TrafficNN: CNN Based Road Traffic Conditions Classification

BY

Faisal Al Mamun
ID: 171-15-8628

Monjurur Kader Shipu
ID: 171-15-8644

AND

Shamim Hossen Razu
ID: 161-15-7367

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Nishat Sultana
Lecturer
Department of CSE
Daffodil International University

Co-Supervised By

Mr. Md. Aynul Hasan Nahid
Lecturer
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

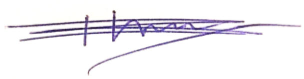
DHAKA, BANGLADESH

MAY 2021

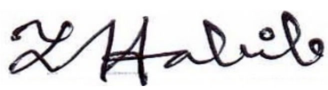
APPROVAL

This Project titled "**TrafficNN: CNN Based Road Traffic Conditions Classification**", submitted by Faisal Al Mamun, Monjurur Kader Shipu and Shamim Hossen Razu to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 01.06.2021

BOARD OF EXAMINERS



Dr. Touhid Bhuiyan **Chairman**
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University



Md. Tarek Habib **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Saiful Islam **Internal Examiner**
Senior Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Md Arshad Ali **External Examiner**
Associate Professor
Department of Computer Science and Engineering
Hajee Mohammad Danesh Science and Technology University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Nishat Sultana, Lecturer, Department of CSE,** Daffodil International University and co-supervision of **Mr. Md. Aynul Hasan Nahid, Lecturer, Department of CSE**, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

Nishat Sultana
Lecturer
Department of CSE
Daffodil International University

Co-Supervised by:

Mr. Md. Aynul Hasan Nahid
Lecturer
Department of CSE
Daffodil International University

Submitted by:

Faisal Al Mamun
ID: 171-15-8628
Department of CSE
Daffodil International University

Monjurur Kader Shipu
ID: 171-15-8644
Department of CSE
Daffodil International University

Shamim Hossen Razu
ID: 161-15-7367
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes it possible to complete the final year project/internship successfully.

We are really grateful and wish our profound indebtedness to **Nishat Sultana**, **Lecturer**, Department of CSE, Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "Deep Learning" to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to Dr. Touhid Bhuiyan, Professor and Head**,** Department of CSE, for his kind help to finish our project and also to other faculty members and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

Traffic congestion can affect different socio-economic aspects of a country. In order to perform better than traditional human effort in traffic control, an automated system must be developed. For reducing the traffic congestion, traffic conditions classification can be considered as the first way to monitor the traffic control system. So, detection of traffic conditions is crucial for building a smart traffic control system to prevent traffic jam escalation. Deep neural network is the area of machine learning that is being widely used for interpreting and analyzing visual data. This branch has an extensive use over image classification. So, modern deep learning approaches can help in detection of road traffic conditions especially Convolutional Neural Network (CNN). We suggest a novel model in this paper that can classify road traffic conditions using CNN. Our proposed model 'TrafficNN' classifies five different road traffic conditions with an accuracy of 82%. To train and test our model, we use our own traffic conditions images dataset. To verify the efficiency of our model, we compare it with several pre-trained models like- VGG16, ResNet50, InceptionV3 and DenseNet121. The comparison result proves the significance of our model to extend its successful application for developing an automated traffic controlling system in the near future.

TABLE OF CONTENTS

CONTENTS	PAGE

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
Introduction

1.1 Introduction

Traffic congestion management is an overhead for almost every big city in the world. Especially countries like Bangladesh where some roads are still under development and other roads are narrow compared to large volumes of vehicles. There are several factors and these have some adverse effects on different socio-economic aspects. Research shows that for 240 working days the cost of traffic congestion is 4495.6544 million USD in Dhaka the capital of Bangladesh [1]. Traffic congestion in Dhaka consumes approximately 5 million working hours per day, according to a report by the Accident Research Institute of BUET [2]. Between January 2009 and April 2019, at least 25,526 people have died and nearly 20,000 people have been seriously injured in road accidents in Bangladesh [3]. According to Bangladesh's Transport Ministry, vehicle accidents have climbed by more than 51% in 2019 and the death rate has climbed by more than 17% over 2018 [4]. So, we can see that traffic congestion has a significant effect on the various socio-economic aspects of a country.

Nowadays, traffic congestion issues impact the transport system in cities and create severe problems in many countries. The optimization of heavy traffic congestion, particularly with multiple junction nodes, is a major issue that needs to be addressed [5]. It is difficult to reduce road traffic congestion with existing traffic control systems, as it is operating manually most of the time. However, it is important to have a smart traffic management system in place to maintain a reliable transportation system.

Many research has already been done to reduce traffic congestion and ensure a reliable smart traffic control system. Many of them proposed to use sensors and IoT based devices in every vehicle to know road traffic conditions, which is pretty expensive and complex to implement. Artificial Intelligence (AI), Machine Learning (ML) and Deep Learning (DL) have been used extensively in this field of study in recent years. But nobody tries to classify traffic conditions on the basis of real-life scenarios.

We are using Convolutional Neural Network (CNN) in this study, which is a part of Deep Learning (DL). We proposed a CNN based model named as 'TrafficNN' to dedicatedly classify road traffic conditions which can be used to automatically control road traffic congestion in the near future. In order to develop an efficient, low cost and real time smart traffic management system, we propose the use of CNN to resolve several defects and enhance traffic management based on the classification of road traffic conditions.

1.2 Motivation

Traffic Jam is a very common phenomenon in our life, especially for those who live in mega cities like- Dhaka. Undoubtedly the traffic jam problem hinders our development effort from individual and national perspective. Existing inefficient traffic signal control produces plenty of issues, including excessive delays and energy waste. Because of this we lost a lot of time in traffic jams every day. Sometimes getting late because of traffic can take a life away from this world. We all know the value of time, how fast the world is getting updated, how fast people are getting digitized. Without giving the value of time no nation can be developed, so our main intention is to give people of our country more time to express themselves, work more which will undoubtedly cause a great development for any country or for any nation undoubtedly. So indeed, in all perspectives it's very important to build such a system on roads that can save our time, make our life easy to reach anywhere we need. To solve those problems, we want to make a CNN based model that can help to build smart traffic control systems.

1.3 Rationale of the Study

As we want to solve the problem of traffic jams, we try to find what other peoples do to improve this existing traffic control system. Thousands of works on traffic control management have undoubtedly been completed utilizing various methodologies. And we find that nobody tries to build a less expensive and easily applicable system on the basis of real-life road traffic conditions. We know in recent times Artificial Intelligence (AI), Machine Learning (ML) and Deep Learning (DL) show significant results in solving many problems around many fields. In our study, we work with Convolutional Neural Network (CNN) which is a part of deep learning to improve these existing situations.

1.4 Research Questions

When we first think about solving the problem of the existing traffic system, we can't understand what our first task is. And how we will solve this problem. Completing this project was quite difficult for us. At that time, we thought of some confusing questions.

- What algorithm should we use?
- Which programming language is perfect to implement our problems?
- What amount of data do we need?
- Is it possible to get good accuracy from our model?
- Is there enough time for completing our study?

1.5 Expected Outcome

We began our initiative with the goal of addressing one of the most pressing concerns in both developing and wealthy countries: traffic congestion. It is impossible to create new roads or infrastructure to match the traffic over the night. Our goal is to figure out how to reduce traffic congestion and get people to their destinations as quickly as possible. After deciding our field of study as machine learning and deep learning specifically as CNN, we expected to use CNN in classification of road traffic conditions. We want to build a CNN based dedicated model to classify real life traffic images which can be used as a part of smart traffic controlling. There is some specific target of our project which is given below.

- Road traffic conditions classification
- Can be used to build smart and efficient traffic management system
- Can give special corridor for emergency vehicles
- Can detect and inform road accidents in real time through AI

1.6 Report Layout

Chapter 1: Introduction

Introduction, motivation, rationale of the study, research questions and the expected outcome of our project will be discussed in this chapter. To give a gentle introduction to our project, here we discuss those topics briefly.

Chapter 2: Background Study

In chapter 2, we will discuss what has already been done by the previous worker/researchers in this area of traffic management, what is the problem of their approaches and why our method is better than their methods. We will also define some scope of the problem and highlight some challenges that we face during this project.

Chapter 3: Research Methodology

In chapter 3, our proposed model will be discussed theoretically. But before that we will mention some important procedures like data collection, data augmentation, data preprocessing, etc. We will also provide some theoretical information about Convolutional Neural Network (CNN), so that everybody can understand this concept. We will address some minimum requirements for this analysis at the end of chapter three.

Chapter 4: Experimental Results and Discussion

Chapter 4 discusses the performance comparison of our 'TrafficNN' models with some popular pre-trained models such as VGG16, ResNet50, InceptionV3 and DenseNet121. After that we discuss complete result analysis of all models including our model.

Chapter 5: Conclusion and Future Works

In Chapter 5, we will make a conclusion by summarizing our study. Some future works regarding our proposed model will be discussed to show its potentiality in the field of traffic management system.

CHAPTER 2

Background Study

2.1 Introduction

In this section, we'll explain why we're researching machine learning to automate the process of identifying different types of traffic congestion on a road using computer vision and artificial intelligence, specifically Deep Learning approaches. Before moving towards our works, here we will discuss related works in the field of smart traffic control systems. Here we will discuss others' research papers, their method, their strategies and problems about their works. We will make a comparative analysis of what we learn from other projects and what problems they have. Then it will be clear why we are doing this project.

2.2 Related Works

Here we will discuss related works in the field of smart traffic control systems. In 2020, S. R. Mugunthan proposed a model based on Li-Fi technology for communication between vehicles and traffic signals. In his model every car has to be equipped with two Li-Fi devices for vehicle to vehicle interaction and also the traffic signals have to be equipped with Li-Fi devices in order to generate proper signals. But the main problem here is Li-Fi technology can be affected by sunlight and other light on road [6]. Priyadharshini and Manikandan presented an automatic traffic control system in 2019 where PIR sensor is used to calculate vehicular density and Arduino is used to count the number of vehicles. And then that information is used to generate traffic lights [7]. In 2015, Sundar and et al., proposed an intelligent traffic control system where a RFID tag is attached with each vehicle. This RFID tag communicates with traffic controllers for counting the number of vehicles, determining traffic congestion, ambulance clearance and sending messages about stolen vehicles to police using the ZigBee module [8]. In 2018, Tarun Kumar and Dharmender Singh Kushwaha proposed a model for creating green corridors for emergency vehicles by placing RFID modules in emergency vehicles. They create green corridors by finding least, moderate and heavy congestion using OpenCV based on surveillance camera images [9]. Biru Rajak and Dharmender Singh

Kushwaha used WiMax technology along with RFID to create routes for emergency vehicles [10]. In 2017, Ramkumar Eswaraprasad and Linesh Raja proposed an IoT based Traffic Management where they use Hybrid Artificial Neural Network with Hidden Markov Model (HANN-HMM) to make better decisions about the traffic management [11]. All of these sensors and IoT based models are expensive and complex to implement. It is also unrealistic to put sensors and devices in every vehicle on the road.

In 2019, Lam et al. presented a traffic congestion detection method by counting the number of vehicles based on online images. To assess two degrees of congestion, namely normal and congested, the image correlation coefficient has been used along with a threshold for the number of vehicles detected [12]. In 2018, Jiyong Chung and Keemin Sohn suggested a supervised learning method to calculate traffic density where the number of vehicles is collectively counted [13]. In 2018, Teresa Pamuła presented a model for road traffic condition classification based on video traffic data where four levels of traffic conditions can be determined based on number of vehicles [14].

In 2020, Vinothkanna proposed to use deep learning in order to detect road conditions to plan optimal paths for self-driving cars [15]. In 2018, Wang et al. presented a Convolutional neural Network (CNN) model named as 'TrafficNet' for the monitoring of unnecessary traffic congestion based on existing surveillance systems. This model classifies traffic conditions as congestion and non-congestion with an accuracy of 90% [16]. Manchanda and et al., presented a Hybrid-CNN model adding Convolutional neural Network (CNN) and Support vector machine (SVM) together. They classify traffic congestion as dense and sparse traffic and also add accident and fire class to classify risks for road safety [17]. In 2018, Kurniawan and et al., presented a CNN based traffic congestion detection model. This model classifies CCTV footage images as jammed and not-jammed. [18]. Cui and et al., presented a CNN model for recognizing highway traffic congestion. They use highway surveillance camera images to classify as congestion and non-congestion [19]. In 2018, Chakraborty and et al., proposed a Deep Convolution Neural Networks based model. They classify CCTV images as two classes named as-congested and non-congested using YOLO, SVM and Deep-CNN [20]. Bautista and et al., proposed a model for vehicle detection in low resolution traffic videos using

Convolutional neural network. They classify vehicles in seven different classes [21]. For explaining non-recurring traffic congestion, Sun and et al., presented a Deep Neural Network. They classify non-recurring traffic congestion as three classes named as- accident, sports game and adverse weather [22]. Willis and et al., prepared a Deep Convolutional Network for traffic congestion classification and they also classify surveillance camera image as uncongested and congested [23]. A Deep Neural Networks for traffic flow prediction based on the traffic performance index was proposed by Yi et al. in 2017 as an indicator of traffic flow conditions [24].

2.3 Comparative Analysis and Summary

After analyzing those previous work, we observed that all sensors and IoT based models are expensive and complex to implement. They want every vehicle on the road to use an IoT sensor in it which is quite unrealistic especially in a country like- Bangladesh. And all of those models that count the number of vehicles based on surveillance camera images are not classifying traffic conditions properly. They just classify traffic conditions as congestion and non-congestion which is not a proper description of road traffic conditions. Most of the Convolutional Neural Network (CNN) based deep learning model also classifies traffic images as congestion and non-congestion categories. So, we can say no previous researcher tries to classify traffic conditions based on real life traffic situations. We proposed a CNN based model that can classify traffic conditions images based on real life applicable scenarios. So that it can be used for a smart traffic management system in near future.

2.4 Scope of the Problem

As traffic congestion has an effect on the different socio-economic aspects of a country. So, we try to solve this problem using new technology. Our study finds that no previous research has been done to classify road traffic conditions on the basis of real life traffic situations. And their methods have several problems to implement in real life scenarios. So, we try to build a new dataset to build a new model to classify road traffic images.

2.5 Challenges

To finish our project, we had to overcome a number of obstacles. Our first challenge was to select a proper algorithm or method to make a model that can help in traffic conditions images classification. We finally found CNN that can help us in doing this job. Then our next challenge was collection of data. As we know small data cannot make a good CNN model, so we try to find as much data as possible. But sadly, we found no dataset regarding this which can full-fill our requirement of traffic conditions situations. Then we decide to collect data manually from the internet and augment them to make an acceptable amount of data to train a CNN model. Our next challenge was training our model on a good computational machine. As we didn't have any high-end GPU to train our model, we tried to find a solution for that because it takes several hours to train a model on local CPU. Then finally we found a solution named 'Google Colab', it gives us a free cloud GPU to train our model in the shortest time. So, we can say that our task was difficult, but we successfully completed it.

CHAPTER 3

Research Methodology

3.1 Introduction

In this chapter 3, we are going to elaborate the workflow of our new approach to classify road traffic conditions images. For image classification, we use convolutional neural networks in this study, because it is the most powerful and proven method of deep learning to classify images. We are going to use our own traffic images dataset to train our proposed model. Here we will discuss our proposed CNN based 'TrafficNN' model, as well as about CNN for better understanding of our model.

3.2 Research Subject and Instrumentation

A simple definition of our study field is given by the research subject. We introduce and design our model in this section, gather perfect data, plan and train our model, discuss results, and then apply our model to function. To complete our task we used Windows 10 platform, Python programming language with many packages like Keras, TensorFlow, numpy, pandas, skit-learn, seaborn, cv2, matplotlib etc. Google Colab is used to train and test our model. It provides a free cloud GPU which is used by us to train our model efficiently within a short amount of time. We prefer Python because of its simple syntax and vast amount of use in complex algorithms readability for machine learning applications.

3.3 Data Collection Procedure

To train our suggested model in this study, we used a new dataset. We construct the new dataset by manually collecting images and publicly available information from the internet. This dataset has five classes of road traffic conditions images such as low traffic, medium traffic, high traffic, emergency traffic and traffic accident. This dataset has a total of 2500 images, among them 2000 images are used for training and 500 images are used for testing purposes. Each class has 500 images and we considered 80% data for training and remaining 20% data for testing purpose. So, for each class we have 100 images for testing and the remaining 400 images are for training.

Figure 3.1: A small part of dataset

3.4 Data Augmentation

Data augmentation is commonly used when training data is limited. We know training a CNN on a small dataset makes it vulnerable to overfitting, so we use data augmentation to overcome that. We gathered 100 images of each class as raw data. The dataset was artificially stretched to discourage overfitting. After collecting 500 images for five classes (100 images per class), we augmented all the main data using 4 different augmentation technique, these techniques are given below:

- Rotation Range (±30 degree)
- Horizontal Flip

- Height Shift Range (15%)

- Width Shift Range (15%)

We have done those augmentation using TensorFlow 2.0 and Keras API via ImageDataGenerator class. So, after data augmentation, we now have a total of 2500 images to use in our proposed model, whereas before we only had 500.

3.5 Data Preprocessing

All data images must have a fixed pixel in our proposed model. So, we resized all data with a fixed resolution of 256 x 256 pixels. To normalize our data, we reduce pixel values by dividing it with 255 and get the pixel values from 0.0 to 1.0. When passing images to our proposed model, we use RGB color channels which helps CNN to easily identify features.

3.6 Convolutional Neural Network (CNN)

A Convolutional Neural Network, also known as CNN, is a Deep Learning algorithm which takes images as input, learns various important attributes from an image and is able to distinguish them between different classes [25]. A standard CNN consists of single or multiple blocks of convolutional and pooling layer, then one or more fully connected layers after that [26].

A digital image has a set of pixels arranged in a grid-like manner that contains pixel values to show how bright each pixel is and what color it contains [27]. CNN takes this grid-like topology as input and processes it through a convolutional layer, then the pooling layer extracts specific features from it and then finally the output of this layer works as input of a fully connected layer which finally classifies the images through various trained neurons. These layers are organized in such a way that they define simple patterns (lines, curves, etc.) first and then more complicated patterns (faces, objects, etc.). One can make sight to computers by using a CNN. CNN is outstanding in machine learning problems. Specifically, the applications concerned with image data, such as the largest image classification data set (ImageNet), computer vision, and in natural language processing (NLP) and the outcomes obtained were very impressive [28].

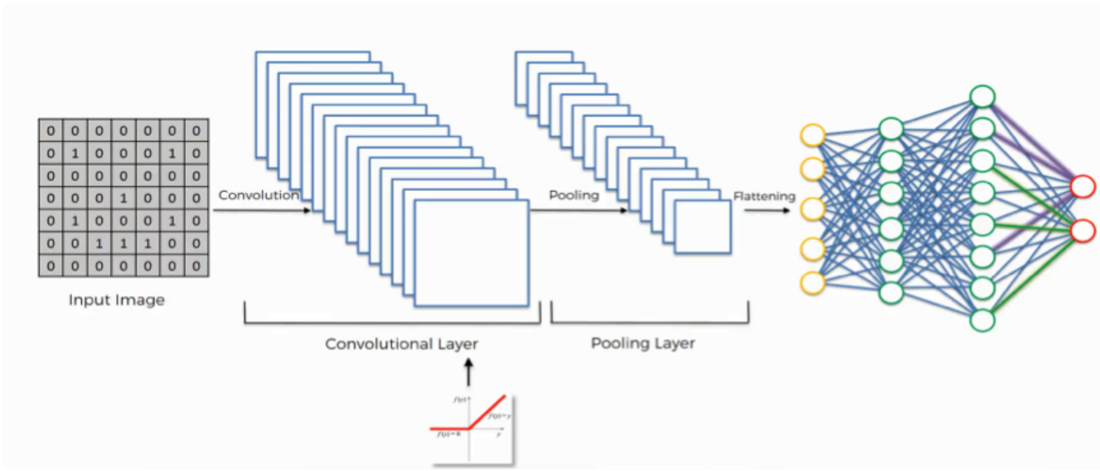


Figure 3.2: Convolutional Neural Network Architecture

3.6.1 Convolutional Layer

The convolution layer is CNN's key building block that does most of the computational heavy lifting. The main goal of this layer is to extract high level features from the input dataset. These layers are composed of filters (kernel) and feature maps. A filter also known as Kernel, is a small matrix used for feature extraction. The feature map is created after applying a filter to the previous layer [29].

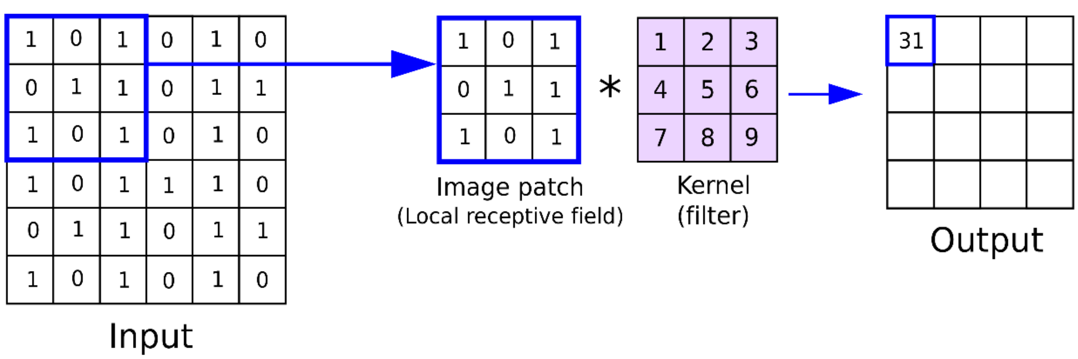


Figure 3.3: Feature Extraction using Kernel

3.6.2 Pooling Layer

To build a new collection of the same number of pooled feature maps, the pooling layer operates separately on each feature map. This means that the pooling layer would often reduce the size of each feature map by a factor of 2, such as halving each dimension, reducing the number of pixels or values to one quarter of the size of each feature map [30].

In the pooling operation, two common functions used are:

- Max Pooling: Calculate the maximum value the pooling window holds.
- Average Pooling: Calculate the average value the pooling window holds.

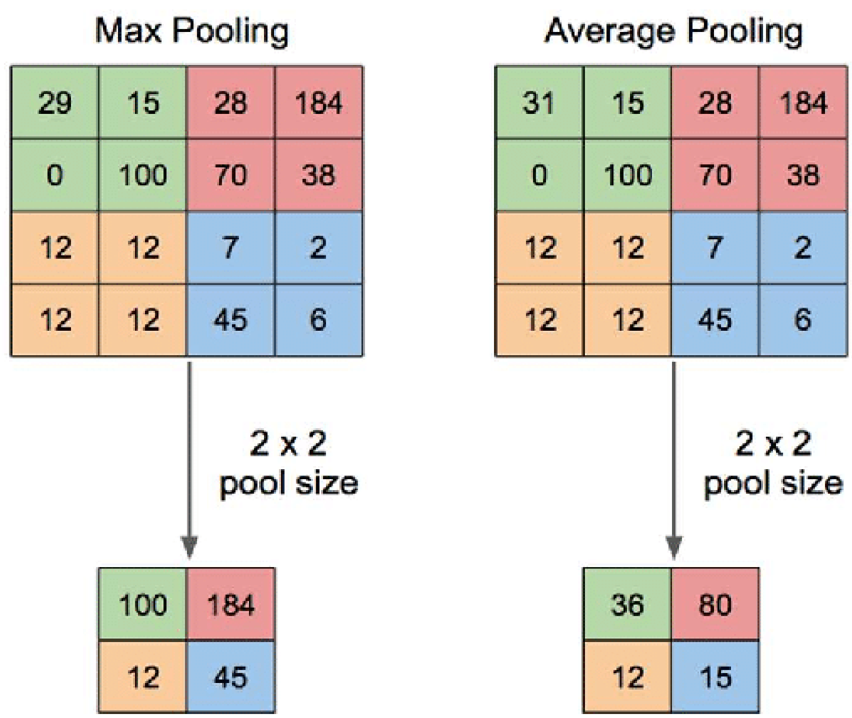


Figure 3.4: Pooling Layer

3.6.3 Flatten Layer

Flatten Layers convert the image into a feature vector. To create a single long feature vector, we flatten the output of the convolutional layers. In order to add it to the next layer, flattening alters the information into a one-dimensional array. This means that we place all the pixel data in one row and align it with the fully connected layer [31].

3.6.4 Fully Connected Layer

Fully connected layers are the final layers in the CNN structure, which could be one or more layers and positioned after a series of convolution and pooling layers. It is a normal flat feed-forward neural network layer. In this layer, neurons have complete contact with all neurons in the prior and subsequent layers. For the output probabilities of class predictions, these layers may have a non-linear activation function or a softmax activation.

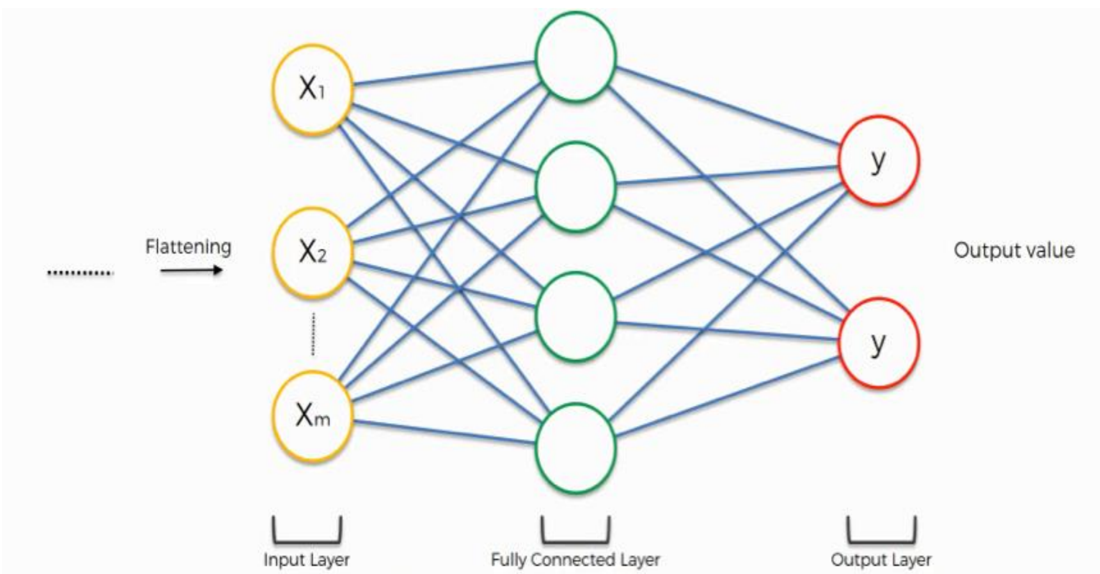


Figure 3.5: Fully Connected Layer

3.6.5 Padding

Padding is a term which refers to the number of pixels that are added to the image as it is processed by the CNN filter. If we want the output to be the same size as the input then we use padding as “same” and if we want no padding then we use padding as “valid”.

3.6.6 Rectified Linear Units (ReLU)

In deep learning models, the Rectified Linear Unit (ReLU) is the most widely used activation function. It acts as an activation function which returns 0 for any negative value, but returns the same value for any positive value. Therefore, it may be written as f(x) = max (0, x) [32]. It is a non-linear activation function used to increase non-linearity in convolutional neural networks.

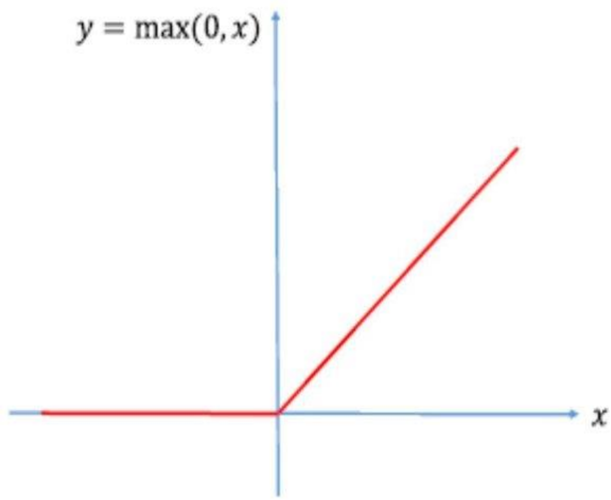


Figure 3.6: ReLU Activation Function

3.6.7 Batch Normalization

We know it is important to normalize data before passing it to a model. We generally do this operation on data, so it only works at the very first layer of the neural network. If the data becomes not normalized at some later layer then that layer has to deal with un-normalized data which is not as good as normalized data. So, to make the data at every layer normalized we use batch normalization. It has the impact of stabilizing the learning process and significantly decreasing the number of training epochs required. It acts as a regularization technique which can help to reduce overfitting.

3.6.8 Dropout

CNNs have a habit of overfitting. Dropout is a technique of regularization which can assist to minimize overfitting. During training, randomly selected neurons are ignored where dropout technique is used. That means their participation in the activation of downstream neurons on the forward pass is temporarily omitted and any weight adjustments are not added to the neuron on the backward pass [33].

3.6.9 Softmax

Softmax is an activation function that is used for multicategory classification in the output layer of neural network models. This function produces a vector describing the probability distributions of a list of possible outcomes [34]. In a multi-class problem, it assigns decimal probabilities to each class. Such decimal probabilities have to sum up to 1.0. This additional constraint allows training to converge more faster than it otherwise would.

3.7 Proposed 'TrafficNN' Model

We'll now talk about our recommended model for reaching the desired result. A simplified picture of our proposed 'TrafficNN' model is shown in figure 3.2. We take inspiration from the VGG network where they do multiple convolution layers before they do pooling. We have followed the pattern of increasing the number of feature maps at each subsequent convolutional layer. Every convolution layer in our model has the same padding because without the same padding the image is going to shrink after each convolution which will eventually make the image too small to do this many convolutions. And as usual we use the ReLU activation function with convolution layer and dense layer.

We actually have 6 groups of convolution and pooling. And each of this group has two convolutions before max pooling. For every convolutional layer, we have a constant size filter of 3x3. We have increased the number of feature maps at each subsequent group of convolution and pooling layers. After every convolution layer, we added a batch normalization layer. Finally, before each fully connected (FC) dense layer, we added a dropout layer for regularization. We use softmax as an activation function in the final dense layer to get the output as probability of each class.

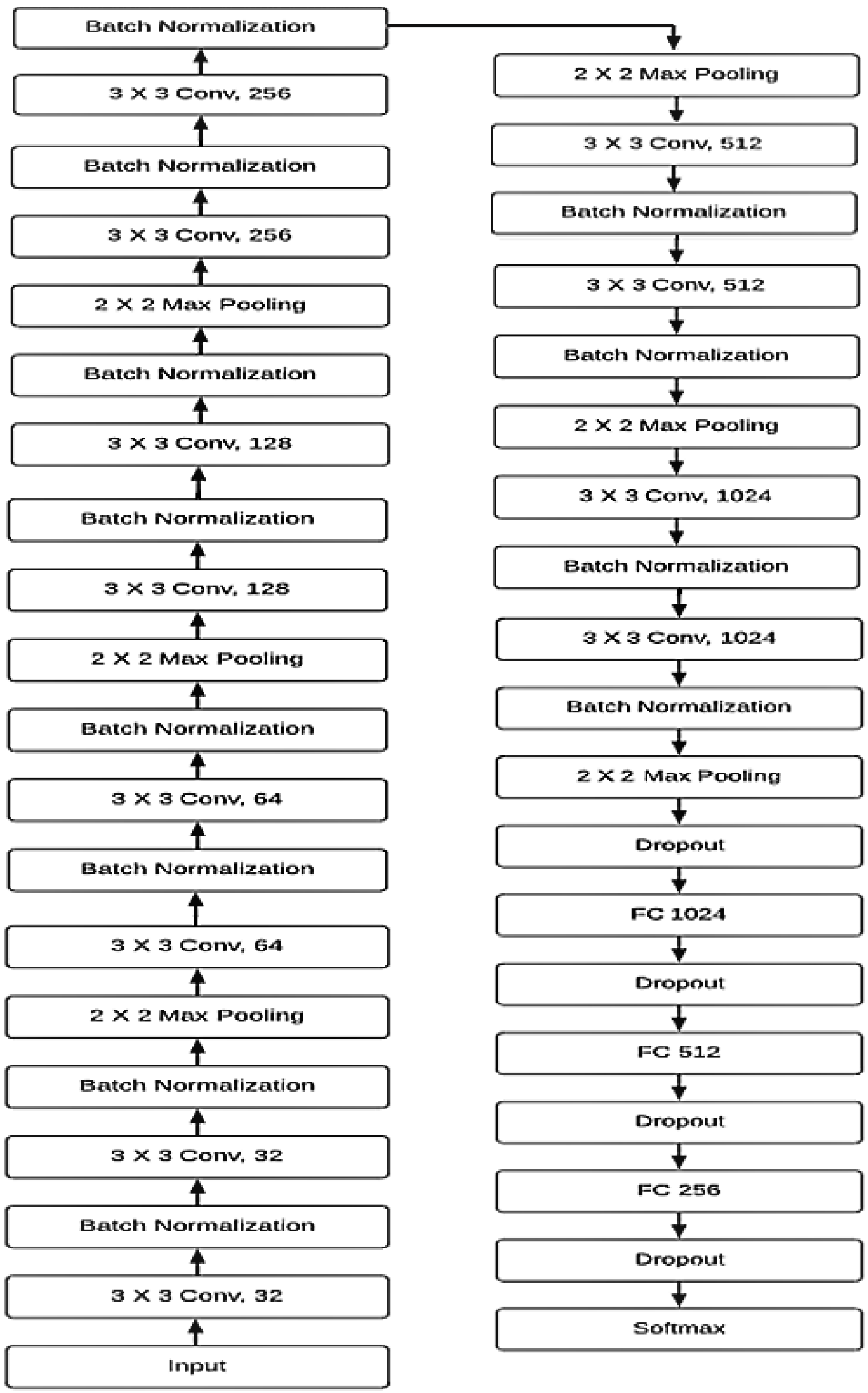


Figure 3.7: TrafficNN Architecture

3.8 Training the Model

To train our model we use our own traffic conditions images dataset. Our dataset has 2500 images, among those 80% images are used for training and 20% images are used for testing. All training procedures were done on Google Colab using free cloud GPU provided by google. It takes around 30 minutes to train our model on this cloud GPU. We update our model several times and alter the optimizer, learning rate, and loss function in order to improve accuracy and reduce loss as much as feasible. To minimize the loss function, we use Adam optimizer to train our model. We used batch size of 25 and 100 epochs to train our proposed model. After doing several types of hyperparameter tuning for several times, we finally found the best performance of our model.

3.9 Implementation Requirements

A list of prerequisites for such traffic image classification work has been produced after a thorough examination of the relevant statistical or theoretical concepts and methods. The following items are likely to be required:

Hardware/Software Requirements

- Operating System (Windows 7 or above)
- Hard Disk (minimum 500 GB)
- Ram (Minimum 8 GB)
- GPU (Recommended)

Developing Tools

- Python Environment
- Google Colab
- Jupyter Notebook

CHAPTER 4

Experimental Results and Discussion

4.1 Introduction

In this section, we will discuss how we train our proposed ‘TrafficNN’ model and what performance we get from it. We will also compare our model’s performance with some pre-trained models like- VGG16, ResNet50, InceptionV3 and DenseNet121.

4.2 Performance Evaluation

Our ‘TrafficNN’ model was trained on our own traffic dataset where 80% images were used as training sets and the other 20% as test sets. So, we have 400 images in each class and a total of 2000 images for training. We have done all the coding and training of our model in Google Colab notebook using the free cloud GPU provided by google. We use Python based TensorFlow 2.0 library along with Keras API to build, train and test our model.

We use Adam as an optimizer which can handle sparse gradients on noisy problems [35]. For ‘TrafficNN’ we used the Adam optimizer with a learning rate of 0.000004. We find this learning rate through hyper parameter tuning using TensorFlow callbacks methods. We use EarlyStopping and LearningRateScheduler as callbacks methods to find perfect learning rate for our given model.

We have used sparse categorical cross entropy as a loss function. The goal of our training was to reduce the cross entropy as much as possible. So, to train the model and observe the performance, we used batch size of 25 and 100 epochs. Our proposed ‘TrafficNN’ model gives 97% accuracy on the training set and 82% accuracy on the test set. Figure 4.1 represents the test and validation accuracy and figure 4.2 represents test and validation loss for our model.

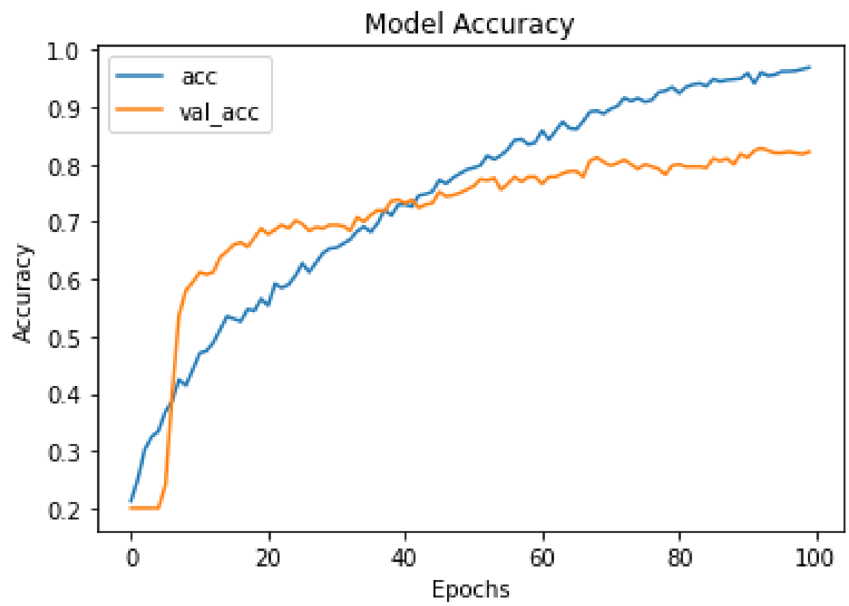


Figure 4.1: TrafficNN training and validation accuracy

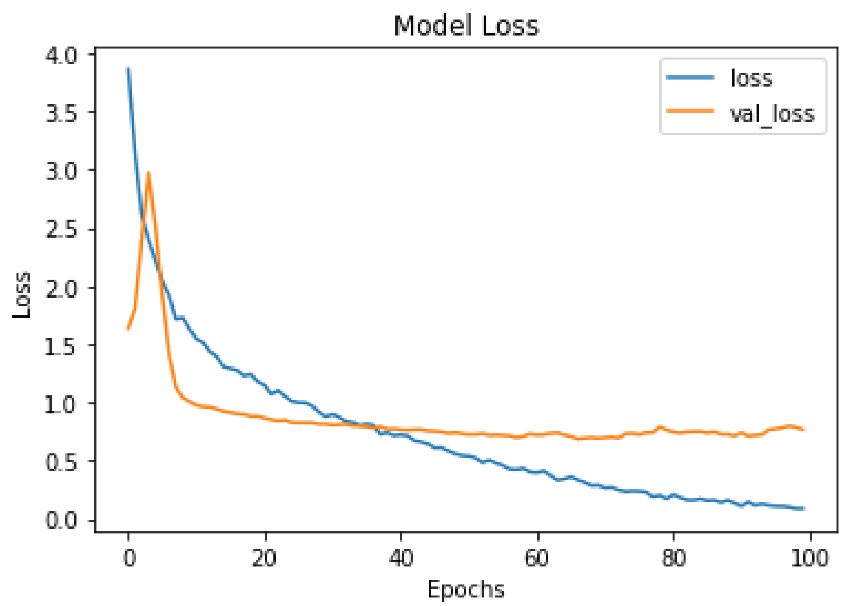


Figure 4.2: TrafficNN training and validation loss

4.3 Result Discussion

We determined f1-score, precision and recall from working on a test dataset which contains 500 images (100 images per class). After classification only 89 images are misclassified and the rest of them give correct predictions.

Many of our traffic conditions images looked so similar for vehicle count, lighting condition and similar type of vehicles. It actually happened due to less difference between features of our determined classes. We also assume that if we have more data, then this would not happen. Despite that, this model also did pretty well for those cases. From table 4.1, we can see the classification report which proves the optimal performance of our 'TrafficNN' model.

TABLE 4.1: TRAFFICNN CLASSIFICATION REPORT

Class Name	Precision	Recall	F1-Score	Support
Emergency Traffic	0.83	0.90	0.87	100
High Traffic	0.90	0.87	0.88	100
Low Traffic	0.83	0.79	0.81	100
Medium Traffic	0.81	0.68	0.74	100
Traffic Accident	0.75	0.87	0.81	100

Indeed, we got 82% accuracy on the test dataset for our 'TrafficNN' model which was quite optimal for unseen data. We are showing the result with the confusion matrix within the given figure 4.3.

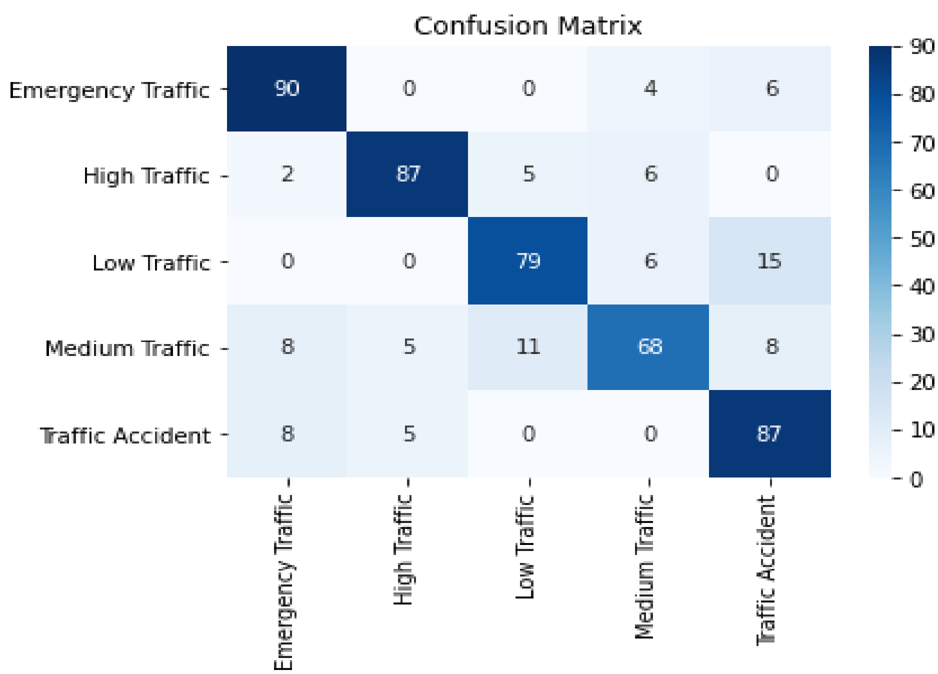


Figure 4.3: Confusion matrix for TrafficNN

From the Confusion matrix above, we can deduce that our model has produced a large number of True Positives and a relatively small number of False Positives.

4.4 Comparison using Transfer Learning

In this part, we compared different pre-trained models with our 'TrafficNN' model such as VGG16, ResNet50, InceptionV3 and DenseNet121. All of those existing models used in this experiment were pre-trained on the ImageNet dataset by default. We also use a transfer learning technique called fine tuning, to re-train those models on our own traffic images dataset. We use same the number of epochs, batch size and learning rate for fine tuning those pre-trained models. But this time we use a different learning rate and number of epochs from 'TrafficNN' training. Because in this transfer learning approach, that learning rate and number of epochs does not work well and shows a huge amount of overfitting.

TABLE 4.2: COMPARISON OF ROAD TRAFFIC CONDITIONS CLASSIFICATION

Model	Training Accuracy	Test Accuracy	Learning Rate	Number of Epochs
TrafficNN	97%	82%	0.000004	100
VGG16	90%	65%	0.001	10
ResNet50	94%	76%	0.001	10
InceptionV3	91%	68%	0.001	10
DenseNet121	90%	75%	0.001	10

From table 4.2, We observed that all of those pre-trained models give at least 90% accuracy on the training dataset. But they show different test accuracy, as different amounts of noise occur for the test dataset. Finally, we observed that under almost the same conditions, every pre-trained model's training accuracy is satisfied, but their test accuracy shows overfitting signs. We also observed that our 'TrafficNN' model performs much better with less overfitting and a test accuracy of 82%.

4.4.1 VGG16

VGG16 gives a good training accuracy of 90% for our traffic conditions images dataset, but it shows a large amount of overfitting as its validation accuracy is just about 65%. From figure 4.3 we can see the training and test accuracy of the VGG16 model.

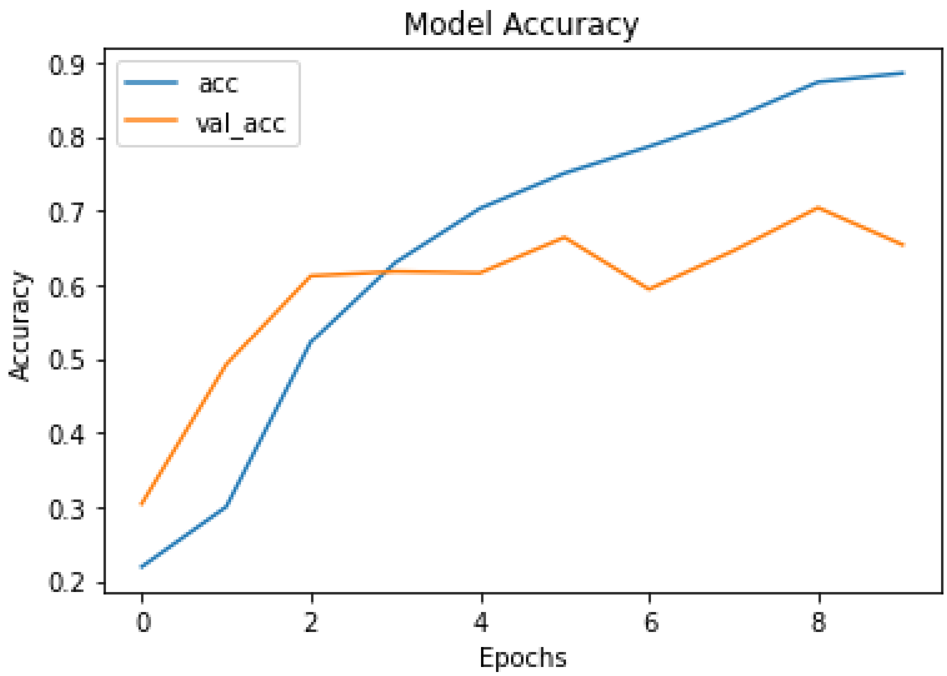


Figure 4.4: VGG16 training and validation accuracy

4.4.2 ResNet50

For our traffic conditions images dataset ResNet50 gives a very good training accuracy of 94% which is highest among all pre-trained models. But it makes a very noisy result for test accuracy which is about 76%. We can see training and validation accuracy of ResNet50 from figure 4.4.

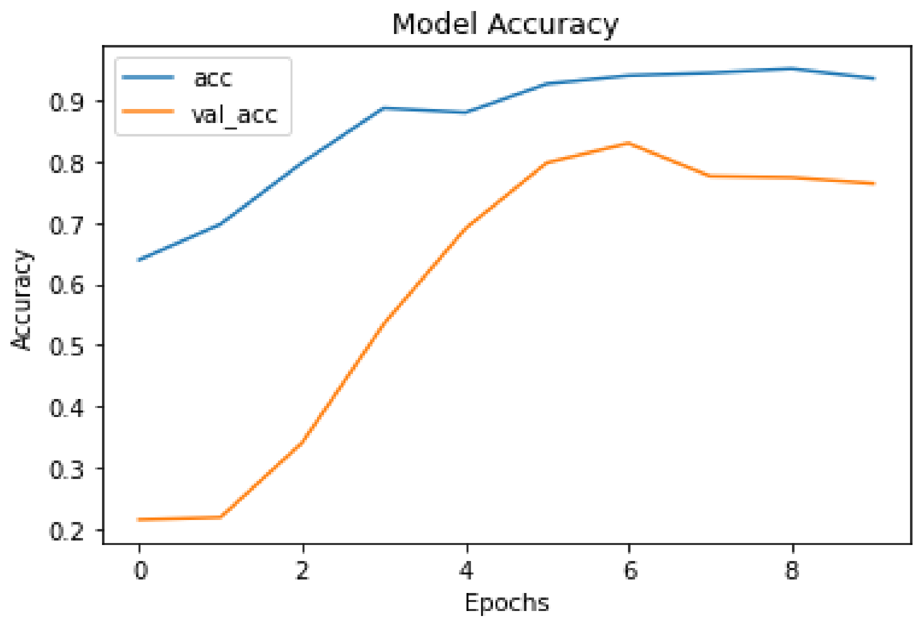


Figure 4.5: ResNet50 training and validation accuracy

4.4.3 InceptionV3

InceptionV3 gives 91% training accuracy on our traffic conditions images dataset. But like other pre-trained models it shows an overfitting sign because it just gives 68% accuracy on the test set.

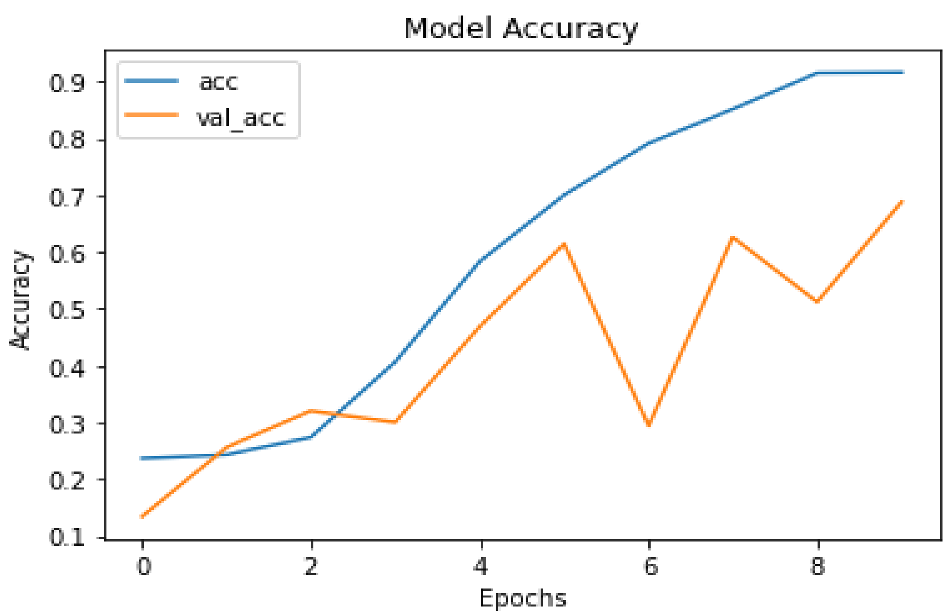


Figure 4.6: InceptionV3 training and validation accuracy

4.4.4 DenseNet121

DenseNet121 performed very well for our dataset and gave 90% training and 75% validation accuracy. In figure 4.6 we see some difference between training accuracy and validation accuracy.

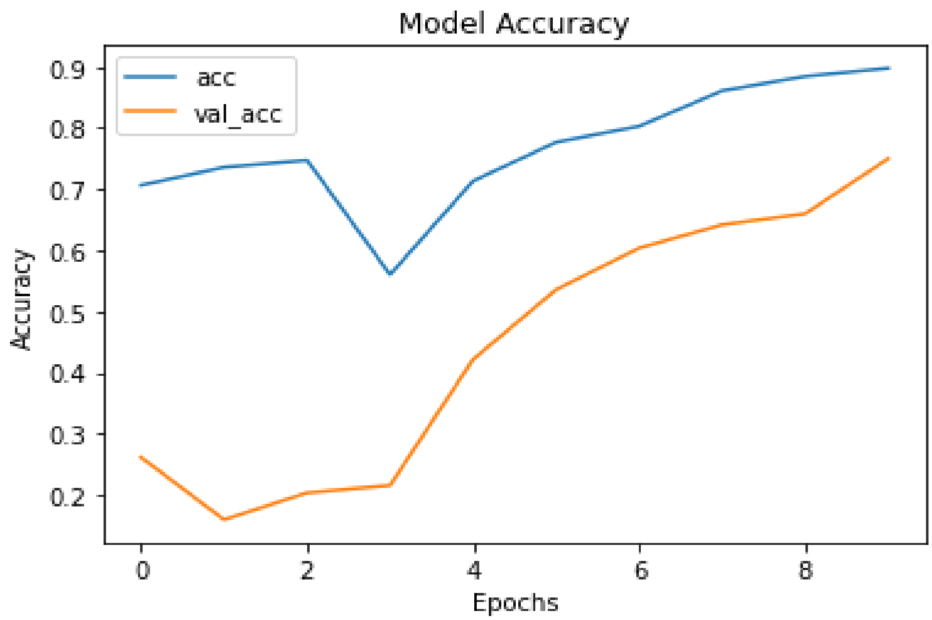


Figure 4.7: DenseNet121 training and validation accuracy

CHAPTER 5

Conclusion and Future Works

5.1 Summary of the study

There is no doubt that there is a lot of research work done in this field. Basically, we want to solve the problem of traffic jams. So, we use a deep learning algorithm called convolutional neural network (CNN) to give a solution to this problem. We proposed a CNN based architecture named as 'TrafficNN' to classify several traffic conditions. We made a traffic conditions images dataset to train our proposed model. And finally, we find an optimal result of our model. We expect that this study will help to build a smart traffic control system in near future along with Artificial Intelligence (AI).

5.2 Conclusions

We know our existing traffic control system causes traffic accidents, traffic jams and road congestion that puts heavy loads on businesses and works because of bad traffic monitoring systems. There is no doubt that a lot of research work has happened in this field. But nobody tries to build a CNN based dedicated architecture to study road traffic conditions properly. We proposed a Convolutional Neural Network (CNN) based 'TrafficNN' architecture to detect traffic conditions on the road. In this paper, we have determined the traffic conditions of a road by classifying into five different classes named as low traffic, medium traffic, high traffic, emergency traffic and traffic accident. Our proposed model classifies road traffic conditions with an accuracy of 82%. We have compared our 'TrafficNN' model with some pre-trained models like- VGG16, InceptionV3, ResNet50, DenseNet121 and find that our 'TrafficNN' model gives highest accuracy among them.

This model along with deep learning and Artificial Intelligence (AI) can be incorporated into current surveillance systems to detect and report road traffic conditions automatically. Then we will try to predict road traffic conditions with the help of surveillance cameras present on the roads. In case, there is congestion on that road, then it can divert the upcoming traffic to another route to save the time of upcoming vehicles to prevent more congestion.

5.3 Future Works

In our dataset all the images are of good resolution. But in the future, we will try to use blur images, low resolution images, distorted images and surveillance camera images as our traffic image dataset. We will also try to integrate more classes to accurately analyze road traffic conditions.

In the future, this work can be a part of a smart traffic control system. Where our model will work as an Artificial Intelligence (AI) system to control traffic lights smartly. It can also give special corridors for emergency vehicles like- ambulance, fire service van, police car to move quickly. Then it can be used to detect accidents just after happening, through the Artificial Intelligence (AI) monitoring system with the help of our proposed model.

REFERENCES

[1] S. Chakraborty, “Traffic congestion in Dhaka city and its economic impact,” Traffic congestion in Dhaka city: Its impact on business and some remedial measures, 2010.

[2] Colossal loss: BUET study finds traffic jam eats up 5 million working hours a day in Dhaka, available at <<https://www.thedailystar.net/frontpage/colossal-loss-1553002>>, last accessed on 26-05-2021 at 09:45 AM.

[3] 25,526 people killed in road accident in the last 10 years, available at <<https://www.dhakatribune.com/bangladesh/nation/2019/06/12/25-526-people-killed-in-road-accident-in-the-last-10-years>>, last accessed on 26-05-2021 at 02:13 PM.

[4] Bangladesh: Alarming rise in road crashes, available at << https://www.aa.com.tr/en/asia-pacific/bangladesh-alarming-rise-in-road-crashes/1692643#>>, last accessed on 26-05-2021 at 03:05 PM.

[5] I. Isa, N. Shaari, A. Fayeez, and N. Azlin, “Portable wireless traffic light system (pwtls),” International journal of research in engineering and technology, vol. 3, no. 2, pp. 242–247, 2014.

[6] S. Mugunthan, “Concept of li-fi on smart communication between vehicles and traffic signals,” Journal: Journal of Ubiquitous Computing and Communication Technologies June, vol. 2020, no. 2, pp. 59–69, 2020.

[7] K. Priyadharshini and S. Manikandan, “Automatic traffic control system based on the vehicular density,” 2019.

[8] R. Sundar, S. Hebbar, and V. Golla, “Implementing intelligent traffic control system for congestion control, ambulance clearance, and stolen vehicle detection,” IEEE Sensors Journal, vol. 15, no. 2, pp. 1109–1113, 2014.

[9] T. Kumar and D. S. Kushwaha, “An approach for traffic congestion detection and traffic control system,” in Information and communication technology for competitive strategies, Springer, 2019, pp. 99–108.

[10] B. Rajak and D. S. Kushwaha, “Traffic control and management over iot for clearance of emergency vehicle in smart cities,” in Information and Communication Technology for Competitive Strategies, Springer, 2019, pp. 121–130.

[11] R. Eswaraprasad and L. Raja, “Improved intelligent transport system for reliable traffic control management by adapting internet of things,” in 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS), IEEE, 2017, pp. 597– 601.

[12] C.-T. Lam, H. Gao, and B. Ng, “A real-time traffic congestion detection system using on-line images,” in 2017 IEEE 17th International Conference on Communication Technology (ICCT), IEEE, 2017, pp. 1548– 1552.

[13] J. Chung and K. Sohn, “Image-based learning to measure traffic density using a deep convolutional neural network,” IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 5, pp. 1670–1675, 2017.

[14] T. Pamula, “Road traffic conditions classification based on multilevel filtering of image content using convolutional neural networks,” IEEE Intelligent Transportation Systems Magazine, vol. 10, no. 3, pp. 11–21, 2018.

[15] M. R. Vinothkanna, “A survey on novel estimation approach of motion controllers for self-driving cars,” Journal of Electronics, vol. 2, no. 04, pp. 211–219, 2020.

[16] P. Wang, L. Li, Y. Jin, and G. Wang, “Detection of unwanted traffic congestion based on existing surveillance system using in freeway via a cnn-architecture trafficnet,” in 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA), IEEE, 2018, pp. 1134–1139.

[17] C. Manchanda, R. Rathi, and N. Sharma, “Traffic density investigation & road accident analysis in India using deep learning,” in 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), IEEE, 2019, pp. 501–506.
[18] J. Kurniawan, S. G. Syahra, C. K. Dewa, et al., “Traffic congestion detection: Learning from cctv monitoring images using convolutional neural network,” Procedia computer science, vol. 144, pp. 291–297, 2018.
[19] H. Cui, G. Yuan, N. Liu, M. Xu, and H. Song, “Convolutional neural net- work for recognizing highway traffic congestion,” Journal of Intelligent Transportation Systems, vol. 24, no. 3, pp. 279–289, 2020.
[20] P. Chakraborty, Y. O. Adu-Gyamfi, S. Poddar, V. Ahsani, A. Sharma, and S. Sarkar, “Traffic congestion detection from camera images using deep convolution neural networks,” Transportation Research Record, vol. 2672, no. 45, pp. 222–231, 2018.
[21] C. M. Bautista, C. A. Dy, M. I. Manalac, R. A. Orbe, and M. Cordel, “Convolutional neural network for vehicle detection in low resolution traffic videos,” in 2016 IEEE Region 10 Symposium (TENSYMP), IEEE, 2016, pp. 277–281.
[22] F. Sun, A. Dubey, and J. White, “DxNAT—Deep neural networks for explaining non-recurring traffic congestion,” in 2017 IEEE International Conference on Big Data (Big Data), IEEE, 2017, pp. 2141–2150.
[23] C. Willis, D. Harborne, R. Tomsett, and M. Alzantot, “A deep convolutional network for traffic congestion classification,” in Proc NATO IST- 158/RSM-010 Specialists’ Meeting on Content Based Real-Time Analytics of Multi-Media Streams, vol. 2017, 2017, pp. 1–11.
[24] H. Yi, H. Jung, and S. Bae, “Deep neural networks for traffic flow pre- diction,” in 2017 IEEE international conference on big data and smart computing (BigComp), IEEE, 2017, pp. 328–331.
[25] A comprehensive guide to convolutional neural networks — the eli5 way, available at << https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 >>, last accessed on 03-02-2021 at 10:05 PM.
[26] F. Sultana, A. Sufian, and P. Dutta, “Advancements in image classification using convolutional neural network,” in 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), IEEE, 2018, pp. 122–129.
[27] Convolutional neural networks explained, available at <<https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939 >>, last accessed on 03-02-2021 at 01:13 AM.
[28] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in 2017 International Conference on Engineering and Technology (ICET), IEEE, 2017, pp. 1–6.
[29] Crash course in convolutional neural networks for machine learning, available at << https://machinelearningmastery.com/crash-course-convolutional-neural-networks/>>, last accessed on 04-02-2021 at 11:13 PM.
[30] A gentle introduction to pooling layers for convolutional neural networks, available at << https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/>>, last accessed on 04-02-2021 at 11:47 PM.
[31] The most intuitive and easiest guide for convolutional neural network, available at <<https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480>>, last accessed on 04-02-2021 at 12:25 AM.
[32] Rectified linear units (relu) in deep learning, available at <<https://www.kaggle.com/dansbecker/rectified-linear-units-relu-in-deep-learning>>, last accessed on 04-02-2021 at 01:17 AM.
[33] Dropout regularization in deep learning models with keras, available at << https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/>>, last accessed on 04-02-2021 at 04:03 AM.

[34] B. Yuan, “Efficient hardware architecture of softmax layer in deep neural network,” in 2016 29th IEEE International System-on-Chip Conference (SOCC), IEEE, 2016, pp. 323–326.
[35] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.

APPENDIX

We faced numerous challenges in completing our project, the first of which was deciding on a methodology. This project was so challenging because only some work was done before with different datasets and different ways. And nobody tries to solve the problem in the way that we have done it. As a result, we didn't receive much information support from anywhere. Another major difficulty was data gathering, which posed a significant problem for us. There was no dataset on this type of road traffic conditions available. But we overcome this challenge and successfully complete our work. We gather data from a variety of web sources and alter it to train our model. We developed a new CNN model with a higher accuracy rate for traffic conditions classification. We have trained and compared several other traditional CNN models with our own model and the best classification result was successfully obtained by our model. Our development and thesis work were difficult but it was very interesting too.

Final Report

ORIGINALITY REPORT

11%	7%	5%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	2%
2	"Emerging Technologies in Data Mining and Information Security", Springer Science and Business Media LLC, 2021 Publication	1%
3	machinelearningmastery.com Internet Source	1%
4	doctorpenguin.com Internet Source	1%
5	Md Tohidul Islam, B.M. Nafiz Karim Siddique, Sagidur Rahman, Taskeed Jabid. "Food Image Classification with Convolutional Neural Network", 2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), 2018 Publication	1%
6	www.computer.org Internet Source	<1%

7	"Information and Communication Technology for Competitive Strategies", Springer Science and Business Media LLC, 2019 Publication	<1 %
8	Submitted to Bahcesehir University Student Paper	<1 %
9	Mario Milicevic, Krunoslav Zubrinic, Ines Obradovic, Tomo Sjekavica. "Chapter 19 Application of Transfer Learning for Fine-Grained Vessel Classification Using a Limited Dataset", Springer Science and Business Media LLC, 2019 Publication	<1 %
10	Submitted to University of Wales Institute, Cardiff Student Paper	<1 %
11	mc.ai Internet Source	<1 %
12	Submitted to GGS IP University Delhi Student Paper	<1 %
13	Loan N. N. Do, Neda Taherifar, Hai L. Vu. "Survey of neural network-based models for short-term traffic state prediction", Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2018 Publication	<1 %

14	Submitted to Napier University Student Paper	<1 %
15	Submitted to Florida Atlantic University Student Paper	<1 %
16	export.arxiv.org Internet Source	<1 %
17	Submitted to Daffodil International University Student Paper	<1 %
18	Submitted to UNITEC Institute of Technology Student Paper	<1 %
19	Submitted to University of Mindanao Student Paper	<1 %
20	thinkspace.csu.edu.au Internet Source	<1 %
21	www.spiedigitallibrary.org Internet Source	<1 %
22	agri.ckcest.cn Internet Source	<1 %
23	napier-surface.worktribe.com Internet Source	<1 %
24	vuir.vu.edu.au Internet Source	<1 %
25	Zhang, Qing-qing, Yong Liu, Jie-lin Pan, and Yong-hong Yan. "Continuous speech	<1 %

recognition based on convolutional neural network", Seventh International Conference on Digital Image Processing (ICDIP 2015), 2015.
Publication

26	alg.csie.ncnu.edu.tw Internet Source	<1%
27	dokumen.pub Internet Source	<1%
28	repository.tudelft.nl Internet Source	<1%
29	towardsdatascience.com Internet Source	<1%
30	www.semanticscholar.org Internet Source	<1%
31	Hongsuk Yi, HeeJin Jung, Sanghoon Bae. "Deep Neural Networks for traffic flow prediction", 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), 2017 Publication	<1%
32	Ramkumar Eswaraprasad, Linesh Raja. "Improved intelligent transport system for reliable traffic control management by adapting internet of things", 2017 International Conference on Infocom	<1%

Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS), 2017
Publication

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		