MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Project Report

on

Traffic Signs Recognition

Submitted By:

Surbhi Kumari 0901CS191127

Sneha Nigam 0901CS191120

Faculty Mentor:

Mr. Mir Shahnawaz Ahmad

Assistant Professor , Computer Science and Engineering

Submitted To**:**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
GWALIOR - 474005 (MP) est. 1957
MAY-JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Project Report

on

Traffic Signs Recognition

A project report submitted in partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

Surbhi Kumari 0901CS191127

Sneha Nigam 0901CS191120

Faculty Mentor:

Mr. Mir Shahnawaz Ahmad

Assistant Professor, Computer Science and Engineering

Submitted To:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
GWALIOR - 474005 (MP) est. 1957
MAY-JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that **Surbhi Kumari** (0901CS191127) has submitted the project report titled "**Traffic Signs Recognition"** under the mentorship of **Mr. Mir Shahnawaz Ahmed**, in partial fulfillment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

Mr. Mir Shahnawaz Ahmad
Faculty Mentor
Assistant Professor,
Computer Science and Engineering

Dr. Manish Dixit
Professor and Head,
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that **Sneha Nigam** (0901CS191120) has submitted the project report titled "**Traffic Signs Recognition"** under the mentorship of **Mr. Mir Shahnawaz Ahmed**, in partial fulfillment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

Mr. Mir Shahnawaz Ahmad
Faculty Mentor
Assistant Professor,
Computer Science and Engineering

Dr. Manish Dixit
Professor and Head,
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

DECLARATION

I hereby declare that the work being presented in this project report, for the partial fulfillment of requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Mr. Mir Shahnawaz Ahmad, Assistant Professor**, Computer Science and Engineering

I declare that I have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.

Surbhi Kumari
0901CS191127
3RD Year,
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

DECLARATION

I hereby declare that the work being presented in this project report, for the partial fulfillment of requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Mr. Mir Shahnawaz Ahmad, Assistant Professor**, Computer Science and Engineering

I declare that I have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.

Sneha Nigam
0901CS191120
3RD Year,
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary/interdisciplinary project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Department of Computer Science and Engineering, for allowing** me to explore this project. I humbly thank **Dr. Manish Dixit**, Professor and Head, Department of Computer Science and Engineering, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance of **Mr. Mir Shahnawaz Ahmad, Assistant Professor** , Computer Science and Engineering for his continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

Surbhi Kumari
0901CS191127
3RD Year,
Computer Science and Engineering

Sneha Nigam
0901CS191120
3RD Year,
Computer Science and Engineering

ABSTRACT

Nowadays, there is a lot of attention being given to the ability of the car to drive itself. One of the many important aspects for a self- driving car is the ability for it to detect traffic signs in order to provide safety and security for the people not only inside the car but also outside of it. The traffic environment consists of different aspects whose main purpose is to regulate flow of traffic, make sure each driver is adhering to the rules so as to provide a safe and secure environment to all the parties concerned. .

In this era of Artificial Intelligence, humans are becoming more dependent on technology. With the enhanced technology, multinational companies like Google, Tesla, Uber, Ford, Audi, Toyota, Mercedes-Benz, and many more are working on automating vehicles. They are trying to make more accurate autonomous or driverless vehicles. You all might know about self-driving cars, where the vehicle itself behaves like a driver and does not need any human guidance to run on the road. This is not wrong to think about the safety aspects—a chance of significant accidents from machines. But no machines are more accurate than humans. Researchers are running many algorithms to ensure 100% road safety and accuracy. One such algorithm is Traffic Sign Recognition that we talk about in this blog.

When you go on the road, you see various traffic signs like traffic signals, turn left or right, speed limits, no passing of heavy vehicles, no entry, children crossing, etc., that you need to follow for a safe drive. Likewise, autonomous vehicles also have to interpret these signs and make decisions to achieve accuracy. The methodology of recognizing which class a traffic sign belongs to is called Traffic signs classification.

Keyword: Artificial Intelligence, Traffic Signs, Convolutional Neural Networks, Deep Learning

सार

आजकल, कार की खुद को चलाने की क्षमता पर बहुत ध्यान दिया जा रहा है। सेल्फ ड्राइविंग कार के लिए कई महत्वपूर्ण पहलुओं में से एक यह है कि न केवल कार के अंदर बल्कि इसके बाहर भी लोगों को सुरक्षा और सुरक्षा प्रदान करने के लिए ट्रैफिक संकेतों का पता लगाने की क्षमता है। यातायात के वातावरण में विभिन्न पहलू होते हैं जिनका मुख्य उद्देश्य यातायात के प्रवाह को विनियमित करना है, सुनिश्चित करें कि प्रत्येक चालक नियमों का पालन कर रहा है ताकि सभी संबंधित पक्षों को एक सुरक्षित और सुरक्षित वातावरण प्रदान किया जा सके। . आर्टिफिशियल इंटेलिजेंस के इस दौर में इंसान टेक्नोलॉजी पर ज्यादा निर्भर होता जा रहा है। उन्नत तकनीक के साथ, Google, टेस्ला, उबेर, फोर्ड, ऑडी, टोयोटा, मर्सिडीज-बेंज जैसी बहुराष्ट्रीय कंपनियां और कई अन्य कंपनियां स्वचालित वाहनों पर काम कर रही हैं। वे अधिक सटीक स्वायत्त या चालक रहित वाहन बनाने की कोशिश कर रहे हैं। आप सभी सेल्फ-ड्राइविंग कारों के बारे में जानते होंगे, जहां वाहन खुद एक ड्राइवर की तरह व्यवहार करता है और उसे सड़क पर चलने के लिए किसी मानवीय मार्गदर्शन की आवश्यकता नहीं होती है। सुरक्षा पहलुओं के बारे में सोचना गलत नहीं है - मशीनों से महत्वपूर्ण दुर्घटनाओं की संभावना। लेकिन कोई भी मशीन इंसानों से ज्यादा सटीक नहीं होती। 100% सड़क सुरक्षा और सटीकता सुनिश्चित करने के लिए शोधकर्ता कई एल्गोरिदम चला रहे हैं। ऐसा ही एक एल्गोरिथम है ट्रैफिक साइन रिकग्निशन जिसके बारे में हम इस ब्लॉग में बात करते हैं। जब आप सड़क पर जाते हैं, तो आपको ट्रैफिक सिग्नल, बाएं या दाएं मुड़ने, गति सीमा, भारी वाहनों का नो पासिंग, नो एंट्री, बच्चों के क्रॉसिंग आदि जैसे विभिन्न ट्रैफिक संकेत दिखाई देते हैं, जिनका आपको एक सुरक्षित ड्राइव के लिए पालन करने की आवश्यकता है। इसी तरह, स्वायत्त वाहनों को भी इन संकेतों की व्याख्या करनी होती है और सटीकता प्राप्त करने के लिए निर्णय लेने होते हैं। ट्रैफिक साइन किस वर्ग से संबंधित है, यह पहचानने की पद्धति को ट्रैफिक साइन वर्गीकरण कहा जाता है।

कीवर्ड: आर्टिफिशियल इंटेलिजेंस, ट्रैफिक साइन्स, कन्वेन्शनल न्यूरल नेटवर्क्स, डीप लर्निंग

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1: PROJECT OVERVIEW

1.1 Introduction

In this era of Artificial Intelligence, humans are becoming more dependent on technology. With the enhanced technology, multinational companies like Google, Tesla, Uber, Ford, Audi, Toyota, Mercedes-Benz, and many more are working on automating vehicles. They are trying to make more accurate autonomous or driverless vehicles. You all might know about self-driving cars, where the vehicle itself behaves like a driver and does not need any human guidance to run on the road. This is not wrong to think about the safety aspects—a chance of significant accidents from machines. But no machines are more accurate than humans. Researchers are running many algorithms to ensure 100% road safety and accuracy. One such algorithm is Traffic Sign Recognition that we talk about in this blog.

When you go on the road, you see various traffic signs like traffic signals, turn left or right, speed limits, no passing of heavy vehicles, no entry, children crossing, etc., that you need to follow for a safe drive. Likewise, autonomous vehicles also have to interpret these signs and make decisions to achieve accuracy. The methodology of recognizing which class a traffic sign belongs to is called Traffic signs classification.

The Traffic signs recognition system processes a traffic sign image extracted from the road scene. Eventually, it should classify that sign into one of 43 categories. In order to make it happen, a Convolutional Neural Network is applied as a type of Deep Neural Network, being trained with 51,888 images beforehand.

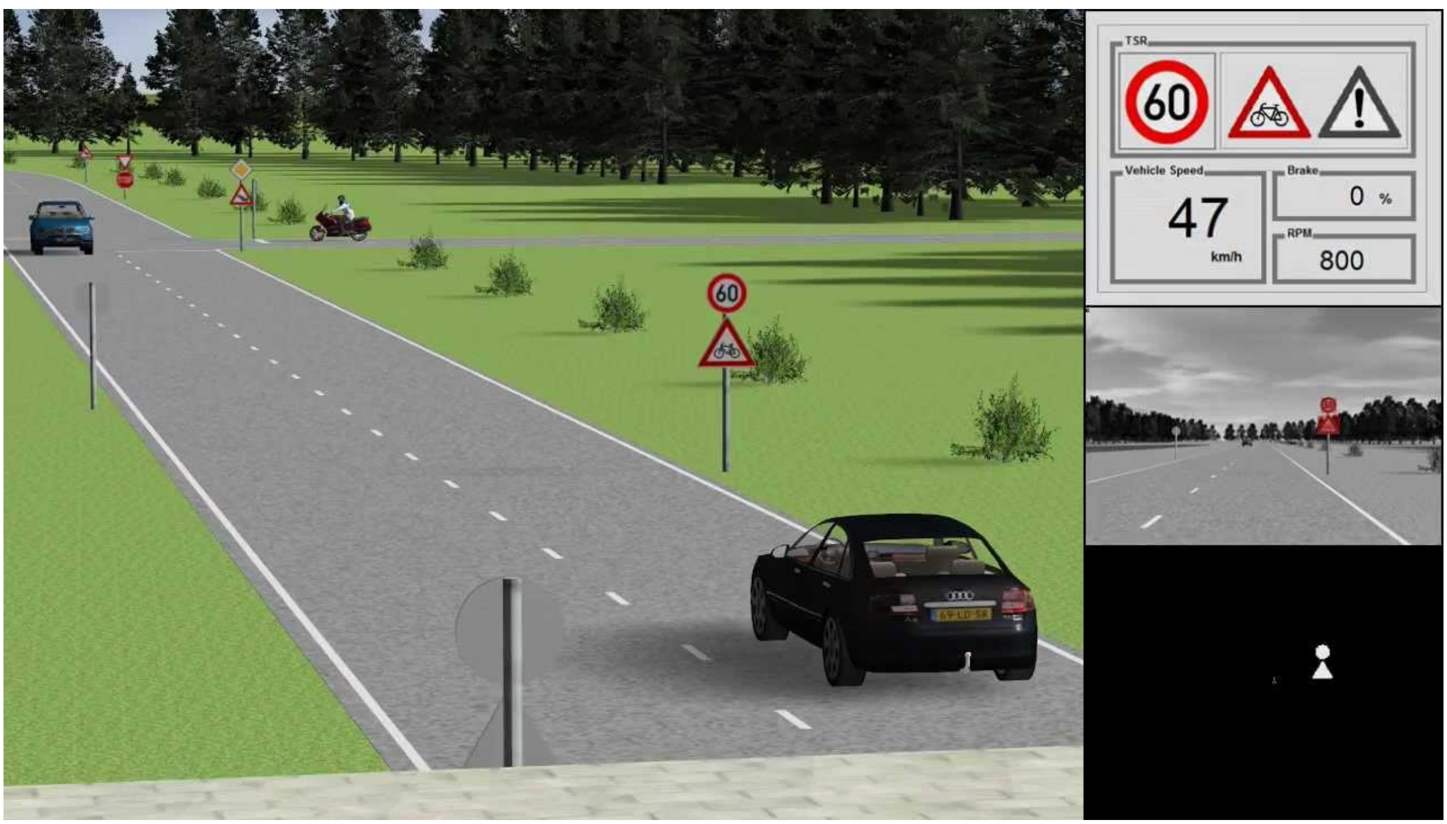

Fig No. 1.1 - Sample of Traffic sign Recognition System

Description	Traffic Sign	Description	Traffic Sign
Turn Left		Danger	
Stop		40km/h	
Only Left		30km/h	
Only Right		Turn Right	
Road Merges Ahead		Pedestrian	
Speed Breaker		Bike	

Fig No. 1.2 – Traffic Sign with Labels

1.2 Objectives And Scope

The use of traffic sign recognition system is very challenging. Since rain, fog, snow etc.affect the whole system. Another thing is the light variation i.e., shadows, sun, clouds etc. The geometrical shape of the object and the perspective is also a big concern. So, we must be able tocome up with a system which can work under light variation and geometrical transformation of the objects in a scene. If we narrow down the scope, we are particularly interested in detection of these signs under low light condition. In short, our objective is to make the system very accurate and efficient.

The goal of the Traffic Sign Recognition project is -

- To build a Deep Neural Network (DNN) which is used to classify traffic signs.
- We should train the model so it can decode traffic signs from natural images using the German Traffic Sign Dataset.
- This data should be firstly preprocessed in order to maximize the model performance.
- After choosing model architecture, fine tuning and training, the model will be tested on new images of traffic signs found on the web.
- To design and construct a computer based system which can automatically detect the road signs so as to provide assistance to the user or the machine so that they can take appropriate actions.

1.3 Project Overview

The project features are divided into four parts which are as follows:

- Explore the dataset
- Build a CNN model
- Train and validate the model
- Test the model with test dataset

1.4 Feasibility

1.4.1 Operational Feasibility

In Operational Feasibility the degree of providing service to requirements is analyzed along with how easy the product will be to operate and maintain after deployment. Along with this other operational scopes are determining usability of product, Determining suggested solution by software development team is acceptable or not etc.
The project is feasible in terms of operations as it can be implemented anywhere with internet connectivity and system to process.

1.4.2 Economic Feasibility

In the Economic Feasibility study, the cost and benefit of the project are analyzed. This means under this feasibility study a detailed analysis is carried out of what will be the cost of the project for development which includes all required costs for final development like hardware and software resources required, design and development cost and operational cost and so on. After that, it is analyzed whether the project will be beneficial in terms of finance for the organization or not.
The project has an economical constraint as the API with more number of request and good internet connectivity require more budget.

1.4.3 Legal Feasibility

In Legal Feasibility study project is analyzed from a legal point of view. This includes analyzing barriers of legal implementation of project, data protection acts or social media laws, project certificate, license, copyright etc. Overall it can be said that Legal Feasibility Study is a study to know if proposed project conform to legal and ethical requirements. The project is feasible legally.

1.5 System Requirements

Windows Based Requirements:

- Computers running Microsoft Windows must meet the following minimum hardware and software requirements.
- Microsoft Windows: 7/8/10/11
- 4 GB RAM minimum, 8 GB RAM recommended
- 1GB of available disk space minimum
- 1280 * 800 minimum screen resolution
- Software Requirement: Python 3.10.4
- Hardware Requirement: Laptop/Computer
- Internet Connectivity

Chapter 2: LITERATURE REVIEW

The first research on traffic sign recognition can be traced back to 1987; Akatsuka and Imai [2] attempted to make an early traffic sign recognition system. A system capable of automatic recognition of traffic sign could be used as assistance for drivers, alerting them about the presence of some specific sign (e.g. a one-way street) or some risky situation (e.g. driving at a higher speed than the maximum speed allowed). It also can be used to provide the autonomous unmanned some specific designed signs. Generally, the procedure of a traffic sign recognition system can be roughly divided of two stages namely detection and classification.

Chapter 3: PRELIMINARY DESIGN

3.1 Dataflow Diagram

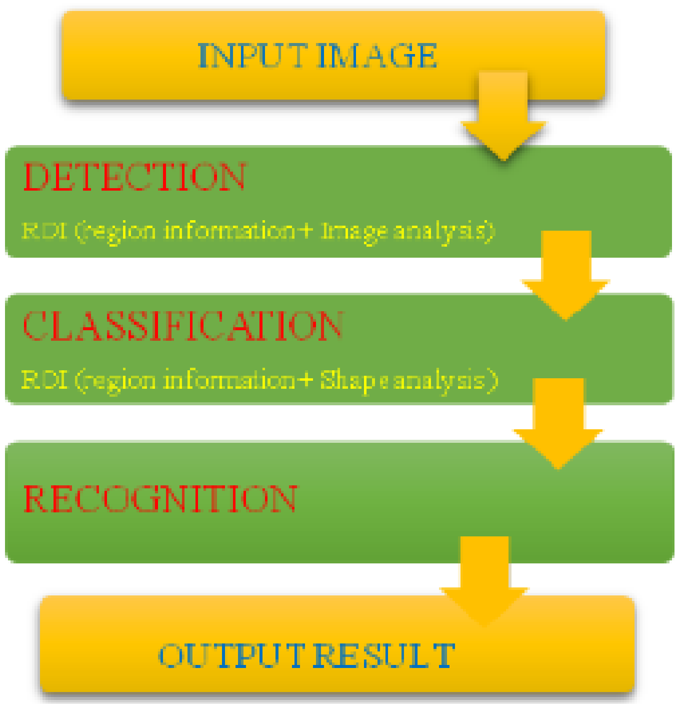

Fig No. 3.1 – Flowchart of Proposed System

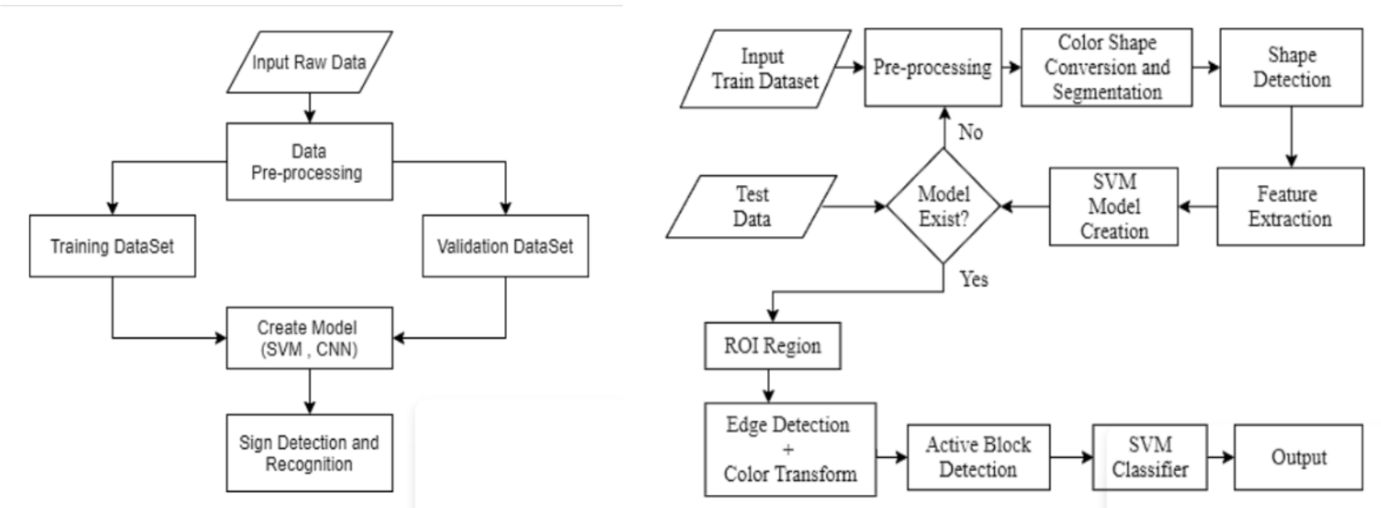

Fig No.3.2 – Flow Chart of working of project

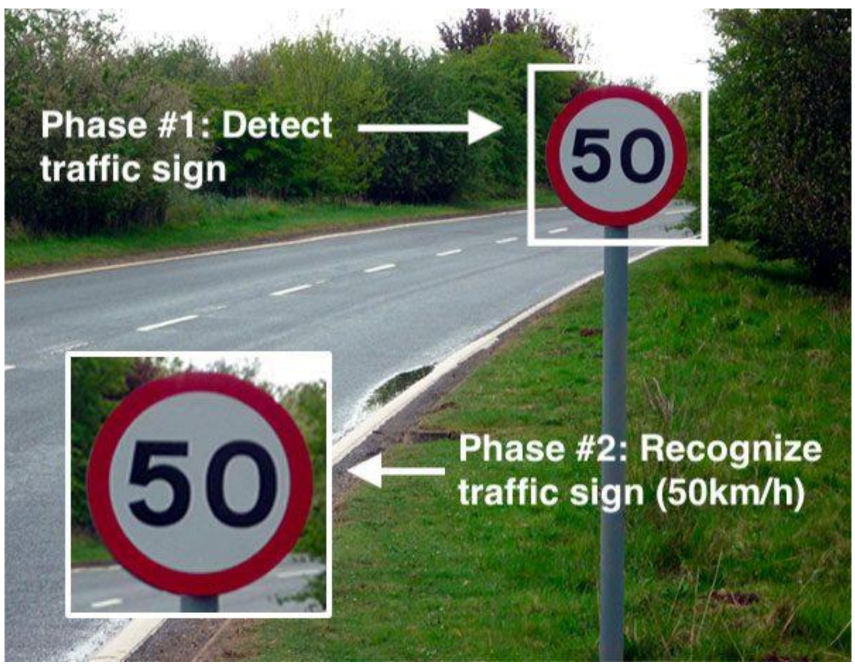

Fig No.3.3 – Detection And Recognition sample

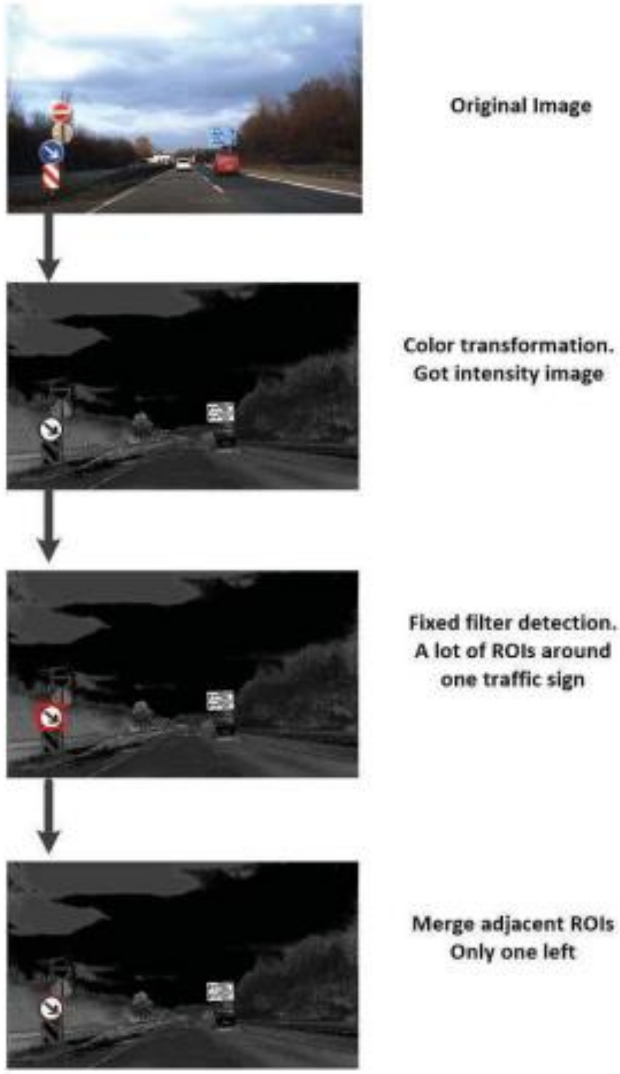

Fig No.3.4 – Detection and Conversion RGB

3.2 Data Exploration

Around 43 subfolders(ranging from 0 to 42) are available in our ‘train’ folder, and each subfolder represents a different class. We have an OS module that helps in the iteration of all the images with their respective classes and labels. To open the contents of ideas into an array, we are using the PIL library.
In the end, we have to store every image with its corresponding labels into lists. A NumPy array is needed to feed the data to the model, so we convert this list into an array.
Now, the shape of our data is (39209, 30, 30, 3), where 39209 represents the number of images,
30*30 represents the image sizes into pixels, and the last 3 represents the RGB value(availability of coloured data).
With the sklearn package, we use the train_test_split() method to split training and testing data.
From the keras.utils package, we use to_categorical method to convert the labels present in y_train and t_test into one-hot encoding.

Fig No. 3.5- Various Traffic Signs in Dataset

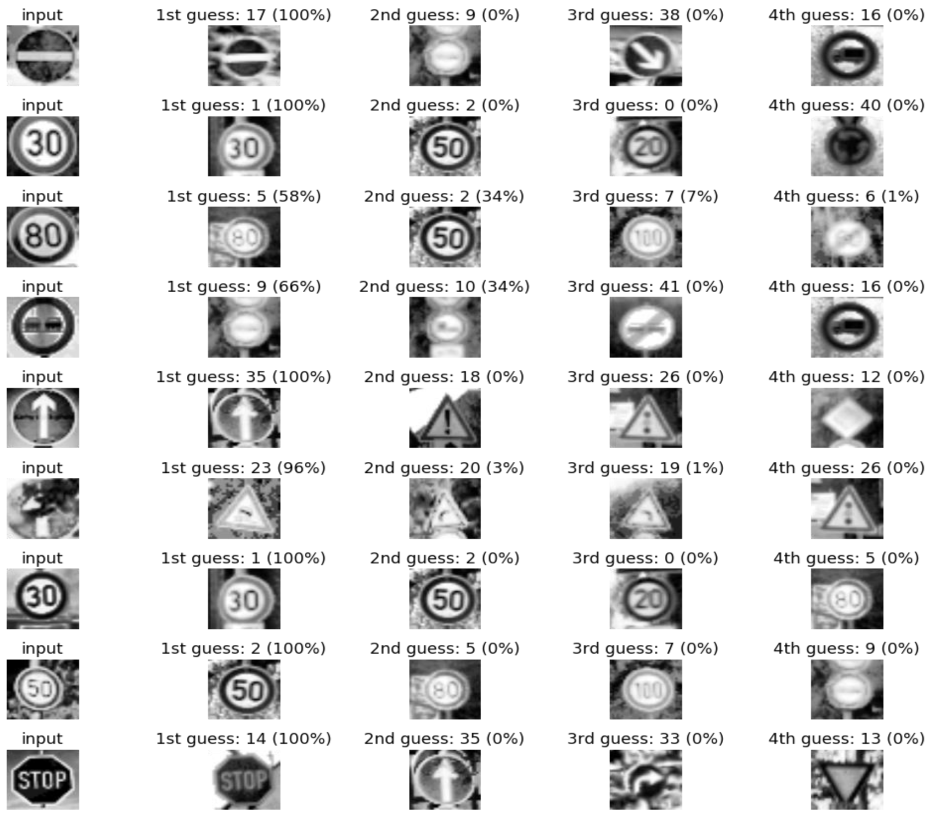

Fig No.3.6 - RGB Images are converted to Grayscale for processing

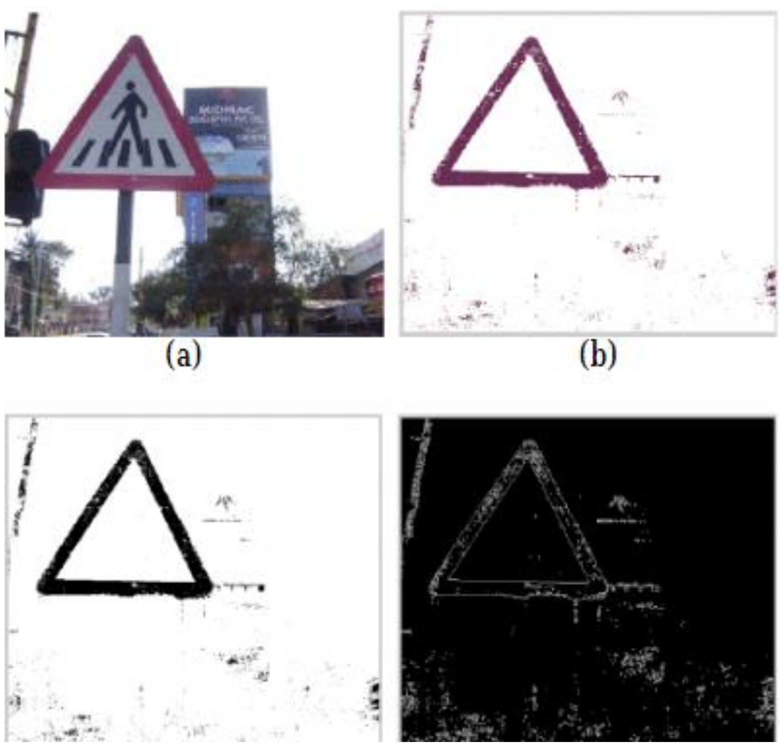

Fig No.3.7- Converted RGB Images to Grayscale

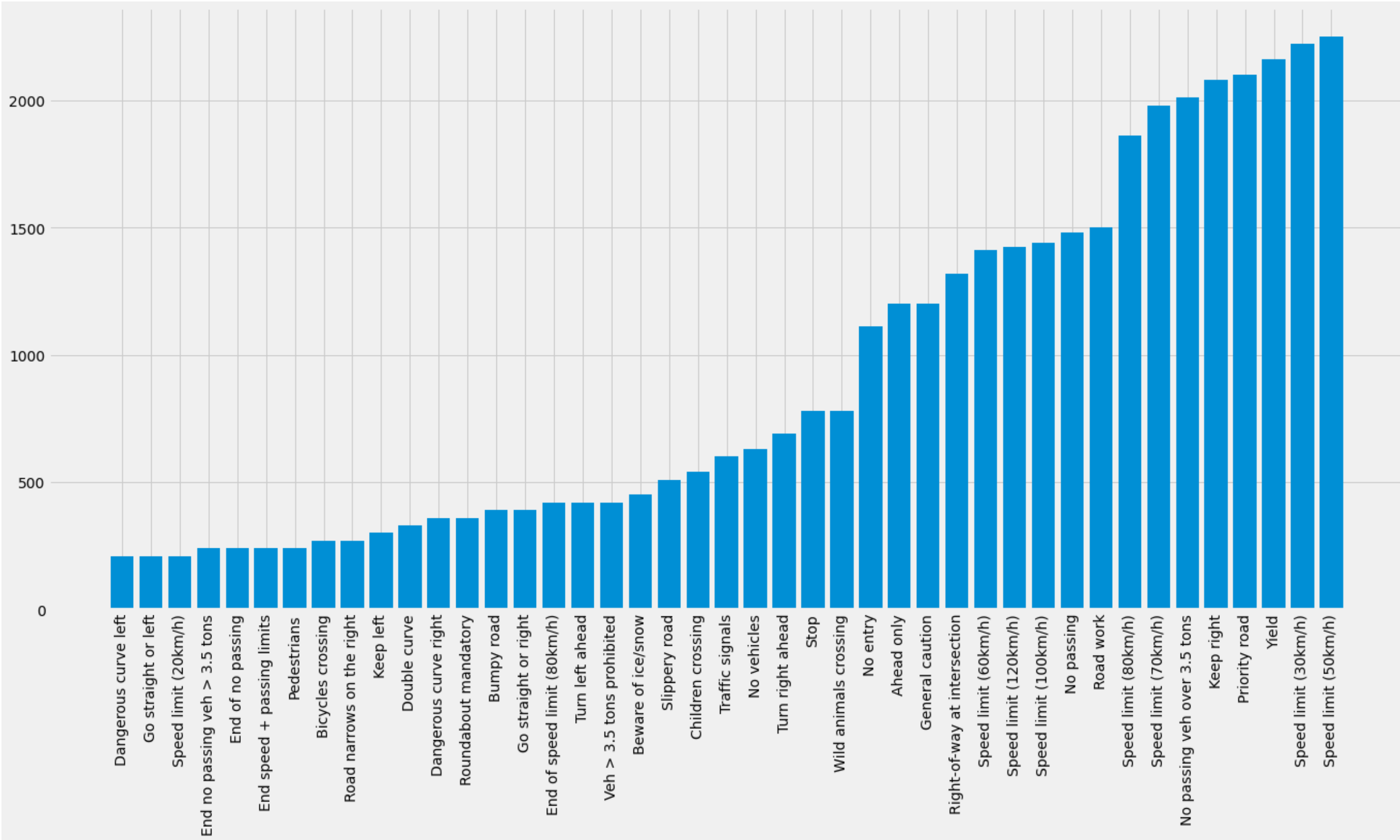

Fig No. 3.8-Plotting the Number of images in each of the 43 classes

3.3 CNN Model Building

The key part to understand, which distinguishes CNN from traditional neural networks, is the **convolution** operation. Having an image at the input, CNN scans it many times to look for certain **features**. This scanning (convolution) can be set with 2 main parameters: stride and padding type. As we see on below picture, process of the first convolution gives us a set of new frames, shown here in the second column (layer). Each frame contains an information about one feature and its presence in scanned image. Resulting frame will have larger values in places where a feature is strongly visible and lower values where there are no or little such features. Afterwards, the process is repeated for each of obtained frames for a chosen number of times.

To classify the images into their respective categories, we will build a CNN model (***Convolutional Neural Network***). CNN is best for image classification purposes.

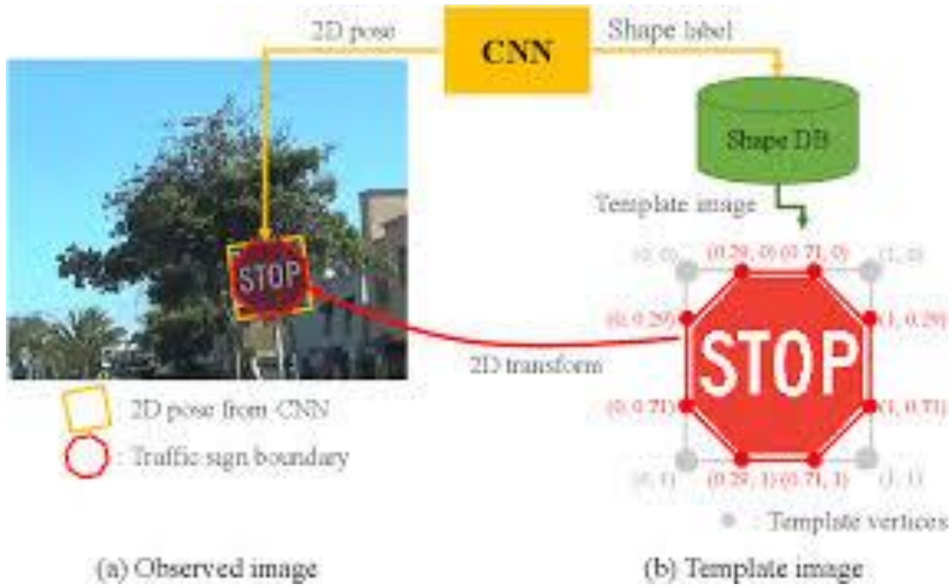

Fig No.3.9 -The architecture of our model

- 2 Conv2D layer (filter=32, kernel_size=(5,5), activation=”relu”)
- MaxPool2D layer (pool_size=(2,2))
- Dropout layer (rate=0.25)
- 2 Conv2D layer (filter=64, kernel_size=(3,3), activation=”relu”)
- MaxPool2D layer (pool_size=(2,2))
- Dropout layer (rate=0.25)
- Flatten layer to squeeze the layers into 1 dimension
- Dense Fully connected layer (256 nodes, activation=”relu”)
- Dropout layer (rate=0.5)
- Dense layer (43 nodes, activation=”softmax”)

We compile the model with Adam optimizer which performs well and loss is “categorical_crossentropy” because we have multiple classes to categorise.

3.3 Model Training And Validation

To train the model, we have used an Adam optimizer with batch size 32 and number of epoch is 15. We followed a simple approach and ran only 15 epochs of the training and observed the validation error trying to set it on minimum level and also due to limitation of computational power. It is very important to consider mainly validation error while improving the model. Decreasing only the error with respect to training data can easily lead to unwanted model over fitting.

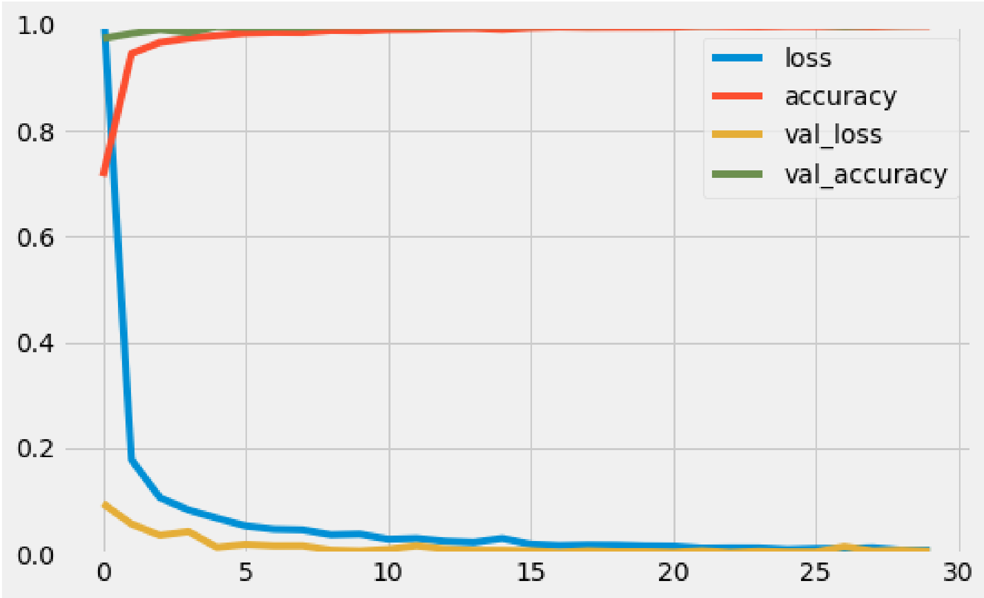

Fig No.3.10 – Evaluating the Accuracy

3.4 Model Testing

Our dataset contains a test folder and in a test.csv file, we have the details related to the image path and their respective class labels. We extract the image path and labels using pandas. Then to predict the model, we have to resize our images to 30×30 pixels and make a numpy array containing all image data. From the sklearn.metrics, we imported the accuracy_score and observed how our model predicted the actual labels.
A folder named" test" is available in our dataset; inside that, we got the main working comma-separated file called" test.csv". It comprises two things, the image paths, and their respective class labels. We can use the pandas' python library to extract the image path with corresponding labels. Next, we need to resize our images to 30×30 pixels to predict the model and create a numpy array filled with image data. To understand how the model predicts the actual labels, we need to import accuracy_score from the sklearn.metrics. At last, we are calling the Keras model.save() method to keep our trained model.

3.5 GUI for Traffic Sign Classifier

We will use a standard python library called flask to build a graphical user interface(GUI) for our traffic signs recognizer. We need to create a separate python file named" gui.py" for this purpose.
Firstly, we need to load our trained model 'traffic_classifier.h5' with the Keras library's deep learning technique. After that, we build the GUI to upload images and a classifier button to determine which class our image belongs to. We create a classify() function for this purpose; whence we click on the GUI button, this function is called implicitly.
To predict the traffic sign, we need to provide the same resolutions of shape we used at the model training time. So, in the classify() method, we convert the image into the dimension of shape (1 * 30 * 30 * 3). The model.predict_classes(image) function is used for image prediction, it returns the class number(0-42) for every image. Then, we can extract the information from the dictionary using this class number.

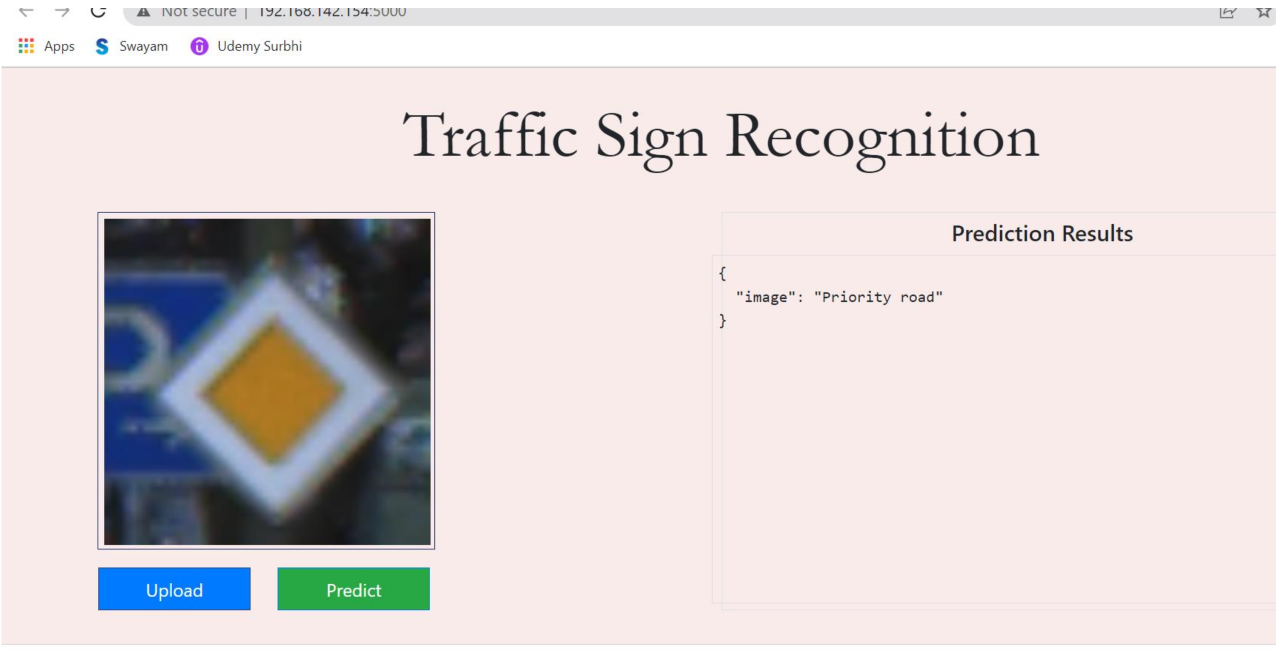

Fig No. 3.11– GUI Recognising Traffic signs

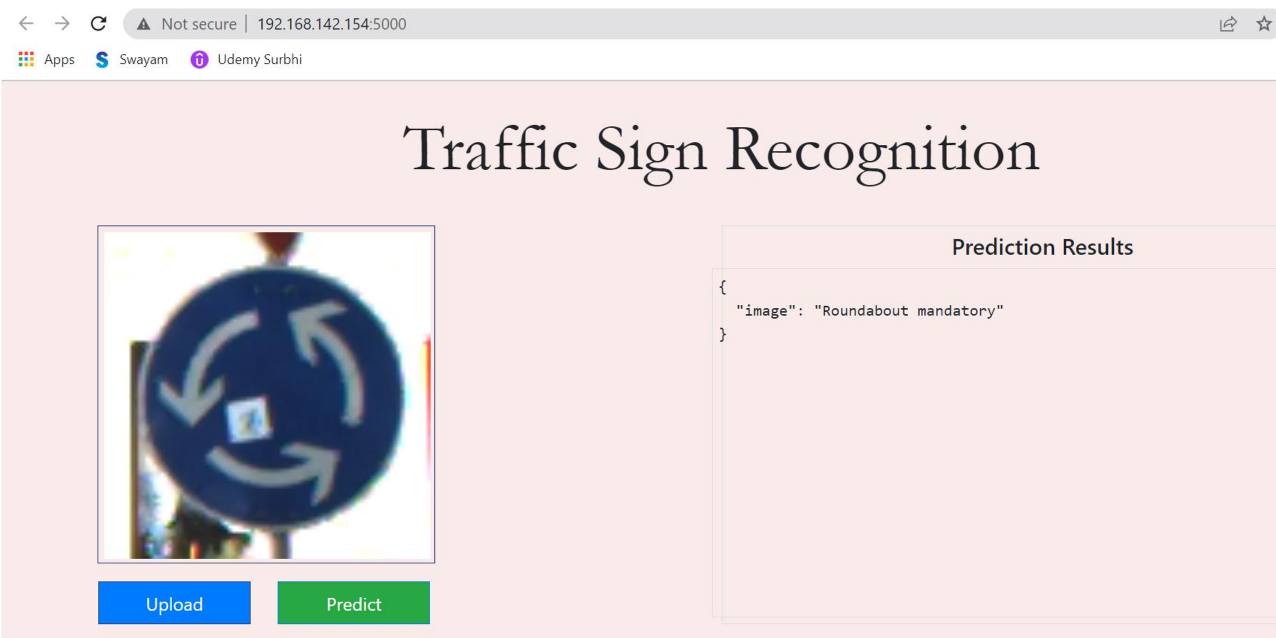

Fig No.3.12 – GUI Recognising Traffic signs

3.6 Tools & Technologies

3.6.1 Python

Python is a high-level, interpreted, interactive, and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactical constructions than other languages.

3.6.2 Machine Learning

Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.

3.6.3 Jupyter Notebook

Jupyter Notebook is a web-based interactive computational environment for creating notebook documents. A Jupyter Notebook document is a browser-based REPL containing an ordered list of input/output cells which can contain code, text, mathematics, plots and rich media. Underneath the interface, a notebook is a JSON document, following a versioned schema, usually ending with the ".ipynb" extension.

3.6.4 Anaconda

Anaconda is a distribution of the Python and R programming languages for scientific computing that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. We used anaconda to get access to Anaconda Navigator and Anaconda Prompt.

3.3.6 Libraries Used

3.3.6.1 Numpy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. NumPy is open-source software and has many contributors.

3.3.6.2 Pandas

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals
.

3.3.6.3 Seaborn

Seaborn is a data visualization library built on top of matplotlib and closely integrated with Pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.

3.3.6.4 Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, Python, Qt, or GTK.

3.3.6.5 Scikit-Learn

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

3.3.6.6 Pickle

Pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What pickle does is that it "serializes" the object first before writing it to file. Pickling is a way to convert a python object into a character stream. The idea is that this character stream contains all the information necessary to reconstruct the object in another python script.

3.3.6.7 Keras

Keras is used for creating deep models which can be productized on smartphones. Keras is also used for distributed training of deep learning models. Keras is used by companies such as Netflix, Yelp, Uber, etc

3.3.6.8 Tensorflow

TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow.

3.3.6.9 Pillow

Python pillow library is used to image class within it to show the image. The image modules that belong to the pillow package have a few inbuilt functions such as load images or create new images, etc.

Chapter 4: FINAL ANALYSIS AND DESIGN

4.1 Result

After that, we developed a system using CNN to evaluate the results with real time images The detection part uses image processing techniques that creates contour on each frame and finds all ellipse or circles among those contours. Then, the detection part marked as categorized traffic sign.

The average success rate is 98.00%, the result being independent of the number of traffic signs that the image contains. The lower rate of success in octagon shaped signals with 88.24%, is due to the fact that in certain images the signature of the octagon is recognized as a circle, so the detected sign is incorrect.

There are recognition problems in low-resolution images as,due to the low pixel rate, the recognized the areas are joined together, resulting in a detection of erroneous ROIs.

Furthermore, very high resolution images, due to the weight in memory, must be previously treated, reducing the storage weight before loading them, because the memory space is problematic and cannot be properly processed by the program. On the other hand, in pictures showing cuts in their area or perimeter due to external agents such as vegetation, climate, etc., the application had difficulty recognizing the area and, in many cases, the area was ruled out due to the fact that it did not have a closed surface.

FINAL TSR

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

	precision	recall	f1-score	support
0	0.98	1.00	0.99	60
1	0.99	1.00	1.00	720
2	1.00	1.00	1.00	750
3	1.00	0.95	0.98	450
4	1.00	0.99	1.00	660
5	0.96	0.99	0.98	630
6	1.00	1.00	1.00	150
7	0.99	1.00	1.00	450
8	1.00	0.99	1.00	450
9	1.00	1.00	1.00	480
10	1.00	1.00	1.00	660
11	0.90	1.00	0.94	420
12	0.99	0.98	0.99	690
13	1.00	0.99	1.00	720
14	0.96	1.00	0.98	270
15	1.00	1.00	1.00	210
16	1.00	1.00	1.00	150
17	1.00	0.99	1.00	360
18	1.00	0.96	0.98	390
19	0.97	1.00	0.98	60
20	0.97	1.00	0.98	90
21	0.82	0.72	0.77	90
22	0.96	0.76	0.85	120
23	0.92	1.00	0.96	150
24	1.00	0.98	0.99	90
25	0.97	0.99	0.98	480
26	0.96	1.00	0.98	180
27	0.94	0.50	0.65	60
28	1.00	1.00	1.00	150
29	1.00	1.00	1.00	90
30	1.00	0.89	0.94	150
31	1.00	0.98	0.99	270
32	1.00	1.00	1.00	60
33	1.00	1.00	1.00	210
34	1.00	1.00	1.00	120
35	0.98	1.00	0.99	390
36	1.00	1.00	1.00	120
37	1.00	1.00	1.00	60
38	1.00	0.99	1.00	690
39	0.94	0.99	0.96	90
40	0.93	0.99	0.96	90
41	1.00	1.00	1.00	60
42	1.00	1.00	1.00	90
accuracy			0.98	12630

22	0.96	0.76	0.85	120
23	0.92	1.00	0.96	150
24	1.00	0.98	0.99	90
25	0.97	0.99	0.98	480
26	0.96	1.00	0.98	180
27	0.94	0.50	0.65	60
28	1.00	1.00	1.00	150
29	1.00	1.00	1.00	90
30	1.00	0.89	0.94	150
31	1.00	0.98	0.99	270
32	1.00	1.00	1.00	60
33	1.00	1.00	1.00	210
34	1.00	1.00	1.00	120
35	0.98	1.00	0.99	390
36	1.00	1.00	1.00	120
37	1.00	1.00	1.00	60
38	1.00	0.99	1.00	690
39	0.94	0.99	0.96	90
40	0.93	0.99	0.96	90
41	1.00	1.00	1.00	60
42	1.00	1.00	1.00	90
accuracy			0.98	12630
macro avg	0.98	0.97	0.97	12630
weighted avg	0.98	0.98	0.98	12630

Fig No. 4.1 Classification Report

4.2 Result Analysis

In this section we are going to discuss about result which was obtained from and CNN. In CNN method we achieved 98.00% training accuracy and 97.00% validation accuracy. The training accuracy, validation accuracy and loss of CNN model is visualized in figure.

Accuracy			0.98	12630
Macro Avg	0.98	0.97	0.97	12630
Weighted Avg	0.98	0.98	0.98	12630

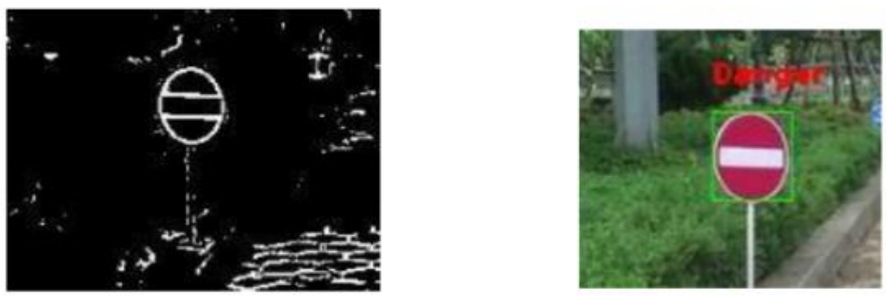

Fig No.4.2 –Recognized Danger Traffic sign

Fig No.4.3 –Recognized Speed 30km/hr Traffic sign

4.3 Application

Traffic sign recognition (TSR) is one of the most important background research topics for enabling autonomous vehicle driving systems. Autonomous driving systems require special handling of input data: there is no time for complex transformations or sophisticated image processing techniques, they need a solid and real-time analysis of a situation. This challenge get more difficult to meet in a city like environment where multiple traffic signs, ads, parking vehicles, pedestrians, and other moving or background objects make the recognition much more difficult. While numerous solutions have been published, solutions are tested on auto ways, country-side, or at a very low speed. In this paper, we give a short overview on main problems and known strategies to solve these problems, and we give a general solution to tackle real-time issues in urban traffic sign recognition.

4.4 Problems Faced

- Lighting conditions – there are differences in acquiring images by daylight and night, or by influencing of source of Light. Thus shade of colors of objects can be seen differently by the illumination changes. Issues cause also a reflection from any light source such as sun in daylight or street lighting in night. Shape based approach for traffic sign detection seems to be a good choice for solving this problem.
- Weather conditions – captured image is influenced by raining, snowing or occurrence of a fog. For example, traffic signs can be shrouded by snow or be poorly visible in a fog.
- Damage – traffic signs can be damaged not only by sun-shine, but also by vandalism or weather over time (strong breeze, storm, raining). They can be then dirty, scribbled over, tilted, rusty etc.

4.5 Limitations

- The limitations of this project are mainly reflected in four aspects. Firstly, our dataset does not cover all the classes of traffic signs in “German Traffic Sign Recognition Benchmark(GTSRB)” which is mainly due to the limited time and physical resources. Thus, only several and the most common used traffic signs are contained in our dataset for training networks for recognition tasks. The practicability of this project will suffer from the partial types of traffic-sign data and the project temporally stuck in the experimental phase.
- Secondly, though the experiments in this project were performed both on the two stage and on-stage neural networks (Faster R-CNN with VGG16 and YOLOv5), there are 51 still a lot of models that are worthy to be estimated so that a more comprehensive results will be summarized and the conclusions will be more instructional in this field.
- Thirdly, there are only several metrics used to evaluate the performance of the two models. Several meaningful measures, such as F1 score and the areas under Precision Recall curve, could be considered as the additional measures to help us choose appropriate models for recognizing traffic signs.
- Lastly, we only explored this problem from a limited researching angle. There are a lot of real-world difficulties while recognizing traffic signs like we mentioned at the beginning of this report, including illuminant issue, rotations, partial occasions and physical damages, etc. However, in this project, we only focused on tackling with the problem of recognizing smaller size of traffic signs. There are still a lot of researching spaces in this field to be mined.

CHAPTER 5: CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

A new traffic sign recognition system has been presented in this paper. The application software developed in this work recognizes and classifies traffic signs from an input image. The image processing techniques used in this software include a preprocessing stage, regions of interest detection, potential traffic sign detection, according to the traffic sign shape patterns, and finally, the recognition and classification of these potential traffic signs according to a database of traffic sign patterns. The performance of this application depends on the quality of the input image, in relation to its size, contrast and the way the signs appear in the image. With this consideration, the percentages of recognized signs for this application are high.

As a result of signs classification, the following features were revealed:

• The approach using image pre -processing is sensitive to changes in environmental conditions;

• Approaches have a high robustness in when classifying signs with noises;

• The approach using a convolution neural network requires a wider selection to enhance the ability to correctly annotate values without overtraining, what in its turn in the approach with color filtering falls on adding and modifying algorithms;

• The approach utilizing color filtering has a higher potential for scaling and implementation compared to convolution networks due to a lower requirements for computational resources;

• Extension in convolution networks becomes available through increasing the number of filter layers.

5.2 Future Scope

- The traffic signs focus on reduction of the traffic load on existing road network through various travel demand management measures.
- Traffic signs should remove the encroachments, congestion and improve the traffic signal, road condition and geometrics features at intersections.
- The traffic signs should be as a guidance or speaker on a road network.
- Traffic sign reduce the traffic congestion along the road and also provide facilities for the road users.
- Road signs notify road users of regulations and provide warning and guidance needed for safe, uniform and efficient operation.
- The purpose of road signs is to promote road safety and efficiency by providing for the orderly movement of all road users on all roads in both urban and non-urban areas.
- As further work, a neural network could be implemented in order to obtain more accurately the empirical parameters used in the application. Furthermore, the application could be optimized by implementing an embedded hardware for use in active applications.

REFERENCES

- Kaoutar Sefrioui Boujemaa, Afaf Bouhoute, Karim Boubouh and Ismail Berrada, Traffic sign recognition using convolutional neural networks in International Conference on Wireless Networks and Mobile Communications (WINCOM) 2017 IEEE
- Amal Bouti, Mohamed Adnane Mahraz, Jamal Riffi, Hamid Tairi, Road sign recognition with Convolutional Neural Network in International Conference on Intelligent Systems and Computer Vision (ISCV). 2018 IEEE
- Prashengit Dhar, Md. Zainal Abedin, Tonoy Biswas, Anish Datta, Traffic Sign Detection- A New Approach and Recognition Using Convolution Neural Network in IEEE Region 10 Humanitarian Technology Conference (R10-HTC) 2017 IEEE.
- Yann Lecun, Leon Bottou, Yoshua Bengio, Patrick Haffner, Gradient-Based Learning Applied to Document Recognition in PROC, Of the IEEE, November 1998.
- Faming Shao, Xinqing Wang,* Fanjie Meng, Ting Rui, Dong Wang, and Jian TangReal-Time Traffic Sign Detection and Recognition Method Based on Simplified Gabor Wavelets and CNNs
- Shopa, P., Sumitha, N., & Patra, Traffic sign detection and recognition using OpenCV in International Conference on Information Communication and Embedded Systems (ICICES2014).
- Faming Shao, Xinqing Wang, Fanjie Meng, Ting Rui, Dong Wang, and Jian Tang, Real-Time Traffic Sign Detection and Recognition Method Based on Simplified Gabor Wavelets and CNNs in PMC Sensors (Basel) 2018 Oct.