TRAFFIC SIGN RECOGNITION

A

Project Report

Submitted for the partial fulfilment of

B.Tech. Degree

in

COMPUTER SCIENCE & ENGINEERING

by

SRISHTI KASHYAP (1900520139004)

TEHREEM ARSHAD (1900520139005)

VIVEK PANDEY (1900520139006)

Under the supervision of

PROF. VINEET KANSAL

DR. ADITI SHARMA

Department of Computer Science and Engineering

Institute of Engineering and Technology
Dr. APJ Abdul Kalam Technical University, Lucknow, Uttar Pradesh

May, 30

Declaration

We believe that this submission is our job and, within our knowledge and beliefs, materials previously published or created by someone else, or the awarding of a diploma or diploma for the following reasons: We hereby declare that we do not include the materials accepted for the serious error of the university or other university unless recognition is given in the text. This project was not submitted to another institution to require a different degree.

Submitted by: - Date:24/05/2022

(1) Name: Srishti Kashyap

Roll No: 1900520139004

Branch: Information Technology

Signature:

(2) Name: Tehreem Arshad

Roll No: 1900520139005

Branch: Information Technology

Signature:

(3) Name: Vivek Pandey

Roll No: 1900520139006

Branch: Information Technology

Signature:

Certificate

This is to certify that the project report entitled **"Traffic Sign Recognition"** presented by **Srishti Kashyap, Tehreem Arshad, and Vivek Pandey** for the award of Bachelor of Technology in Information Technology, is a record of work carried out by them under my supervision and guidance at the Department of Computer Science and Engineering at Institute of Engineering and Technology, Lucknow. It is also certified that this project has not been submitted to any other institute for the award of any other degrees to the best of my knowledge.

Prof. Vineet Kansal
Department of computer science and Engineering
Institute of Engineering and Technology, Lucknow

Dr. Aditi Sharma
Department of Computer science and Engineering
Institute of Engineering and Technology, Lucknow

Acknowledgment

At the outset, we are indebted and grateful to our Lord who blessed us with innumerable favors to reach the present academic heights. We thank God for giving us kind parents so that we were able to gain knowledge through their compassionate sacrifice and tenacious efforts.

We would like to express our sincerest gratitude to **Prof. Vineet Kansal** and **Dr. Aditi Sharma** of the Institute of Engineering and Technology, Lucknow for all the wisdom and guidance that they had shown us throughout this year. They had always been gracious and respectful in all stages of our project and without their support, this project would not have been possible.

We are deeply thankful to our Supervisor (Prof. Vineet Kansal) and co-supervisor (Dr. Aditi Sharma) for their cooperation and for giving us valuable suggestions and guidance on our project and for making it correct every time we needed. without their guidance, this project would not have been possible.

Our deep sense of appreciation goes to our parents, sisters, brothers, and friends for their continuous support, care and blessings.

Last but not least we are also thankful to our various teachers who taught us various subjects throughout this course which help us to reach this stage.

Project Members:

SRISHTI KASHYAP (1900520139004)

TEHREEM ARSHAD (1900520139005)

VIVEK PANDEY (1900520139006)

Table of Contents

PAGE NO

ABSTRACT

"Traffic sign recognition and classification" plays an important role in our daily lives today. These days, road signs are everywhere on the road. Despite this, drivers frequently make mistakes. While driving on the road, it is quite difficult to perceive and detecttraffic signs. This may happen due to the following conditions; if the weather conditions are not good and the signboards become blurry, covered by snow, there is heavy rainfall, signboards are of different sizes, and they may be rotated by certain angles. Drivers may misunderstand traffic signs. This leads to accidents, fatal accidents, and damage to the vehicle. To solve this problem, this project introduces a concept called traffic sign recognition. This model is built using CNN to extract images and classify road signs. There are different types of traffic signs, For Example, Speed limit, no entry, traffic lights, left or right turns, children crossing, overtaking of large vehicles, etc. An important part of the Intelligent Transport Systems is the Traffic Sign Recognition System (TSRS).

Accurate and effective recognition of traffic signs can increase driving safety. Traffic signs have certain properties that can be used for recognition and classification. For example, color, shape, and shading are important attributes that help drivers get road information. The colors used for traffic signs in each country are almost the same, consisting of colors such as red, blue, and yellow, and fixed shapes such as circles, triangles, and rectangles. The traffic sign recognition system is implemented using the Convolutional Neural Network. (CNN). The CNN model for this project will be used for classification. The project uses the Kaggle German Traffic Sign Recognition Benchmark as a dataset.

ACRONYMS

CNN	Convolutional Neural Network
ADAS	Advanced driver assistance systems
SDC	Self-Driving Car
CV	Computer Vision
ITS	Intelligent Transformation system
TSR	Traffic Sign Recognition
GTSRB	German Traffic Sign Recognition Benchmark
ROI	Region of interest
IJCNN	International joint Conference of Neural Network

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Traffic Sign Recognition is an important technology that is used to detect and classify traffic signs. There are various traffic signs present as we move on the road such as children crossing, road work, 20-speed limit, zebra crossing, no entry, U-turn, etc., we need to follow these traffic signs for driving safely or to move safely on the roads. Whether we talk about self-driving cars or human driving cars both the safety of the vehicle and the driver is an important aspect for which to follow the traffic rules are important. A convolution Neural Network is a Deep Learning Algorithm that helps to implement this concept of Traffic Sign Recognition with the help of image classification by taking an image as an input and classifying it with an appropriate label.

1.1 Motivation

Traffic sign detection and classification is an important technology, as it helps drivers in understanding signs and following traffic rules, as well as contributing to the development of autonomous driving systems. Traffic signs present on the roads provide proper assistance to the drivers which helps them to be alert in advance for the speed regulations, road constructions, slippery conditions, etc. These traffic signs can be categorized as follows, namely warning, regulatory, direction signs, and information. Road signs are placed on the road in various sizes and shapes. Some traffic signs are very blurry and small, covered with snow and heavy rain, and can rotate at a specific angle so that the driver will not recognize or misunderstand them. Even in this state, we are aiming to recognize traffic signs. Traffic sign classification is a technique that uses images as input to recognize traffic signs such as "U-turn destination", "speed limit 40", and "caution". These signs guide the driver. Therefore, traffic sign recognition is a challenging topic and a valuable topic in traffic engineering research.

1.2 Objective and scope

Traffic signs are placed next to, above, or adjacent to highways, roads, paths, or other routes to guide and warn the flow of traffic, including cars, bicycles, pedestrians, riders, and other travelers. , And a device to regulate. As customer safety becomes more and more important, traffic sign recognition (TSR) has become one of today's research topics for improving driving safety. In addition, the resulting system is unaffected by human problems such as fatigue and

insomnia. This has improved driving safety.

To this end, TSR systems have been developed primarily to reduce the chance of overlooking important traffic signs on the road. At first glance, the problem seems easy to solve. However, the human brain performs steps based on all the experience of visual data, so computers cannot easily perform the same process. Instead of these experiences, a basic understanding of the characteristics of traffic indicators can be used. Color and shape information makes up the bulk of these attributes. This knowledge is not enough to distinguish road signs from others, but you can improve segmentation by combining intelligence and knowledge. Automatic traffic sign recognition systems need to solve problems such as sign recognition and location in crowded scenes, sign feature recognition, and character classification. Implementing traffic sign classification to identify traffic signs more quickly. To locate Traffic Sign Detection, this detects the traffic sign as itapproaches us and alerts and notifies us. To learn how to recognize traffic signs from real-timewebcams to detect traffic signals.

1.3 Organization

The report is divided into five chapters. Chapter 1 gives a brief idea about the project. It tells about the Objective and Motivation of the project. Chapter 2 is the Literature Review which provides depth analysis of all the Research Paper read during the project to explore the topic and Documentation that were used in making this project. Chapter 3 is an explanation of the methodology used in this project which gives an analysis of how the project will be working when put under various test case scenarios. Chapter 4 shows the experimental result and all the possible output. Chapter 5 is concluding the project and possible future work.

CHAPTER 2

LITERATURE REVIEW

For the development of a model that can recognize traffic signs through images, previous research work on the way is reviewed to get the knowledge about the work that had already been done in this field. The last decade has shown growth trends in the development of Intelligent Transport Systems (ITS), especially ADAS and Self-Driving Vehicles (SDC). In these systems, the recognition and recognition of traffic signs is one of the difficult tasks faced by researchers and developers. This issue has been addressed as a problem of recognizing, recognizing, and classifying objects (traffic signs) using CVs, but there are still challenges.

2.1 Traffic Signs

Understanding every traffic sign is the most important aspect for all road users. The main rules and regulations of the road are communicated through traffic signboards that are placed on the sides of the roads using that can be understood easily within seconds. In addition, everyone who wants to apply for a driver's license must first understand all of the traffic signs to pass the driving theory test. The development of self-driving vehicles using the concept of traffic sign recognition for driving safely on the road is very important as it is very useful in recognizing roads, vehicles, pedestrians, and traffic signs. The basic principle of a driving assistance system for traffic sign recognition is to recognize the sign, interpret its meaning, and then send the information to the driver (via the windshield, screen, or smartphone projection) or even more. Is to send to the vehicle itself. Then perform the action without the need for a human decision. The 43 classes that have been used in this project are displayed below that show the basic traffic signs that are used on the roads to provide information to the driver for his safety, to prevent accidents, deaths, and, damage to the vehicle.

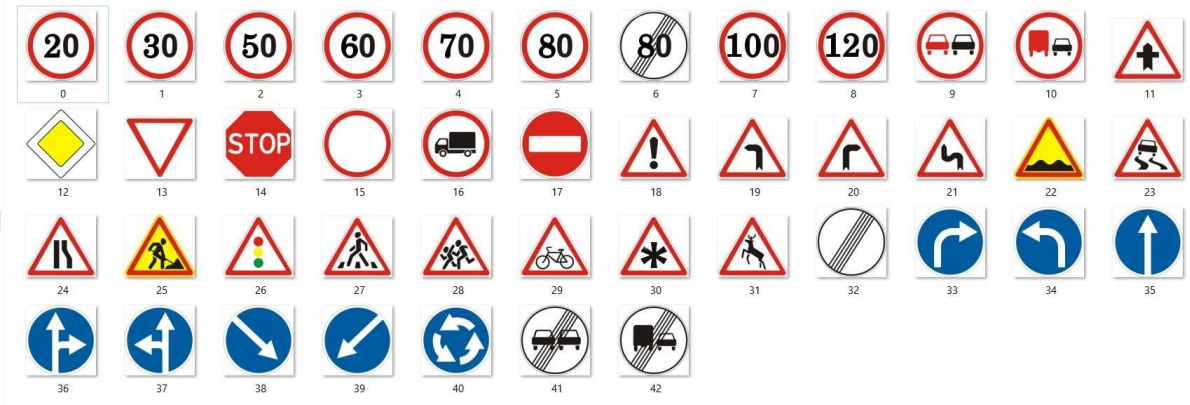


Figure 2.1: Traffic Sign of GTSRB

2.2 Related Work

Deep learning is a branch of machine learning that includes a "convolutional neural network" used in this project to build a model that classifies images with appropriate labels. Deep learning algorithms mimic the behavior of the human brain, but at a much smaller scale. Image classification is used to extract features from an image to identify trends in a dataset.

Ying sun, Pingshu Ge and Dequan Liu (2019) offered a system for traffic sign recognition on account of deep learning that was used to classify circular traffic signs. This method was used to identify traffic signs by uniting image preprocessing, traffic sign detection, recognition, and classification. The accuracy of the system proposed is 98.2 percent, according to the test results [1].

Md Tarequl Islam (2019) proposed a system that focuses to detect and classify traffic signs that are circular and triangular in shape. The background color of the traffic signs used was either red or blue. A set of 28 traffic signs was used for the proposed model. The dataset used was taken from the UK traffic sign image database. Firstly, the image was blurred using gaussian blur for smoothing of the image i.e., if we talk in terms of image processing the sharp edges of the image are eliminated and then converted in the HSV (Hue saturation Value) color space. To eliminate the blue and red colors from the image masking was done. It is a process in which certain portions of the image are revealed are some are hidden. The accuracy of the system was moderate and can be improved by trying different neural network structures. To find the accuracy used the street images of the UK and Bangladesh. Generally, the system was divided

into two stages namely detection and classification. But this is not used for real-time it just classifies the image [2].

Vaibhavi Golgire (2021) proposed a traffic detection and recognition system and a method for extracting traffic signs from complex images of nature, processing them, and alerting the driver with voice commands. It is designed to allow drivers to make quick decisions. The proposed architecture is divided into three parts, the first part is image preprocessing including quantification of the input file of the dataset, the second part is to determine the input size for learning purposes, and The last part is scaling. Information on learning steps. The algorithm used classifies the symbols recognized during the recognition process [3].

Wenhui li, Daihui, and Shangyou Zeng (2019) worked on both a traditional Convolution Neural Network and an improved Convolution Neural Network and used the German Traffic Sign Recognition Benchmark (GTSRB) dataset and the Belgium Traffic Sign Dataset. The comparison of both the Algorithms was done and the results showed that the accuracy of the model build using the improved CNN was much higher as compared to the traditional CNN. The improved CNN uses fewer parameters, smaller models, and easier training i.e., this algorithm was superior to the traditional CNN method in traffic identification.

CHAPTER 3

METHODOLOGY

3.1 Design and Implementation

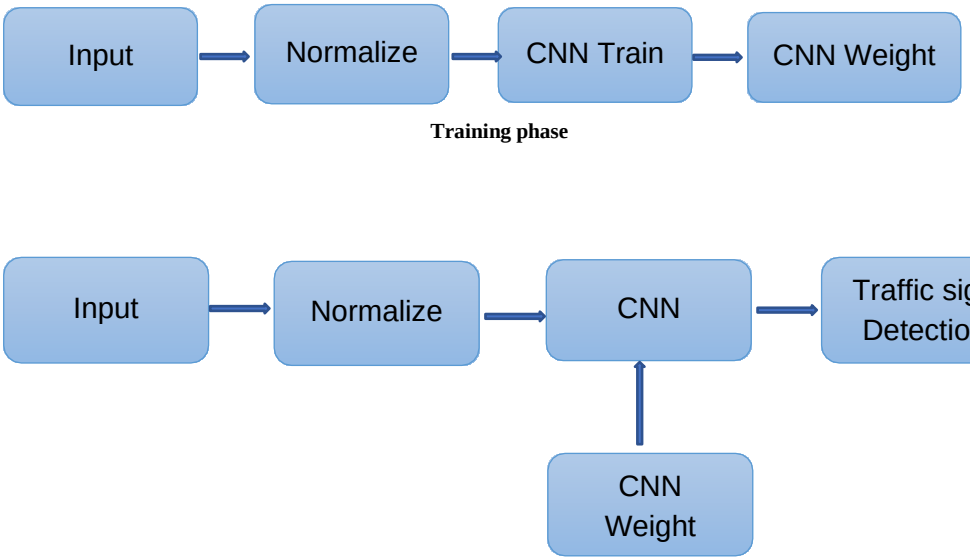


Figure 3.1: (b) Testing phase

3.1. 1 Proposed Model

The architecture of a **"Convolutional Neural Network"** consists of an input layer, some convolutional layers, a pooling layer, a flattening layer, some fully-connected layers, and an output layer. CNN is a multi-layered network that functions similarly to the human brain. Multiple neurons make each layer of CNN.

1. Convolution Layer:

This layer is the primary layer withinside the CNN and a main constructing block withinside the convolution process. It makes use of convolution to apprehend diverse functions in a given photo, i.e., it's miles used for function extraction with the assist of filters. It scans the complete pixel grid after which does a dot product. A filter, every now and then referred to as a kernel, is a function derived from numerous functions in an enter photo that we need to apprehend. For example, withinside the case of aspect detection, we'd have separate filters for curves, blur, and sprucing the photo. As we development deeper into the network, greater state-of-the-art features emerge as visible.

2. Pooling Layer:

This layer is used to downsample the features of the image. Reduces the dimensions of large images while preserving important details. Reducing the size of the image reduces the computational power and time required to process the image. Two types of pooling are available. Depending on the situation, you can use maximum or average pooling. Maximum pooling uses the maximum value in the feature map, and averaging takes the average of all pixels.

3. Activation Function:

An Activation Function is used in the convolution layer which transforms the given input into the required output that has a certain range. This layer introduces non-linear properties to the network so that the model can predict complex patterns in the data as in the case of images, audio, video, and, sound. It aids in determining which data should be processed further and which should not. The weighted sum of the inputs is used as the input to the activation function that produces the output. This step is important because a model without an activation function is a simple linear regression model with limited learning ability. The model here uses the ReLU activation function which replaces all the negative values from zero and if the value is positive then the output is itself the input. This function hasa certain range and is defined by the following equation.

Rectified Liner Unit: F(u) = max (0, u)

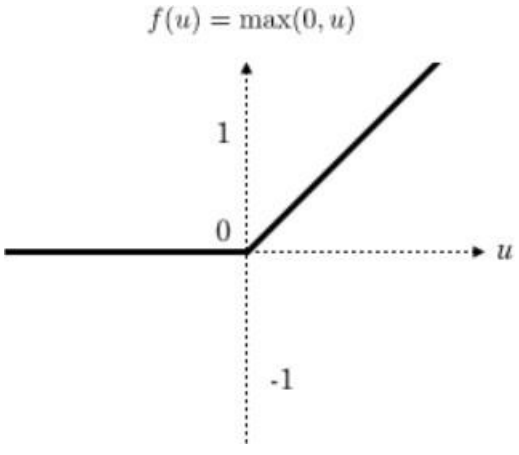


Figure 3.2: Rectified Liner Unit Activation Function

4. Dropout Layer:

The dropout layer is used to prevent the problem of overfitting and reduces the complexity of the model. The problem of overfitting occurs when the model tries to fit more data than what is actually required and tries to capture every detail fed. As a result, it captures noise and inaccurate data from the dataset hence the accuracy and performance of the model are reduced. An overfitted model does not function well.

The Dropout layer used with the convolution neural network is a mask that is used to nullify the contribution of some of the neurons towards the next layer and all the others are left unmodified.

If the Dropout layer is not present then the learning is influenced by the first batch of the training samples in a disproportionately higher manner. Hence the presence of dropout layer in the training of the CNN models is important.

The combination of the convolution layer, pooling layer, and dropout layer is called phase learning and is repeated multiple times to improve the training.

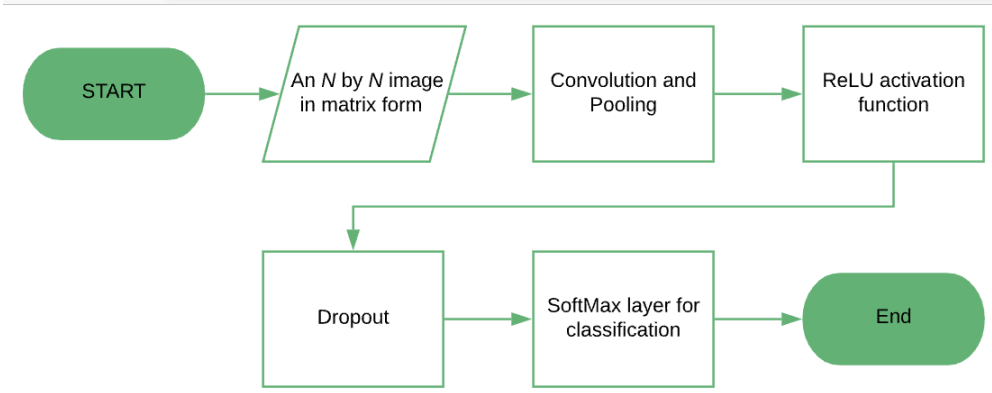


Figure 3.3 Using dropout with CNN

5. Flattening Layer:

Pooling layers are output in the form of 3D feature maps.The function of the flatteninglayer is to transform the 3D data from the pooling layer into a continuous one-dimensional vector. The output of the fattening layer is fed into the artificial neural network that is connectedto the convolution neural network that was built so far. Fully-connected layer for further processing.

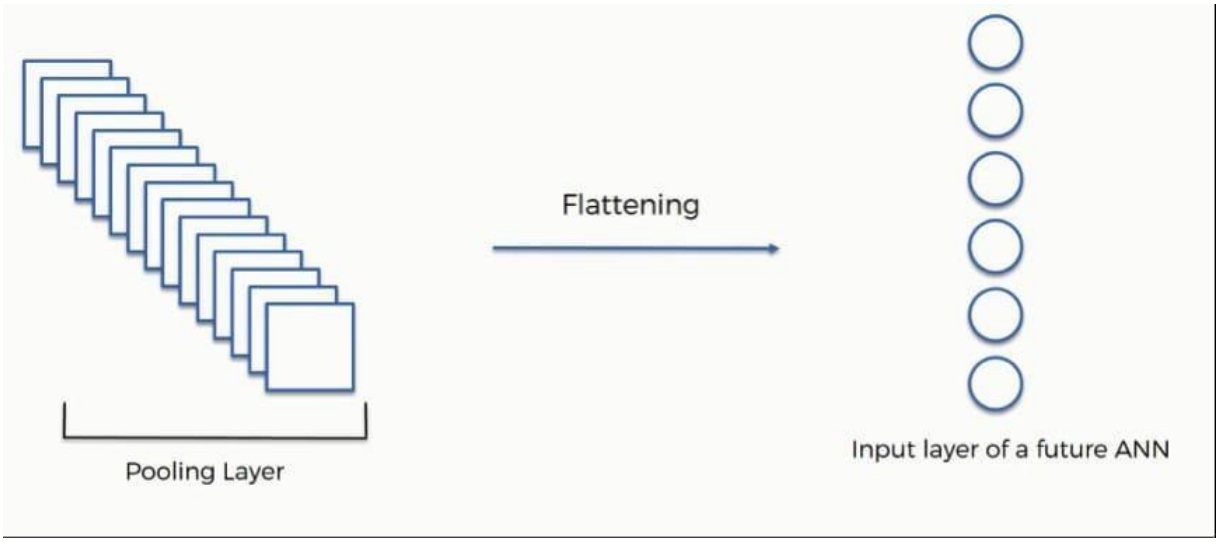


Figure 3.4 Flattening Process

6. Fully connected Layer:

A fully connected layer involves chaining an existing convolutional neural network to an artificial neural network. This step is called the fully connected step because a fully connected layer takes the place of the hidden layer in the Artificial Neural Network. Each neuron in the fully connected layer is connected to every other neuron in the next layer. It combines the features with extra attributes to get a more accurate classification prediction.

7. Dense Layer:

This is the last layer used as the output layer. There are as many nodes as there are classes used in the model. This model uses 43 classes, so here the number of nodes is 43. The high density layer uses the SoftMax activation function used for probability values from 0 to 1, so the model can predict which class has the highest probability.

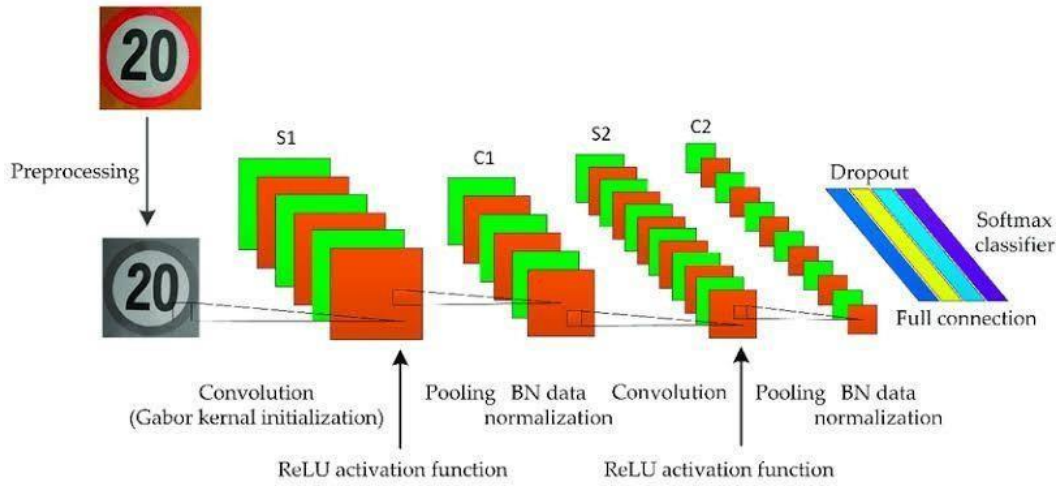


Figure 3.5 Working of CNN Model

3.1.2 Data Augmentation

Image enhancement is a way to modify an existing image to generate additional data for the model training process. In other words, it is a technique to artificially increase the datasets that can be used to train deep learning models. Deep learning neural network models trained with more data will be more powerful, and extended approaches can provide variations to images and enhance the ability of fitting models to generalize what they have learned to new images.

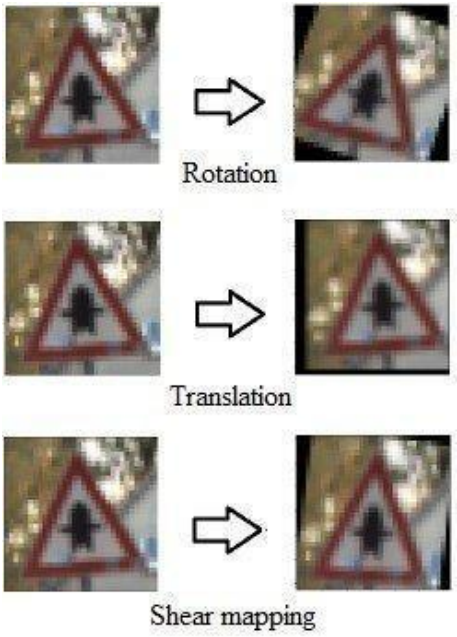


Figure 3.6 Data Augmentation

Data augmentation is a technique that helps in training the model in a way that the model in such a way that it results in more skillful models. Data augmentation provides variations in the existing images. Here the ImageDataGenerator class in the Keras library of python is used to do data augmentation. Using data augmentation different transformations are applied to the images in the dataset such as shearing, zoom in, flip, shift, rotation, and brightness to produce a wide range of variations in the images.

This algorithm randomly selects an image using only classes with less than 1000 images and performs one of the conversion operations(Figure 3.6). The resulting image will be added to the same class until the number of elements reaches a bias of 1000 images.

Figure 3.7 Result After Applying Data Augmentation

3.2 Requirement Analysis

- **Language used:** Python (3.10.0)
- **Libraries used:** NumPy (1.22.3)
  TensorFlow (2.9.0)
  Matplotlib (3.5.2)
  Keras (2.9.0)
  SkLearn
- **Platform used:** Jupyter Notebook

3.3 Dataset

Kaggle GTSRB dataset is used for training and testing. The data used in this project will be the GTSRB German traffic sign recognition benchmark. The German Traffic Sign Benchmark is a multi-class single image classification challenge held at the International Joint Session on Neural Networks (2011). The benchmark has the following characteristics.

- Contains over 40 classes
- Single image classification
- Over 50,000 images in large, realistic datasets

The GTSRB dataset contains Throughout the network, GTSRB and self-designed traffic sign data sets were used for trainingand testing. The dataset was further split into 34799 training, 4410 validation, and 12630 test images.

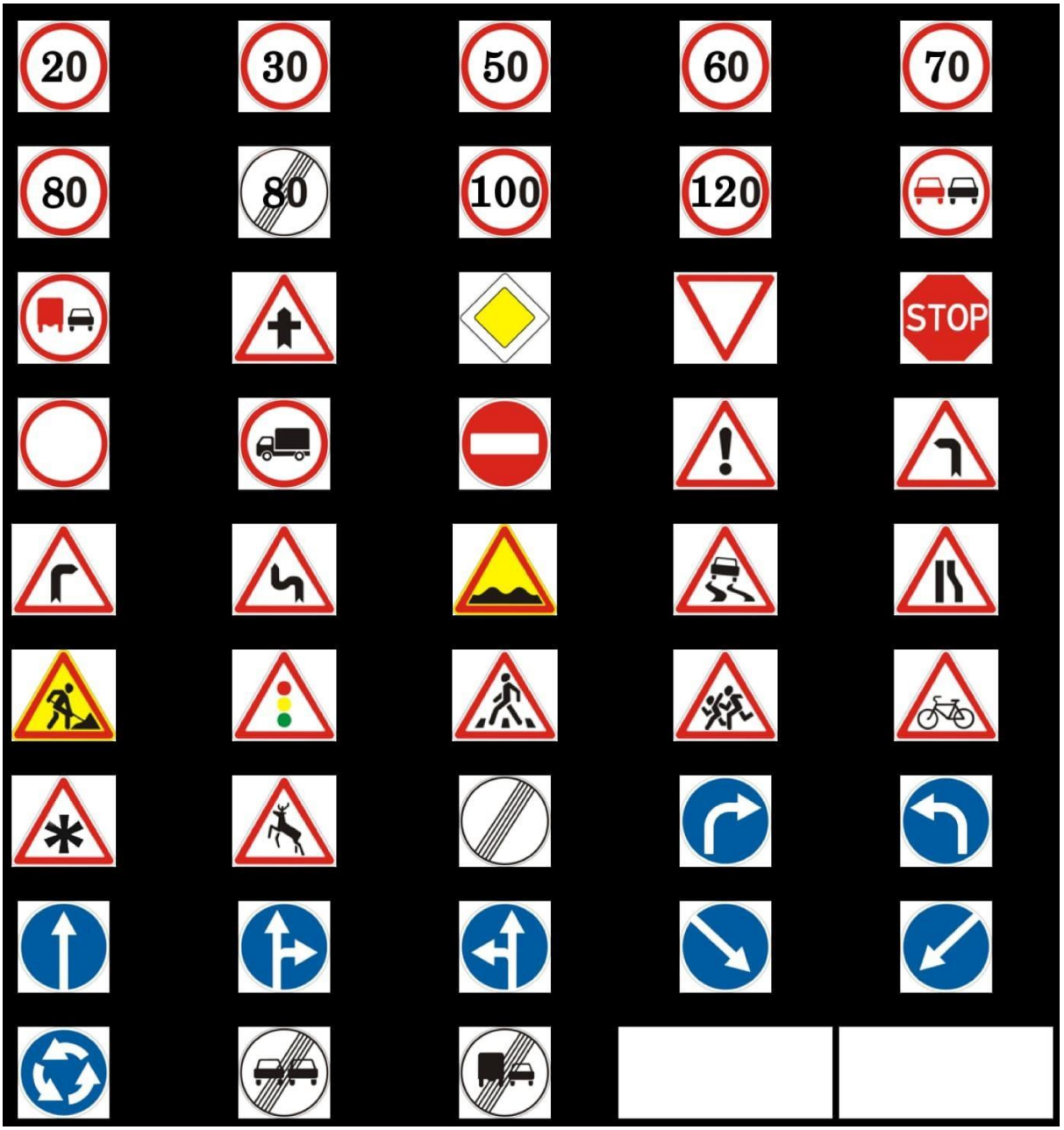


Figure 3.8 GTSRB Dataset

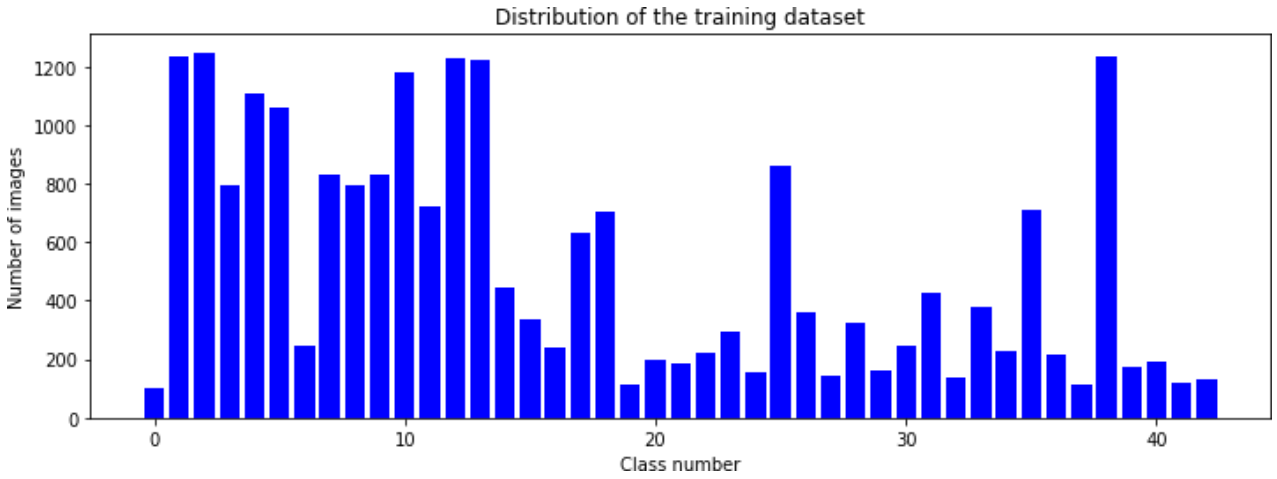


Figure 3.9 Distribution of the training dataset

3.4 Steps for Traffic Sign Classification

Convolutional neural networks are used to recognize and classify traffic signs. Input pre-processing is done before classification to eliminate noise, reduce complexity, and improve the accuracy of the method used. Traffic sign recognition is important in traffic sign recognition applications. Misidentified signs cannot be classified and recognized to warn the driver.

3.4.1 Image Processing

Gray Scale Conversion: When an image is taken it is in the RGB format, it consists of three channels for red, green, and blue which increases the complexity of the model. If we remove the redundant details from the images in some situations this will help us save space and reduce the computing complexity. One way to do this is to convert the color image to grayscale because only one channel is used. Grayscale conversion is performed because colors are not always used to identify and recognize images of multiple objects. Grayscale is sufficient to identify such artifacts. Color images contain more detail than black-and-white images, which can add unnecessary complexity and occupy more storage space. Enhance the input image with histogram equalization to make improvement in the contrast of the image, make the bright portion brighter and the dark portion darker, and then normalize the value between 0 and 1 instead of 0 to 255.

3.4.2 Detection

The purpose of traffic sign detection is to find the areas of interest (ROI) where a traffic sign is more likely to be discovered and to confirm the sign's presence hypothesis. Due to the vast scale of detection in a single image, the first detection phase of a traffic sign recognition system involves high expenses. Prior information on the sign location is meant to be trimmed to save space. ROI stands for Return on Investment. ROI uses the form of the traffic sign to locate it in the image.The backdrop image is detected as an undesirable signal and marked as a black pixel. A considerable amount of the image can be discarded based on these assumptions. Trafficsigns are made with a specific color and shape to make them more easily identifiable.

Traffic sign detection methods are inherently dependent on the nature of the data for which they were developed. The approaches that are followed in the detection stage have traditionally used Shape-based methods.

3.4.3 Classification

The detected traffic signs must be identified and categorized. CNN is mostly used to classify data. The GTSRB data set is used to train and test CNN, which is the most important part of the training and testing process. CNN is a multilayered network that functions similarly to the human brain. Multiple neurons make up each layer of CNN. Each neuron receives input to complete the task, and some actions and outputs are passed as input to the next neuron. The main purpose of the convolutional layer of a convolutional neural network is feature selection. When it comes to classification, CNNs are better than traditional machine learning.

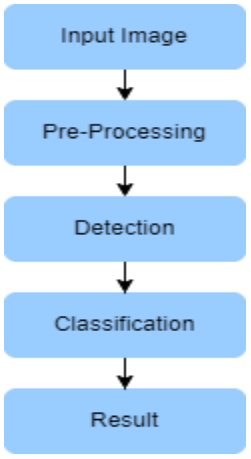


Figure 3.10 Steps of Traffic Sign Recognition

3.5 Flow chart

A Flowchart is the representation of the workflow of the process. A flowchart is defined as the diagrammatic representation of an algorithm or it can be defined as the step-by-step procedure for solving a problem. They are used for various purposes such as documenting, analyzing, designing or managing a program flow in various fields.

The flow chart shows the steps as different types of boxes and their order by connecting the boxes with arrows. This flowchart shows the workflow for the entire project.

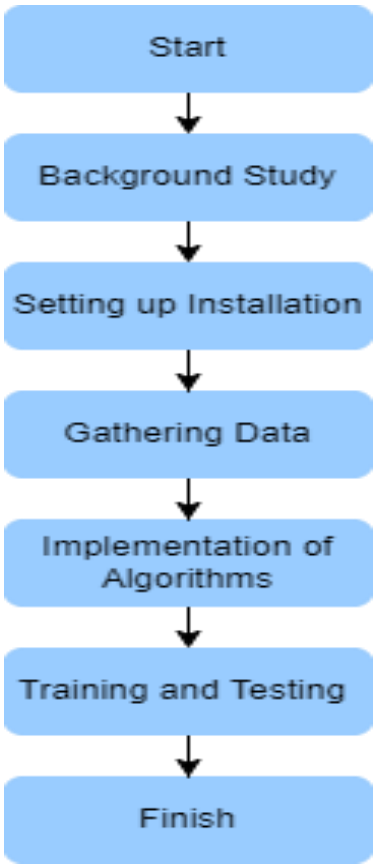


Figure 3.11 Flowchart of the overall process

3.6 Implementation

3.6.1 Designing a CNN Model

- A convolutional neural network (CNN) is a class of deep learning neural networks. CNN represents a major advance in image recognition.
- These are most commonly used to analyze visual images and often work behind the scenes of image classification.

- CNNs use very little pre-processing
- This means that they can learn the filters that have to be hand-made in other algorithms
- A CNN works by extracting features from images. This eliminates the need for manual feature extraction
- The features are not trained, They are learned, while the network is being trained on a set of images

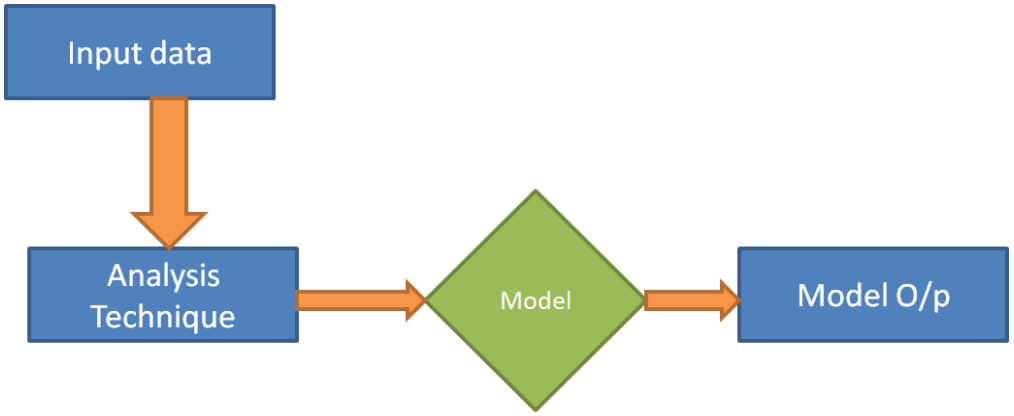


Figure 3.12 Basic Building block of model

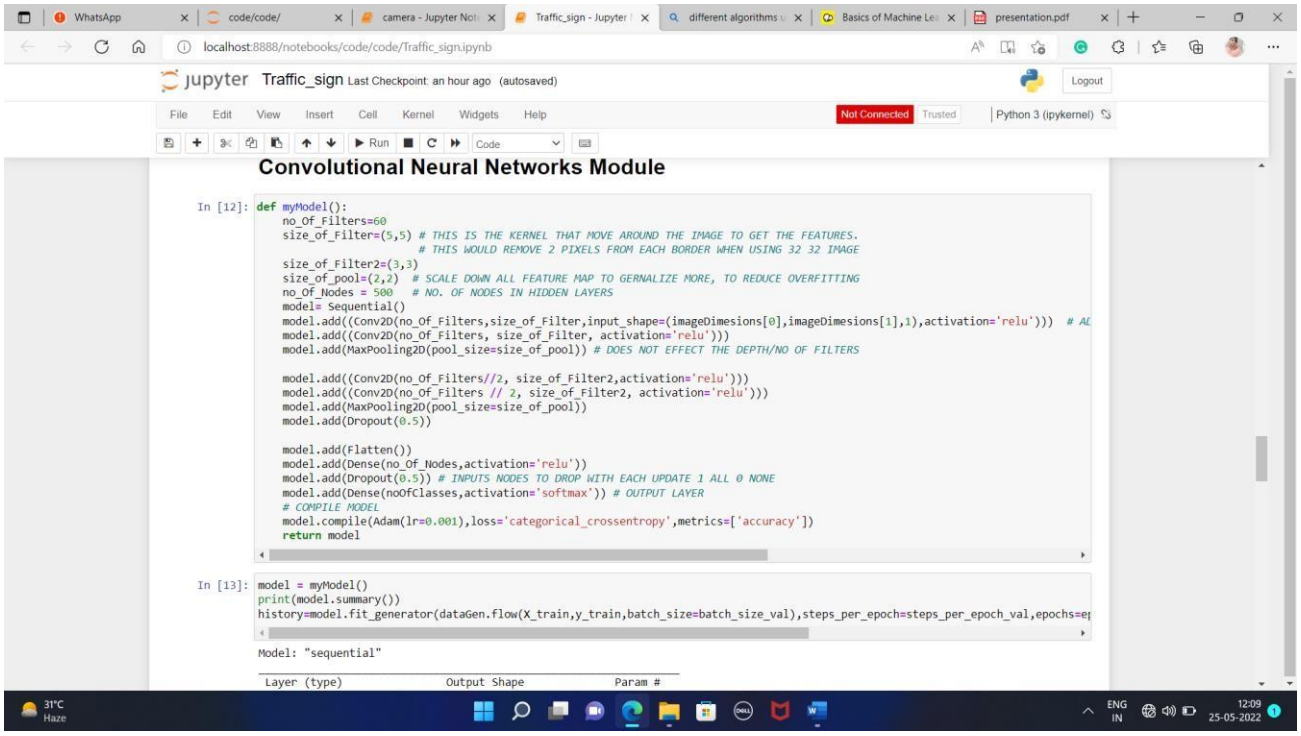
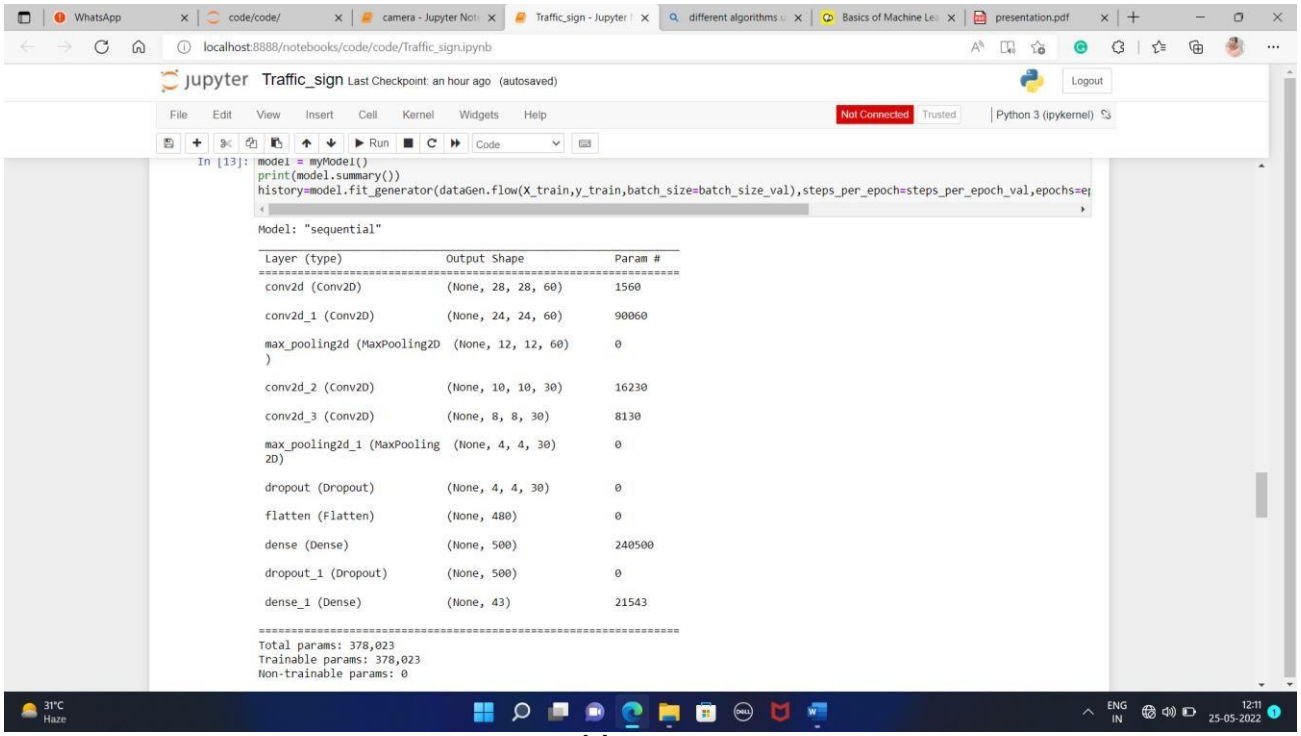


Figure 3.13 Implementation of CNN Model

3.6.1 Training a CNN Model



(a)

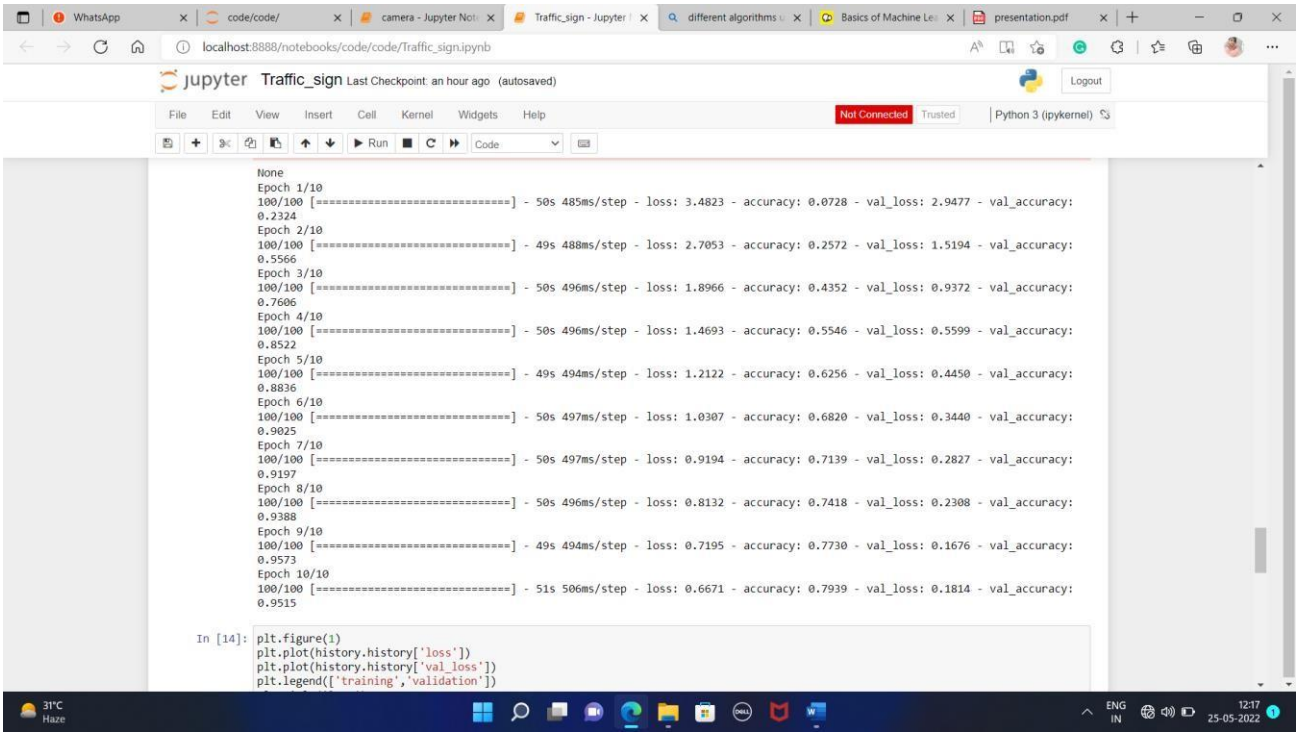


Figure 3.14 (b) Training of CNN Model

3.7 Activity Diagram

Activity diagrams are the unified modeling diagrams that are used to depict the flow of activities of the system. Here shows an activity diagram that describes the flow of activities in the project to get the desired results.

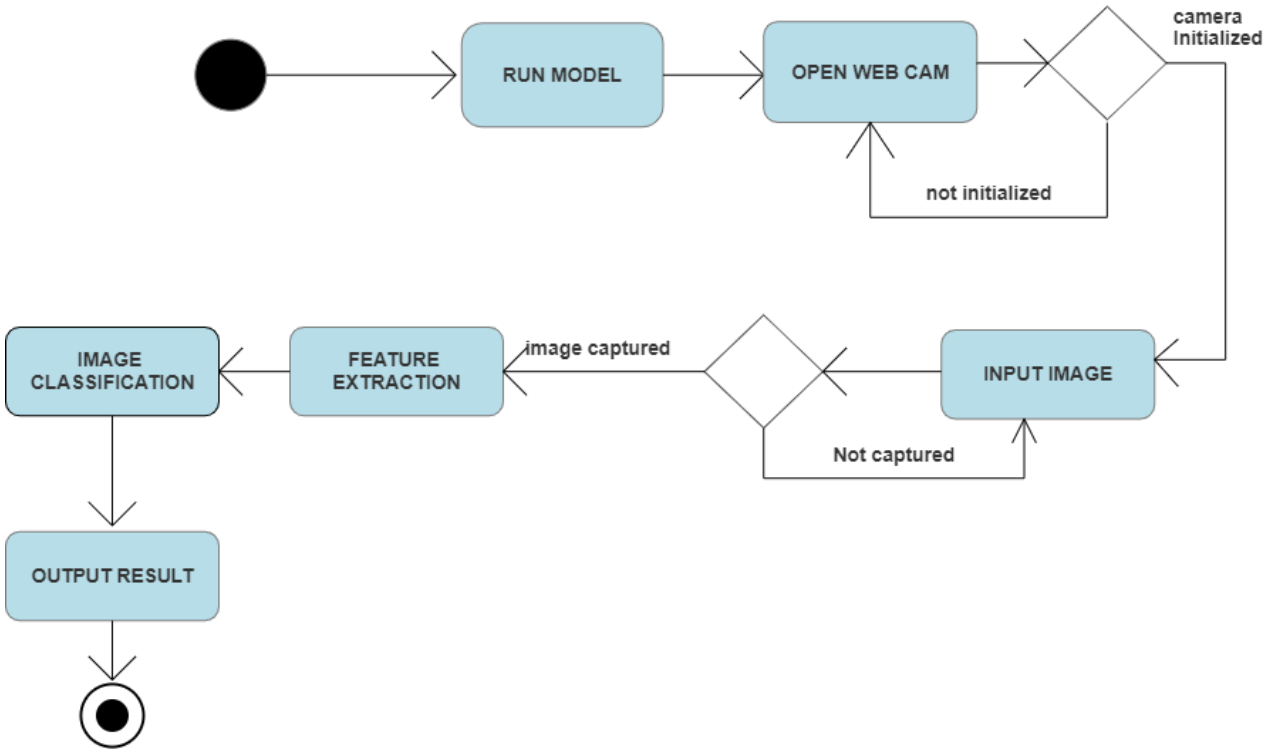


Figure 3.15 Activity diagram

CHAPTER 4
RESULT AND DISCUSSION

CNN architecture for Traffic Sign recognition as mentioned above is implemented by using Python and different libraries of python are being used for different purposes such as NumPy, Keras, TensorFlow, and OPENCV, Matplotlib, and skLearn. After creating the model architecture, training, and validating the model, use model.fit () to train the model. The stack size I tried is 100. And after 10 epochs, the accuracy was almost stable. The accuracy of the training dataset for the model is 97.8. Use matplotlib to plot the graph for accuracy and loss.

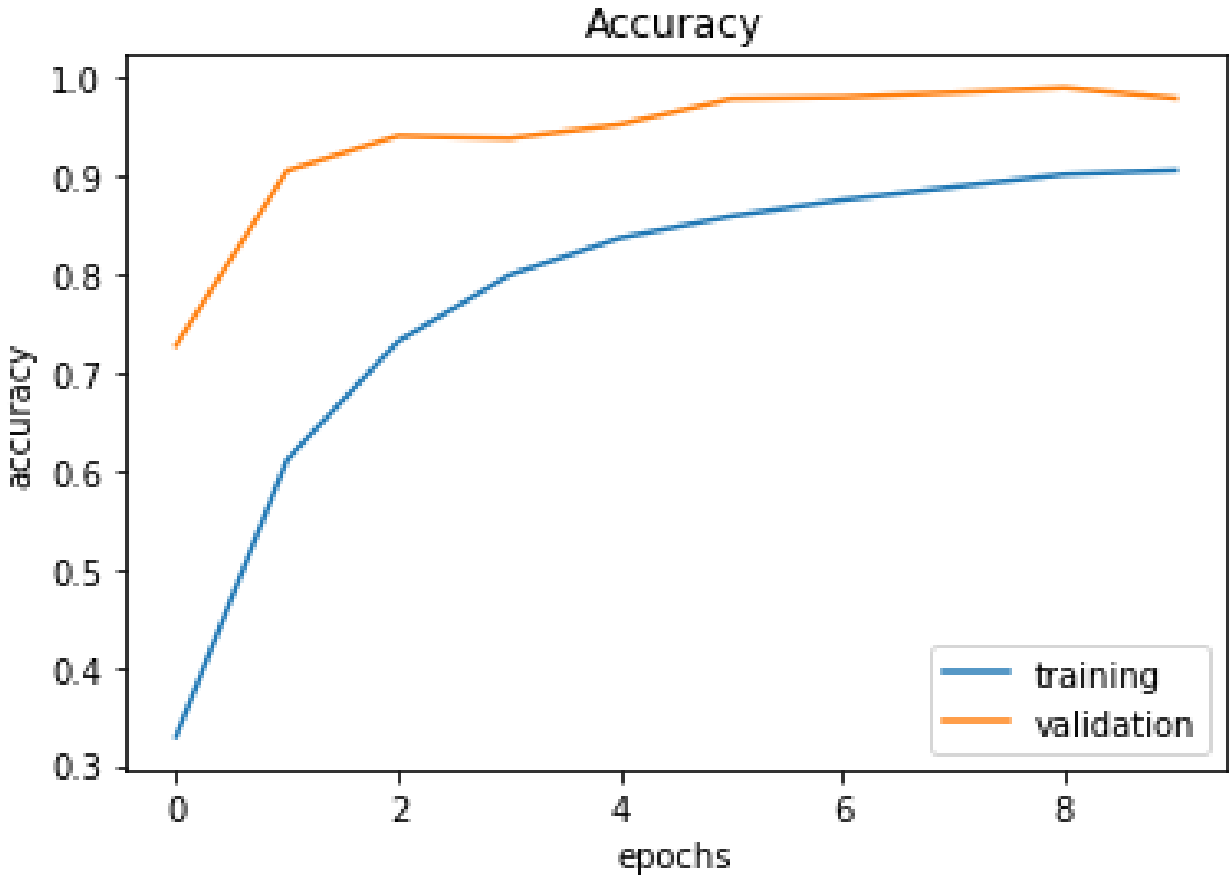


Figure 4.1: Accuracy graph of the model

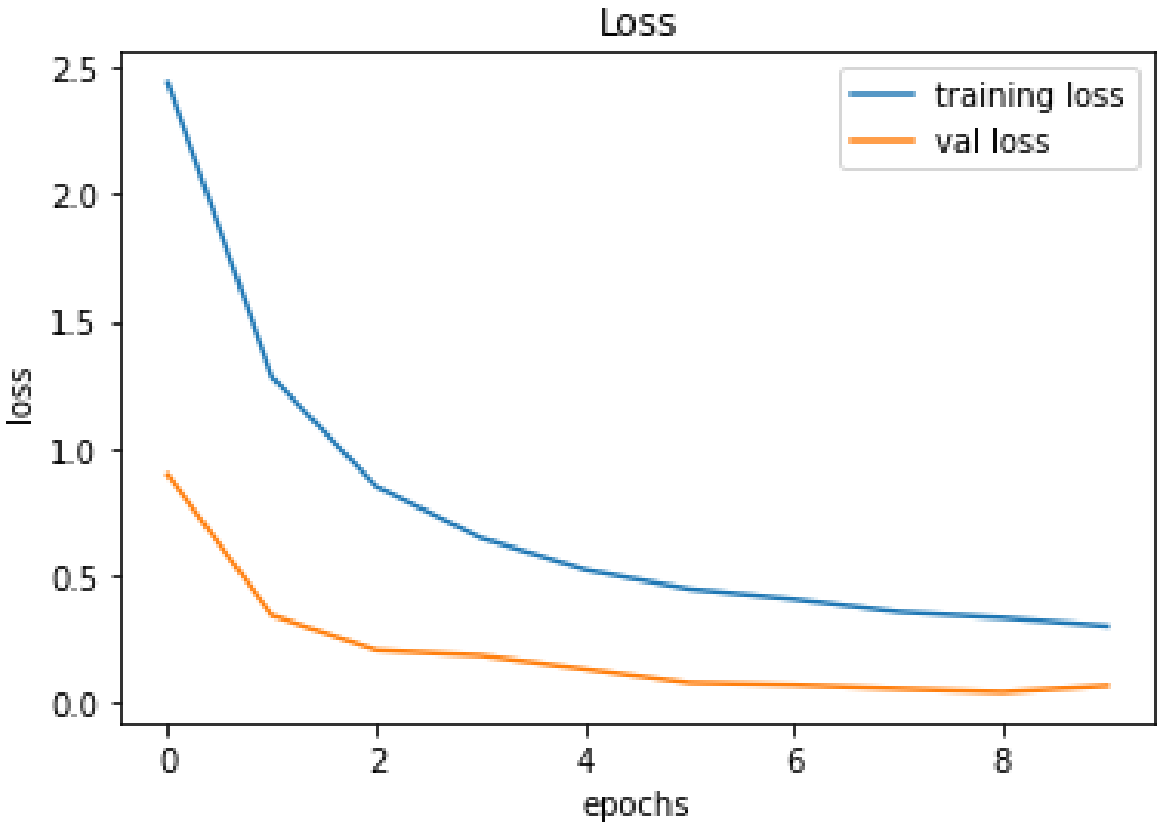


Figure 4.2: Loss graph of the model

CNN Classifier is then used to classify images taken from a webcam on Laptop. First, it uses OpenCV to recognize the characters in each frame of the webcam feed. The area of the image that contains the size is scaled to 32x32 and passed as input to the CNN. The network publishes a list of softmax scores for 43 classes of traffic signs. The character with the highest probability is displayed on the screen. Some classification results using a webcam are shown below:

Figure 4.3 show the snapshot taken from the laptop while doing testing on the model which shows traffic sign detected in the webcam belonging to class 28 among the 43 classes used in thisproject. Class 28 is children crossing and the accuracy it is giving is 70.12% which is an averageaccuracy. Here accuracy is depending on the number of images present in a particular class. Other snapshots are also displayed below which have a quite high accuracy giving good results.

Figure 4.3 Output is detected with class 28 which is children crossing

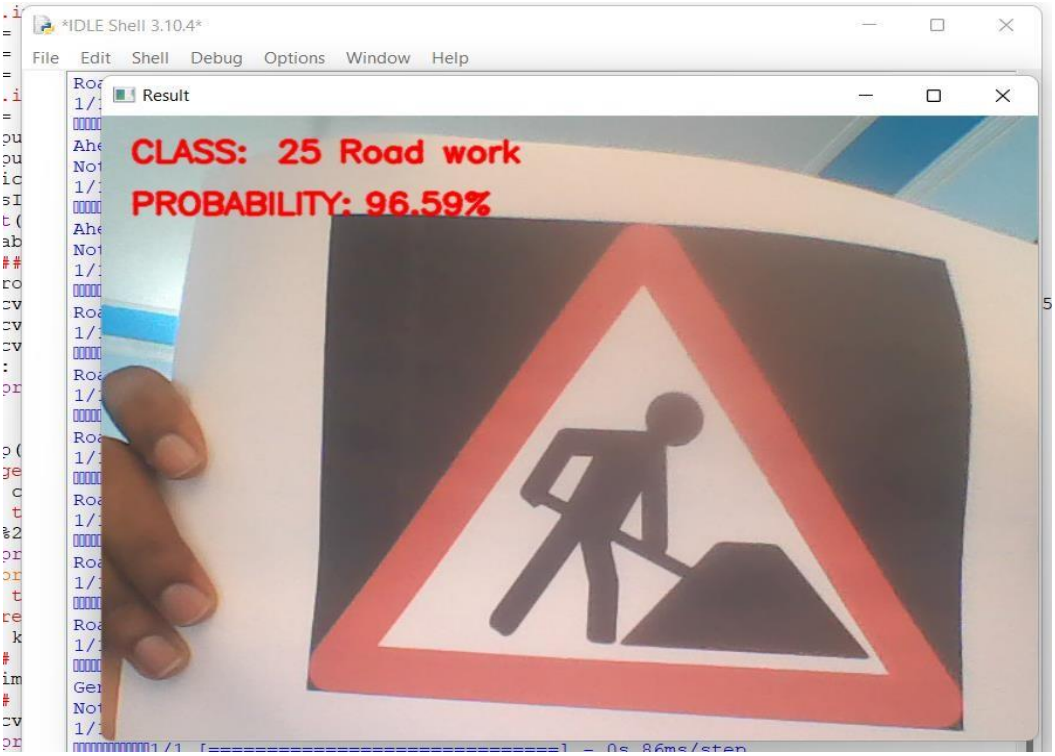


Figure 4.4 Output is detected with class 25 which is Road Work

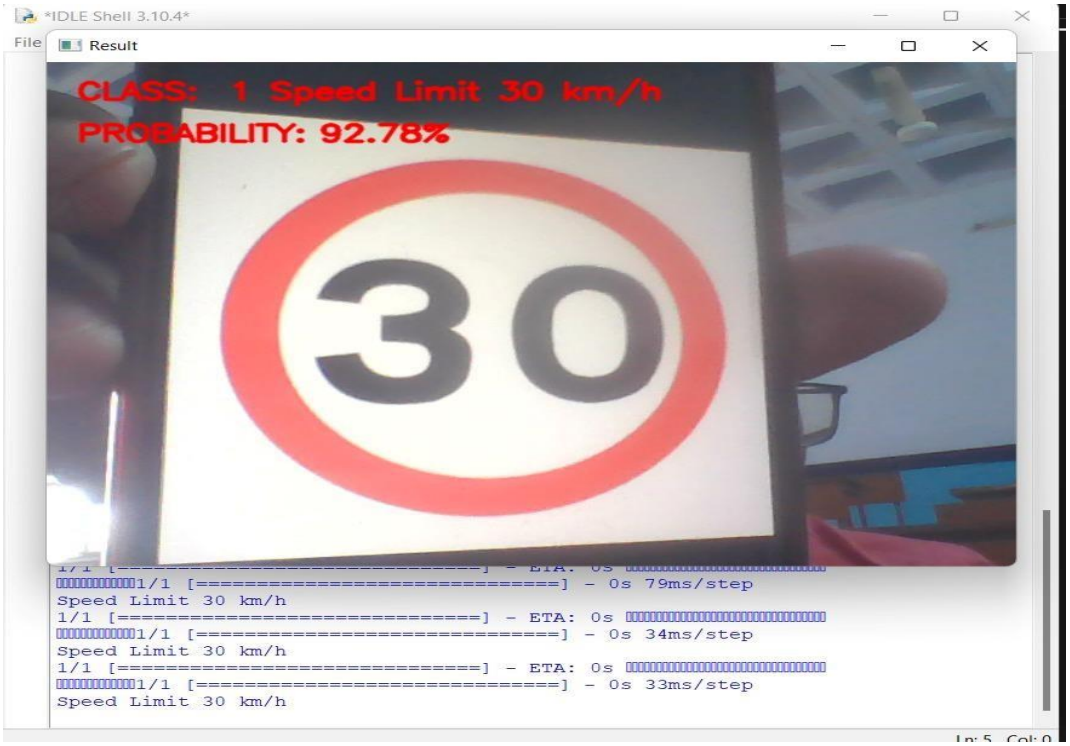


Figure 4.5 Output is detected with class 1 which is a speed limit of 30 km/h

Figure 4.6 Output is detected with class 5 which is a speed limit of 80 km/h

Figure 4.7 GUI of model

Figure 4.8 Upload an image to classify

Figure 4.9 Output is detected with class 38 which is keep right

Figure 4.10 Output is detected with class 14 which is stop

Figure 4.11 Output is detected with class 23 which is slippery road

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Conclusion

The proposed model is an effective method for classifying and recognizing traffic signs. Even though the traffic signs are designed in such a way that they are easily recognized by the human drivers but sometimes due to the weather conditions it is blurry, covered by snow, there may be heavy rainfall, there may be fog in the winters or they may be small in size, or improperly displayed, or rotated by certain angles. So, it becomes challenging to interpret the traffic signs correctly for the driving person. Introducing this mechanism helps in the detection and classification of traffic signs much more quickly and alerts us. Firstly the, preprocessing of the image is done. Secondly, traffic sign is detected based on the shape feature and here Convolution Neural Network architecture is used for traffic sign recognition. Finally, the visitors signal class and popularity are carried out primarily based totally on GTSRB. There are 43 classes of traffic signs that are tried to be recognized. In this project, a convolution neural network is implemented to classify traffic signs i.e., 20km, 30km, 40km, 50km, 60km, 70km, 80km, end of the 80 km, no passing, road work, children are crossing, General caution, slippery road, and many more classes are there. Favorable prediction and accurate recognition of traffic signs is achieved through continuous training and testing of network models. Experimental results show that the accurate recognition rate of traffic signs reaches an accuracy of 98.8%. The proposed algorithm has better accuracy, better real-time performance, stronger generalization ability, and higher training efficiency than other algorithms. The accurate detection rate and average processing time are greatly improved. The same concept of traffic sign recognition is used in Automatic Driving cars to provide them instructions while driving on the road safely. It also helps to understand different traffic signs and can be helpful to blind persons to detect and classify the traffic signs using this concept and then provide them with voice instructions to drive on the roads safely. In this way, we canensure the safety of the driver while driving on the roads to prevent accidents, deaths and damage to the vehicle.

5.2 Future work

It is important to note that there is no specific formula for building a neural network that is guaranteed to work. Achieving acceptable verification accuracy requires different network architectures and a lot of trial and error for different challenges. For this reason, neural networks are commonly referred to as "black-box algorithms". However, there are some places that can be improved.

- Number and composition of convolution layers
- Number and placement of high-density layers
- Dropout rate in dense layers

In the future, the comprehensiveness and error prevention detection of traffic sign recognition algorithms can be further optimized and improved to take advantage of the overall performance of the algorithm. It may also include integrating other algorithms into a CNN model such as ANN (Artificial Neural Network) to show that the accuracy provided by the CNN is more accurate is as compared to the Artificial Neural Network Algorithm.

Also in future work, it can include the Graphical user interface that can be provided as the front end of the model. This, can include two options either the user can capture the live image using the webcam and it will detect and classify the image captured or the user can choose the image from the local device that can be classified and the model will provide the predicted output to the user. It can also include voice instruction to make the system more accurate and reliable for the safety of the driver and the vehicle.

References

[1] Ying Sun and Pingshu Geˆand Dequan Liu,"Traffic Sign Detection and Recognition Based on Convolutional Neural Network,", IEEE. DOI:10.1109/CAC48633.2019.8997240,2019

[2] Md Tarequl Islam,"Traffic sign detection and recognition based on convolutional neural networks", International Conference on Advances in Computing, Communication and Control (ICAC3),DOI: 10.1109/ICAC347590.2019.90367842019.

[3] Vaibhavi Golgire," Traffic Sign Recognition using Machine Learning: A Review", International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181, 05, May-2021.

[4] Wenhui Li, Daihui Li and Shangyou Zeng, "Traffic sign Recognition with a small Convolutional Neural Network", IOP Publishing Ltd, 2019.

[5] Zhilong He, Zhongjun Xiao, Zhiguo Yan, "Traffic Sign Recognition based on Convolutional Neural Network", Chinese Automation Congress, IEEE 2020.

[6] Yanzhao Zhu, Wei Qi Yan, "Traffic Sign Recognition based on Deep Learning", Auckland University of Technology, CBD, 2022.

[7] Prashengit Dhar , Md. Zainal Abedin , Tonoy Biswas and Anish Datta,"Traffic Sign Detection- A New Approach and Recognition Using Convolutional Neural Network" ,IEEE Region 10 Humanitarian Technology Conference (R10-HTC),DOI: 10.1109/R10-HTC.2017.8288988,2017.

```
import tkinter as tk
from tkinter import filedialog
from tkinter import *
from PIL import ImageTk, Image

import numpy

from keras.models import load_model
import warnings
window = Tk()

model = load_model('Trafic_signs_model.h5')
warnings.filterwarnings("ignore", category=DeprecationWarning)

#dictionary to label all traffic signs class.
classes = { 1:'Speed limit (20km/h)',
            2:'Speed limit (30km/h)',
            3:'Speed limit (50km/h)',
            4:'Speed limit (60km/h)',
            5:'Speed limit (70km/h)',
            6:'Speed limit (80km/h)',
            7:'End of speed limit (80km/h)',
            8:'Speed limit (100km/h)',
            9:'Speed limit (120km/h)',
           10:'No passing',
```

```
           19:'General caution',
           20:'Dangerous curve left',
           21:'Dangerous curve right',
           22:'Double curve',
           23:'Bumpy road',
           24:'Slippery road',
           25:'Road narrows on the right',
           26:'Road work',
           27:'Traffic signals',
           28:'Pedestrians',
           29:'Children crossing',
           30:'Bicycles crossing',
           31:'Beware of ice/snow',
           32:'Wild animals crossing',
           33:'End speed + passing limits',
           34:'Turn right ahead',
           35:'Turn left ahead',
           36:'Ahead only',
           37:'Go straight or right',
           38:'Go straight or left',
           39:'Keep right',
           40:'Keep left',
           41:'Roundabout mandatory',
           42:'End of no passing',
           43:'End no passing veh > 3.5 tons' }
```

```
window.geometry('600x500')
window.title('Traffic sign Recognition')

window.configure(background='#1e3e64')

heading = Label(window, text="Traffic sign Recognition",padx=220, font=('Verdana',20,'bold'))
heading.configure(background='#143953',foreground='white')
heading.pack()

sign = Label(window)
sign.configure(background='#1e3e64')

value = Label(window,font=('Helvetica',15,'bold'))
value.configure(background='#1e3e64')

def classify(file_path):
    global label_packed
    image = Image.open(file_path)
    image = image.resize((30,30))
    image = numpy.expand_dims(image, axis=0)
    image = numpy.array(image)
    #print(image.shape)
    pred = model.predict(image)
    pred1 =pred
    pred1 = numpy.max(pred)

    pred = numpy.argmax(pred)
    print(pred)
    sign = classes[pred+1]
```

```
def show_cb(file_path):
    classify_b=Button(window,text="Classify Image",command=lambda: classify(file_path),padx=20,pady=5)
    classify_b.configure(background='#147a81', foreground='white',font=('arial',10,'bold'))
    classify_b.place(relx=0.6,rely=0.80)

def uploader():
    try:
        file_path = filedialog.askopenfilename()
        uploaded = Image.open(file_path)
        uploaded.thumbnail(((window.winfo_width()/2.25),(window.winfo_height()/2.25)))
        im = ImageTk.PhotoImage(uploaded)

        sign.configure(image=im)
        sign.image=im
        value.configure(text='')
        show_cb(file_path)
    except:
        pass

upload = Button(window,text="Upload an image",command=uploader,padx=10,pady=5)
upload.configure(background='#e8d08e', foreground='#143953',font=('arial',10,'bold'))
upload.pack()
upload.place(x=100, y=400)

sign.pack()
sign.place(x=230,y=100)
```