Non-vehicles by VGG16

The EfficientNet model was trained and evaluated using the validation set.The accuracy achieved by this model on the test set is 98.95%. I tested the predicted Vehicles and Non-vehicles in the dataset shown in fig4.9.

Figure 4.9: Predicted Vehicles & Non-vehicles by EfficientNet

4.3 Discussion

As we can see that after conducting experiments on the models,the VGG16 model gives the highest detection accuracy amongst the three models.

Chapter 5

Conclusion

5.1 Concluding Remarks

In this study, we compared the performance of three different models for vehicle detection: Basic CNN, VGG16, and EfficientNet. After conducting trials and analysing the findings, we discovered that the VGG16 model attained the highest accuracy of the three models.

The VGG16 model outperformed the others in detecting vehicles from images because to its deeper architecture and more complicated feature extraction capabilities. It learned and represented sophisticated patterns and attributes connected with cars well, yielding accurate predictions.The basic CNN model, despite its simpler design, fared reasonably well in vehicle detection. It successfully learned and collected essential information from the photos, but its performance was slightly lower than the VGG16 model.The EfficientNet model also has given a satisfying output but less acuuracy in comparison to the other two models.

Based on these results, we can conclude that for the given vehicle detection task, the VGG16 model is the most suitable and effective choice in terms of accuracy.

5.2 Future Work

The future work could include examine the possibility of fine-tuning the models on a bigger and more precise vehicle identification dataset in order to improve their performance on the target task.We can also extend the project by precisely establishing the spatial location of observed vehicles within the image using approaches such as bounding box regression or region-based convolutional neural networks (R-CNN).Furthermore, the thesis concentrated on detecting a single vehicle. As a result, the existing CNN might be upgraded in the future to detect numerous vehicles at once.

Bibliography

[1] Azimjonov, J. and Özmen, A. (2021). A real-time vehicle detection and a novel vehicle tracking systems for estimating and monitoring traffic flow on highways. *Advanced Engineering Informatics*, 50:101393.

[2] Cepni, S., Atik, M. E., and Duran, Z. (2020). Vehicle detection using different deep learning algorithms from image sequence. *Baltic Journal of Modern Computing*, 8(2):347–358.

[3] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[4] Husain, A. A., Maity, T., and Yadav, R. K. (2020). Vehicle detection in intelligent transport system under a hazy environment: a survey. *IET Image Processing*, 14(1):1–10.

[5] Plemakova, V. (2018). Vehicle detection based on convolutional neural networks. *University of Tartu*.

[6] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[7] Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR.

[8] Tang, T., Zhou, S., Deng, Z., Lei, L., and Zou, H. (2017). Arbitrary-oriented vehicle detection in aerial imagery with single convolutional neural networks. *Remote Sensing*, 9(11):1170.

[9] Wang, H., Yu, Y., Cai, Y., Chen, X., Chen, L., and Liu, Q. (2019). A comparative study of state-of-the-art deep learning algorithms for vehicle detection. *IEEE Intelligent Transportation Systems Magazine*, 11(2):82–95.

[10] Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710.