Traffic Sign Recognition Using Deep Learning

by

Afia Binte Amir
15301039
Umme Habiba Nisa
15301044
Ali Ashab Shafi
15301099
Md. Rafid-Ur-Reza
16101229

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
December 2019

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Afia Binte Amir 15301039	Umme Habiba Nisa 15301044
Ali Ashab Shafi 15301099	Md.Rafid-Ur-Reza 16101229

Approval

The thesis titled "Traffic Sign Recognition using Deep learning" submitted by

1. Afia Binte Amir (15301039)
2. Umme Habiba Nisa (15301044)
3. Ali Ashab Shafi (15301099)
4. Md. Rafid-Ur-Reza (16101229)

Of Fall , 2019 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on December 24, 2019.

Examining Committee:

Supervisor:
(Member)

Dr. Jia Uddin
Associate Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Annajiat Alim Rasel
Lecturer
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Mahbubul Alam Majumdar
Professor and Chairperson(CSE)
Department of Computer Science and Engineering
Brac University

Abstract

Traffic sign recognition plays a significant role in modern automated driver assisting systems and showing information about safety measures. It is a technology that allows users to recognize traffic signs in real-time, typically in videos, or sometimes just in photos. Poor identification of traffic signs cause road accidents. Moreover In adverse situation like heavy rain,foggy weather or sleepy driver can misidentify a traffic sign that may cause the death of hundreds of people. As a result identification of traffic signs properly has become an obligatory topic for research. In this research, we have used convolutional neural network for detecting and classifying the road signs accurately. We have proposed five Keras models of CNN and compared their results. The main challenge of this research is dealing with noise in images such as ads, parked vehicles, pedestrians, and other moving objects or background objects that made the recognition much more difficult. Not only the objects but also various environmental issues like the reflection of light, rainfall, fog etc has affected the research. In order to conduct this research we have collected our own data-set. We roamed around Dhaka city and clicked pictures of the traffic signs as there is no benchmark data-set available in the perspective of Bangladesh. For 500 images this model gives out an accuracy of 63%. There have been many researches in this field but our one is unique as it is tested on our own collected data-set on Bangladesh's perspective. Recognizing traffic signs has become a part of our daily essentials as road safety depends on it, on a large scale which made it an obligatory topic for research.

Keywords: Models; Image processing; CNN; Detection; Recognition; Traffic Sign; Detection; Prediction; Train loss; Parameters; Keras; Inception V3; Inception-Resnet; Mobilenet; Renet50; Xception; Over-fitting

Acknowledgement

All praise to the Almighty Allah, the most beneficent and the most merciful who blessed us with guidance, strength and ability to complete this research.
Furthermore, we would like to express our gratitude to our advisor Dr. Jia Uddin sir for his kind support and advice throughout our thesis work. His contribution and guidance helped us to overcome our problems and make this research work successful.
Finally, we are thankful to our friends and parents for their continuous support throughout this journey. A special thanks to BRAC University for providing us with the opportunity to complete our thesis and finish our Bachelors Degree.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

$ADAS$ Advanced Driver Assistance System

$ADES$ Automatic Driver Evaluation System

ANN Artificial Neural Net

AR Augmented Reality

CNN Convolitional Neural Net

CSV Comma Separated Values

CVA Computer Vision Applications

DSS Driver Support System

EVS Explaine Varience Score

FC Fully Connected

$GTSRB$ German Traffic Sign Recognition Benchmark

HOG Histogram of Oriented Gradient

KL Karhunen-Ioeve

MAE Mean Absolute Error

MLP Multilayer Perceptron

MSE Mean Squared Error

NN Neural Network

$PLSA$ Probabilistic Latent Semantic Analysis

$R-CNN$ Regional Convolutional Neural Network

$ResNet$ Residual Network

ROI Region of Interest

$SIFT$ Scale Invariant Feature Transform

$SURF$ Speeded Up Robust Features

SVM Support Vector Machine

TS Traffic Sign

TSR Traffic Sign Recognition

Chapter 1

Introduction

Traffic signs or road signs are the signs we see everyday at the side of the roads. The purpose of these signs are to give instructions to the pedestrians as well as drivers. The oldest traffic sign is dated back to 1668 which was placed in Lisbon, Portugal by the order of King Peter II of Portugal in order to regulate traffic at that time. Since then the traffic volume is expanding. Many countries have readjusted traffic signs in to simplified pictorial forms. Slowly it became a must to recognise these signs. With the growth of technology the importance of TSR (traffic sign recognition) became inevitable. TSR is a technology that enables to control and guide traffic for road safety purposes.

1.1 Inspiration

Traffic Sign is one of the most indispensable components of sleek flowing of traffic now a days. It also provides road safety by communicating road conditions and reminding the road-safety rules to the drivers. TS recognition and detection is very important for building automated, intelligent driver support systems, intelligent vehicle development and road maintenance. With the help of this detection system, there is a scope to inform the drivers about the road conditions and reduce road accidents. As a result there is a wide range of intelligent driving systems now a days that require the perfect detection method. Detection of the TS might be difficult sometimes due to various weather conditions such as - rainy or cloudy weather, night environment etc. This technology regulates the position of the road sign and what it is in real time. This makes the task complicated and requires a very high accuracy. In order to achieve a higher and more accurate detection system CNN is used.
In recent years self driving cars have taken center stage which is only possible because of CNN. A research conducted by Kaijing Shi, Hong Bao and Nan Ma CNN gives an excellent estimation on vehicle detection base on CNN [30]. Another paper presents an analytical study of comparing the two major approaches: color segmentation and convolutional neural network (C-CNN) approach and fast region proposal convolutional neural network (Fast R-CNN) approach. In the C-CNN method a set of regions of interest (ROIs) is applied for color tresholding on the input image. These methods imply several techniques to improve training and testing speed while also increasing detection accuracy [24]. We are inspired by such research works for conducting our thesis.

1.2 Objective

The aim of this thesis is to predict a TS recognition system based on Convolutional Neural Network (CNN). Our main objective is to collect real time images of the TS in the perspective of Bangladesh on various weather conditions. After collecting the images we have converted the digital images into a data-set. Later we used that data-set for building pretraining and training models. We also aim to apply numerous pre-trained models based on Convolutional Neural Network (CNN) [13]. The models will predict the traffic signs and thus we will make a comparative analysis between various models. Finally, we analyzed the reasons why performance varies between models having tested with the same dataset.

1.3 Thesis Outline

In this research an in depth study is presented in multiple distinctive strategies and techniques that are associated with traffic sign recognition. For our thesis we have used a new approach and methodology to recognize TS in real-time
In chapter 1 we gave a brief introduction of the traffic sign recognition. Chapter 2 discussed some previous research works and their procedures whereas chapter 3 highlighted on CNN and its feature extraction process. Moreover, in chapter 4 we discussed the proposed model and its implementation. Chapter 5 discussed the result analysis. Finally, in Chapter 6 we concluded our research and gave a brief overview of the future.

Chapter 2

Literature Review

TSR has become an integral part of modern research as it paves the way for a safer transportation system. Detecting and recognizing traffic signs properly has always been a challenge due to various environmental conditions. This technology is mainly adapted by numerous automobile companies. In order to detect TS properly, various methods have been used. TSR systems have evolved rapidly over the past ten years. Extracting the ROI that contains the TS is the main goal of a TSR system. A research done by researchers Prashenjit Dhar and Zainal Abedin states that HSV color model is used to extract color information from the images. Thus the features of the image can be extracted by using ROI. The images then are classified again with CNN [26]. The detection method can be divided into three main categories such as color-based, shape-based and learning-based methods. Moreover, classification methods can be divided into two main categories: learning methods based on hand-crafted features (HOG, BRISK, SIFT,LBP, SURF) and deep learning methods. According to these categories, some previous research work on the TSR system is discussed in the following sections.

2.1 Detection method

As acknowledged earlier, there are three main detection categories: color-based, shape-based and learning-based methods. The appropriate method to apply is chosen considering the system requirements and the problem [29]. A few works are discussed that used these detection categories

2.1.1 Colour-based method

For different traffic signs, there are different color specifications. Red, blue and yellow is the most commonly seen traffic sign colors. In this method, the dominant color based portion is applied to find the ROI. The research paper of Tam T. Le and Son T. Tran gives a clear visualization of color detection and segmentation base on SVM (Support Vector Machine) on their paper [7]. They use SVM for retrieving candidate regions for real-time TS processing. An input vector of the SVM block of pixels is used as which is used in color classification. A group of nearby pixels is adopted for increasing the dimension of each vector.
On the other hand, Luis David Lopez and Olac Fuentes used a set of Gaussian models to segregate the TS from its background [4]. A constant homogeneous color

space called CIELab is used for measuring the similarities of the original color (what is viewed by a human observer) and the reproduction of it. thus the system is color sensitive. Also, Gaussian models were implemented for automated detection of mobile targets using color information.
Color threshoding technique is sometimes used in color based detection system. In order to separate TS from background noise, this technique works perfectly. Moreover, RGB color space, HIS, HSV are used for TS detection. An innovative color model named Eigen-Color was introduced by Xie and Li-Feng Liu in their research [5] . Their idea was inspired from the KL (Karhunen-Ioeve) transform hypothesis. This hypothesis suggests to differentiatet road sign color pixels from the background.

2.1.2 Shape-based method

Color segmentation is not treated as an absolute feature for the detection method because of its sensitivity to discrete factors. for example the distance of the target, weather condition, sunlight reflection, variance of daylight, etc. Contrarily Shape-based methods are further robust than color-based methods. the computation time on this approach depends on the numbers of detected edges which may give a very high computation time.
G. Loy and N. Barnes stated a new approach for detecting geometrical shapes effectively [2]. The symmetric nature of the shapes such as triangular, square and octagonal are used together with the pattern of edge orientations. This method returns the location and size of the shape detected. The results in still images show a 95% detection rate.
Another paper by Alfes and Bram presents a system based on edge orientation histograms for road sign detection [3]. The edge orientation histograms are steadfast. It entails of features like contrast invariant and scale that can be used efficiently using integral images. To select features based on the implicit transmission function of the designer's template to the object's appearance in the image a learning method is familiarized. The system is intelligent to detect 85% of the objects on from 12 pixels width and 95% for objects on from 24 pixels width at a truncated false alarm rate

2.1.3 Learning-based method

Machine learning has become compulsory for learning to solve problems. Traffic signs detection is one of the key technologies in automatic driving, which uses artificial feature extraction and machine learning algorithms. In a research based on deep learning for detecting traffic signs, an advanced method called Faster R-CNN is used [31]. Faster R-CNN is an improved version of R-CNN. It combines region proposals with CNNs. As a result, represents the highest level in the field of object detection in recent times.
S. Houben and J. Stallkamp described locating a traffic sign properly as a very challenging computer vision task [11]. Their approach aims to remove the inconsistency between the absence of comprehensive and unbiased identification of the previous methods. In this paper, a real-world benchmark data set is introduced for TS detection. Viola-Jones detector based on Haar features, HOG descriptors are used in

this research. Their algorithms are the best-performing algorithms in the IJCNN competition.
Another research incorporates a top-down and bottom-up visual operation to recognize TS [6]. This paper explains the concept of implementing a completely parallel image analysis approach. The bottom-up approach uses the improved model of Saliency-Based Visual Attention. On the other hand, the top-down stage looks out for saliency regions based on the HOG features. This approach achieves robustness to a great extent.
A group of Korean researchers proposed a strong detection and recognition method that perform well against illumination changes [28]. There is a lot of previous research work that does not show promising results under low lighting conditions. That's, why the introduced a General Purpose Graphics Processing Unit (GPGPU), based on real-time traffic sign detection techniques. In order to evolve ADAS systems in the near future robust methods for traffic sign detection are becoming essential.

2.2 Classification method

Image classification is the mechanism categorizing an image considering its visual content and according to its group with the help of a computer vision application. An image classification model should identify images properly and divide the images into groups or types. Detecting an object and classifying it properly might seem like an easy task but for computer vision applications it is a vigorous task. Classification methods can be divided into two subdivisions- learning methods based on various hand-crafted features and deep learning methods. These two divisions are discussed in the below sections.

2.2.1 Hand crafted feature method

A research conducted by Mrinal Haloi, suggests that every image is a periodic collection of visual topics [14]. His algorithm uses an exceptional classification technique based on PLSA (probabilistic latent semantic analysis). This method is important for feature depiction. This algorithm is tested onGTSRB (German traffic sign recognition benchmark) and delivers a very outstanding result.
High performance of computer vision and machine learning is essential for sign detection and recognition system. This system mainly consists of three parts- detection, feature extraction, and classification. In their paper, Samira El Margae and Berraho Sanae focused on feature extraction with Local Binary Pattern (LBP) method [12]. The LBP method is block based. They implemented the LBP method with the K-NN classifier. Tested on GTSRD, this method has proven to influence a very high accuracy in terms of image classification.

2.2.2 Deep learning-based method

Deep learning is making revolutionary changes in image classification methods and bringing more accurate detection and classification systems. One such system is proposed by Rongqiang Qian and Bailing Zhang who developed a high performance classification system with deep convolutional neural network (CNN) [16]. Their system demonstrates a very high performance rate and recognition accuracy. As their

system is tested on a Chinese traffic sign dataset, their main target was to classify signs according to digits, English letters and Chinese characters. Their work was inspired by R-CNN.
For building, a perfect TSR system, high accuracy and short processing time are intensely important. Lotfi Abdi and Aref Meddeb discover one such techniques using deep CNN architecture [22]. In their research, they focused on Augmented reality (AR) and cascade deep learning. For maximum road-safety and driver comfort, they have used the AR that provides the driver with visual feedback. On the other hand, cascade deep learning and AR covers augmented virtual objects which enhance the driving experience. The researchers experimented by combining the Haar Cascade and deep convolutional neural networks. The results indicate that combined learning increases the ability of the detection system.
Real-time traffic sign detection and classification system are essential for the automatic Driver Evaluation System (ADES). The system proposed by Kemal Kaplan and Caner Kurtul uses affine transformation coefficients as genetic algorithm parameters for sign detection [10]. Their results show that the neural networks are better in classification than SVMs. Their research uses two machine learning approaches that are tested for sign classification- the neural network-based classification and the Levenberg-Marquardt learning technique. Their model can linearly separate classes of objects. However, the errors on the detection stage showed an adverse effect on the classification stage.

Chapter 3

CNN and feature extraction

3.1 Convolutional Neural Network (CNN)

CNN is a normalized version of multilayer perceptrons.It specializes in 2D matrix like images detection. These images consist of 2D matrix of pixels on which we run the CNN algorithm. For recognizing the image or to classify it, CNN takes brain as a motivation. In a human brain the visual area consists of two kind of cells- simple cells and complex cells. Simple cells aid to feature detection whereas complex cells combine local features from small spatial neighbourhood. Spatial pooling helps with translational invariant features. Humans combine all the different local features that they scan through their eye for classifying the image.That's how we see a new image. CNN adopts this process in order to classify images accuraely.

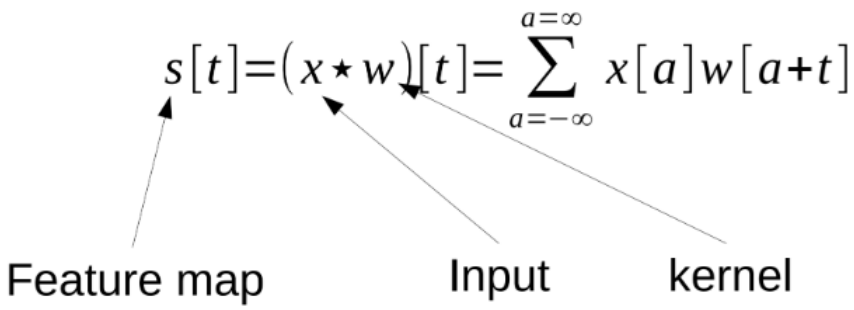


Figure 3.1: Convolution function

A convolutional neural network works with a convolution. A convolution mathematically refers to a function that is derived from two other functions by using integration. It describes how the shape of a function can be modified by another function. For example, if we imagine an input I and an argument, the kernal K will produce an output that will express the modification of the shape of one is by another.From the figure 2.1, we get an image "x". The image is a 2D array of pixels that contain different color(RGB) channels. "w" is the kernel (feature detector) through which we get the output after applying feature map. A feature map is a mathematical operation that is used to compute similarity between two signals. In order to identify the edges of an image, a feature detector or filter is used. The convolution operation does the main task of determining the edges of an image.

Generally it is assumed that the convolution function is always zero. This results in the implementation of infinite summation as a summation over a finite number of array elements which we can see from the below stated equation.

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \tag{3.1}$$

From the equation we see a convolutional function for 2D array, where I means the 2D array and K represents the Kernal.
For simplifying the implementation of convolution in machine learning, we can rewrite the equation number 3.1 using a cross correlation function. A cross correlation function is used in almost all of the neural networks. As a convolution is unstable, the equation 3.1 can be written as the equation 3.2.

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n) \tag{3.2}$$

The equation 3.2 shows the cross correlation function. in order to stabalelize the algorithm the range of valid values of m and n is used. As a result less variation takes place.

3.1.1 How CNN works

CNN is considered as the regularized version of fully connected networks. Earlier we learned that CNN takes inspiration from the brain for functioning. Each neuron in a single layer is connected to all the neurons in the next layer. However, in a neuron is completely independent in a single layer and don't share any connection with other neurons of that layer .As a result, the network becomes fully connected with every layer which also makes CNN prone to data overfitting. The last FC (fully connected) is recognized as the output layer. Moreover, CNN regularizes data with the hierarchical pattern. It assembles the complex patterns using more simplified patterns. CNN takes image as an input for processing and classifying it. Computers recognize images as an array of pixels which depends on the image resolution.It will divide the resolutions into height, width and dimension.

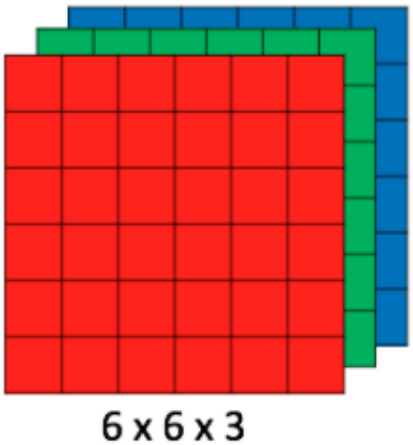


Figure 3.2: Array of RGB matrix

From figure 3.2 we see an image of h x h x d (h = height, w = width, d = dimension)array of matrix of RGB. Here the values are 6 x 6 x 3 [32]. Now for example if we take an image sized 224 x 224, the neurons will calculate 224x224x3 weights. here the dimension will remain 3 as it define the RGB color channel. What CNN mainly does is to pass every input image through a series of convolutional layer that includes filters. This helps CNN to reduce the image size without loosing any information. The layers that include filters are called Kernals.The images are passed through the kernals for training and testing various CNN models.

3.1.2 Architectural Overview of CNN

A convolutional network is made of several layers. The layers are stacked together in order to form a full convolutional architecture. Every layer of this network reconstructs one volume of activations to to another which is done by a differentiable function. Each layer has a straightforward API that converts the 3D input volume to a 3D output volume. The differential function is used to do so. Generally three types of layers are used to build convolutional architecture: Convolutional layer, pooling layer and FC layer. By piling up these layers the original image is transformed to the final class score. Some layers may or may not contain parameters. Usually the convolution layer and the FC layer contain parameters. These parameters are used not only in activation of the input volume, but also as the weights and biases of the input. On the other hand, the ReLu and Pooling layers work to implement a fixed function. So that the parameters from the previous layers can be trained. Gradient descent technique is used to calculate the class results in the convolution layer that depends on the labels of each image in the training set.

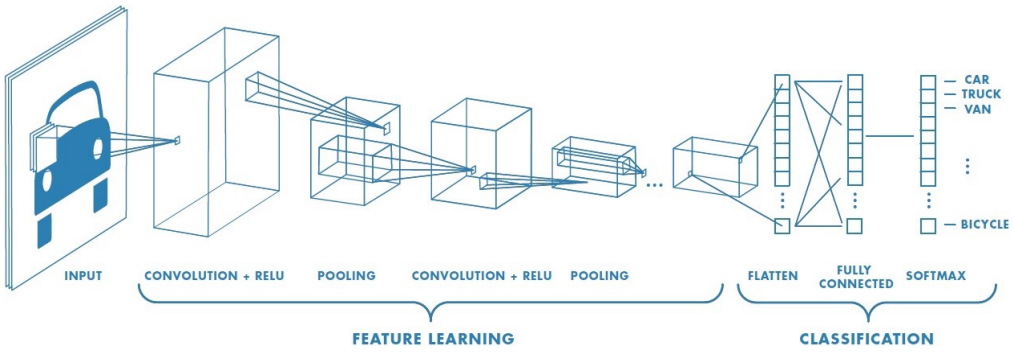


Figure 3.3: Different layers of CNN

From the figure 3.3, we can get a clear visualization of the different layers of CNN. Each input image pass through these layers and Kernals, Pooling, FC and lastly applying Softmax function for classifying an object. This gives out a probabilistic values between 0 and 1. The figure 3.3 represents the complete flow of CNN from processing an input image to classifying the images based on their values. The different layers of CNN are described below:

Input Layer

This layer reads the image. As a result, the input layer has nothing to learn. It only provides the shape of the input image. The input holds the raw pixel values

of an image. For example, if we consider a 64x64 image, the input will be 64x64x3. Here, the height 64, width 64 and with three color channels RGB. So, there are no parameters learn in this layer.

Convolutional Layer

CNN is implemented in this layer.This makes it the basis of a CNN as it is responsible for most of the complex computation. This layer is where CNN actually learns. So, certainly, we have parametres that are the weight matrices. These parameters comprise a set of learnable filters. In order to calculate the parameters, we have to multiply the height, width, and account for the filters. We also have to consider the bias term for each filter. So the number of parameters in a convolutional layer would be -

$$((m * n) + 1) * k) \tag{3.3}$$

In equation 3.1, m= shape of the width of the filter. n= shape of the height of the filter and k= number of filters. It is considered that a convolutional layer has "k" feature maps as output. We added the 1 as the bias term for every single filter.

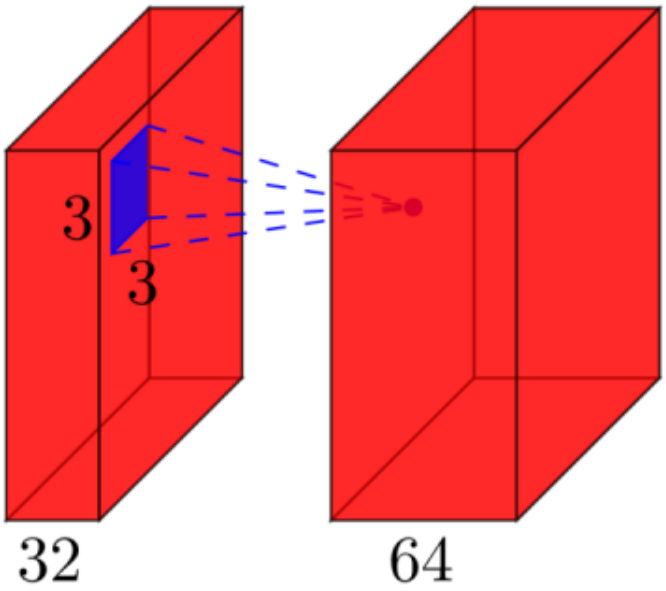


Figure 3.4: Input and output of a Conv layer

From the figure 3.4, We see that the input has I=32 feature maps. k=64 feature maps will be given as outputs and the filter size is n=3 and m=3. It is crucial to keep in mind that we take a 3*3 filter because the input is of 3 2 dimensions. So the output of the first convolution layer is 64 different 3*3*32 filters.

Pooling Layer

In this layer no parameters are announced because it only calculates a fixed function of the given input. This layer helps to reduce image dimension size for reducing amount of parameters. As a result overfitting does not occur. A pooling layer is usually found in-between successive convolutional layers.
A pooling layer is also invariant to translation [8]. This means that if the input is changed at a very small portion, the pooled output remains the same. This feature of the pooling layer helps detecting common characteristics in the input image. For example, edges or colors of an image.

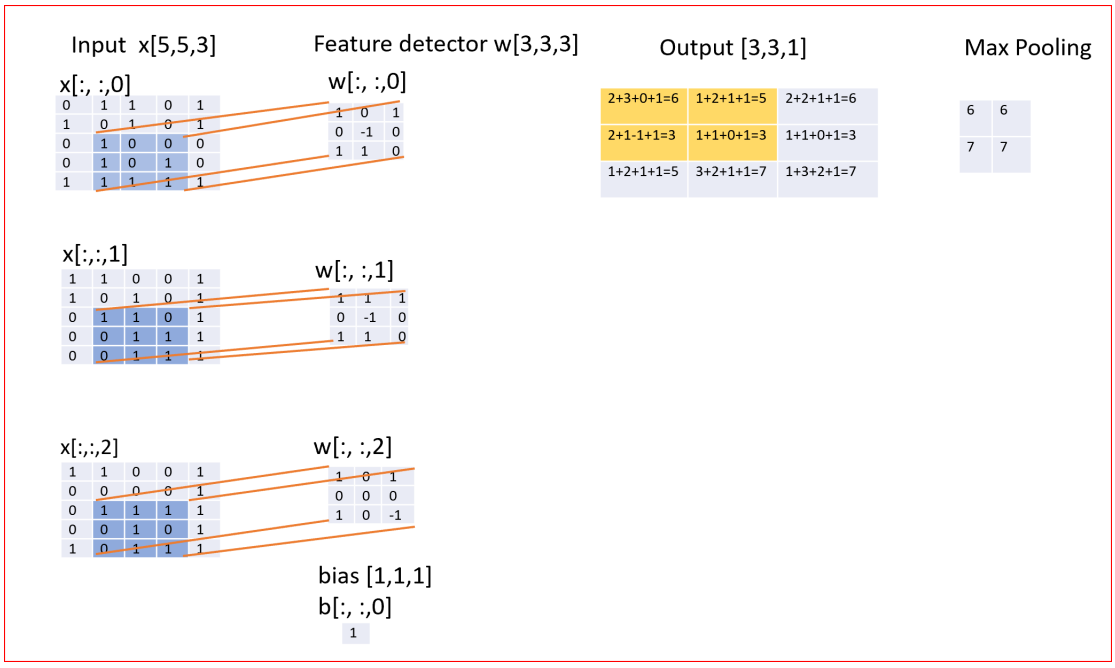


Figure 3.5: Input and output of a Conv layer

Moreover, the max-pooling function produces a better result if compared to min or average pooling. Max pooling sums up the output over a whole other similar input.

ReLu Layer

Rectified Linear Unit(ReLU) is a non-linear operation. reLu means rectified linear unit. This idea is implemented in the convolutional network for introducing non-linearity. As the real-world data requires the convolutional network to be non-negative-linear values, what ReLU does is to transform all the negative numbers into zeros.The output of this layer is shown in equation 3.2.

$$f(x) = max(0, x) \tag{3.4}$$

From the equation 3.4, it is clear that the ReLU function takes any integer value 'x' and returns zero if the integer value is negative.

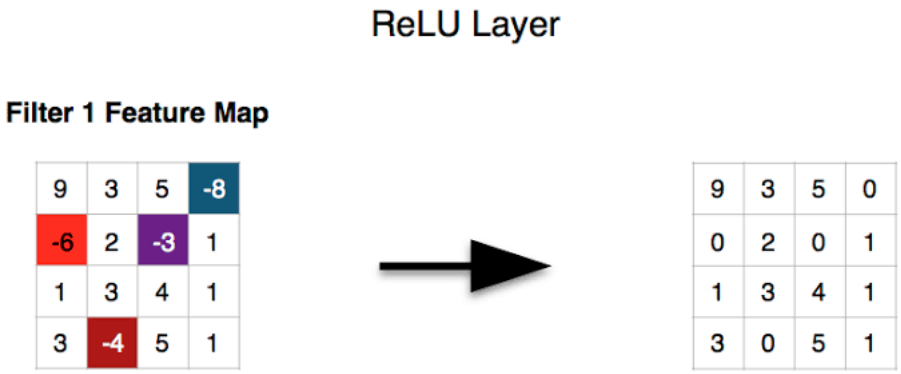


Figure 3.6: ReLU layer

Besides the ReLU function there are also other activation functions like- tanh or sigmoid functions that can used for adding non-linearity to the network. But according to experiments and research, ReLU works best for CNN [17].

Fully-connected Layer

The fully connected layers in CNN are fully connected to each other as the name suggests. This lays is in the last stage of CNN which is completely connected with the previous layers.It portrays the feature vector for the input. From the figure 3.5 we can see the structre of a FC layer.

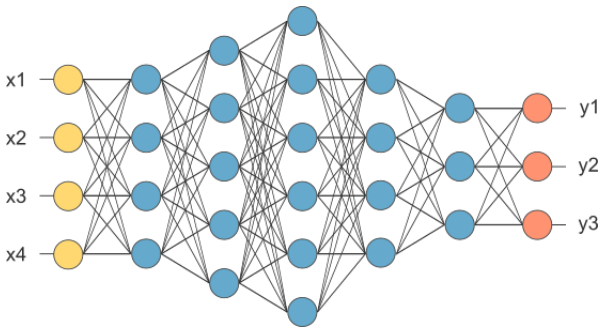


Figure 3.7: Fully connected layer

Neurons of a fully connected layer has all previous layer activation connection. Matrix multiplication with a basis offset id used for activating the connections. When the network gets trained, the feature vectors regulate the loss. This feature vector is also used for classification, regression and feeding input into other layers.

Output Layer

Output layer is the final layer of a CNN. in this layer produces an approximation of each class according to the given input image. This layer uses the softmax activation function that maps the final dense layer and produces vector output that sums up in one output. It will denote whether each element is in certain classes or not.

3.2 Data set description

Benchmark image dataset is vital for conducting a good research. As our research is based on the road signs of our country- Bangladesh, we have to use a data-set that contains Bangladeshi traffic signs. However there is no benchmark dataset available for bangladeshi traffic signs. Hence, we had to collect our own data-set. We moved around various main streets of Dhaka city to collect the footage of traffic signs. We found that some particular signs were not uniform for a particular road sign. For example, we saw that there were "Pedestrian-crossing" sign was inconsistent in different streets of Dhaka; it had contrastive shapes of the frame, independent fonts, different type of image too.

Figure 3.8: Collected data set

Figure 3.9: Collected data set

Table 3.1: Class values

Class number	Class name
1	Road cross
2	Left turn
3	No hydraulic horn
4	No bus stop
5	Bus Stop
6	///// blank
7	No rickshaw
8	right turn
9	One way
10	Caution
11	No right turn
12	No parking
13	30 km
14	40 km
15	U turn
16	Mosque
17	Side road
18	Speed breaker
19	Petrol pump
20	Parking
21	No parking 1
22	No horn
23	Hospital
24	No road crossing
25	Parallel parking
26	Rail-line
27	No truck
28	No car
29	No u turn
30	Both ways
31	50 km
32	No left turn

In future, we plan to get more datasets in diverse weathers like rainy season, foggy winter etc. also in the various time of the day for getting different light conditions and expanding collecting dataset outside of Dhaka city so that we can get a more enriched and diverse dataset of road signs to do further advance research.

3.3 Workflow

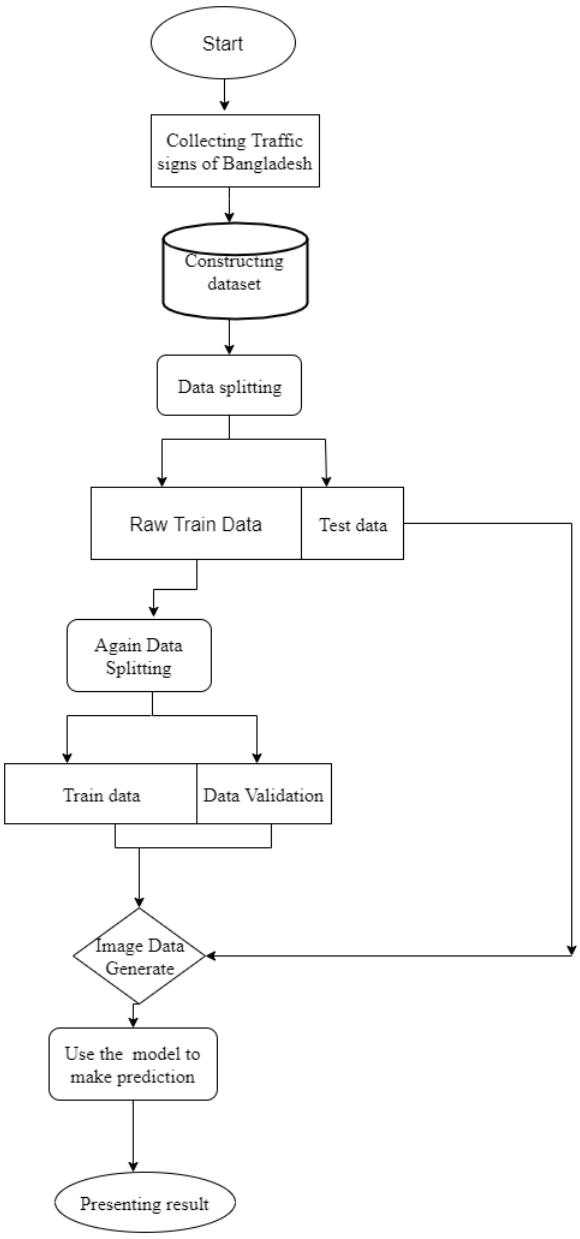


Figure 3.10: Flowchart of the Proposed Model

Our proposed model recognizes Bangladeshi Traffic signs. We have collected our data set by taking pictures of the road signs of Dhaka city for implementing the

model. Figure 3.10 describes the complete workflow of the proposed model. At first we used the data(images) for training our model. We split the data into two parts- raw training data and test data. Then we again split the data into two parts- training data and validation data. After that we pass our train data through the image generator function. We also pass the test data through the same function. Finally we use the generated data with the prediction model and present results.

3.4 Pre-processing

3.4.1 Image pre-processing techniques

For implementing our model, first we have to process our data. To do that image processing is required. Image processing in a technique through which a consistent set of data is produced. Image processing helps to improve the detection systems by reducing variation among data sets, remove misinterpretation and enhancing some valuable features [1]. In image processing systems the images are interpreted as 2D signals. Image processing is essential for pre processing the data set because it helps to extricate objects in an image, measuring the objects of an image and visualizing the invisible objects of an image.
Image processing is usually of two type- digital image processing and analogue image processing. For our research work we will be using digital image processing. It helps to manipulate digital images given as data sets.

3.4.2 Image pre-processing with Keras

Keras is a very powerful deep learning library that is used for data augmentation and building machine learning models. The image data generator class in Keras generates batches of tensor image data with real time data augmentation. First of all all the images are converted into 224 x 224 pixels as it is the standard input for CNN models. We have used some parameters like horizontal_flip,vertical_flip,height_shift_range, width_shift_range,zoom_range,fill_mode etc.

3.5 Compiling the model

After the pre-processing stage we compiled our models. For compiling the models we have excluded the fully connected layer. It is necessary to do so because it requires a complete image to function but with a reduced version of an image it might not be able to perform properly. An active FC layer might result in overfitting. Thus the object will not be classified properly. As for our models the image shape or input size is fixed at 224x 224 x 3, the FC layer is prevented from activating.

Chapter 4

Proposed model implementation

For this research we have used five NN models from Keras. We ran our data-set on these models in order to compare the results. The details is discussed in the following sections.

4.1 Inception V3

In the development stages of the CNN classifier, the Inception network brought an important breakthrough. Before the concept of Inception, what CNNs did, is to just stack convolution layers deeper and deeper for better performance. On the other hand, the Inception network was complex. It pushed performance in terms of both speed and accuracy. the continuous modification of this network leads to the discovery of several versions of the network. For this research, we will be using Inception version 3.

4.1.1 Inception V3 concept

Inception V3 is a CNN model that is trained especially for image classification. In the

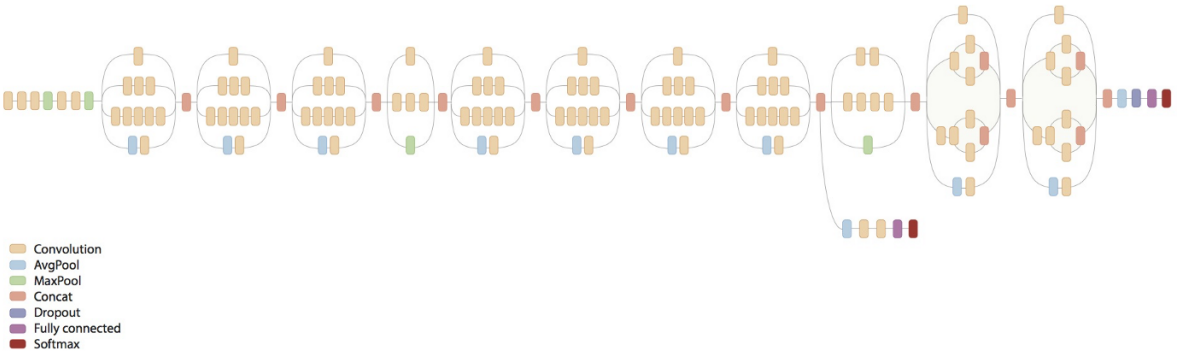


Figure 4.1: Inception V3 architecture

Inception model, the additional classifiers subsidize at the end of a training process. Usually, the accuracy are saturated near the end. That's why in the Inception V3 model the BatchNorm is used in the additional classifiers[20]. Moreover in this

model, a regularizing component is added to the loss formula which prevents the network from overfitting. This process is called label smoothing. Besides Inception V3 also uses 7x7 factorized convolutions. This model also integrates the upgrades of the previous models of Inception.

4.1.2 Applying Inception V3 on our dataset

After preparing our model, we ran 500 steps per epoch and 10 epochs per model. For the Inception V3 model we see a greater valid loss then train loss.

Table 4.1: Train loss and Valid loss(Inception V3)

Train loss	Valid loss
62.3223	118.1029
45.0415	251.5956
54.9919	141.7624
43.0990	57.4983
36.2884	274.7885
39.7707	447.9269
38.1633	628.4097
32.7210	70.7147
41.5034	964.8751
36.5510	155.6451

From the table 4.1 we see that the train loss data has slowly decreased whereas the valid loss data has a very high value. The higher value of validation loss indicates that over-fitting has occured. Over-fitting is a situation where a highly complex model allows a perfect classification of the training samples. It is highly implausible to give a very good classification result. So, Inception V3 model allowed a perfect classification which resulted in data over-fitting.

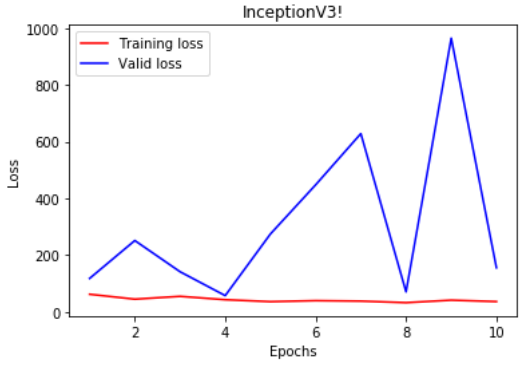


Figure 4.2: Train loss and valid loss Inception V3

The figure 4.2, shows a graph between the trainloss and valid loss data. We can see that irregular and sudden rise and fall of the valid loss data. The training loss data is comparatively steady.

4.2 Resnet 50

Deep Residual net(ResNet) is one of the most groundbreaking discoveries in deep learning. In ResNet, several layers are stacked which helps to train hundreds or even thousands of layers. Even though training this huge amount of data ResNet gives still produces compelling performance. Due to this fascinating performance, many CVA(computer vision applications) have been encouraged. Especially the object detection and face recognition work remarkably well in this model. Since the discovery of ResNet in 2015, many changes have been made to its architecture for better results[27].

4.2.1 Resnet concept

We know that shallow network seems to work well. We used the weights of the shallow network to construct a much deeper network and whenever we have gap, we do identity mappings. This is just the construction.The main idea of ResNet is making a deeper network out of a shallow network by copying weights in the shallow network and setting other layers in the deeper network to identity mapping. General convention suggests going deeper for training and testing errors in ResNet.But simply stacking the layers together does not make the layer deeper. Deep networks are hard to train because the non-linearity is inversed along with the depth. However, this seems to be a problem because generally, this does not seem to work as we go deeper.The more we go deep the more saturated result we get. As a result, we face issues with training and testing. The reason behind this problem is called vanishing gradient problem [15]. In this problem the gradient is back-propagated to the previous layers which repeats multiplication among layers. Correspondingly the gradient becomes infinitively small in size.That is not alleviated and so it is an optimization problem basically. From the figure 4.1 we can visualize this problem.

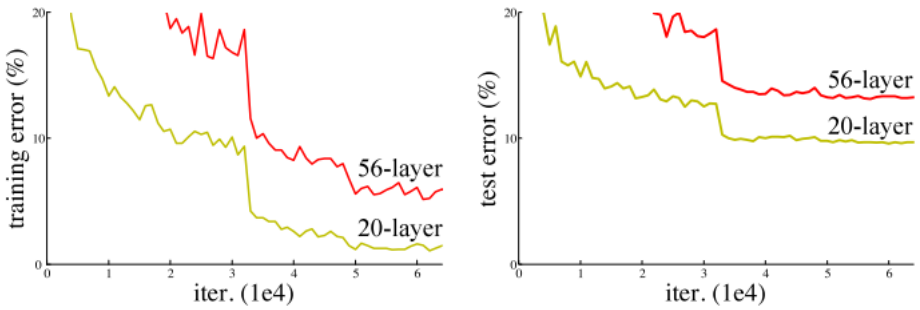


Figure 4.3: Increment of network depth resulting worse performance

To get over this problem, the ResNet paper introduced “Skip connections” which is basically identity mapping or replication. This provides an alternative path for the gradient flow and makes training possible. The number of parameters is also sometimes increase with the kind of approach as you go deeper, you have more layers.

4.2.2 Applying Resnet 50 on our dataset

For ResNet 50 we have used 10 epochs and 500 steps per epoch. We kept the epochs and steps per epoch same for each of the models in order to compareamong them.

Table 4.2: Train loss and Valid loss(ResNet50)

Train loss	Valid loss
65.8007	146.6959
52.5009	58.9105
40.3297	63.0124
40.1097	100.4593
46.4111	83.5137
34.1146	76.4893
35.5693	69.8227
30.7061	59.7766
30.1588	68.5390
24.4571	34.4596

From the table 4.2 we can see that the train loss and valid loss data has given closer output then the Inception V3. This indicates that the ResNet50 network worked perfectly for our dataset. ResNet 50 is able to perfrom at its best in comparatively smaller dataset. That's why the ResNet50 give out a very good accuracy.

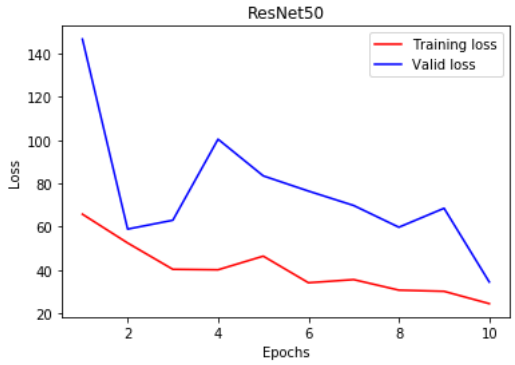


Figure 4.4: Train loss and valid loss ResNet 50

From the figure 4.4, we can see that in the second and last epoch the train loss and valid loss data is very close to each other. If the train loss and valid loss data are close to each other, there's a very chance of getting a very high accuracy.

4.3 Inception-Resnet V2

The Inception V3 architecture has achieved an exquisite performance with a surprisingly low computational cost. The recent discovery of residual networks that has a more traditional architecture also performed exceptionally well in a challenging environment. This model of CNN is pretrained in ImageNet dataset which has more than a million of images. Somehow its performance was identical to the Inception-v3 model. Inception-ResNet is the study of two ideas combined together.

4.3.1 Inception-Resnet V2 concept

Inception-Resnet v2 is simply the combination of the Inception and the Residual Network structure.In this structure, multiple sized convolution filters are merged with the residual network. It is designed in such a way to bypass the degradation problem that occurs for the deep structure. This model is used to ease the difficulties of training a deep neural network. Instead of learning unrefereed functions, this model only learns from the layers that have the reference to the layer input. These networks are easy to optimize. On the other hand, the Inception model is exceedingly prone to change. A pure Inception variant without residual connections gives out the same recognition result as Inception-ResNet-v2. In fact, sometimes a Residual-Inception model outperforms the similar Inception models without residual connections such as- Inception V3 and Inception V4 [19].

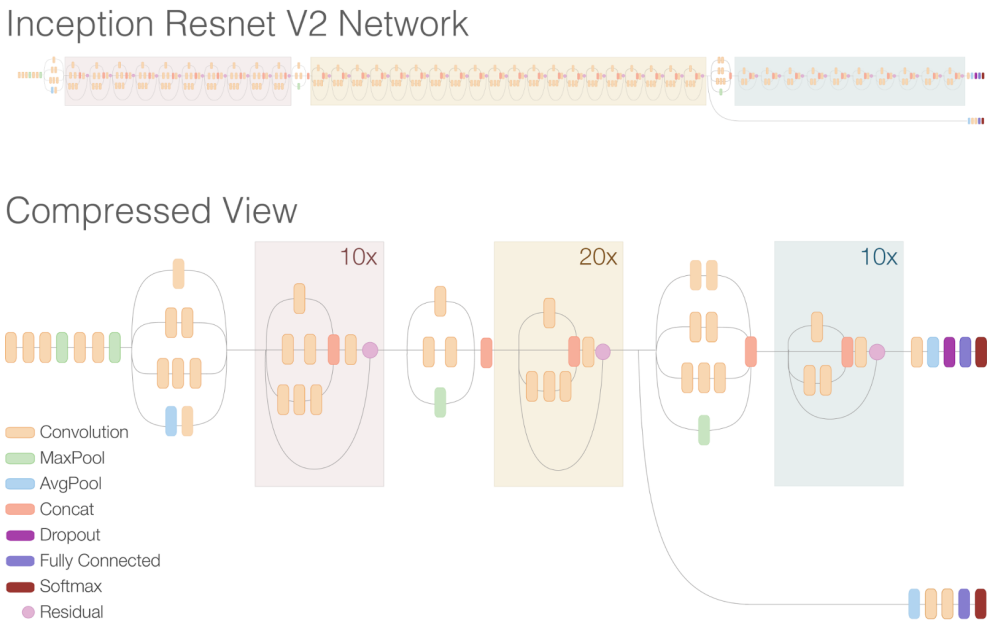


Figure 4.5: Inception-ResNet V2 architecture

From figure 4.3, we can see that the top of the second Inception-ResNet V2 figure, the full Inception-ResNet V2 network is expanded. This network is considered deeper than the previous Inception V3 networks. The Inception-ResNet V2 architecture is more authentic than the previous networks. This model also requires approximately twice the memory as well as the computation time compared to Inception V3.

4.3.2 Applying Inception-Resnet on our dataset

Similar to the previous models, we have used 10 epochs and 500 steps per epoch in this model as well.
From the table 4.3, we visualize that some valid loss and train loss data is close to each other. As a result, it is not due to overfitting. InceptionRestnet model uses the core concept of Residual network.

Table 4.3: Train loss and Valid loss(Inception-ResNet V2)

Train loss	Valid loss
46.4809	60.9243
38.6178	35.2266
27.6253	34.1973
28.3849	44.4474
32.6751	63.8055
29.5776	40.4417
27.9130	53.7313
27.2952	41.4022
26.8861	50.6517
26.6391	34.3154

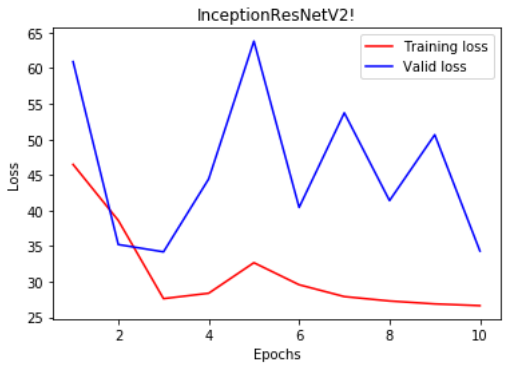
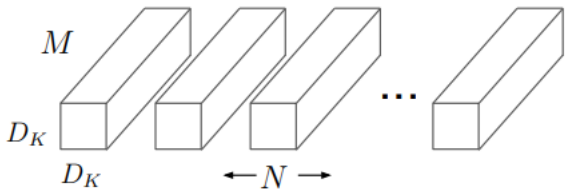


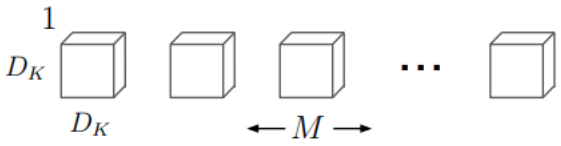
Figure 4.6: Train loss and valid loss Inception-ResNet V2

4.4 Mobilenet

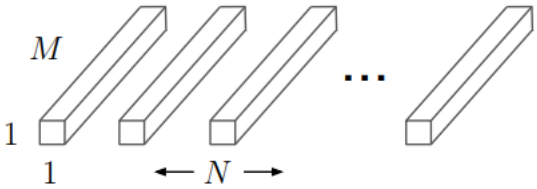
The MobileNet network is basically proposed by Google. The significance of using this network is the advantage of parameter number reduction in NN. This is one of the fastest models in terms of performance and a significantly lower space-consuming model. The MobileNet gives an opportunity to solve image classification and detection problems efficiently.



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters



(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

Figure 4.7: Depthwise separable convolution

4.4.1 Mobilenet concept

In order to explain the architecture of the MobileNet, first of all, the perception of depthwise separable convolution should be clear. It is used by MobileNet.That's why MobileNet has a very lightweight architecture [33]. If we try to distinguish between the network that has normal convolution with a depthwise separable convolution, with the same depth in the network, we will find that that the depthwise separable convolution outstandingly lessens the parameter quantity.
From the figure 4.4, we can see that depthwise separable convolutions does not combine all three color channels rather performs a single convolution on each colour channel. This effects filtering of the input channels. In other words, MobileNets depthwise convolution applies a single filter for each input channel. However, the pointwise convolution. The pointwise convolution is a 11 convolution that is applied in combining the outputs. The depthwise separable convolution has two layers - a

layer for filtering and another one for combining. This helps to reduce computation and model size.

4.4.2 Applying Mobilenet on our dataset

As we kept the epochs and steps per epochs same for all the model,so, for this model it is also 10 epochs with 500 steps per epoch.

Table 4.4: Train loss and Valid loss(MobileNet)

Train loss	Valid loss
35.6542	65.1221
26.5699	34.7975
23.7361	52.3003
23.7620	27.3313
21.5263	37.9076
24.8763	61.6929
23.1744	27.0103
19.5940	27.0088
18.5548	42.2702
20.9273	47.1078

The table 4.4 shows the train loss and valid loss data for the MobileNet model. this model gives out the best result on our dataset. from the table it is visible that the difference between train loss and valid loss data is not very high compared to other models. This indicates that MobileNet model works well for our data-set.

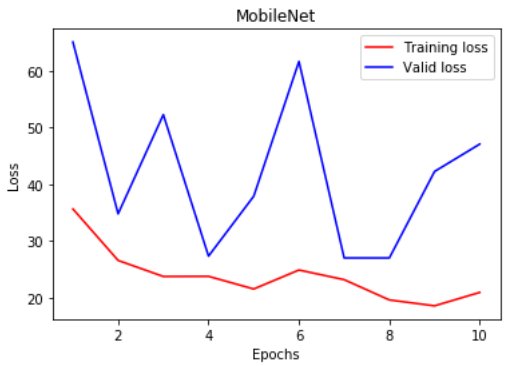


Figure 4.8: Train loss and valid loss MobileNet

From the figure 4.8, we see that the in most of the epochs, the difference between valid loss and train loss data are very low. As we know if train loss and valid loss data is approximately similar compared to other models then there is a chance of having a better result. Because of that MobileNet model worked excellently under our small dataset.

4.5 Xception

Xception a model developed by Google which stands for “Extreme Inception”. This model represents an comparatively stronger version of Inception architecture. This

model also uses a depthwise separable convolution and gives out further better even result than Inception V3. Xception presents a clarification of Inception models in CNN. It is an intermediate stepin-between regular convolution and the depthwise separable convolution operation which is followed by a pointwise convolution [25].

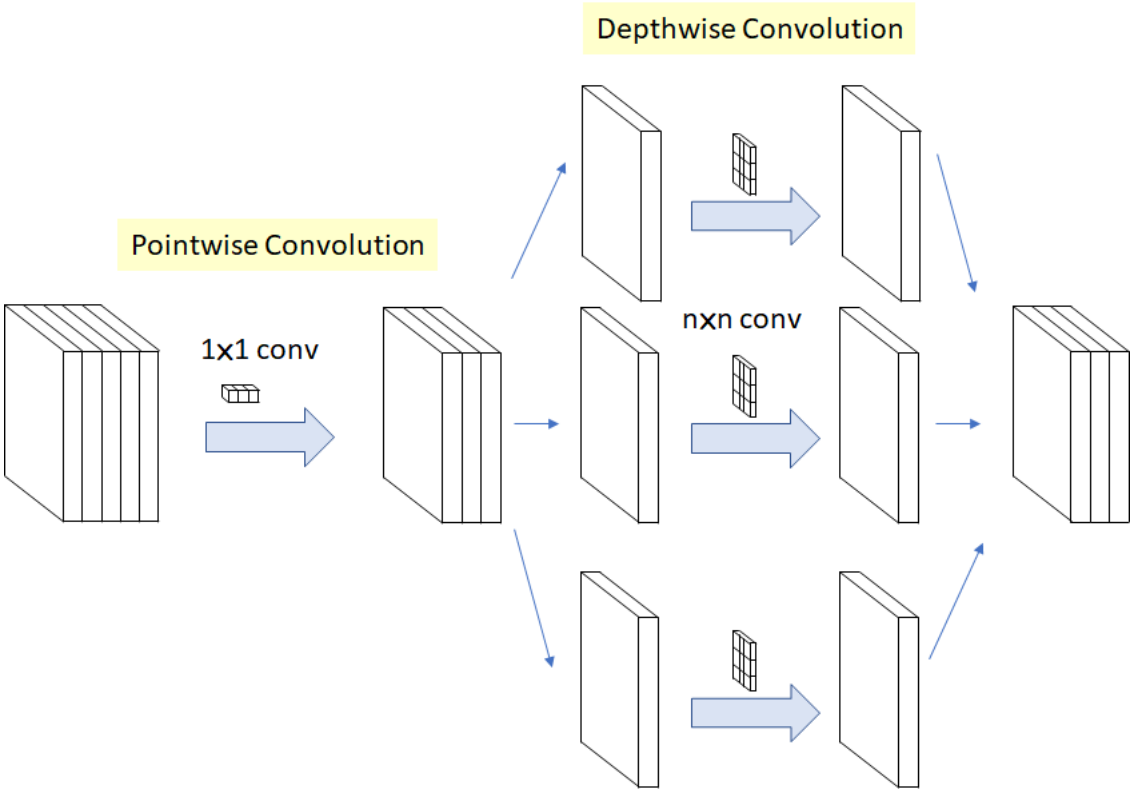


Figure 4.9: Modified depthwise separable convolution

4.5.1 Xception concept

A Xception architecture is a linear stack of depthwise separable convolution layers with residual con-nections. It has 36 convolutional layer that forms the feature extraction base of this network.This model uses the modified depthwise separable convolution hypothesis. Modified depthwise separable convolution performs 1x1 convolution first then channel-wise spatial convolution whereas an original depthwise separable performs channel-wise spatial convolution first. Xception architecture depends on two major points- one is the depthwise separable convolution and another is shortcuts in convolutional blocks (similar to the ResNet architecture). In a word the Xception model architecture consists of depthwise separable convolution blocks and Maxpooling together linked with the shortcuts of ResNet application. This model architecture has a finite number of trainable parameters contrast to an identical depthwise classical convolutions.

4.5.2 Applying Xception on our dataset

As same as the previous models, we have used 10 epochs and 500 steps per epoch. the table 4.5 shows us the trainloss and valid loss data for our Xception model.

Table 4.5: Train loss and Valid loss(Xception)

Train loss	Valid loss
43.9929	716.7349
32.5064	108.2320
28.0942	53.0001
24.9578	102.3378
22.7562	119.2220
24.2169	42.0844
22.2047	32.3283
22.0193	84.6817
22.4495	34.0924
25.6874	32.4876

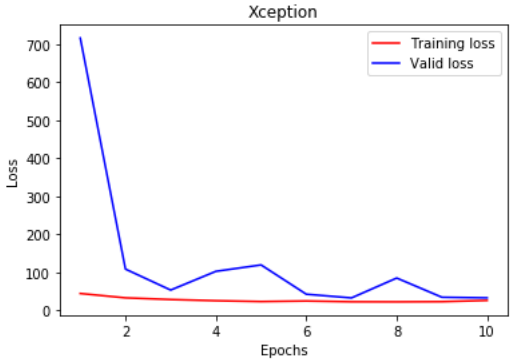


Figure 4.10: Train loss and valid loss Xception

From the figure 4.10, we see that our train loss and validation loss are close to each other. This shows that the Xception model worked well for our dataset.

Chapter 5

Result analysis

5.1 Comparison among the CNN models

For detecting and recognizing the traffic signs in an effective manner, we have used five CNN models. They are Inception V3, MobileNet, ResNet 50, Inception-ResNet V2 and Xception. This section views a comparative study among the given results by these models. Here we have used three metrics for measuring accuracy in CNN models. They are - mean absolute error (MAE), mean squared error (MSE) and explained varience score (EVS). From the table 5.1 below, we see the results of different models.

Table 5.1: Results of different models

Model	**Mean Absolute Error**	**Mean Squared Error**	**Explained Variance Score**
MobileNet	4.6919	37.2801	0.5710
Inception V3	2.3792	27.8953	0.5028
ResNet 50	4.1039	31.5095	0.6308
Inception-ResNet V2	3.8293	29.7352	0.6223
Xception	3.8314	33.1771	0.6291

Mean absolute error

MAE of a CNN model assigns the average of the absolute values of each prediction error. It indicates error on all instances of the test data-set. The difference between the original value and the prediction value of a particular instance is called the prediction error. From the table we see that our Inception-Resnet V2 model gave a MAE of 3.8293 metrics.among all the models Inception-ResNet V3 gave the less error in terms of MAE. Moreover, not only Inception-ResNet but also Xception model gave a similar MAE. Xception model showed MAE of 3.8314 metric which is almost similar to the Inception-ResNet model. This means our Inceprtion-Resnet model and Xception model will work more effectively than the other models if the accuracy is calculated in MAE.In terms of MAE accuracy search, highest value suggests that the model is less effective.

Mean squared error

MSE of a CNN model defines the mean squared error of the prediction results. It is kind of similar to MAE. But the difference if that MSR calculates square

difference for each point. The square difference is computed between the target and the predicted values. After that the average of those values are calculated. The high value in this method indicates the less effective model. From the table 5.1, we see that the values of the mean squared errors of the models. Furthermore, the smallest value 29.7352 is shown by Inception-ResNet V2 which illustrates it as the best model for our data-set.

Explained variance score

Variance score means the difference between the average of predicted values and the observed values. For our system we have calculated our EVS.
Inception V3 gave out the less accuracy among the five models we have used. we see that Inception V3 has an EVS of 50% .
On the other hand, ResNet 50 gave the highest accuracy that is 63% . ResNet architecture works well in smaller data-set thus worked well for our data-set.
MobileNet gave the accuracy of 57% . worked well for our data-set. This is the smallest accuracy we have got.
Inception resnet V2 and Xception both gave the same EVS which is about 62% . Both worked well for our data-set.
If we collect more data and run our algorithm the results will change. more over if the epochs and steps per epoch changes the results will also vary.

Table 5.2: Output,shape and the parameters

Layer	**Inception V3**	**MobileNet**	**Inception ResNet**
Image Shape	(224,224,3)	(224,224,3)	(224,224,3)
Model Convolution shape	(5,5,2048)	(7,7,2024)	(5,5,1536)
Model Convolutional parameters	21802784	3228864	54336736
Global avg pooling	2048	1024	1536
Dense_13 (shape)	1024	1024	1024
Dense_13(parameters)	209817	1049600	1573888
Dense_14(shape)	1	1	1
Dense_14(Parameters)	2025	1025	1025
Total parameters	(23,901,985)	(4,279,489)	(55,911,649)
Trainable parameters	(23,867,553)	(4,257,601)	(55,851,105)
Non-trainable parameters	(34,432)	(21,888)	(60,544)

Table 5.3: Output,shape and the parameters

Layer	**ResNet 50**	**Xception**
Image Shape	(224,224,3)	(224,224,3)
Model Convolution shape	(7,7,2048)	(7,7,2048)
Model Convolutional parameters	23587712	20861480
Global avg pooling	2048	2048
Dense_5 (shape)	1024	1024
Dense_5(parameters)	2098176	2098176
Dense_6(shape)	1	1
Dense_6(Parameters)	1025	1025
Total parameters	(25,686,913)	(22,960,681)
Trainable parameters	(25,633,793)	(22,906,153)
Non-trainable parameters	(53,120)	(54,528)

Table 5.4: Result comparison of the previous works

Author Name	**Dataset**	**Methods**	**Accuracy**
Boujemaa Et al. [24]	GTRSB Dataset	C-CNN, R-CNN	95% & 94%
Dhar Et al. [22]	Own Dataset	ANN, KNN, CNN	93.9% , 71% & 97%
G. Loy and N. Barnes [2]	Google	Novel shape-based techniques	95%
Lopez and Fuentes [4]	none	Gaussian model	95%
T.T.Le Et al. [7]	Real time video	SVM and Hough transform	92%
K.Kaplan Et al. [10]	Real time images	SVM	87%
Houben Et al.[11]	GTSB	Viola-Jones method	88%
Kwangyong Lim Et al. [28]	LISA US traffic sign dataset	GPGPU-based real-time traffic sign detection	89.5%
Aghdam Et al. [23]	GTSRB	New Conv Net	99.23%
Xie Et al. [21]	GTSRB	Cascade CNN	97.94%
Cirean Et al. [9]	GTSRB	MLP, HOG CNN	99.15%
Qian Et al. [18]	GTSRB	CNN, MLP	98.86%

Chapter 6

Conclusion

About millions of people die every year in road accidents all over the world. In Bangladesh road accidents are increasing at an alarming rate. Most of the accidents occur due to the lack of traffic sign recognition. In order to improve that state traffic sign recognition has become mandatory. The main purpose of this thesis work is to propose a Traffic sign detection system on the perspective of Bangladesh. While conducting this research we found some major problems with the traffic signs in our country. We compared five Keras models of CNN on our own data-set. We did a comparative study between these models and it turns out Mobile net gives out the best accuracy for our data-set which is about 63%.

6.1 Future works

There has been a lot of research work done for image detection and traffic signs detection using CNN and other neural networks. But if we look closely, we can see that Tesla, Volvo and few other vehicle manufacturing companies are trying to implement proper image and traffic sign detecting technologies which will finally lead us toward full autonomous transportation system. Again, for our work, we have collected our own data sets based on our weather conditions. So, there is a huge opportunity to improve and solve any problem that may occur in automated transportation system in the future.
In our research work, we got just about average accuracy rate for the algorithms we have applied. It is because we used a small number of data sets and a fixed no of class and also data sets were collected in bright sunlight conditions. So, we can gather more data sets in other roads of Bangladesh and also in different weather and lighting conditions. Then we can achieve more precise accuracy rate. Again, as our work was based on our country's road, the same road sign was given different shape, font, color which gave us a complicacy but we believe the results can be more improved through further research.

Bibliography

[1] J. R. Jensen and K. Lulla, "Introductory digital image processing: A remote sensing perspective", 1987.

[2] G. Loy and N. Barnes, "Fast shape-based road sign detection for a driver assistance system", in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 1, Sep. 2004, 70–75 vol.1. DOI: 10.1109/IROS.2004.1389331.

[3] B. Alefs, G. Eschemann, H. Ramoser, and C. Beleznai, "Road sign detection from edge orientation histograms", in *2007 IEEE Intelligent Vehicles Symposium*, IEEE, 2007, pp. 993–998.

[4] L. D. Lopez and O. Fuentes, "Color-based road sign detection and tracking", in *Image Analysis and Recognition*, M. Kamel and A. Campilho, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 1138–1147, ISBN: 978-3-540-74260-9.

[5] L.-W. Tsai, J.-W. Hsieh, C.-H. Chuang, Y.-J. Tseng, K.-C. Fan, and C.-C. Lee, "Road sign detection using eigen colour", *IET computer vision*, vol. 2, no. 3, pp. 164–177, 2008.

[6] Yuan Xie, Li-Feng Liu, Cui-Hua Li, and Yan-Yun Qu, "Unifying visual saliency with hog feature learning for traffic sign detection", in *2009 IEEE Intelligent Vehicles Symposium*, Jun. 2009, pp. 24–29. DOI: 10.1109/IVS.2009.5164247.

[7] T. T. Le, S. T. Tran, S. Mita, and T. D. Nguyen, "Real time traffic sign detection using color and shape-based features", in *Intelligent Information and Database Systems*, N. T. Nguyen, M. T. Le, and J. Świątek, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 268–278, ISBN: 978-3-642-12101-2.

[8] D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition", in *International conference on artificial neural networks*, Springer, 2010, pp. 92–101.

[9] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, "A committee of neural networks for traffic sign classification", in *The 2011 international joint conference on neural networks*, IEEE, 2011, pp. 1918–1921.

[10] K. Kaplan, C. Kurtul, and H. L. Akİn, "Real-time traffic sign detection and classification method for intelligent vehicles", in *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012)*, IEEE, 2012, pp. 448–453.

[11] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of traffic signs in real-world images: The german traffic sign detection benchmark", in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, Aug. 2013, pp. 1–8. DOI: 10.1109/IJCNN.2013.6706807.

[12] S. El Margae, B. Sanae, A. K. Mounir, and F. Youssef, "Traffic sign recognition based on multi-block lbp features using svm with normalization", in *2014 9th international conference on intelligent systems: theories and applications (SITA-14)*, IEEE, 2014, pp. 1–7.

[13] Y. Wei, W. Xia, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, "Cnn: Single-label to multi-label", *arXiv preprint arXiv:1406.5726*, 2014.

[14] M. Haloi, "A novel plsa based traffic signs classification system", *CoRR*, vol. abs/1503.06643, 2015. arXiv: 1503.06643. [Online]. Available: http://arxiv.org/abs/1503.06643.

[15] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015. arXiv: 1512.03385 [cs.CV].

[16] Rongqiang Qian, Bailing Zhang, Yong Yue, Zhao Wang, and F. Coenen, "Robust chinese traffic sign detection and recognition with deep convolutional neural network", in *2015 11th International Conference on Natural Computation (ICNC)*, Aug. 2015, pp. 791–796. DOI: 10.1109/ICNC.2015.7378092.

[17] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network", *arXiv preprint arXiv:1505.00853*, 2015.

[18] R. Qian, Y. Yue, F. Coenen, and B. Zhang, "Traffic sign recognition with convolutional neural network based on max pooling positions", in *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, IEEE, 2016, pp. 578–582.

[19] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning", *CoRR*, vol. abs/1602.07261, 2016. arXiv: 1602.07261. [Online]. Available: http://arxiv.org/abs/1602.07261.

[20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[21] K. Xie, S. Ge, Q. Ye, and Z. Luo, "Traffic sign recognition based on attribute-refinement cascaded convolutional neural networks", in *Pacific rim conference on multimedia*, Springer, 2016, pp. 201–210.

[22] L. Abdi and A. Meddeb, "Deep learning traffic sign detection, recognition and augmentation", in *Proceedings of the Symposium on Applied Computing*, ACM, 2017, pp. 131–136.

[23] H. H. Aghdam, E. J. Heravi, and D. Puig, "A practical and highly optimized convolutional neural network for classifying traffic signs in real-time", *International Journal of Computer Vision*, vol. 122, no. 2, pp. 246–269, 2017.

[24] K. S. Boujemaa, I. Berrada, A. Bouhoute, and K. Boubouh, "Traffic sign recognition using convolutional neural networks", in *2017 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, Nov. 2017, pp. 1–6. DOI: 10.1109/WINCOM.2017.8238205.

[25] F. Chollet, "Xception: Deep learning with depthwise separable convolutions", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[26] P. Dhar, M. Z. Abedin, T. Biswas, and A. Datta, "Traffic sign detection — a new approach and recognition using convolution neural network", in *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, Dec. 2017, pp. 416–419. DOI: 10.1109/R10-HTC.2017.8288988.

[27] V. Fung. (2017). An overview of resnet and its variants, [Online]. Available: https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035 (visited on 07/17/2017).

[28] K. Lim, Y. Hong, Y. Choi, and H. Byun, "Real-time traffic sign recognition based on a general purpose gpu and deep-learning", *PLoS one*, vol. 12, no. 3, e0173317, 2017.

[29] Y. Saadna and A. Behloul, "An overview of traffic sign detection and classification methods", *International Journal of Multimedia Information Retrieval*, vol. 6, pp. 1–18, Jun. 2017. DOI: 10.1007/s13735-017-0129-8.

[30] K. Shi, H. Bao, and N. Ma, "Forward vehicle detection based on incremental learning and fast r-cnn", in *2017 13th International Conference on Computational Intelligence and Security (CIS)*, Dec. 2017, pp. 73–76. DOI: 10.1109/CIS.2017.00024.

[31] Z. Zuo, K. Yu, Q. Zhou, X. Wang, and T. Li, "Traffic signs detection based on faster r-cnn", in *2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, Jun. 2017, pp. 286–288. DOI: 10.1109/ICDCSW.2017.34.

[32] Prabhu. (2018). Understanding of convolutional neural network (cnn) — deep learning, [Online]. Available: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148 (visited on 11/19/2019).

[33] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.