UNIVERSITY PARTNER

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE
KATHMANDU

Project and Professionalism
(6CS007)

Project Report
Traffic Signs Recognition

Student Id	: 2049827
Student Name	: Shiv Kumar Yadav
Group	: L6CG1
Supervisor	: Sarjil Napit
Reader	: Yogesh Bikram Shah
Cohort	: 5
Submitted on	: 26-04-2022

Declaration Sheet

Award Title: ***BSc(Hons) Computer Science***

Declaration Sheet

(Presented in partial fulfillment of the assessment requirements for the above award.)

This work or any part thereof has not previously been presented in any form to the University or to any other institutional body whether for assessment or for other purposes. Save for any express acknowledgements, references and/or bibliographies cited in the work. I confirm that the intellectual content of the work is the result of my own efforts and of no other person.

It is acknowledged that the author of any project work shall own the copyright. However, by submitting such copyright work for assessment, the author grants to the University a perpetual royalty-free license to do all or any of those things referred to in section 16(I) of the Copyright Designs and Patents Act 1988. (viz. to copy work; to issue copies to the public; to perform or show or play the work in public; to broadcast the work or to make an adaptation of the work).

Student Name: SHIV KUMAR YADAV

Student ID Number: 2049827

Signature: Shiv

Date: April 26, 2022

Abstract

We can see the different types of traffic sign images located on the side of the road or at a certain height. some signs are easily recognized but some signs are difficult to recognize. This creates certain problems while driving and maybe drivers do not drive safely. So to enhance safety driving practices I have made the system called ‘The traffic’. This project proposes a system for recognizing road and traffic signs to aid drivers in recognizing traffic signs, to create an inventory of them that will assist highway engineers in updating and maintaining them, as well as the authority of the traffic department in maintaining and updating traffic signs. It makes use of photographs that have been uploaded via file explorer. Image upload, color segmentation, and recognition are the three primary stages of the system.

The traffic system used a convolutional neural network (CNN) to classify the traffic sign. Multi-class pictures and different distinct types of moments were used to train and test the classifier. After training and validating the data, the data is fit into the model. Additional features, kernels, parameters, and moment's orders were used to test the CNN performance. The performance rate is around 97 percent on average.

Acknowledgment

I would like to express my deepest thanks to Mr. Sarjil Napil who has been a great mentor since the beginning of the project. His delivery of project materials was soo good and it has helped me throughout the project. I am so glad that we have such a nice person as our supervisor. I would also like to thank our reader Mr. Yogesh Bikram Shah who helped us in maintaining the flow of development. His reviews help us a lot to improve our project. Mr. Biraj Dulal, who is the module leader of this module project and professionalism, I would also like to thank him for giving us a way to work on the fabulous project.

Lastly, I would like to thank all the friends who have been so kind and helped me throughout the development phase of the project.

Table of Contents

1. Introduction

1.1. Project Briefing

1.1.1. Project Introduction

Traffic sign recognition is a web app that helps in recognizing and classifying traffic signs. While traveling, we can see several types of traffic signs that are placed beside the road or at some certain height. So recognizing some traffic signs is easy but some may create problems. Therefore an individual can click that image and upload it to the system for recognition of that particular name of the traffic sign. This will improve the safe driving practices and may the accidents decrease.

ó**The t r f f i e ô** intends to establish a procedure that can detect and categorize road and traffic signs to produce a tally that will allow transportation authorities to renovate and keep up to date traffic signs. It operates by capturing shots by using a camera and uploading recorded shots i.e., images to the system and the system used the AI model to recognize and classify the traffic signs.

1.1.2. Problem Statement

Given humans' limited ability to recognize and comprehend objects, developing a computer-based system that can assist people in their daily lives is a difficult task. Many factors change over time, such as luminance and visibility, and are easily handled by human identification systems but pose severe challenges for computer-based recognition. Looking at the problem of recognizing road and traffic signs reveals that the aim is well stated and appears to be a straightforward problem. Traffic or Road signs are placed in convenient locations, have standard shapes, colors, pictograms, and are well-known. However, several parameters that influence the detecting system's performance must be thoroughly investigated to comprehend the problem completely. Furthermore, traffic signs may be partially or entirely obscured by other things such as automobiles or pedestrians in these photographs. Other issues, such as the existence of similar-looking objects to road signs, such as buildings or billboards, can wreak havoc on the system and make sign detection difficult.

Different countries utilize a variety of colors and pictograms in traffic signs. In addition, the system should be adaptable, which means it should allow for ongoing learning rather than having to redo the training for each country. Finally, traffic sign recognition should be given many sign examples to address all of these issues so that the system can respond appropriately when a traffic sign is met.

1.1.3. Description of System

A system for recognizing road and traffic signs should operate in two modes: the training mode, which involves collecting a set of traffic signs for training and validation, and the prediction model, which consists of the system recognizing a traffic sign that has never been seen before. It comprises many modules that work together to execute this task. These modules are camera, colour segmentation, shape analysis, raw image datasets, feature extraction, and classification. Colour segmentation, recognition, and classification are the three main stages of the system.

The traffic system used a convolutional neural network (CNN) to classify the data. The categorization is done in two stages: first, the rim's shape is classified, and then the sign's contents are classified. Binary pictures and five distinct types of moments were used to train and test the classifier. Different features, kernels, parameters, and moment's orders were used to test the CNN performance. The performance rate is around 97 percent on average.

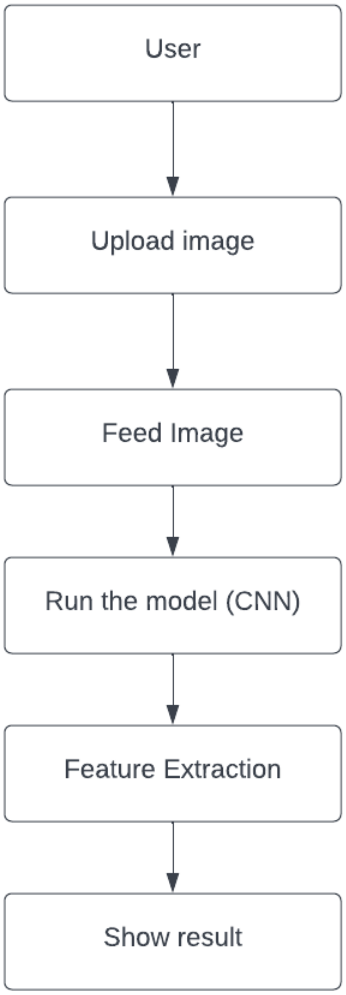


Figure 1 System Architecture

1.1.4. Nepal traffic and road Signs

Road and traffic signs regulate, warn, guide, or educate road users. They improve road traffic quality and safety by ensuring an orderly and logical flow of all traffic, including pedestrians.

Traffic and road signs are intended to be easily recognized by drivers due to their distinct designs and colours.

Nepalese road signs are divided into five groups:

1.1.4.1. Warning Signs

It is generally an equilateral triangle in shape signs with red colour on the border. This sign is placed to warn drivers of the different difficult situations on road.

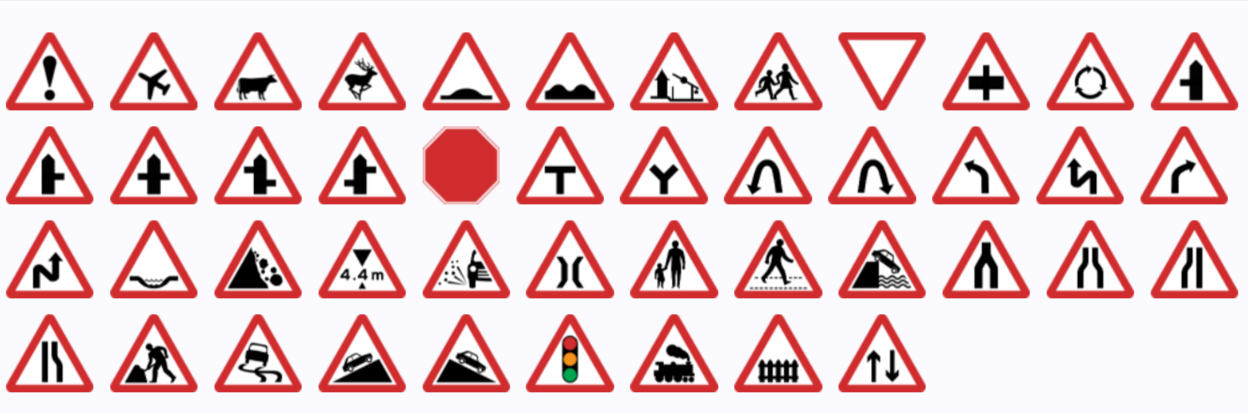


Figure 2 Warning Signs (Traffic-Rules, 2022)

1.1.4.2. Prohibitory Signs

Prohibitory traffic signs are generally circles in shape with a thick red colour border around them. This sign generally indicates certain tasks or work that is prohibited for road users.

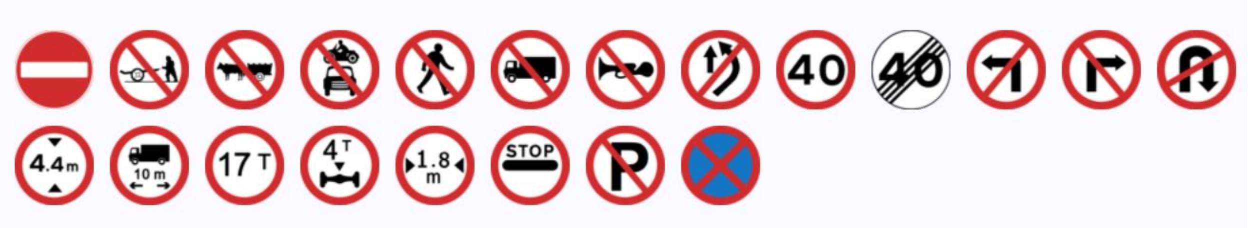


Figure 3 Prohibitory Signs (Traffic-Rules, 2022)

1.1.4.3. Mandatory Signs

A mandatory sign is characterized by a blue circle having some arrows which are used to control the actions of road users.

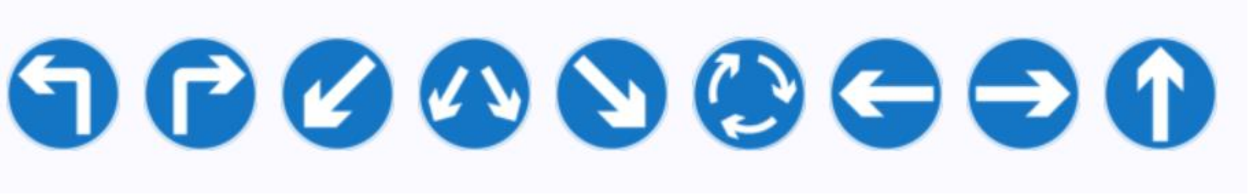


Figure 4 Mandatory Signs (Traffic-Rules, 2022)

1.1.4.4. Information Signs

Information signs give additional information about the start or end of some kind of traffic situation.

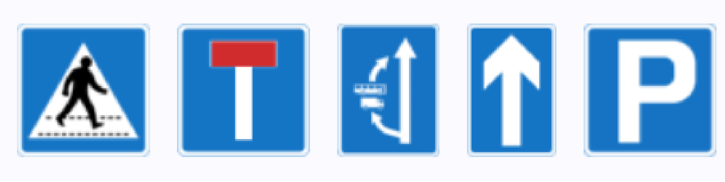


Figure 5 Information Signs (Traffic-Rules, 2022)

1.1.4.5. Priority Signs

Priority signs carry the information about the priority rule i.e., who should be given priority and who has priority.

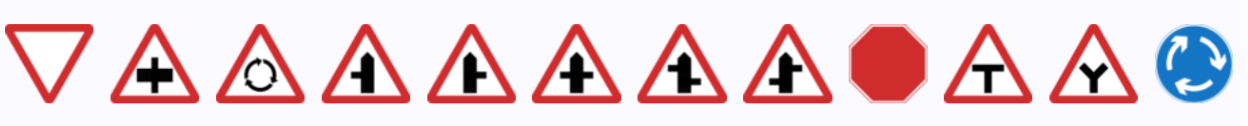
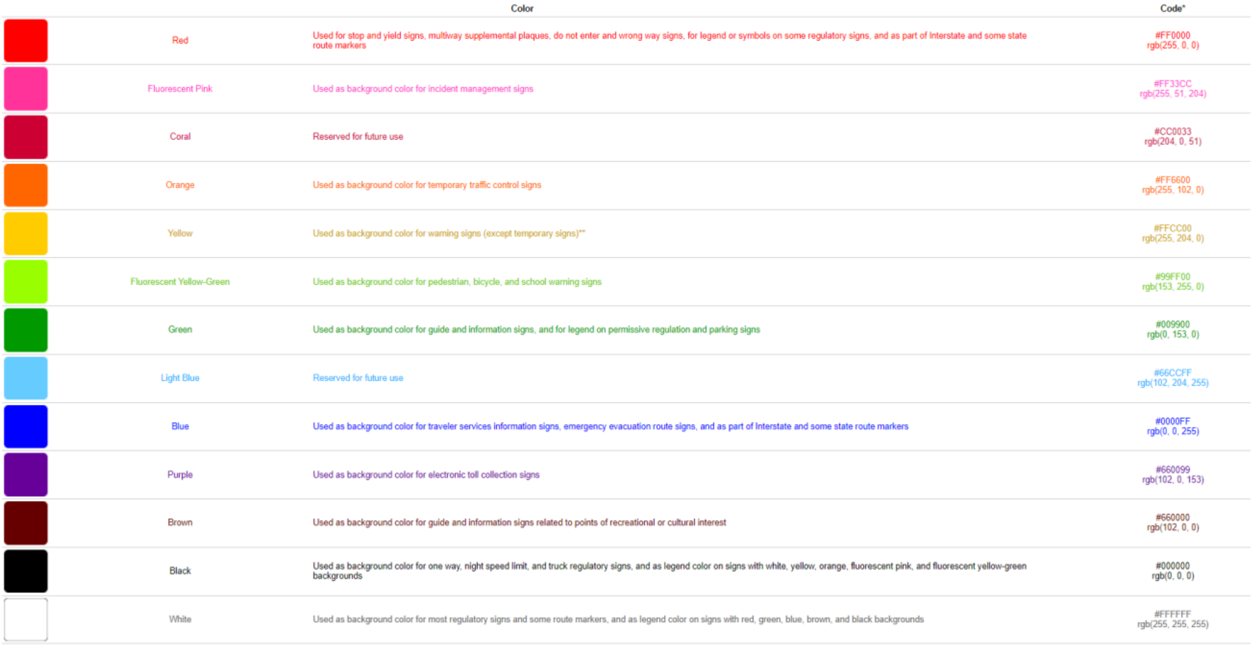


Figure 6 Priority Signs (Traffic-Rules, 2022)

1.1.5. Standard Colours and meaning of colours used for traffic signs

There are different colours used in traffic signs and each colour has its significance and meaning. Every colour has its code. The different colours with their meaning and standard colour code are shown in the table.



		Color	Code*

🟥	Red	Used for stop and yield signs, multiway supplemental plaques, do not enter and wrong way signs, for legend or symbols on some regulatory signs, and as part of Interstate and some state route markers	#FF0000
			rgb(255, 0, 0)

■	Fluorescent Pink	Used as background color for incident management signs	#FF33CC
			rgb(255, 51, 204)

■	Coral	Reserved for future use	#CC0033
			rgb(204, 0, 51)

🟧	Orange	Used as background color for temporary traffic control signs	#FF6600
			rgb(255, 102, 0)

🟨	Yellow	Used as background color for warning signs (except temporary signs)**	#FFCC00
			rgb(255, 204, 0)

■	Fluorescent Yellow-Green	Used as background color for pedestrian, bicycle, and school warning signs	#99FF00 rgb(153, 255, 0)
🟩	Green	Used as background color for guide and information signs, and for legend on permissive regulation and parking signs	#009900 rgb(0, 153, 0)
■	Light Blue	Reserved for future use	#66CCFF rgb(102, 204, 255)
🟦	Blue	Used as background color for traveler services information signs, emergency evacuation route signs, and as part of Interstate and some state route markers	#0000FF rgb(0, 0, 255)
🟪	Purple	Used as background color for electronic toll collection signs	#660099 rgb(102, 0, 153)
🟫	Brown	Used as background color for guide and information signs related to points of recreational or cultural interest	#660000 rgb(102, 0, 0)
⬛	Black	Used as background color for one way, night speed limit, and truck regulatory signs, and as legend color on signs with white, yellow, orange, fluorescent pink, and fluorescent yellow-green backgrounds	#000000 rgb(0, 0, 0)

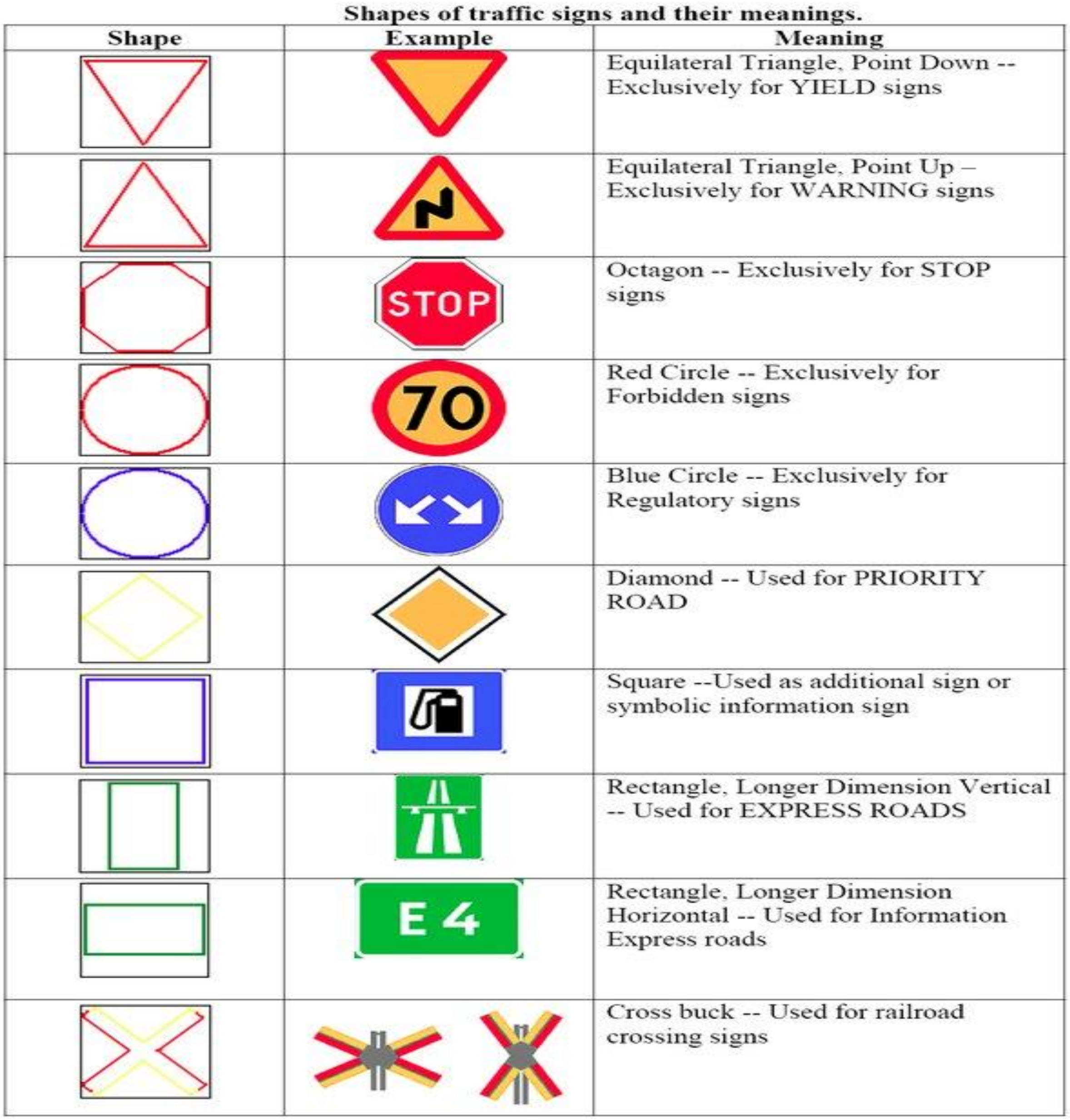
⬜	White	Used as background color for most regulatory signs and some route markers, and as legend color on signs with red, green, blue, brown, and black backgrounds	#FFFFFF
			rgb(255, 255, 255)

Table 1 Standard Colour code and their meaning (Moeur's, 2019)

In the above-mentioned table, we can see that there are different colours such as Red, Blue, Green, Coral, etc have their meaning and colours code. These types of colours and their code are generally used for making several types of traffic signs.

1.1.6. Shapes of traffic signs and their meaning

While recognizing traffic signs, the shapes of traffic signs play an important role. The different shapes are so we can recognize certain traffic signs instantly, without having to read them. For example, when you see an eight-sided traffic sign, you know it’s a stop sign and you must stop, even if there is some reason you can’t read it (or you don’t have the time).



Shapes of traffic signs and their meanings.

Shape	Example	Meaning
		Equilateral Triangle, Point Down -- Exclusively for YIELD signs
		Equilateral Triangle, Point Up – Exclusively for WARNING signs
	STOP	Octagon -- Exclusively for STOP signs
	70	Red Circle -- Exclusively for Forbidden signs
		Blue Circle -- Exclusively for Regulatory signs
		Diamond -- Used for PRIORITY ROAD
		Square --Used as additional sign or symbolic information sign
		Rectangle, Longer Dimension Vertical -- Used for EXPRESS ROADS
	E 4	Rectangle, Longer Dimension Horizontal -- Used for Information Express roads
		Cross buck -- Used for railroad crossing signs

Table 2 Shapes of the traffic sign and their meaning

1.1.7. Possible Difficulties

The environment of the road and its surrounding scenes are very complex. Some road signs are found in a severe conditions such as damaged, aged, disoriented, etc. and such cases, there arises difficulty in recognizing traffic signs and even got wrong results and may face various challenges:

Á As a result of prolonged exposure to sunlight and the reaction of the paint with the air, the colour of the sign fades with time.

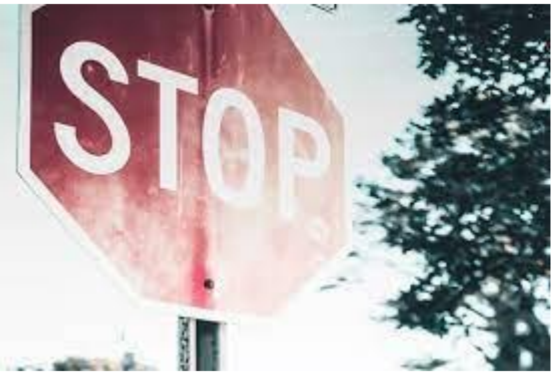


Figure 7 Faded Sign

Á Weather factors such as fog, rain, clouds, and snow have an impact on the visibility of traffic signs.

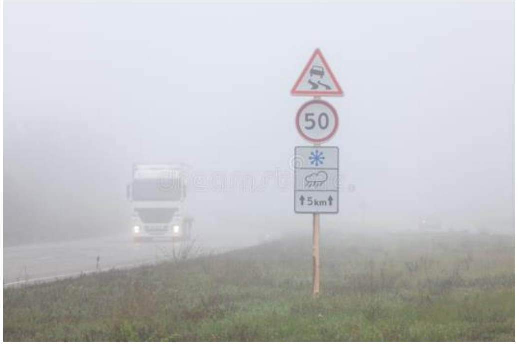


Figure 8 Bad weather

Á Variations in light circumstances, such as shadows, clouds, and the sun, have a significant impact on colour information. Illuminant colour (daylight), illumination geometry, and viewing geometry can all influence it.

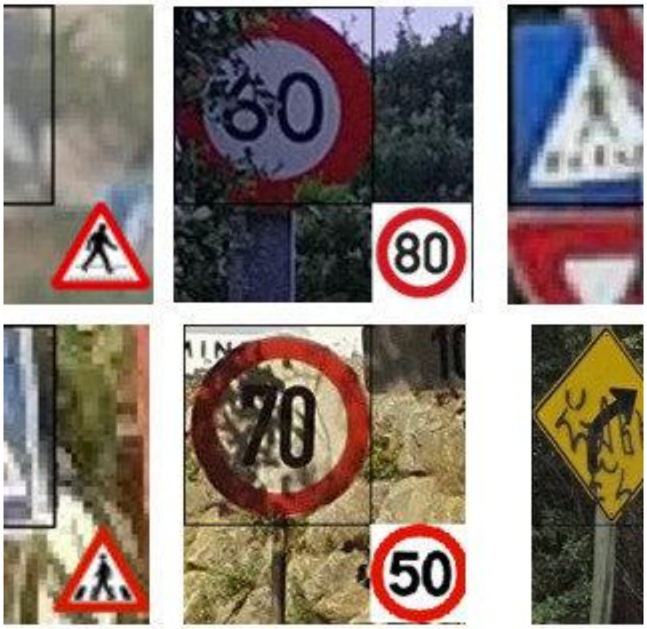


Figure 9 Bad-lighting circumstances

1.1.8. AI Implementation

Several neural networks in the CNN family specialize in the processing of data that has value in matrix form like pixel by pixel, such as images, and are also known as **CNNs** or ConvNet. In a digital image, visual data is encoded in binary form. It contains matrix-like management of pixels with pixel values that specify the lightness and colour of each pixel**.** In most cases, convolutional neural networks are taught as **supervised approaches**, which means that the training data includes both the inputs (such as pictures in an image recognition task) and the labels (i.e. the objects portrayed in the photos).

1.1.8.1. Reason behind using CNN

An image's patterns are the primary focus of CNN's analysis tools. As it digs further, it finds the correct characteristics on its own, so we don't have to offer them. Because of this, we turned to CNN whenever there was an issue with CNN. Convolutional networks are based on biology when it comes to how neurons link. Because of their shared-weights design, they are also known as shift-invariant. It's distinct from MLP since CNN employs several different layers to recognize patterns in pictures fed into it. In contrast to CNN, which uses sparsely connected layers and receives a matrix as input, MLP utilizes densely linked layers (Liao, et al., 2019).

1.1.8.2. Working flow of CNN

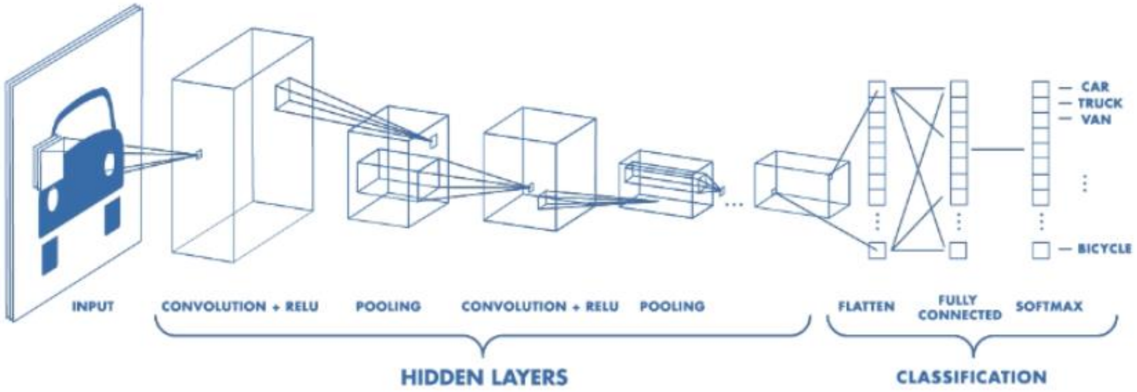


Figure 10 CNN architecture

A CNN generally includes 3 components: a convolutional layer, a pooling layer, and a fully connected layer.

Convolution layer

The convolution layer is the CNN's core. It carries the bulk of the state's computational load. Both the set of training data (kernels) and the limited section (perceptron) of the perceptron are linearly combined in this layer. The kernel is smaller in size and more detailed than an image. In a picture with three (RGB) portals, the module's height and width are small, but the depth extends to all three-color channels.

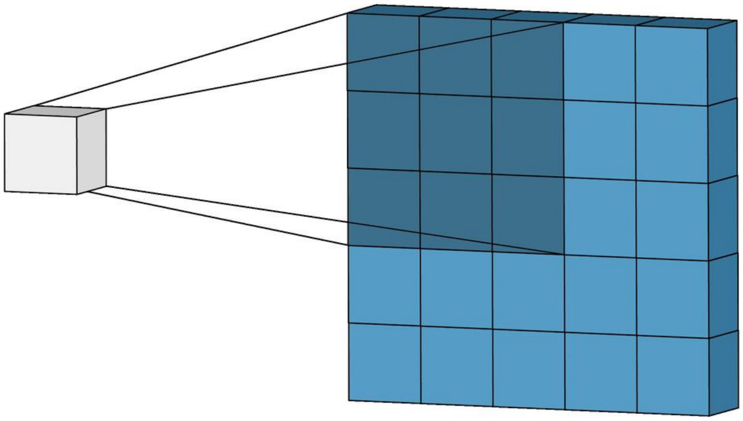


Figure 11 Convolutional layers illustrations

The kernel slides over the picture's height and breadth during the forward pass, providing an image representation of the receptive region. An activation map depicts the kernel's reaction to each spatial point in the picture. A stride is the kernel's slow movement from one place to another.

W x W x D + Dout kernels with spatial size F, stride S, and padding P = output volume size

$$W_{out} = \frac{W - F + 2P}{S} + 1$$

Size Wout + Dout + Dout + Dout will be the final output volume.

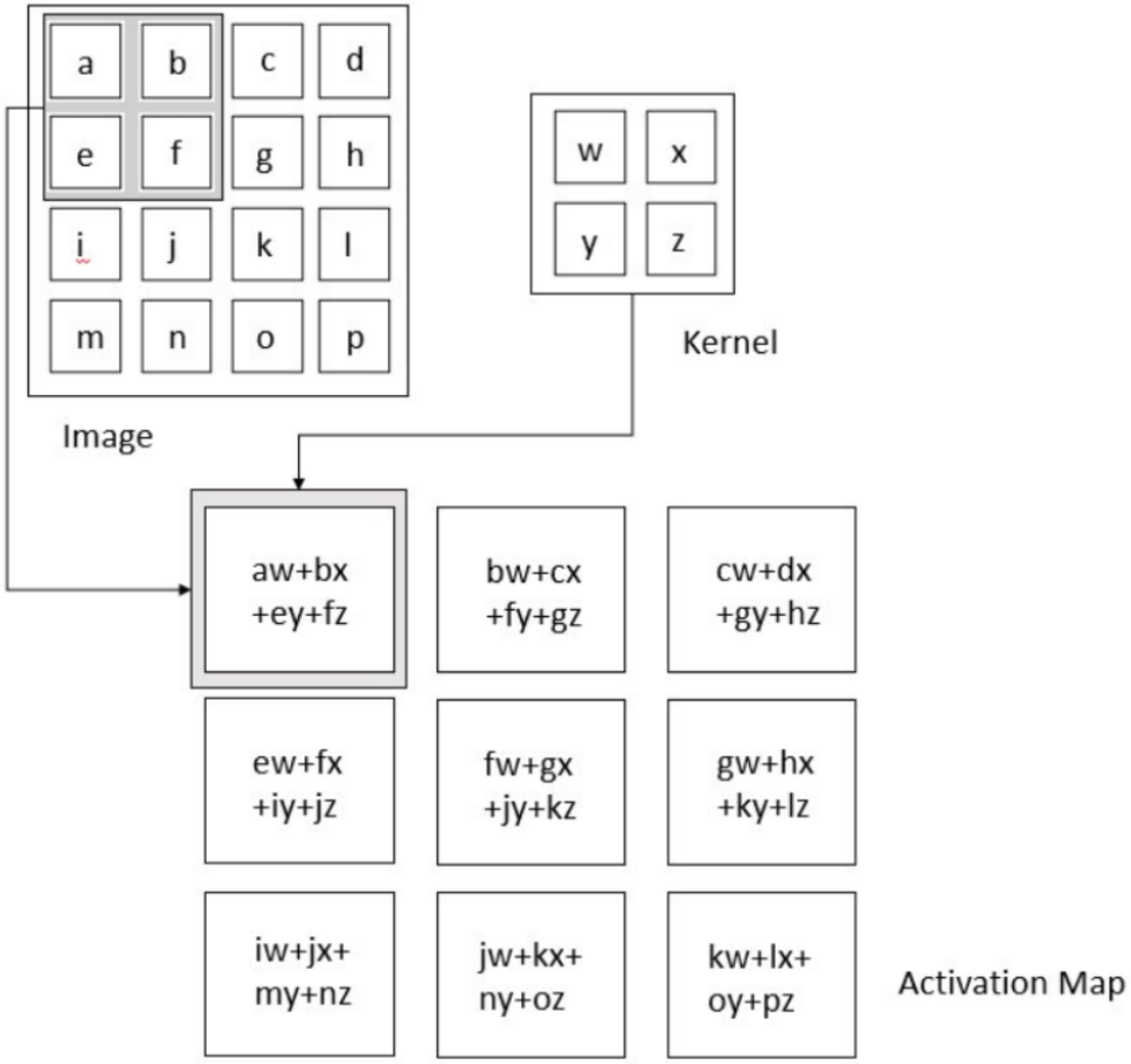


Figure 12 Operation of convolutional layer

Motivation Behind the convolution layer

Convolution employs three basic ideas that propelled machine vision investigators: sparse contact, present sample, and partition representation. Let's analyze each one of them in depth.

Minimalist neural network layers utilize multiplications by a matrix of parameters indicating the link between the intake and outflow unit. This means that every transfer function interacts with every input unit. Therefore, convolution neural networks display sparse interaction. This is done by making the kernel smaller than the input e.g., an image may include millions or thousands of pixels, but when processing it using the kernel we may uncover useful information that is of tens or hundreds of pixels. This suggests that we need to maintain fewer parameters that not only minimize the memory use of the model but also boost the statistical efficacy of the model.

If calculating one feature at a spatial place (x1, y1) is advantageous then it should also be valued at some other geographical point say (x2, y2) (x2, y2) (x2, y2). It indicates that for a single double slice i.e., for creating one activation map, neurons are constrained to apply the same set of weights. In a typical neural network, each member of the weight matrix is utilized once and then never returns, however, a convolution network contains shared parameters i.e., for getting output, weights supplied to one input are the same as weight applied elsewhere.

Due to sampling being utilized to gather, the layers in the convolution neural network will have a property of equivariance to translation. It shows that if we updated the input in a method, the output would likewise be modified in much the same way.

Pooling Layers

Rather than relying on the network's output, the pooling layer generates a statistical summary of the nearby outputs. As a result, the amount of computation and weights required for the representation are reduced. It is performed on each slice of the representation separately.

It is possible to calculate the average of the rectangle's neighborhood, the L2 norm of the neighborhood, and a weighted average based on distance from the center pixel. Max pooling, on the other hand, is the most common method of reporting neighborhood output.

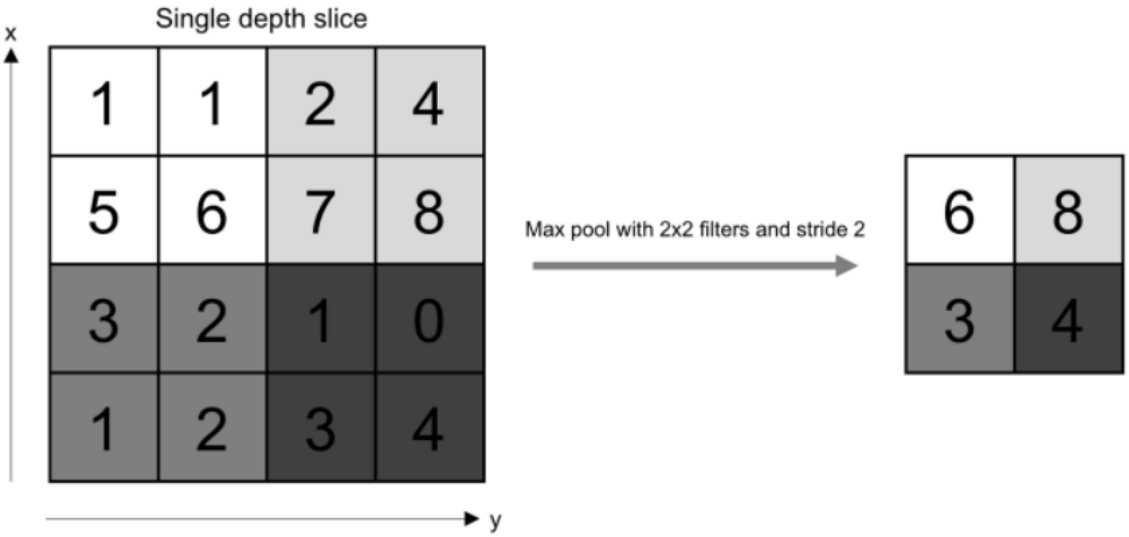


Figure 13 Operation of pooling

To calculate the output volume, we need to know the size of the activation map, the pooling kernel's spatial size F, and the stride S.

$$W_{out} = \frac{W - F}{S} + 1$$

In other words, this will produce an output volume with dimensions equal to (Wout + Wout + D).

Regardless of where an object appears on the frame in a pooled scene, it will be recognizable to the viewer.

Fully connected layers

Neurons in this layer exhibit complete connection with all neurons in the previous and following layer as demonstrated in normal FCNN. This is why it may be calculated as usual by a matrix multiplication followed by a bias effect. The FC layer serves to map the representation between the input and the output.

Activation Function

Since convolution is a linear operation and pictures are far from linear, non-linearity layers are commonly inserted right after the convolutional layer to introduce non-linearity to the activation map.

ReLU

The Rectified Linear Unit (ReLU) has proven increasingly popular in the last few years. It computes the function $f(\kappa)$=max (0,κ). In other words, the activation is essentially a threshold at zero. In comparison to sigmoid and tanh, ReLU is more trustworthy and accelerates the convergence by six times. Unfortunately, a drawback is that ReLU could be fragile throughout training. A huge gradient running across it may update it in such a way that the neuron will never be further updated. However, we may cope with this by creating the right learning rate.

We'll flatten the picture into a column vector once it's been transformed to a Multi-Level Perceptron-friendly format. Neural networks are trained using a **feed-forward neural network** with **backpropagation** done to every training iteration. The model can discriminate between dominant and specific low-level characteristics in pictures and categorize them using the **Softmax** Classification algorithm across a series of epochs.

1.2. Aims

a) To explore a productive solution to help in understanding the name of different traffic signs using AI.

b) To assign the detected traffic indicators to their appropriate sub-classes

c) To develop a user-friendly web application.

1.3. Objectives

a) Datasets of traffic signs will be collected through Kaggle to train AI models to recognize the name of traffic signs.

b) A better understanding of the attributes of road and traffic signs will be developed with the help of TensorFlow, Keras library for image processing for the recognition challenge.

c) Various machine learning models like Convolutional Neural Network(CNN) or Support Vector Machines(SVMs), etc. will be used to train the dataset.

d) The best method for extracting features from traffic signs will be created.

e) An image uploader will be generated to upload an image for further process of recognition.

f) A web application will be created for tracking traffic sign inventory.

1.4. Artefact

When a system is developing, we have to divide the tasks into several groups, that group is collectively called artefact or subsystem. Dividing the system into subsystems will help in the development of a better version of the system.

1.4.1. Functional Decomposition Diagram

A functional decomposition diagram may break any function or project into smaller steps. There are several applications for functional decomposition, from business and industry to computer programming and artificial intelligence.

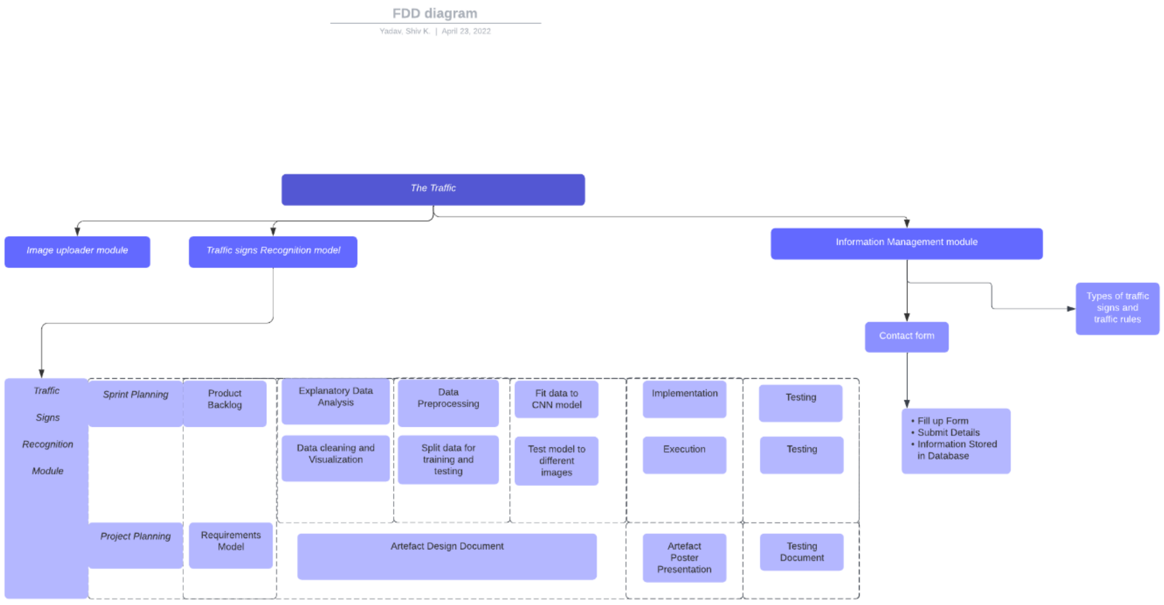

FDD diagram
Yadav, Shiv K. | April 23, 2022
The Traffic
Image uploader module
Traffic signs Recognition model
Information Management module
Types of traffic signs and traffic rules
Contact form
• Fill up Form
• Submit Details
• Information Stored in Database
Traffic Signs Recognition Module
Sprint Planning
Product Backlog
Explanatory Data Analysis
Data Preprocessing
Fit data to CNN model
Implementation
Testing
Data cleaning and Visualization
Split data for training and testing
Test model to different images
Execution
Testing
Project Planning
Requirements Model
Artefact Design Document
Artefact Poster Presentation
Testing Document

Figure 14 Functional Decomposition Diagram

1.4.2. System Explanation

Traffic signs recognition system, which helps in recognizing traffic signs. I have made this system so that everyone can use it. So it does not requires any user authentication. Users can go directly to the web page and see the instruction on how to use the system for recognition. Users can upload an image of a traffic sign from the file explorer and wait for a few seconds to get the result. The system processes the image internally with the help of a machine learning model; the model predicts the label of that particular image and prints the name of the matched title. The uploading of just one picture at a time is possible for users. The system also contains a contact form where users can give feedback or suggestions about anything related to the traffic sign system, the webpage, or any problem they face while using this web app. This system may be beneficial for students also. They can get knowledge about traffic rules and types of traffic signs. They can even upload an image of a traffic sign to get the name of traffic signs. The system is user-friendly and accessible.

1.4.3. Subsystems

1.4.3.1. Image Upload module

This sub-system gives the basic information of the system where users have to upload an image and go through further processing. With changes in lighting conditions, these useless interference zones will have a greater impact on traffic sign identification, increasing the computational complexity of the training network and the rate of traffic sign misrecognition. As a result, image preparation is required.

Methodology: Scrum

Planning: Use case Diagram, Activity Diagram

Design tools: Figma (UI/UX)

Programming Language: Python

Python Framework: Flask (Frontend & Backend)

Coding tools: Visual Studio Code

Test plan: Usability testing

1.4.3.2. Traffic sign recognition module

This is the main sub-system of this project which addresses the academic questions as it helps in recognizing the traffic sign. For this process, several photos are chosen at random from the testing set samples and preprocessed before being fed into the trained network model. The network model outputs the recognition results, displaying the meaning of traffic signs with the highest likelihood.

The statistical recognition results are achieved by comparing the output results to the real reference meanings.

Methodology: Scrum

Design tools: Figma(UI/UX)

Frontend and Backend Development: Flask

Data Requirement: a dataset of pre-collected traffic signs data

Development tools: Visual Studio Code, Jupyter Notebook

Image classification: Convolutional Neural Network (CNN)

Test Plan: Usability testing, Unit testing

1.4.3.3. Traffic Rules and Types module

This sub-system provides the general information which would be knowledge-improving factors for an individual about traffic signs. Traffic signs can be categorized as regulatory, warning, and guide signs. Regulatory signs are often used to inform drivers of their rights and responsibilities. A number of the most typical regulation signs, such as the Stop Sign, and Speed Limit Sign, are likely to be seen on any given route. Drivers are alerted to an unexpected or dangerous road condition or traffic situation with warning signs. This means that misinterpreting a warning sign can be deadly.

Some examples of warning signs are curved with a speed limit, reverse turn, one-way, etc. Travelers can get mileage and directional information from guide signs. They're usually green with white text, although they can be any color. Highway routes, mile markers, exit and entry signs, etc are some examples of guide signs.

Design tools: Figma(UI/UX)

Development tools: Visual Studio Code

Frontend and Backend Development: Flask

Test tools: Usability testing

1.4.3.4. Information Management Module

This subsystem deploys a contact form that includes a name field, email field, and messages field. Users can send their queries and system-related questions through the contact form. The messages are stored in the database. The database used here is ‘MySQL’ and in this system, the database is only used for the contact form. The information is stored in the proper format. The information is only accessible by the developer or admin of this system. The information stored in the database can be used for further different work such as advising traffic police about traffic signs, working on awareness programs on the topic of traffic signs and their impact, etc.

Design tools: Figma (UI/UX)

Planning: ER Diagram

Development tools: Visual Studio Code

Frontend Development: Html, tailwind CSS

Backend Development: Flask

Database: MySQL

1.5. Academic Question

How can AI and machine learning help to assist in safe driving practices and enhance safety in complicated traffic signals?

1.5.1. Explanation of Academic Question

Nowadays we can see many road accidents happening day by day. There are many reasons for that accidents, but one of them is not recognizing traffic signs properly. There are many easily understandable traffic signs, but some are very complicated to understand, and even we don’t know their name. These types of traffic signs do not show any impact on a general people, but they impact the safe driving practices of drivers. As a result, drivers compromised their safety and lost their lives without proper signs of recognition. So I have made a system that solves this problem by helping in recognizing traffic signs. Hence I have a made a question that gives the information about both the problem and solution and the question is **How can AI and machine learning help to assist in safe driving practices and enhance safety in complicated traffic signals?**

1.6. Scope and limitation of project

Scope of project

A project aims to create a system that can recognize and classify road and traffic signs to develop an inventory to help transportation authorities update and maintain traffic signs. It will also enable users to enhance safe driving practices and drive accordingly. Educational materials like users can get to know the several traffic rules and about different types of traffic signs. But said, like this, users will recognize all the traffic signs, so those project developers clarify that they will have **the scope that will allow recognizing all the possible traffic signs.** The project's content will be raised according to the context within the report, but there arises a problem of **limitations** that are mainly not specified by projects.

Limitations of project

A project owner allows one user to use its system to check whether all the things that are said initially are completed or not. The project has certain scope. Will all scopes succeed or not? there always arises a problem that problem is **limitations.** Some of the limitations of the project are mentioned below:

¶ Since the system is about recognizing the traffic signs but it recognizes even the image different than traffic signs.

¶ If the faded image or image has a different background color, it will recognize it but give the wrong name of traffic signs.

¶ The system will not be able to detect the labeled class name if the image have been have a value of transparency.

1.7. Report Structure

Table 3 Report Structure

Elements	**Explanation**
Title page	This section includes the project title and information about the student and his/her supervisor and reader.
Table of contents	It includes the information about every heading and subheading of several aspects like page number, indexing, etc
Introduction	This section includes project briefing, overall working system information, problem domain, and its solution. This section also includes the information on AI implementation i.e. machine learning and algorithm used etc. It also includes the main aspect of the project or system i.e, Aims, objectives, and most importantly Academic Questions and their explanation.
Literature Review	It includes the research papers about the similar system or AI algorithm which are going to be implemented in this system.
Project Methodology	It includes the way of doing our project, and which methodology we are going to use for the system like scrum, waterfall, etc.
Tools and Technology	It includes the programming language and where do we code of backend and frontend and which tools like Github, we are going to use for updating progress.

Artefacts Design	It includes the design of every subsystem like the use case diagram, activity diagram, wireframe, ERD diagram, sequence diagram, etc. it also includes the testing part of the system.
Conclusion	It includes all the answers to aims, objectives, and especially academic questions.
Critical evaluation	It includes the good and bad aspects of the development of the system.
Reference and bibliography	It includes references to all the research we have done throughout the project.
Appendices	It includes the extra information about the system like the user manual, system configuration, etc

2. Literature Review

The facts of our legal challenge are sometimes all we have. We have no notion what the legal problem is, and we may not even be aware of the legal subject or topic. When conducting preliminary research, we strive to get a wide overview of the relevant sources to gain a better knowledge of the issues and to keep track of important cases and statutes.

2.1. Initial Research

There are two basic steps in the identification of traffic signs: detection and recognition. There are three types of research groups in 'detection.' The first set of researchers believes that traffic sign colors are significant information for detecting and classifying traffic signs. The second group argues that just traffic sign shape may be used to detect traffic signs, while the third group believes that color and shape together form the backbone of any road sign detection. As a result, there are three basic ways to detect traffic signs: detection based on color information, detection based on shape information, and detection based on both color and shape information (Cao, et al., 2019).

2.1.1. Color Based Detection of traffic signs

(KhabiriKhatiri, et al., 2021) suggests that pixels in a digital image were segmented into object pixels and background pixels using thresholding. The method works by computing the distance between a pixel color and a reference color in RGB space. If the unknown pixel is close enough to the reference color, it is deemed an object pixel.

2.1.2. Shape-Based Detection of traffic signs

When colors are lacking or difficult to identify, shape-based techniques could be a useful substitute. Shape-based approaches should be able to avoid problems with calling colors for sign recognition while also being robust enough to handle in-plane transformations like translation, scaling, and rotation. These strategies have taken a lot of work to develop, and the results are highly promising.

¶ (Karthika & Murugan, 2020) suggests that HOG (Histogram of Oriented Gradients) is a widely used shape descriptor used for traffic sign recognition. Different strategies for identifying and acknowledging traffic signs are discussed by (Karthika & Murugan, 2020). The shading and shape inspection modules are included in the recognition module.

2.1.3. Color-Shape-Based Detection of traffic signs

It is feasible to use both strategies to detect traffic and road signals by using a mix of color and shape. An adaptive hybrid strategy, on the other hand, can use one technique in certain situations and the other in other situations. Even if this adaptive strategy isn't used, integrating color and shape in any sign detection system has the benefit of utilizing data from both sides of the problem. Because color and shape both communicate information that should not be overlooked, it is also possible to avoid a slew of issues and drawbacks.

¶ (Swapna, et al., 2021) detected the route advice sign using the HSV color space (RGS). Hue, saturation, and value all have automatic thresholds. To select the RGS candidate based on size and shape, a connected component analysis is used.

2.2. Introduction to literature review

Road and traffic signs that use visual/symbolic language describing the road(s) ahead that can be interpreted by vehicles driver are included in this system. In this system, the phrases are interchangeable, and they may also occur together as "road traffic signs" elsewhere. They provide information to drivers that makes driving safer and more convenient. **The essence of this literature review is to focus on how traffic signs recognition system helps in safe driving practices based on an image of traffic signs**.

2.3. Body

2.3.1. ANFIS Model for Image Recognition

The purpose of (Stojcic, et al., 2020) was to investigate the capacity of a picture to detect traffic signals using the Adaptive Neuro-Fuzzy Inference System (ANFIS) model, which combines fuzzy logic with artificial neural networks principles. (Stojcic, et al., 2020) the technique takes advantage of the geometric features of traffic sign symbols as the input value of ANFIS variables. It is recommended to build five distinct models for categorizing the provided sign. The final determination is made using the bulk of the ANFIS model's outputs, and the technique demonstrated a high degree of recognition accuracy and flexibility. (Stojcic, et al., 2020) discusses how to recognize a traffic sign without first obtaining, discovering, and collecting pictures, and video data. (Stojcic, et al., 2020) proposes a technique for detecting rectangular traffic signals based on their geometry. The suggested technique relies on ANFIS models that conduct recognition separately. The benefit of the proposed strategy is that after experimentation, it is feasible to pick several models (sections) as needed to reduce the final identification error. The research does not investigate real-world traffic situations or the whole identification process but rather explores recognition concepts using existing, isolated sign pictures.

Mathematics of ANFIS Theory

A typical fuzzy rule written by Takagi-Sugeno can be written as:

$$\text{IF } x \text{ is A AND } y \text{ is B, THEN } z=f(x,y),$$

Where A and B are the corresponding value to the input variables x and y and z represents the consequence rule. It can be 0 or first-degree polynomial. ‘IF’ represents hypothetical rules.

It consists of five layers as shown below in the figure:

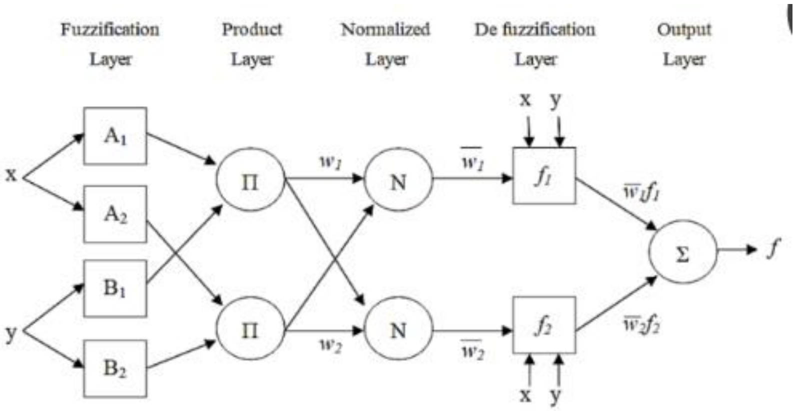


Figure 15 ANFIS MODEL

The first layer represents corresponding fuzzy functions to input variables i.e, the fuzzification layer. The parameters of these layers can be adjusted during the training phase. Input variables X can be defined as:

$\mu_{A_j}(x)$, where *j* (*j*=1,..., 2)

The second layer performs the multiplication of AND operation. The output form of ith form is :

$$w_i = \mu_{A_i}(x) \times \mu_{B_i}(y)$$

The results of the 2nd layer equal the minimal value of two inputs.

The output comes from the second layer is normalized in the third layer where their hidden layers had some mathematical function like this:

$$\bar{w}_i = \frac{w_i}{w_1 + w_2} \qquad (1)$$

Each node in the fourth layer is an adaptable node that performs the following function:

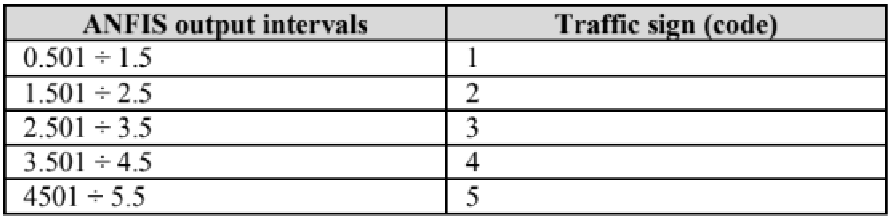
$$\bar{w}_i f_i = \bar{w}_i \left(p_i x + q_i y + r_i \right) \qquad (2)$$

Where pi, qi, and ri are parameters of inference. The sum of all inputs is calculated in fifth layers which are given by:

$$f = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \qquad (3)$$

Let's see some of the ANFIS output:

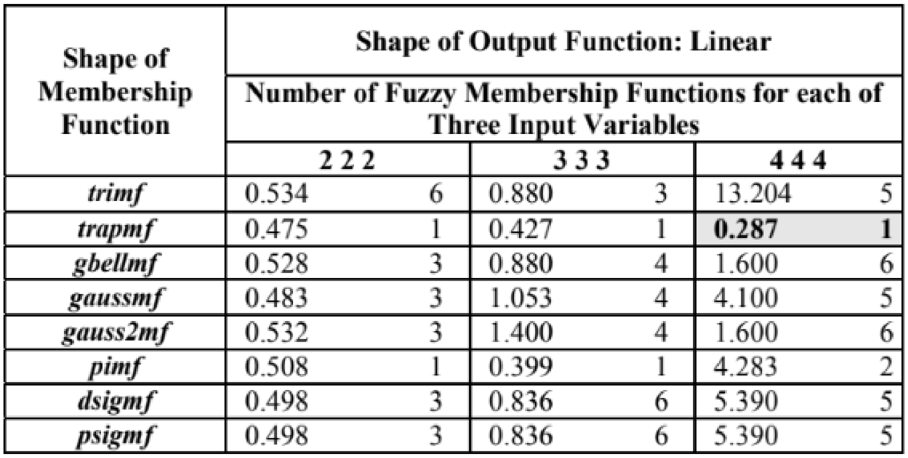
Table 4 ANFIS output For a single input



ANFIS output intervals	Traffic sign (code)
0.501 ÷ 1.5	1
1.501 ÷ 2.5	2
2.501 ÷ 3.5	3
3.501 ÷ 4.5	4
4501 ÷ 5.5	5

In both testing and training, the outputs of individual ANFIS models should be categorical variables that define the traffic sign.

Table 5 ANFIS model 2 output analysis

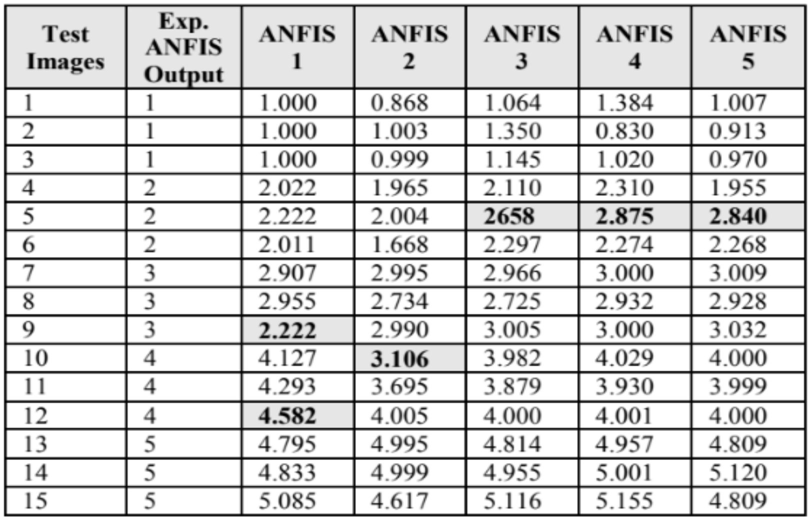


Shape of Membership Function	Shape of Output Function: Linear					
	Number of Fuzzy Membership Functions for each of Three Input Variables					
	2 2 2		3 3 3		4 4 4	
trimf	0.534	6	0.880	3	13.204	5
trapmf	0.475	1	0.427	1	**0.287**	**1**
gbellmf	0.528	3	0.880	4	1.600	6
gaussmf	0.483	3	1.053	4	4.100	5
gauss2mf	0.532	3	1.400	4	1.600	6
pimf	0.508	1	0.399	1	4.283	2
dsigmf	0.498	3	0.836	6	5.390	5
psigmf	0.498	3	0.836	6	5.390	5

Table 4 shows that the ANFIS 2 model's average test error is marginally greater than the preceding case, at 0.287 with the fault of recognizing only one sign. For each input of the trapezoidal form, the chosen model has four fuzzy functions.

Let's see the final summary of the ANFIS model:

Table 6 Comparing five ANFIS model



Test Images	Exp. ANFIS Output	ANFIS 1	ANFIS 2	ANFIS 3	ANFIS 4	ANFIS 5
1	1	1.000	0.868	1.064	1.384	1.007
2	1	1.000	1.003	1.350	0.830	0.913
3	1	1.000	0.999	1.145	1.020	0.970
4	2	2.022	1.965	2.110	2.310	1.955
5	2	2.222	2.004	**2658**	**2.875**	**2.840**
6	2	2.011	1.668	2.297	2.274	2.268
7	3	2.907	2.995	2.966	3.000	3.009
8	3	2.955	2.734	2.725	2.932	2.928
9	3	**2.222**	2.990	3.005	3.000	3.032
10	4	4.127	**3.106**	3.982	4.029	4.000
11	4	4.293	3.695	3.879	3.930	3.999
12	4	**4.582**	4.005	4.000	4.001	4.000
13	5	4.795	4.995	4.814	4.957	4.809
14	5	4.833	4.999	4.955	5.001	5.120
15	5	5.085	4.617	5.116	5.155	4.809

We can see the difference in value in each ANFIS model and only a few models are making errors otherwise remaining are performing accurately.

2.3.2. OpenCV Techniques for Road Sign Recognition

The purpose of (Botekar & Mahalakshmi, 2017) is to examine the challenge of traffic sign identification using deep learning approaches. design is a suggested technique that consists of three modules: a detection module (PostNet) for identifying the traffic sign inside a static picture, a classification module (PatchNet) for categorizing the discovered image patch, and a temporal filter for correcting the identification results. PostNet is a convolutional neural network for binary object recognition that treats all traffic signs as one class and the backdrop as another. In contrast to previous work that relied on static images to identify traffic signs, the suggested temporal filter makes use of contextual information to recover the missing detection zone and rectify the inaccurate classification. (Botekar & Mahalakshmi, 2017) proposed several newer approaches, such as extracting road sign features using histogram-oriented gradient features. Additionally, Hough Transform (HT) is a perception technique that is used to identify circles and rectangles with a high computational cost. The great majority of current systems are composed of classifiers constructed from hand annotations of actual photos. (Botekar & Mahalakshmi, 2017) the objective was to develop a system for detecting and classifying traffic signs. The system is separated into two phases: a detection phase and a classification phase. They had used the database of Indian road signs. The testing findings indicate that the system performs well for ideogram-based sign identification, with an average speed of 25 frames per second and an accuracy of up to 94 percent.

Mathematics Involved

The system overview used by (Botekar & Mahalakshmi, 2017) is presented below:

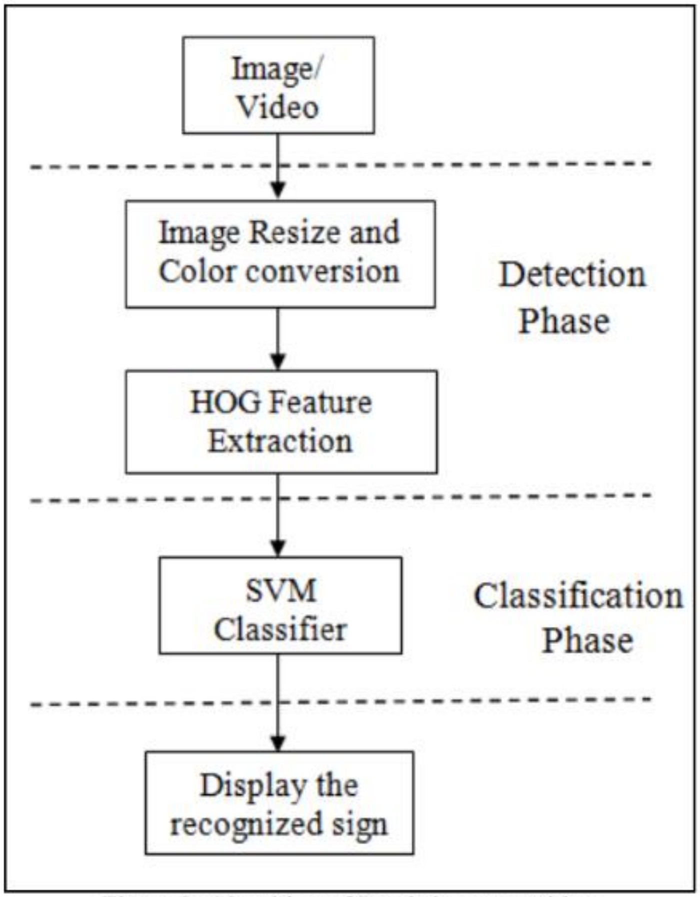


Figure 2: Algorithm of Road sign recognition

The extraction of information includes the three steps

First step: the first step is to calculate the magnitude and direction of images i.e.gradient magnitude and gradient direction which is calculated as:

$$G(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \qquad (1)$$

$$\alpha(x,y) = tan^{-1}\frac{G_y(x,y)}{G_x(x,y)} \qquad (2)$$

It is the process of calculating how fast the value of an image and its direction are changing.

Second Step:

The development of cell histograms is the second stage in the process. Every pixel in the cell casts a weighted vote for an orientation-based histogram channel based on the results of the gradient computation. The histogram channels are evenly spaced from 0 to 180 degrees or 0 to 360 degrees, and the cells can be radial or rectangular. Figure 17 depicts a visual picture of cell development in the image.

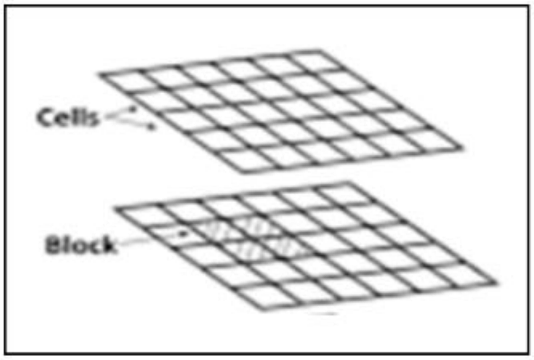


Figure 16 formation of cells

Step 3:

The construction of a descriptor block and normalization is the third stage. The gradient strengths must be locally normalized to account for changes in illumination and contrast, which necessitates grouping the cells together into larger, geographically connected blocks. The block normalization with the descriptor feature vectors is shown in Figure 18. The components of the normalized cell histograms from all of the block regions are concatenated to form the HOG descriptor.

After that, the descriptor feature vectors are sent into a classifier.

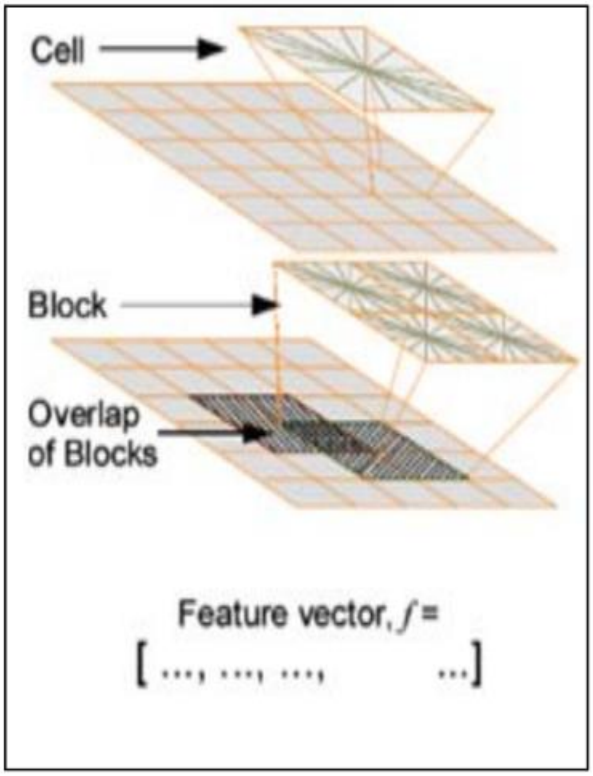
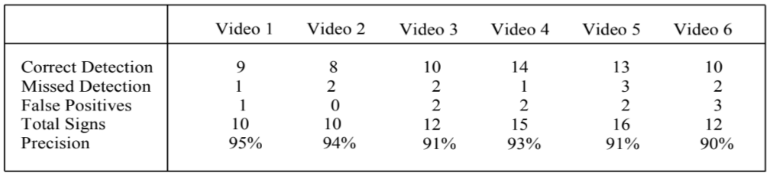


Figure 17 Normalization of Block

Where the number of features can be calculated by using the following formula:

$$N = \left(\frac{R_{width}}{C_{width}} - 1\right) * \left(\frac{R_{height}}{C_{height}} - 1\right) * B * H \qquad (3)$$

We can see the performance analysis done by (Botekar & Mahalakshmi, 2017) in figure 19.



	Video 1	Video 2	Video 3	Video 4	Video 5	Video 6
Correct Detection	9	8	10	14	13	10
Missed Detection	1	2	2	1	3	2
False Positives	1	0	2	2	2	3
Total Signs	10	10	12	15	16	12
Precision	95%	94%	91%	93%	91%	90%

Figure 18 Performance analysis

2.3.3. Lightweight SP-CNN for Image Classification

As part of the recognition process, a lightweight spatial pyramid convolutional neural network (SP-CNN) model that enables visual categorization of traffic signs may be used, according to the authors (Rachmadi, et al., 2018). The ResNet (residual network) CNN architecture, which was first used to classify images in CIFAR10, serves as the basis for the lightweight SP-CNN classifier. The SP-CNN classifier is based on the ResNet CNN architecture. (Rachmadi, et al., 2018) suggested a classifier that is composed of five concurrent convolutional networks, each of which uses a spatial pyramid structure to investigate a cropped region. They have a 12.5 percent overhang on each axis of the level 1 spatial pyramid region's layout, which allows for more fluid transitions between the areas cut in the level 1 spatial pyramid design. Using the NAG (Nesterov Accelerated Gradient) training technique, the suggested classifier was trained using the CIFAR10 weights that were fine-tuned to get the best results. Experiments on the GTSRB (German Traffic Sign Recognition Benchmark) dataset have shown that the lightweight SP-CNN version has a 99.70 percent accuracy rate and an execution time of 60 milliseconds. The suggested classifier, when compared to earlier algorithms, achieves a high degree of accuracy while requiring fewer parameters.

2.3.4. Detection of Image Under Severe Weather Condition

(Wan, et al., 2021) proposed TS-**Yolo** (You Only Look Once), In extreme weather circumstances, a CNN-based algorithm for accurate traffic detection has been developed. To begin, (Wan, et al., 2021) supplemented the data with a copy-paste method, producing a lot of new examples based on known traffic-sign occurrences. **MixConv**, which is based on YoloV5, was utilized to perform the convolutional process with varied kernel sizes, allowing for the easy capture of patterns with varying resolutions. Thus, even though the traffic signs look distorted in form and color at various viewing angles and distances, they may still be spotted. The attentional feature fusion (AFF) module was also used to fuse features that were dependent on attention from the same or other layers, including short and long skip connections, as well as to perform the initial fusion with itself. Thus, in complicated road circumstances or varied light situations, even partly obscured traffic signs may be detected and positioned more precisely using the upgraded model. The AFF module's use allowed for the capturing of high-resolution details on the targets. When (Wan, et al., 2021) used both MixConv and AFF in the TS-Yolo model, the precision was 83.73 percent.

2.3.5. Recognition of Image-Based on CNN

(Tang, et al., 2018) demonstrated a technique for classifying traffic signs that are based on a Convolutional Neural Network architecture and a dataset augmentation approach. A dataset imbalance is a difficulty for a machine learning system since it results in a bias toward classes with more examples. To mitigate the effect of imbalance, (Tang, et al., 2018) use dataset augmentation technologies to provide additional data for classes with fewer samples. This design achieved a validation accuracy of 99.6 percent and a test accuracy of 95.18 percent. They evaluate the strategy using an appropriate picture; experimental findings indicate that the method is successful at classifying traffic signs. The featured work is divided into two sections. First, they compare each dataset augmentation methodology, using the most significant transfer methods and excluding insignificant ones. Second, they attempt to classify using various neural networks and compare them to the two-stage ConvNet used in this article.

2.3.6. AC-RDF for Image Recognition Process

Precision detection of small traffic indicators is especially important for the safety of intelligent transportation systems. (Zhigang, et al., 2021) suggest a recognition system named attentive context region-based detection framework, which is based on a region-based approach (AC-RDF). A pointwise convolutional layer was used to concatenate the target and contextual information, resulting in the attentive context feature (Zhigang, et al., 2021). This feature was designed for the identification of microscopic traffic signals. Instead of focusing on the problem of recognizing small traffic signals in large areas using the GTSRB and GTSDB datasets, they employ the Tsinghua-Tencent 100K benchmark dataset. This is in contrast to previous work that used the GTSRB and GTSDB datasets. The framework for Attentive Context Region-based Detection (Zhigang, et al., 2021) was developed by the researchers (AC-RDF). The recall and accuracy values for the small-size group are 89.3 percent and 84.5 percent, respectively, while the recall and accuracy values for the medium-size group are 94.4 percent and 93.1 percent, and the recall and accuracy values for the large-size group are 88.2 percent and 92.7 percent.

2.4. Conclusion

Recognition of traffic signs is a critical component of an intelligent transportation system along the road, which includes speed limit signs, stop signs, yield signs, and merge signals, among others. The earliest investigation of traffic sign detection dates to 1984 in Japan. Numerous strategies have been proposed by various researchers to construct an effective traffic sign recognition system capable of mitigating and resolving all the above-mentioned concerns. The planned work's ultimate objective is to increase the accuracy and precision with which traffic signs are predicted. To accomplish the aims, deep learning methods can be used. Convolutional Neural Networks can be used to identify and categorize traffic signals, as well as to extract their characteristics from the information. Following feature extraction, the output layer is utilized to predict the pictures of traffic signs and, lastly, to identify the sign's name. The created systems, such as CNN, have shown promising results, with a test accuracy of almost 97 percent (Tang, et al., 2018).

3. Project Methodology

3.1. Introduction

Planning is very much crucial before beginning any project or any activity. Proper planning leads to excellent outcomes else there emerge different issues if work is begun without planning. There are different approaches which we may use according to our needs such as Scrum, waterfall model, etc. for this project planning and to work on a specific assignment, I have utilized Agile, a framework of the scrum. So there comes a concern about why just agile or why only Scrum, why not others.

3.2. Agile Methodology

Other software development approaches rely on the pre-designed process of learning the requirements from customers, organizing the development, creating the software, constructing the product, testing, and final result delivery. This takes a few months or years even. But Agile focuses on releasing developing knowledge and skills in a couple of weeks or the full bit of software in a month and a half.

With Agile best practices adopted throughout the company, they will provide excellent goods to the customers on time to make them pleased. Satisfied customers are the key reason behind the success of Agile. Throughout the Agile process, consumers are engaged in the entire process so that they can obtain information on their product which makes sure that they are fulfilling their expectations.

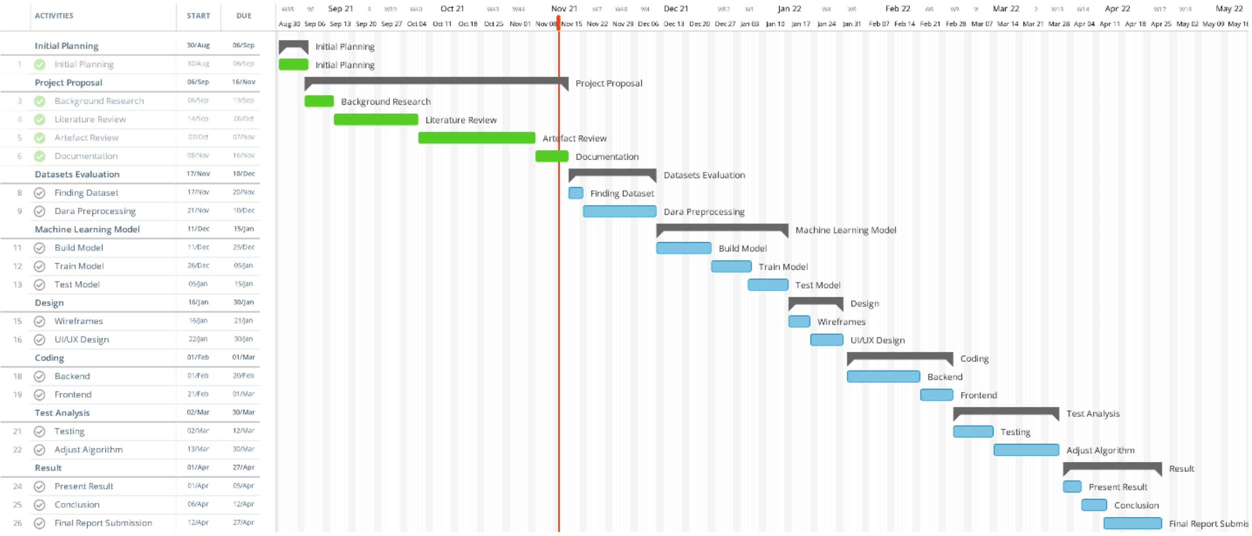
3.2.1. Scrum Methodology

Scrum is a well-known Agile approach that is continually evolving. As a procedure aimed to deliver high-quality outcomes promptly throughout a project, it is adaptive, fast, flexible, incremental, iterative, and effective. Scrum is responsible for ensuring that communication is open and honest throughout the project's lifespan.

To bring particular features to the testing teams before the finish of the development cycle, the full project is split down into smaller chunks. A modest operational software product may be supplied after each interaction and feedback from the stakeholder to expand or change the scope of the project in response. The Agile approach's basic concepts are summarised in these tenets.

Finally, the project is for consumers, not for us. As a consequence, the customer should get precisely what they want, although no one understands what they want at the outset of a project.

3.3. Initial Gantt Chart



	ACTIVITIES	START	DUE
	Initial Planning	30/Aug	06/Sep
1	✓ Initial Planning	30/Aug	06/Sep
	Project Proposal	06/Sep	16/Nov
3	✓ Background Research	06/Sep	13/Sep
4	✓ Literature Review	14/Sep	06/Oct
5	✓ Artefact Review	07/Oct	07/Nov
6	✓ Documentation	08/Nov	16/Nov
	Datasets Evaluation	17/Nov	10/Dec
8	✓ Finding Dataset	17/Nov	20/Nov
9	✓ Dara Preprocessing	21/Nov	10/Dec
	Machine Learning Model	11/Dec	15/Jan
11	✓ Build Model	11/Dec	25/Dec
12	✓ Train Model	26/Dec	05/Jan
13	✓ Test Model	05/Jan	15/Jan
	Design	16/Jan	30/Jan
15	✓ Wireframes	16/Jan	21/Jan
16	✓ UI/UX Design	22/Jan	30/Jan
	Coding	01/Feb	01/Mar
18	✓ Backend	01/Feb	20/Feb
19	✓ Frontend	21/Feb	01/Mar
	Test Analysis	02/Mar	30/Mar
21	✓ Testing	02/Mar	12/Mar
22	✓ Adjust Algorithm	13/Mar	30/Mar
	Result	01/Apr	27/Apr
24	✓ Present Result	01/Apr	05/Apr
25	✓ Conclusion	06/Apr	12/Apr
26	✓ Final Report Submission	12/Apr	27/Apr

Figure 19 Initial Gantt chart

4. Different technology and tools used for the project

It shouldn't seem like we're reinventing the wheel when selecting a project management solution. Even though there are hundreds of possibilities accessible, only a small number of them are employed by top technical teams.

That is not to argue that we should all rush out and start copying the likes of Apple, Facebook, and Amazon. Instead, it should be noted that there is no stigma associated with drawing inspiration from the greats to complete enormous projects and sell goods to millions of people.

4.1. Programming languages

for this project, I will **Python** language for development. Beginners and novices alike will find Python a breeze to learn and use. There are several programming languages, but Python is one of the most user-friendly due to its simple syntax and focus on natural language. Codes created in Python can be written and executed far more quickly than code produced in other programming languages. Not just only that my project related to artificial intelligence, several libraries will be required to develop a machine learning model.

4.2. Framework

I used the **Flask framework** of python because Flask web application code is in most circumstances clear. Flask is easy to get started with as a newbie since there is minimal boilerplate code for having a small project up and running.

4.3. Web Technologies

This section includes Bootstraps, Typescript, Javascript, Html5, tailwind, etc. I have used Html5, javascript as well as tailwind CSS for web development.

4.4. Database

Managing the information that we got from the users is stored in a database. For the database, I have used MySQL database. For connecting the database and python I have used SQLAlchemy where SQLAlchemy is a library that helps in the connection of Python and database.

4.5. UI/UX Design

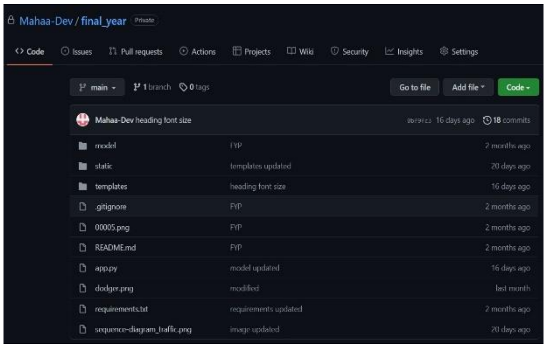
The most important aspect of any project is the design which includes user interface design, wireframe, pages design, and diagrams like use case diagram, sequence diagram, activity diagram, etc. And for the UI/UX design, I have used **Figma** tools. With the help of these designs, the development of the system will be easier and more fasten as we had already mapped the way to complete the project in time.

4.6. IDE

I have used VSCode for coding of frontend and backend whereas jupyter notebook for building a machine learning model. There are many benefits to using VSCode as it provides the IDE for any language.

4.7. Version control

It provides us a platform where we can change or update codes and all the information is stored where and when we updated a code. For that, I have used Github to manage my project or system.



Mahaa-Dev heading font size [illegible] 16 days ago 18 commits

model	FYP	2 months ago
static	templates updated	20 days ago
templates	heading font size	16 days ago
.gitignore	FYP	2 months ago
00005.png	FYP	2 months ago
README.md	FYP	2 months ago
app.py	model updated	16 days ago
dodger.png	modified	last month
requirements.txt	requirements updated	2 months ago
sequence-diagram_traffic.png	image updated	20 days ago

Figure 20 version control

5. Artefact Design

5.1. Diagram

5.1.1. FDD (Functional Deployment Diagram)

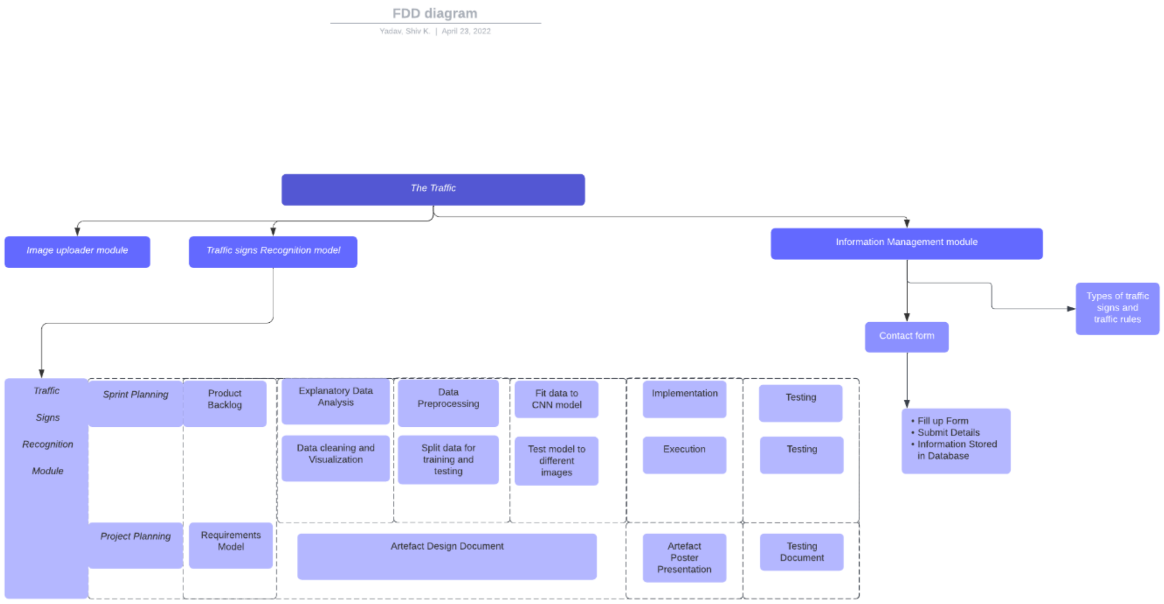

FDD diagram
Yadav, Shiv K. | April 23, 2022
The Traffic
Image uploader module
Traffic signs Recognition model
Information Management module
Types of traffic signs and traffic rules
Contact form
• Fill up Form
• Submit Details
• Information Stored in Database
Traffic Signs Recognition Module
Sprint Planning
Product Backlog
Explanatory Data Analysis
Data Preprocessing
Fit data to CNN model
Implementation
Testing
Data cleaning and Visualization
Split data for training and testing
Test model to different images
Execution
Testing
Project Planning
Requirements Model
Artefact Design Document
Artefact Poster Presentation
Testing Document

Figure 21 Functional Deployment Diagram of system

Image Uploader module (IU)

Table 7 Image uploader module

Requirement Code	Requirement Description	Use Case	Prioritization
IU-F-1.0	The system should allow the user to use an image from file explorer for recognition.	Image upload using file explorer	Must have

IU-F-1.0	The system should allow any format of images	Image format	Should have

Traffic Sign Recognizer model (TSR)

Table 8 Traffic signs recognizer model

Requirement Code	Requirement Description	Use Case	Prioritization
TSR-F-1.0	The system should allow the user to use uploaded images for recognition	Recognition using file explorer images	Must have
TSR-F-1.0	The system should use an ML model for any prediction	Machine learning model	Must have

Information Management Module (IM)

Table 9 Information management module

Requirement Code	Requirement Description	Use Case	Prioritization
IM-F-1.0	The system should allow the user to use the contact form to send messages	Messaging using the contact form	Must have
IM-F-1.0	The system should provide information about the types and rules of traffic signs.	View traffic rules	Should have

5.1.2. Use Case Diagram

Use case diagram of image uploader module

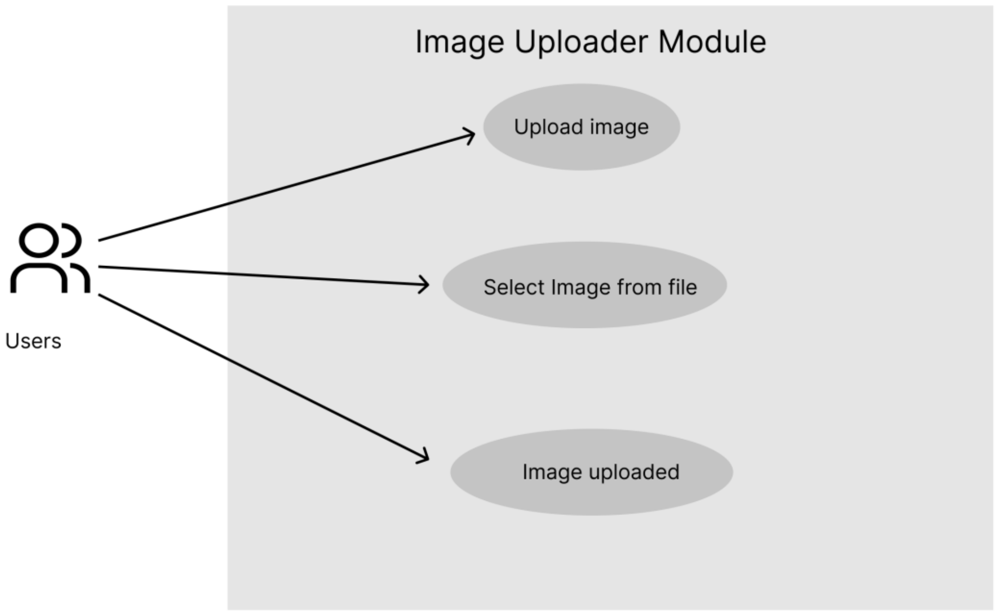


Figure 22 Usecase diagram of image uploader module

Use case diagram for Traffic signs recognition module

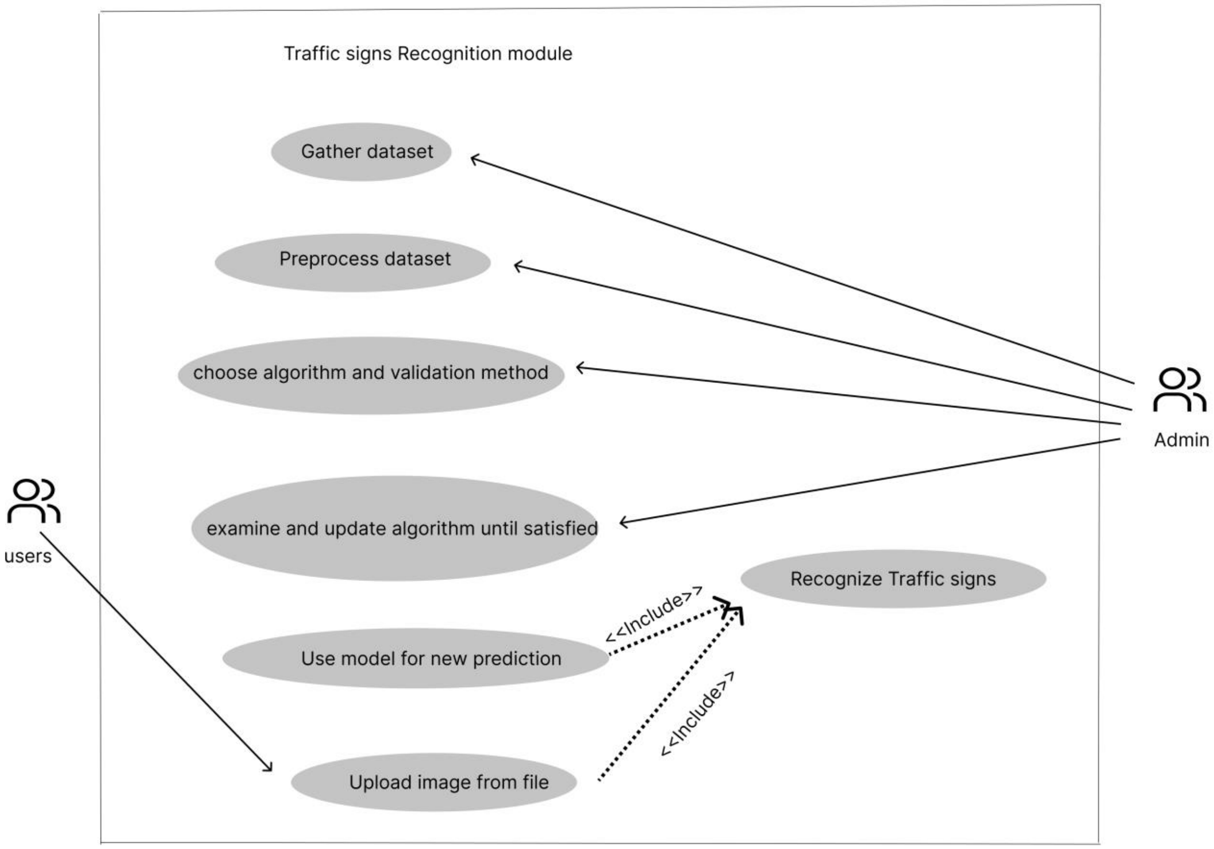

Traffic signs Recognition module
Gather dataset
Preprocess dataset
choose algorithm and validation method
examine and update algorithm until satisfied
Recognize Traffic signs
Use model for new prediction
Upload image from file
<<Include>>
<<Include>>
users
Admin

Figure 23 Use case diagram of Traffic signs recognition module

Use case diagram of the information management subsystem

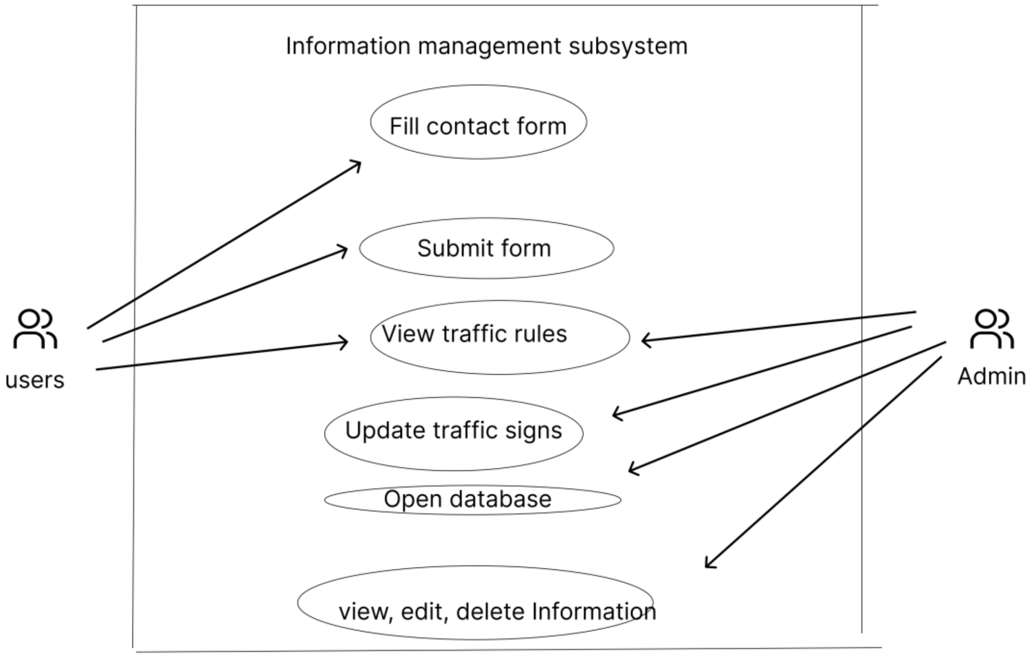


Figure 24 Usecase diagram of the information management subsystem

5.1.3. Activity Diagram

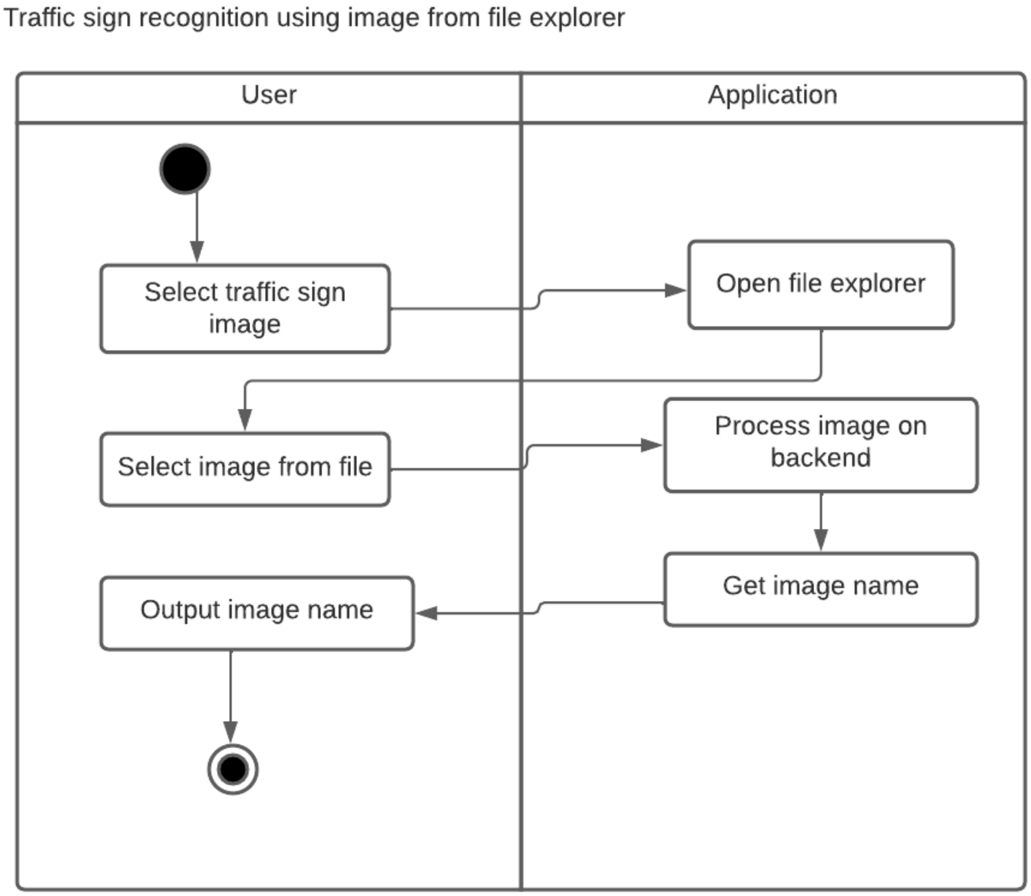


Figure 25 Activity diagram

5.1.4. Sequence Diagram

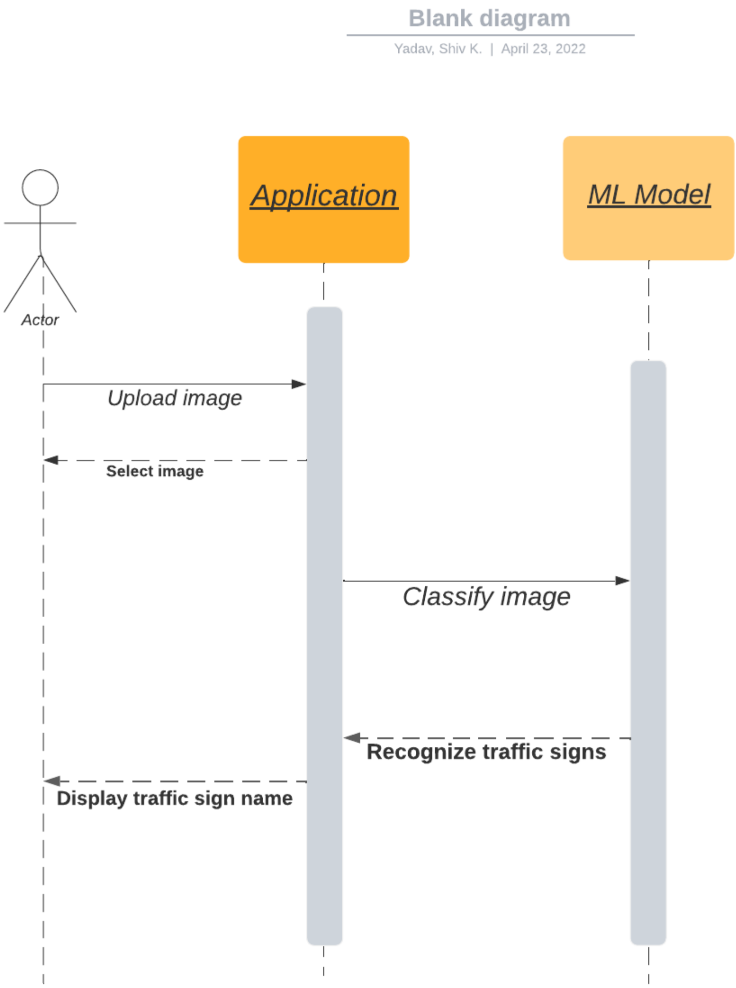


Figure 26 Sequence diagram

5.1.5. ERD Diagram

ERD diagram is used to display the information that is stored in the database which is used to show the data flow diagram. The name of the database used in this system is ‘**traffic**’ and it contains only one table. i.e., ‘**contacts**’

Table 10 Table of the traffic database

Contacts		
Keys	Field	Type
PK	sno	Int(20)
	name	text
	email	text
	msg	Varchar(50)
	date	datetime

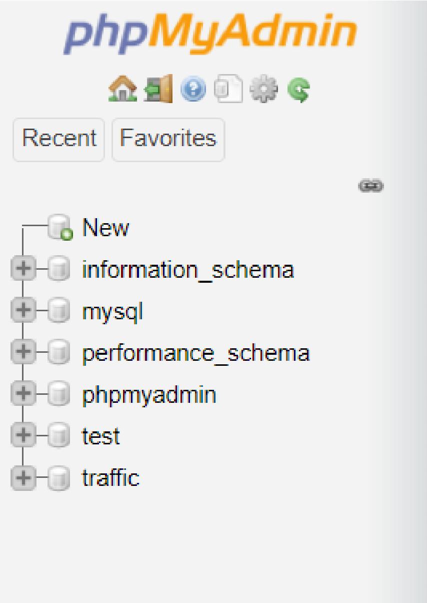
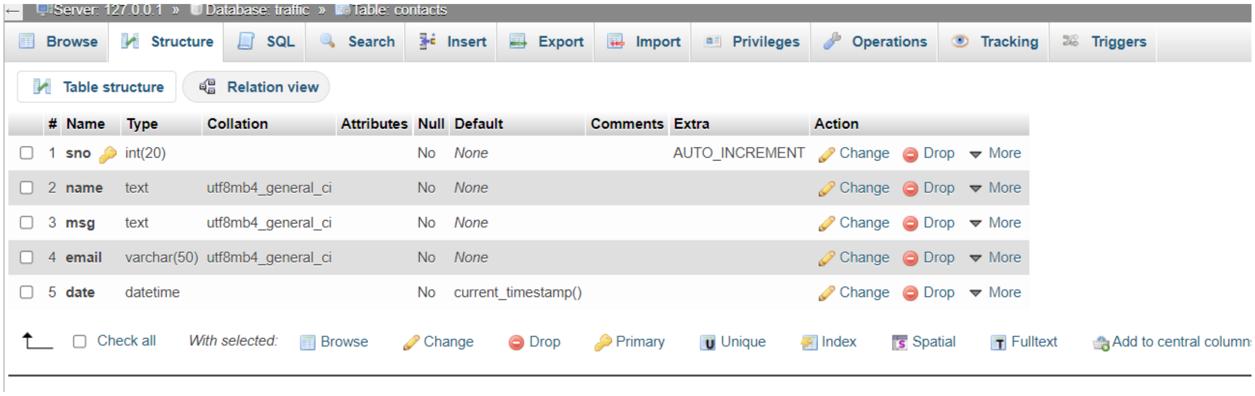


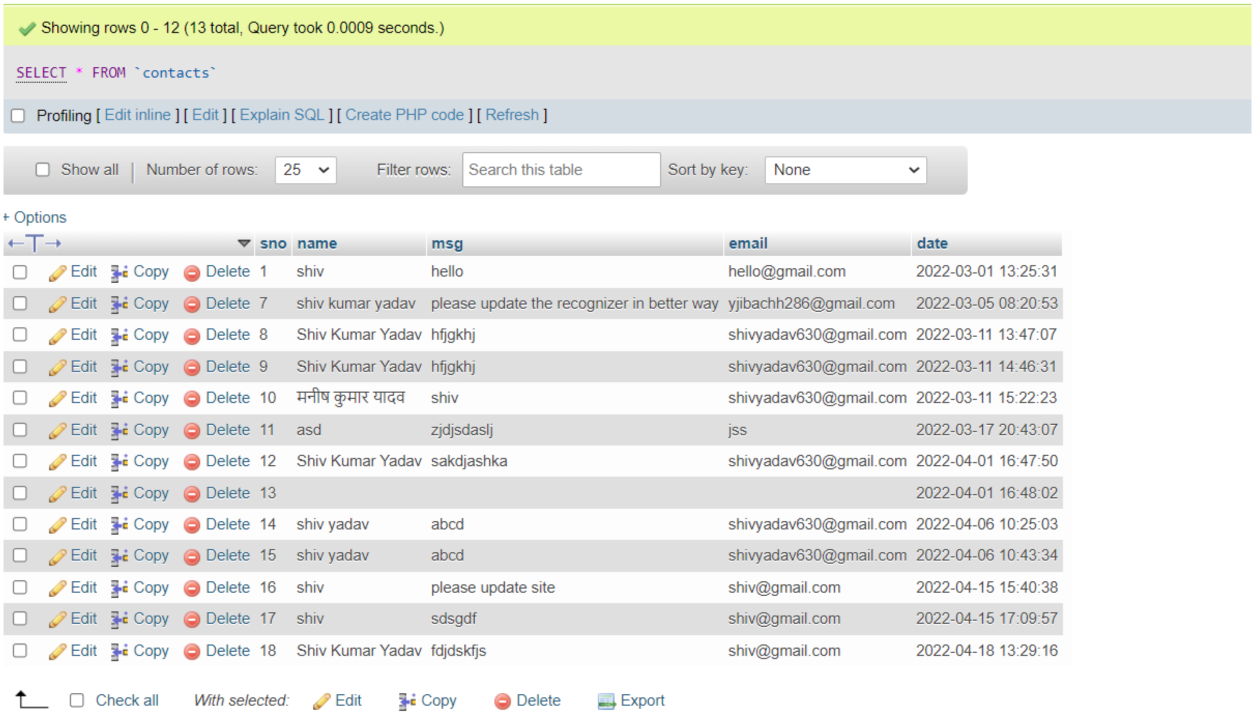
Figure 27 Screenshot of the database created 'traffic'


Server: 127.0.0.1 » Database: traffic » Table: contacts
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers
Table structure
Relation view

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**sno** 🔑	int(20)			No	*None*		AUTO_INCREMENT	Change Drop More
□	2	**name**	text	utf8mb4_general_ci		No	*None*			Change Drop More
□	3	**msg**	text	utf8mb4_general_ci		No	*None*			Change Drop More
□	4	**email**	varchar(50)	utf8mb4_general_ci		No	*None*			Change Drop More
□	5	**date**	datetime			No	current_timestamp()			Change Drop More

□ Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext Add to central column

Figure 28 screenshot of the structure of data stored in a database


✓ Showing rows 0 - 12 (13 total, Query took 0.0009 seconds.)

```
SELECT * FROM `contacts`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
+ Options

		sno	name	msg	email	date
□	Edit Copy Delete	1	shiv	hello	hello@gmail.com	2022-03-01 13:25:31
□	Edit Copy Delete	7	shiv kumar yadav	please update the recognizer in better way	yjibachh286@gmail.com	2022-03-05 08:20:53
□	Edit Copy Delete	8	Shiv Kumar Yadav	hfjgkhj	shivyadav630@gmail.com	2022-03-11 13:47:07
□	Edit Copy Delete	9	Shiv Kumar Yadav	hfjgkhj	shivyadav630@gmail.com	2022-03-11 14:46:31
□	Edit Copy Delete	10	मनीष कुमार यादव	shiv	shivyadav630@gmail.com	2022-03-11 15:22:23
□	Edit Copy Delete	11	asd	zjdjsdaslj	jss	2022-03-17 20:43:07
□	Edit Copy Delete	12	Shiv Kumar Yadav	sakdjashka	shivyadav630@gmail.com	2022-04-01 16:47:50
□	Edit Copy Delete	13				2022-04-01 16:48:02
□	Edit Copy Delete	14	shiv yadav	abcd	shivyadav630@gmail.com	2022-04-06 10:25:03
□	Edit Copy Delete	15	shiv yadav	abcd	shivyadav630@gmail.com	2022-04-06 10:43:34
□	Edit Copy Delete	16	shiv	please update site	shiv@gmail.com	2022-04-15 15:40:38
□	Edit Copy Delete	17	shiv	sdsgdf	shiv@gmail.com	2022-04-15 17:09:57
□	Edit Copy Delete	18	Shiv Kumar Yadav	fdjdskfjs	shiv@gmail.com	2022-04-18 13:29:16

Figure 29 information stored in a database

5.1.6. Wireframe

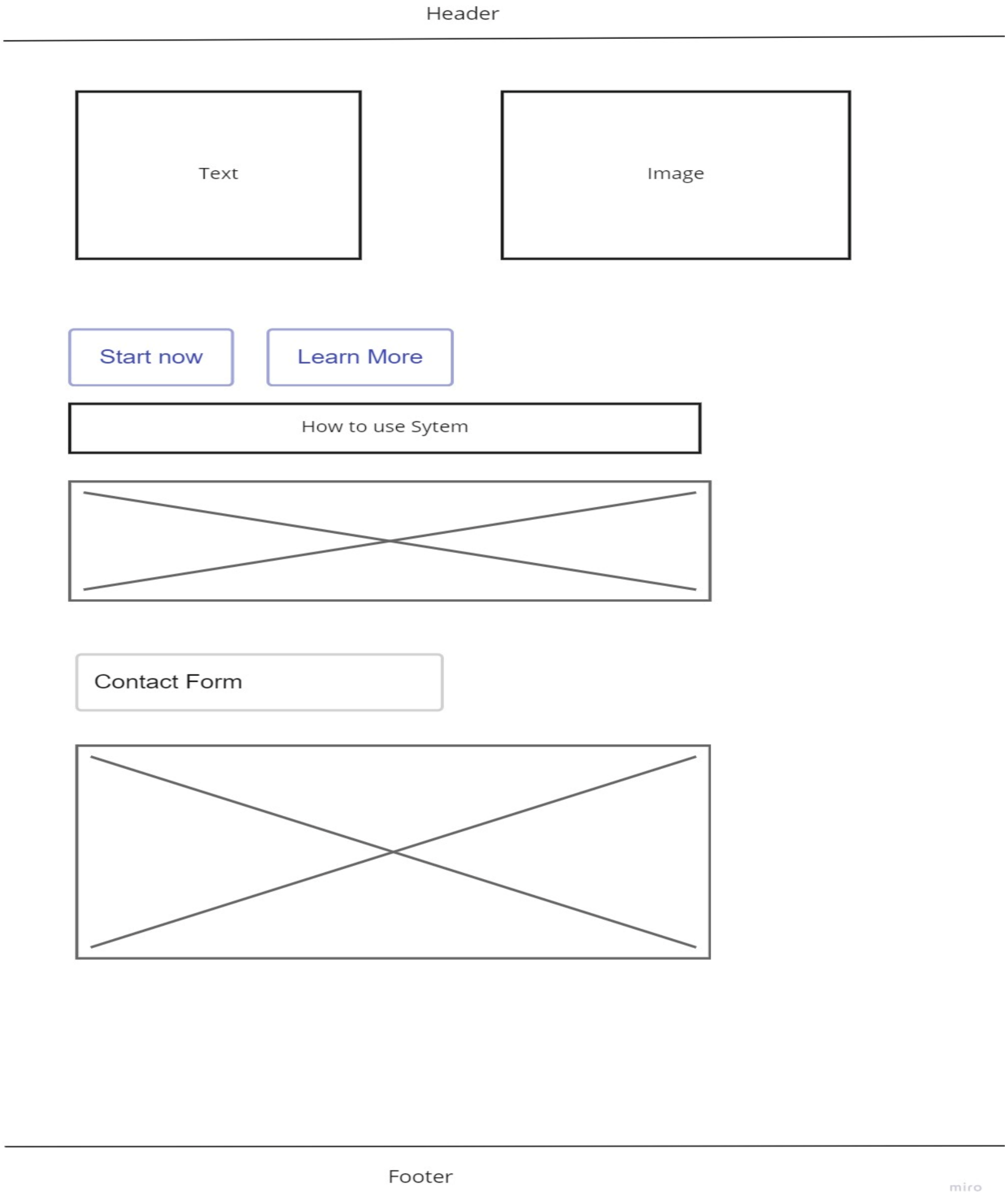


Figure 30 Wireframe -1 Home page

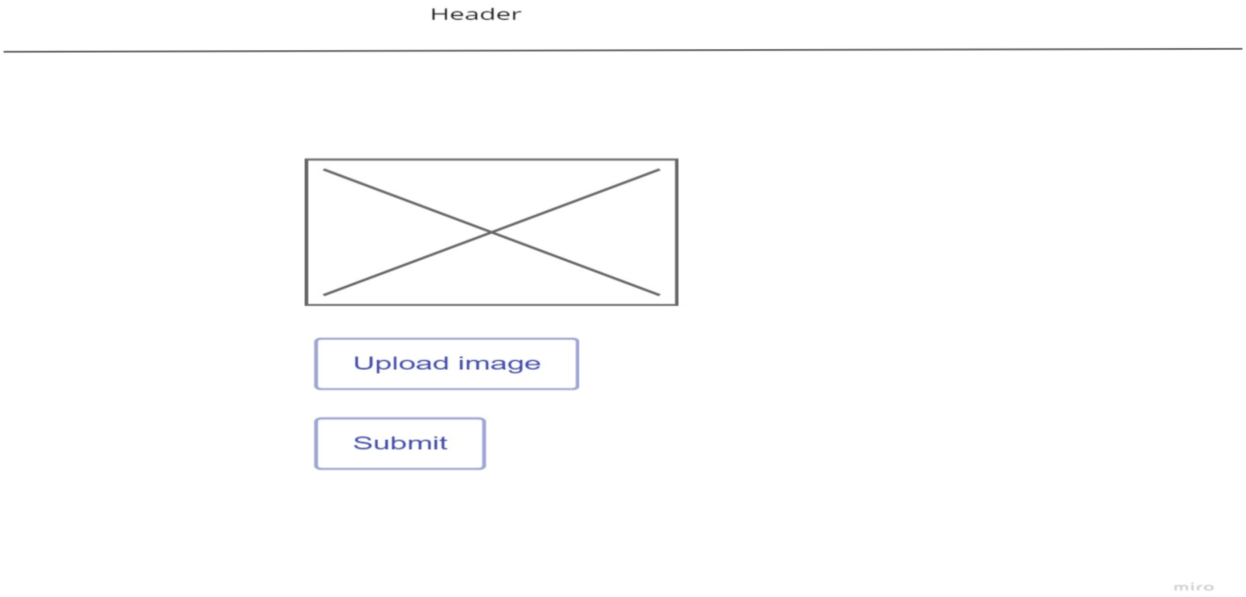


Figure 31 Wireframe-2 Image upload page

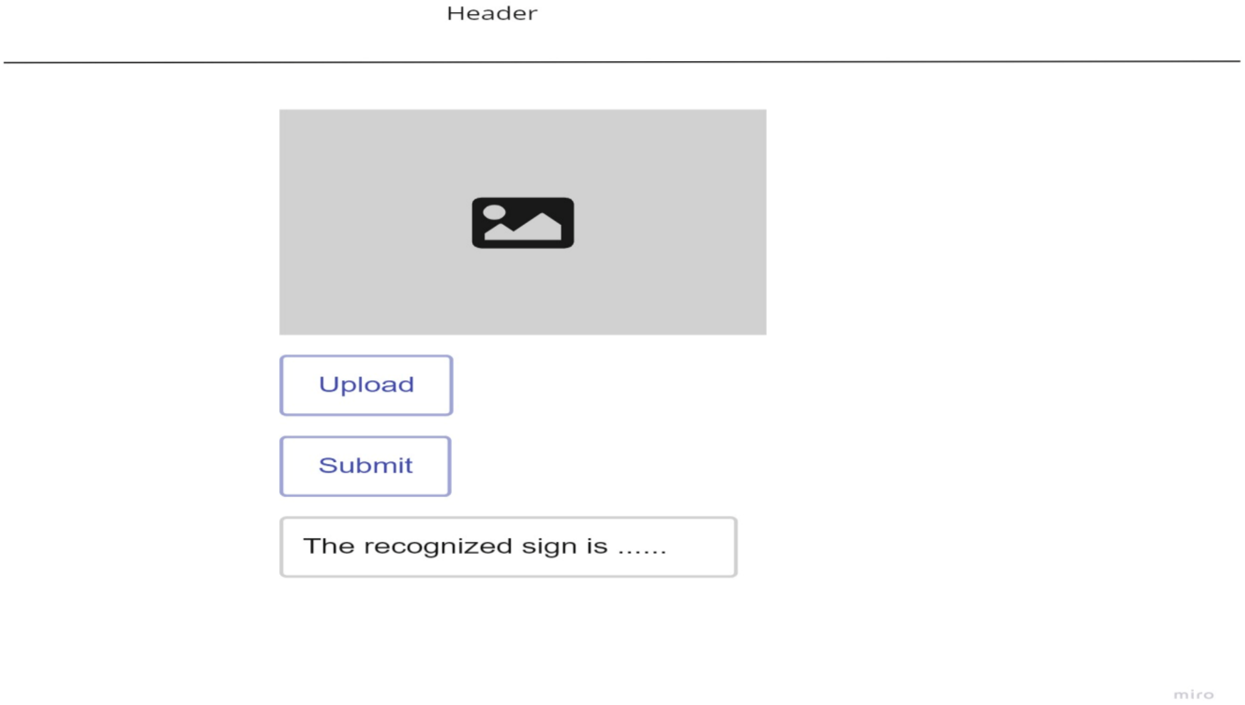


Figure 32 Wireframe-3 Image recognized page

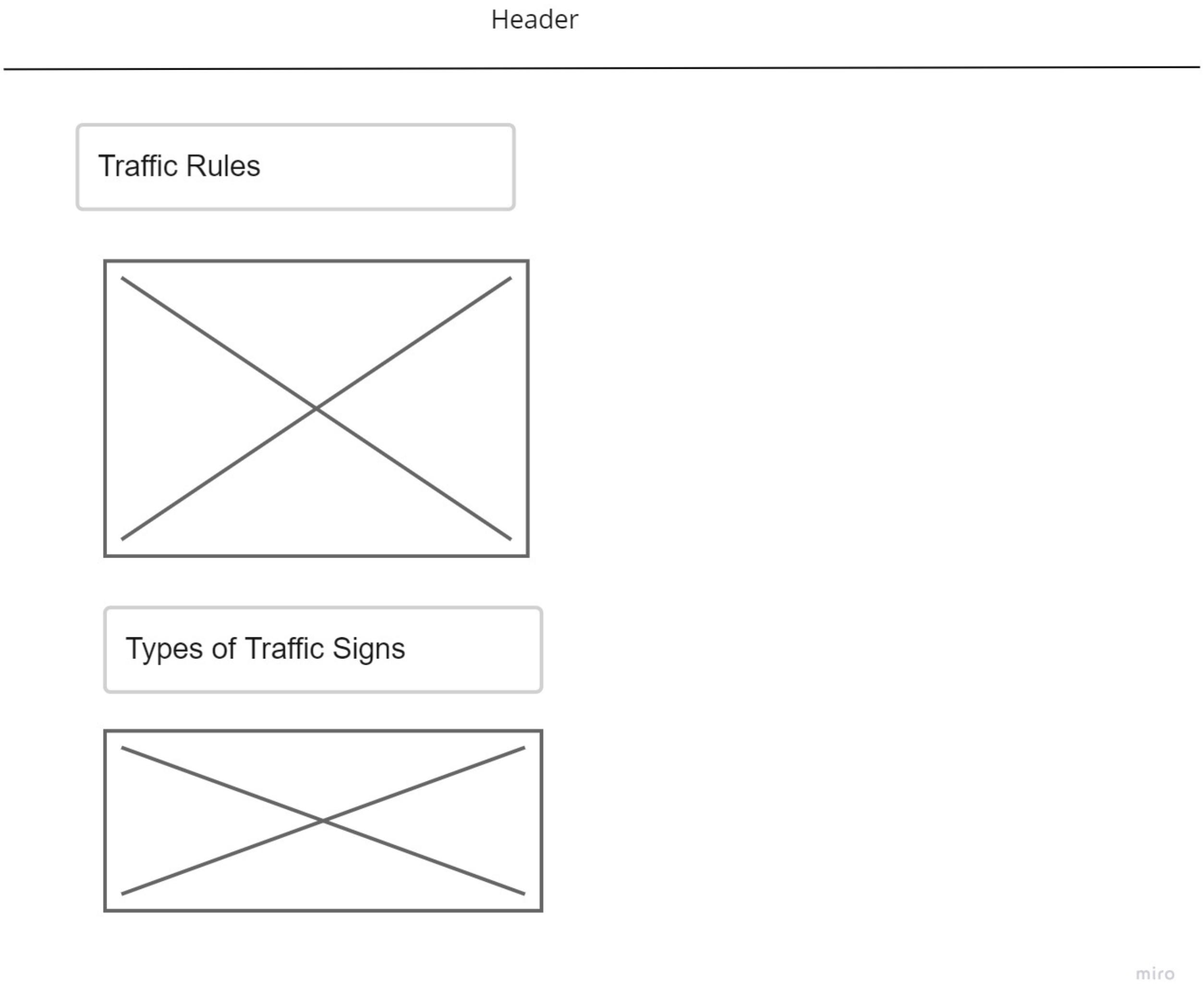
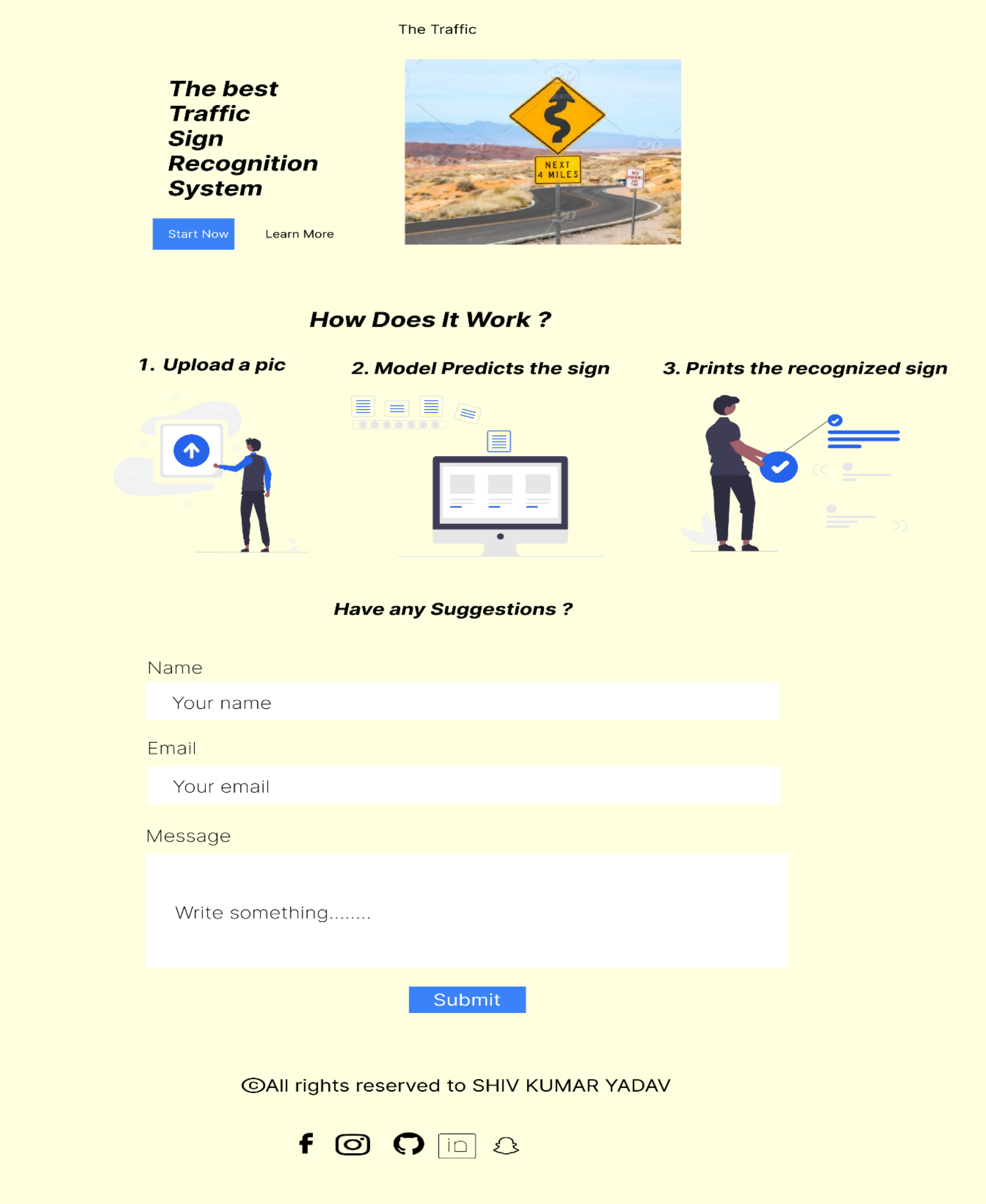


Figure 33 Wireframe-4 learn-more page

5.1.7. Design



The Traffic

The best Traffic Sign Recognition System

How Does It Work ?

1. Upload a pic

2. Model Predicts the sign

3. Prints the recognized sign

Have any Suggestions ?

Name

Your name

Email

Your email

Message

Write something........

©All rights reserved to SHIV KUMAR YADAV

Figure 34 Home page

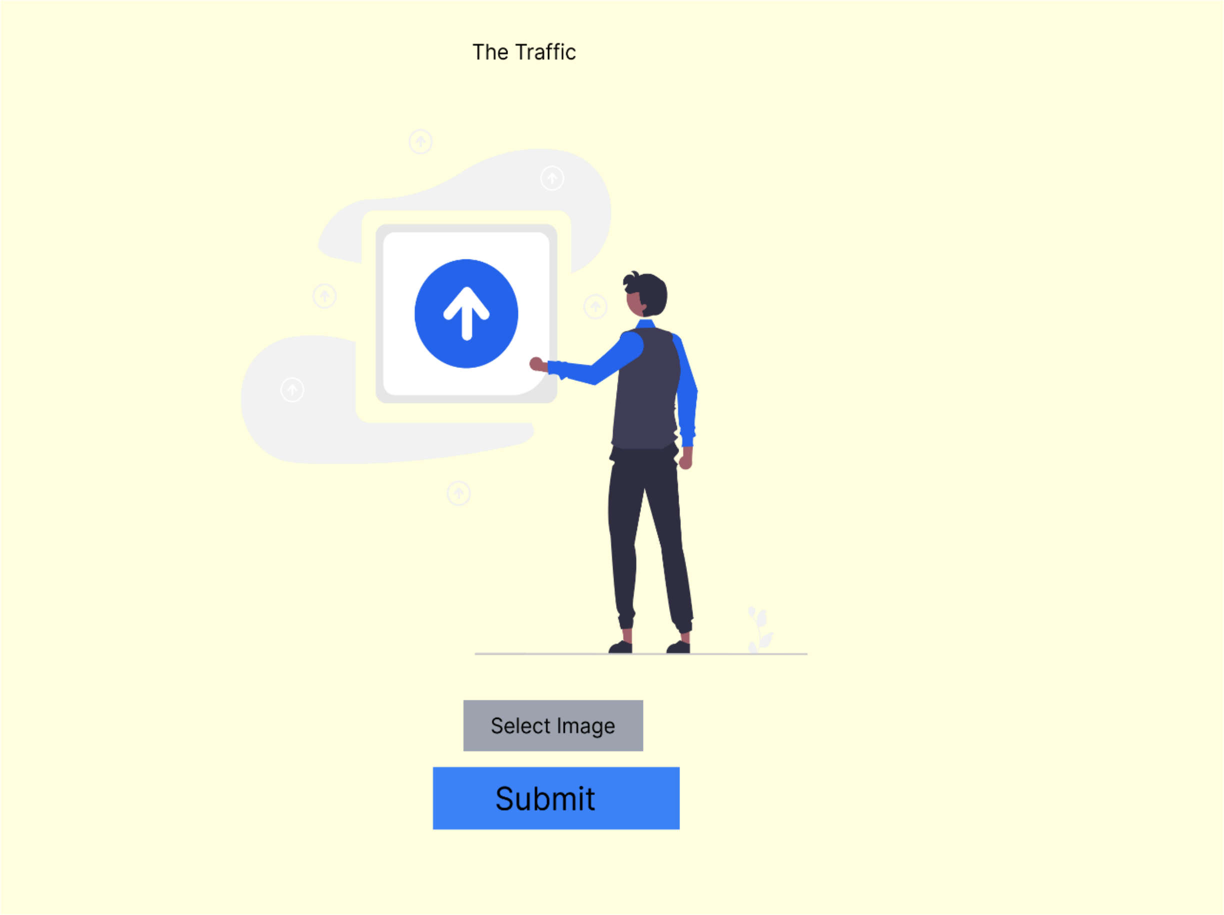
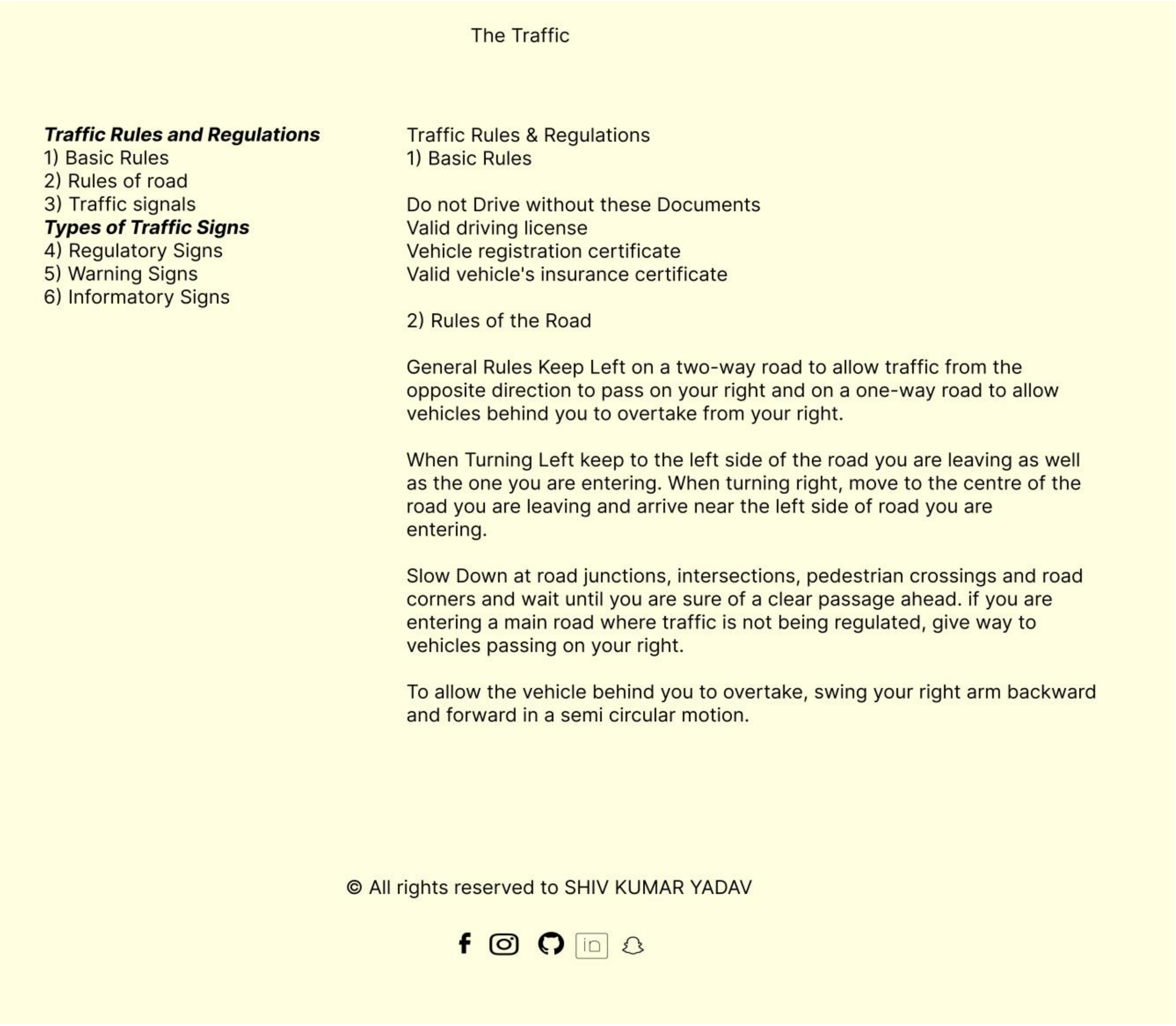


Figure 35 Image uploader page


The Traffic

Traffic Rules and Regulations
1) Basic Rules
2) Rules of road
3) Traffic signals
Types of Traffic Signs
4) Regulatory Signs
5) Warning Signs
6) Informatory Signs

Traffic Rules & Regulations
1) Basic Rules

Do not Drive without these Documents
Valid driving license
Vehicle registration certificate
Valid vehicle's insurance certificate

2) Rules of the Road

General Rules Keep Left on a two-way road to allow traffic from the opposite direction to pass on your right and on a one-way road to allow vehicles behind you to overtake from your right.

When Turning Left keep to the left side of the road you are leaving as well as the one you are entering. When turning right, move to the centre of the road you are leaving and arrive near the left side of road you are entering.

Slow Down at road junctions, intersections, pedestrian crossings and road corners and wait until you are sure of a clear passage ahead. if you are entering a main road where traffic is not being regulated, give way to vehicles passing on your right.

To allow the vehicle behind you to overtake, swing your right arm backward and forward in a semi circular motion.

© All rights reserved to SHIV KUMAR YADAV

Figure 36 Learn more page

5.2. Testing

Table 11 Test cases of image upload module

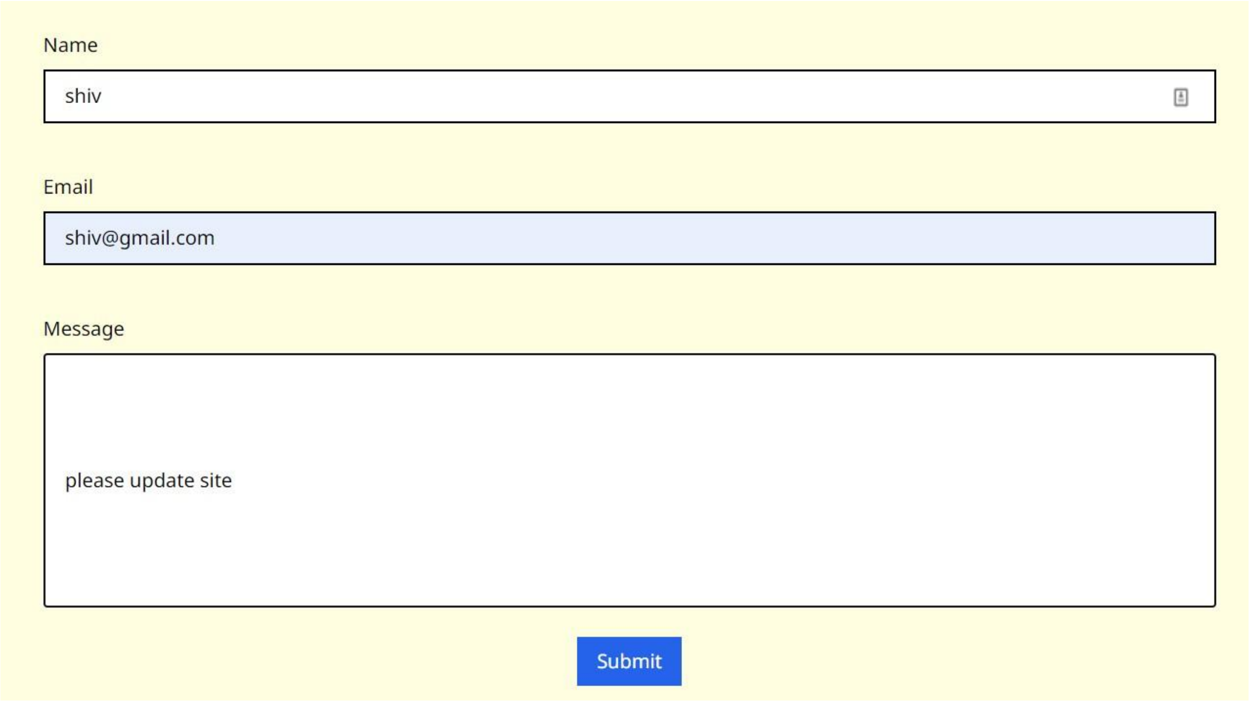
TC Id	Req. Id	Priority	Description	Precondition	Steps	Expected Result	Success/Fail
IU-TC-01	IU-F-1.0	Must have	The user should be able to take an image from the file explorer	1. Should have permission to open file explorer	Click on select image	The button works properly and opens file explorer.	Success
IUS-TC-02	IUS-N.F-1.0	Should have	Uploaded image should be of any format	Should check image format	View the image file types	Jpg png files are uploaded	Success

Table 12 Test case of the Contact form

TC Id	Req. Id	Priority	Description	Precondition	Steps	Expected Result	Success/Fail
IM-TC-01	IM-F-1.0	Must have	The user should be able to fill out the form	1. Should have not left any field empty	Click on the text area of every field	all fields of form should be filled	Success
IM-TC-02	IM-F-1.1	Must have	After filling out the form, users should submit the form	Should check form properly	Click on submit button	Submit button should work properly	Success

Here are some of the screenshots of the success of testing cases.

Figure 37 Testing of the select image button



Name

shiv

Email

shiv@gmail.com

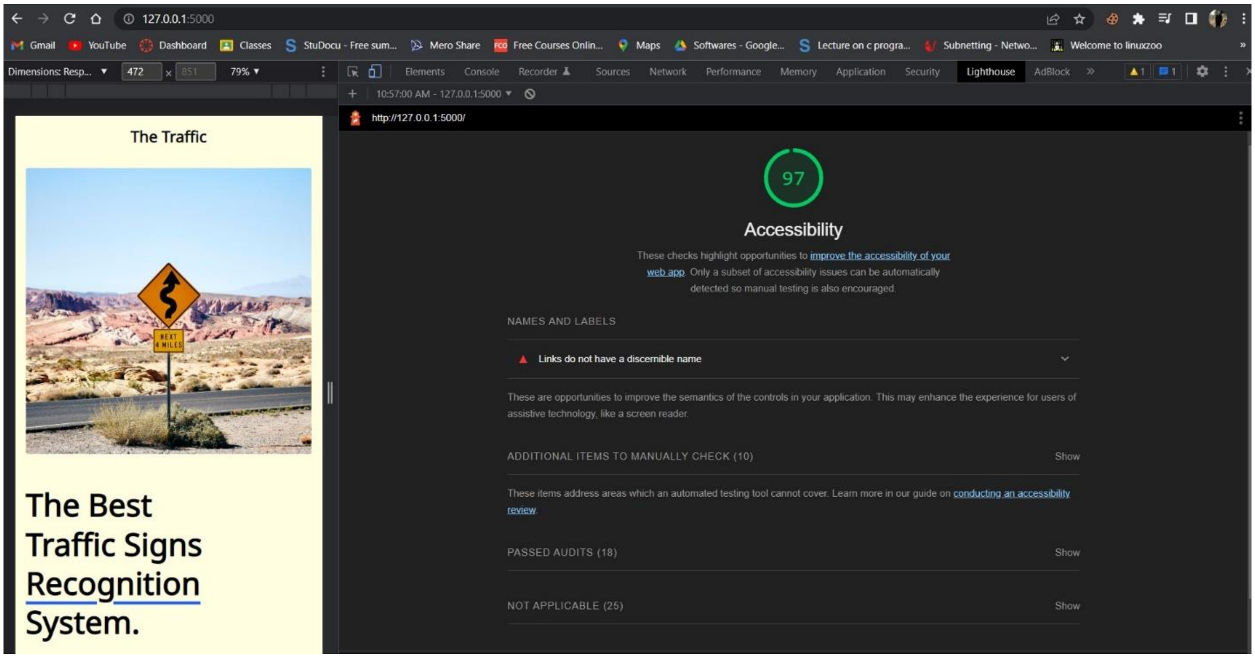
Message

please update site

Figure 38 Testing case of the contact form and submit button

Usability testing

This testing includes the accessibility testing of web pages showing how user-friendly a website is and this testing was done in the **lighthouse**.



The Traffic

The Best Traffic Signs Recognition System.

Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

NAMES AND LABELS

Links do not have a discernible name

These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

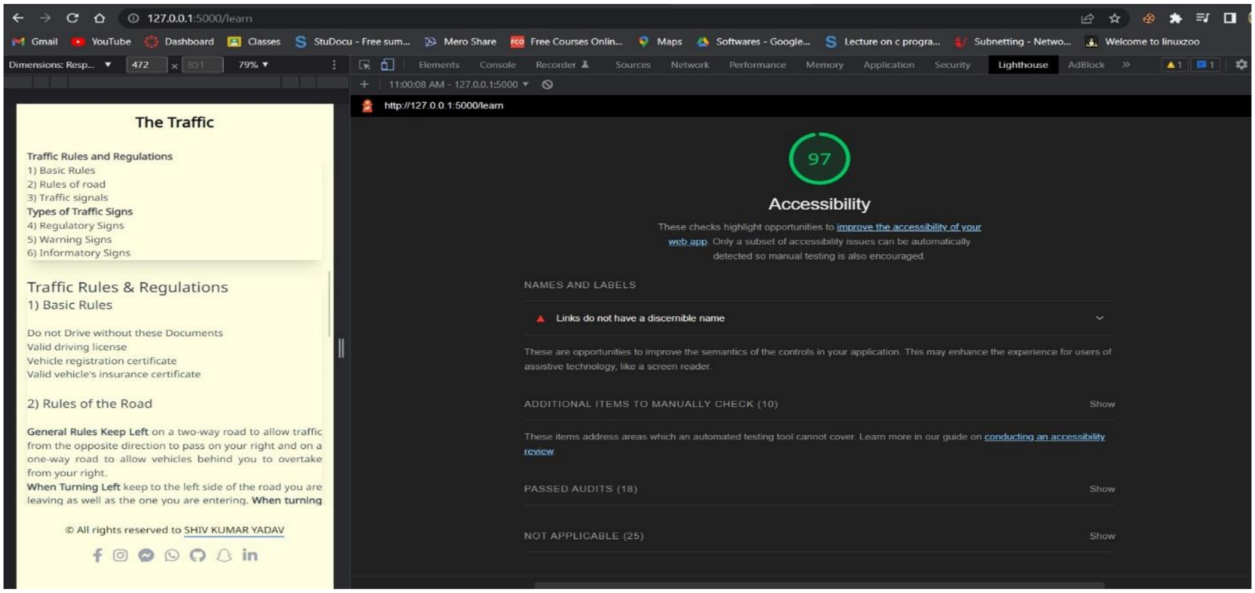
ADDITIONAL ITEMS TO MANUALLY CHECK (10) Show

These items address areas which an automated testing tool cannot cover. Learn more in our guide on conducting an accessibility review.

PASSED AUDITS (18) Show

NOT APPLICABLE (25) Show

Figure 39 accessibility testing of the home page



The Traffic

Traffic Rules and Regulations
1) Basic Rules
2) Rules of road
3) Traffic signals
Types of Traffic Signs
4) Regulatory Signs
5) Warning Signs
6) Informatory Signs

Traffic Rules & Regulations
1) Basic Rules

Do not Drive without these Documents
Valid driving license
Vehicle registration certificate
Valid vehicle's insurance certificate

2) Rules of the Road

General Rules Keep Left on a two-way road to allow traffic from the opposite direction to pass on your right and on a one-way road to allow vehicles behind you to overtake from your right.
When Turning Left keep to the left side of the road you are leaving as well as the one you are entering. **When turning**

© All rights reserved to SHIV KUMAR YADAV

Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

NAMES AND LABELS

Links do not have a discernible name

These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

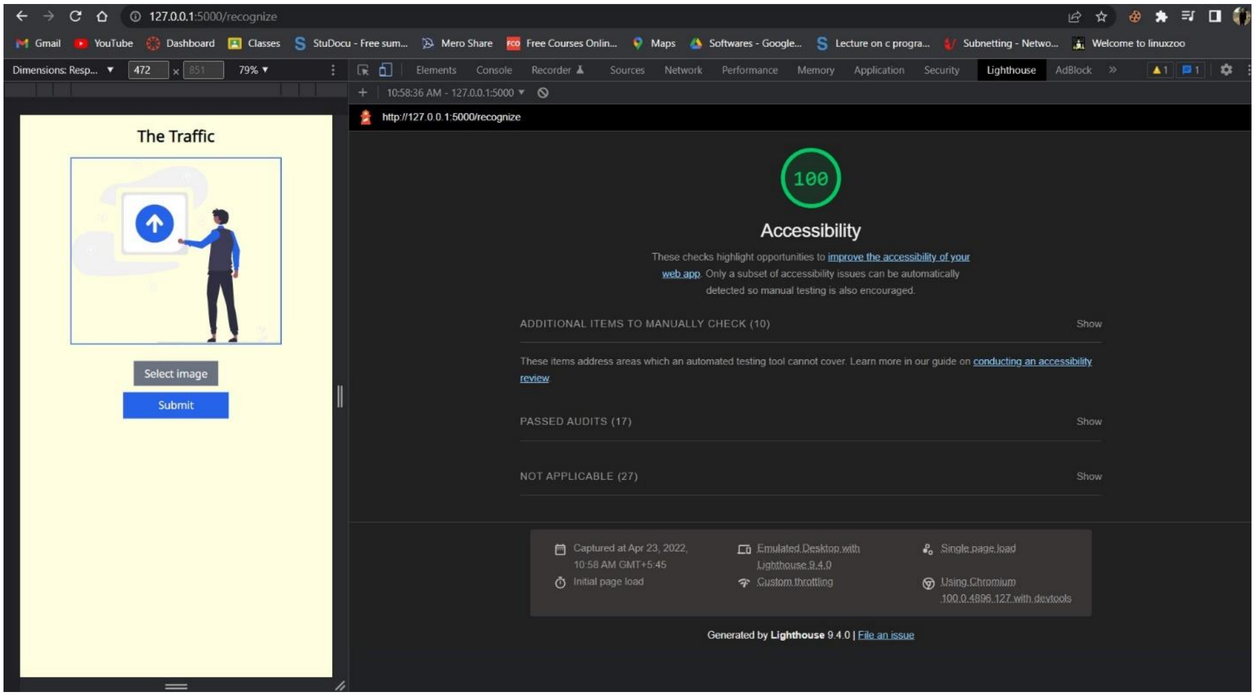
ADDITIONAL ITEMS TO MANUALLY CHECK (10) Show

These items address areas which an automated testing tool cannot cover. Learn more in our guide on conducting an accessibility review.

PASSED AUDITS (18) Show

NOT APPLICABLE (25) Show

Figure 40 accessibility testing of learn_more page



Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

ADDITIONAL ITEMS TO MANUALLY CHECK (10) Show

These items address areas which an automated testing tool cannot cover. Learn more in our guide on conducting an accessibility review.

PASSED AUDITS (17) Show

NOT APPLICABLE (27) Show

Captured at Apr 23, 2022, 10:58 AM GMT+5:45
Initial page load
Emulated Desktop with Lighthouse 9.4.0
Custom throttling
Single page load
Using Chromium 100.0.4896.127 with devtools

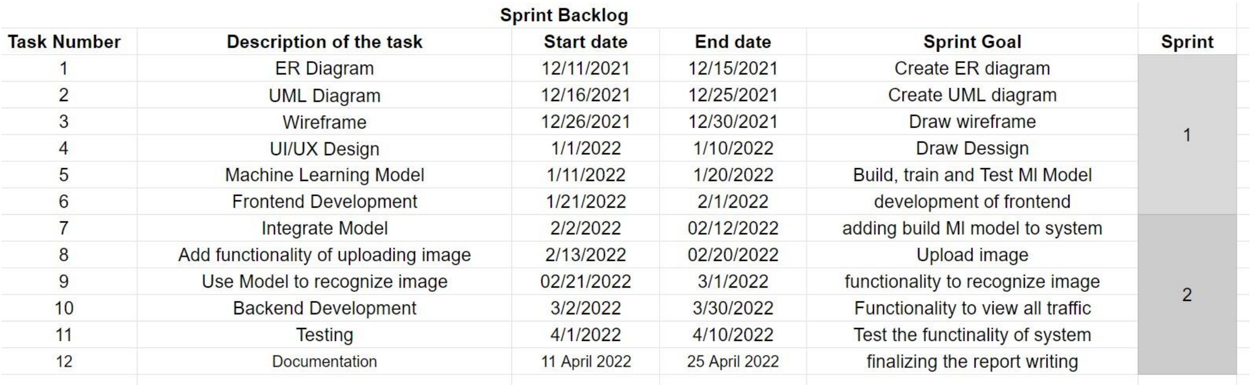
Generated by **Lighthouse** 9.4.0 | File an issue

Figure 41 Accessibility testing of recognition page

5.3 Product Backlog

As a	I want toé	So tha	Acceptance criteria	Priority
User	Select an image from the gallery	I can recognize the traffic signs.	A system with permission to use the storage of a device.	MUST HAVE
User	See the name of the recognized traffic signs.	I can know more about the traffic signs.	A system that recognizes the traffic signs.	MUST HAVE
User	Reupload an image from the gallery	I can re-upload any images of traffic signs.	A system that allows to re-upload the image.	COULD HAVE
User	Fill out the contact form	I can give suggestions or feedback.	A system that allows sending feedback.	MUST HAVE

5.4. Sprint Backlog



Sprint Backlog					
Task Number	Description of the task	Start date	End date	Sprint Goal	Sprint
1	ER Diagram	12/11/2021	12/15/2021	Create ER diagram	1
2	UML Diagram	12/16/2021	12/25/2021	Create UML diagram	
3	Wireframe	12/26/2021	12/30/2021	Draw wireframe	
4	UI/UX Design	1/1/2022	1/10/2022	Draw Dessign	
5	Machine Learning Model	1/11/2022	1/20/2022	Build, train and Test Ml Model	
6	Frontend Development	1/21/2022	2/1/2022	development of frontend	
7	Integrate Model	2/2/2022	02/12/2022	adding build Ml model to system	2
8	Add functionality of uploading image	2/13/2022	02/20/2022	Upload image	
9	Use Model to recognize image	02/21/2022	3/1/2022	functionality to recognize image	
10	Backend Development	3/2/2022	3/30/2022	Functionality to view all traffic	
11	Testing	4/1/2022	4/10/2022	Test the functinality of system	
12	Documentation	11 April 2022	25 April 2022	finalizing the report writing	

Figure 42 Sprint Backlog

5.5. AI Implementation

This section includes data collection information, how a model is developed, how the developed model is evaluated, confusion matrix, ROC curve, etc.

5.5.1. Important Libraries

Everybody in today's world tries to complete work faster but it depends on their approaches as there comes a scenario where we have to write multiples code, if we go on writing code on our own it will take time. If we use libraries we don’t have to write the same code for the different programs as its collection of modules related to python. Some of the important libraries are mentioned below:

5.5.1.1. Numpy

To handle numeric data, Numpy is the most required library for performing a wide range of mathematical functions on arrays.

5.5.1.2. Pandas

To import various data files such as Comma-Separated values files, Microsoft Excel files, JSON files, etc, Pandas library is used. It analyzes data by reading the files.

5.5.1.3. Matplotlib

To understand data distribution in every column or overall dataset visualization of data is most important, for that **matplotlib** is used. It makes things easy to understand by visualizing the data and even makes harder things possible to understand.

5.5.1.4. Tensorflow

It is an open-source artificial intelligence library that helps in the building model. For the project ‘The Traffic’ the dataset used is for classification and TensorFlow is mainly used for the classification dataset. It helps to classify the larger image size that building the model.

5.5.1.5. Keras

When it comes to minimizing cognitive load, Keras follows industry best practices: it provides uniform and straightforward APIs, limits the number of user activities necessary for typical use cases and gives clear and actionable feedback when a user makes a mistake. There are no complicated features in Keras.

5.5.1.6. Opencv

Cv2 is used as a module named import from OpenCV. For image processing and computer vision jobs, OpenCV is a fantastic resource. we may use it to accomplish tasks such as face identification, object tracking, and even landmark detection.

5.5.1.7. PIL

There are a variety of image file formats supported by the Python Imaging Library(PIL), which is an open-source and free extension to the Python language.

5.5.2. Data Collection

For developing an AI system, we need a machine learning model which works as a backend for the system. In developing a machine learning model, we require a dataset. For my system “The Traffic”, I had collected the dataset from Kaggle “GTSRB-German Traffic Sign BenchMark”.

The dataset contains two folders i.e. Train and Test and three CSV files i.e, train.csv,test.csv and meta.csv, etc. There are 39209 total images of traffic signs in the dataset. There are 43 different classes of the image of the traffic sign.

5.5.3. Data Visualization

Data visualization plays important role in the further processing of making a model.

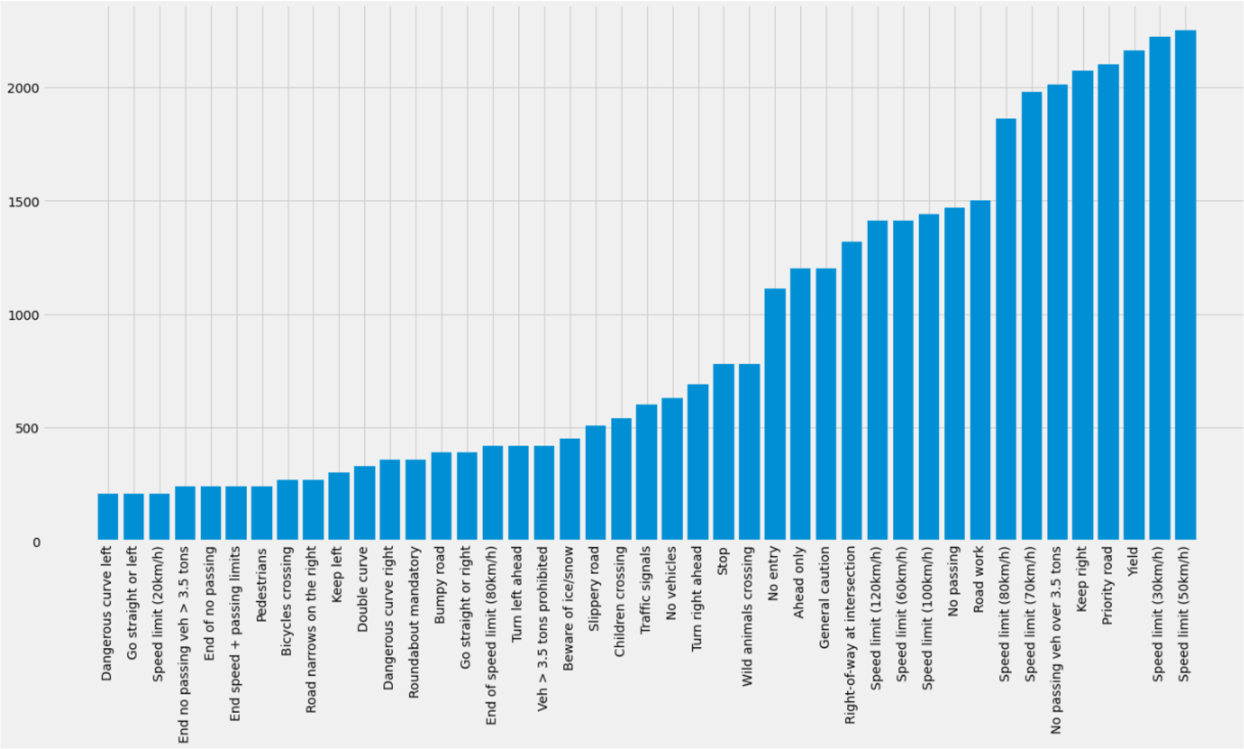


Figure 43 Data Visualization

The above visualization is a bar plot of data showing the number of images according to each class. From the above bar plot, we can see that the number of images is high of speed limit (50km/hr) whereas the dangerous curve left has the lowest number of images.

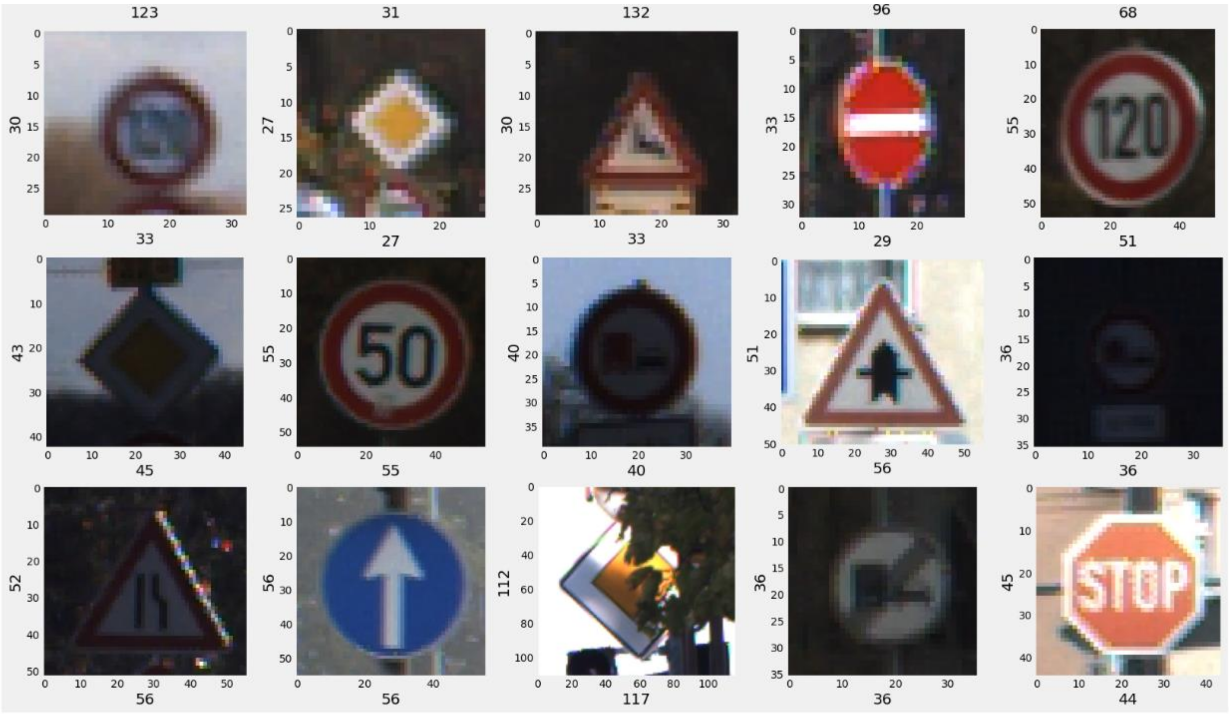
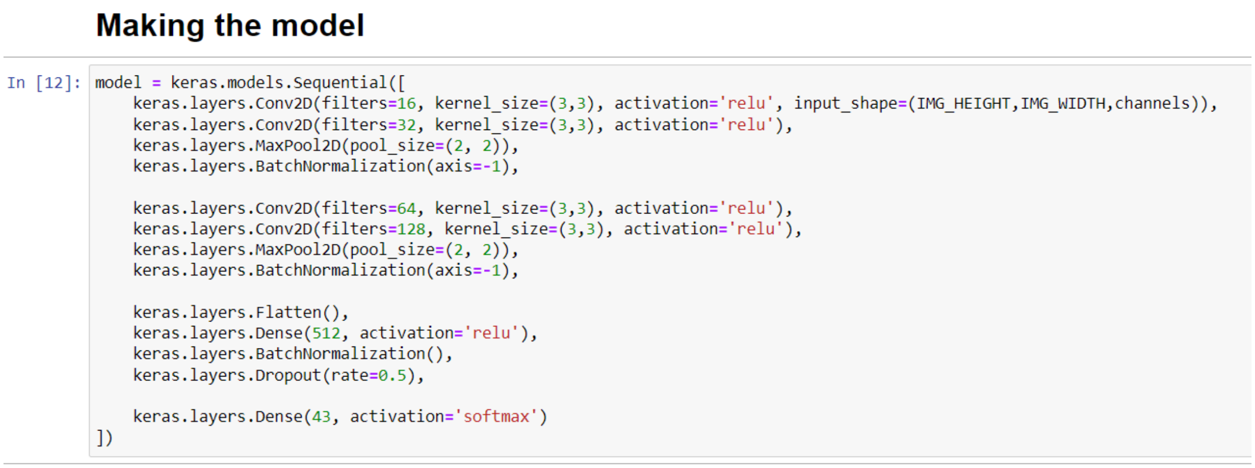


Figure 44 Visualizing random images

The above visualization shows the random images from the test folder of the dataset.

5.5.4. Model Development



Making the model

In [12]:

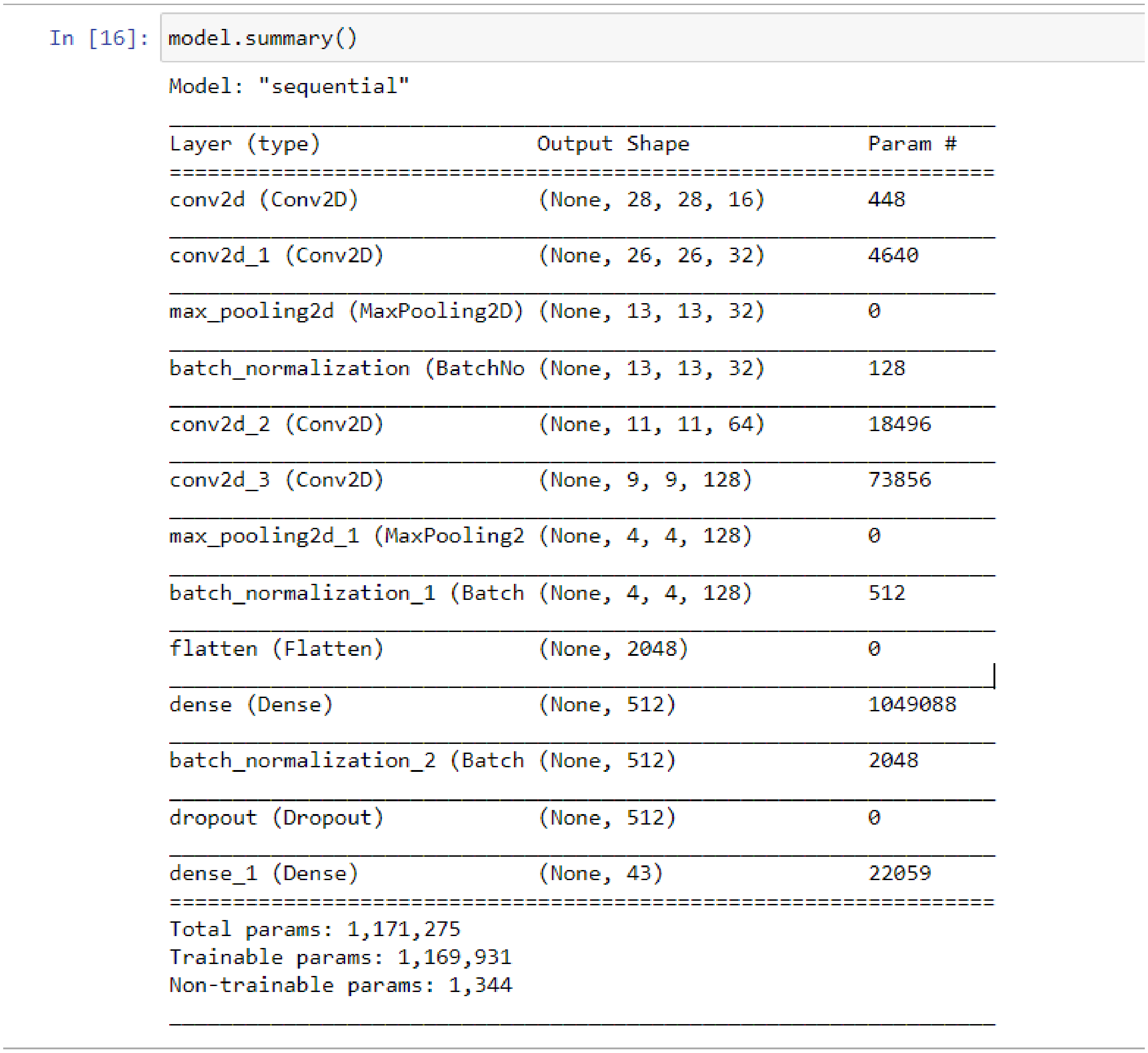
```
model = keras.models.Sequential([
    keras.layers.Conv2D(filters=16, kernel_size=(3,3), activation='relu', input_shape=(IMG_HEIGHT,IMG_WIDTH,channels)),
    keras.layers.Conv2D(filters=32, kernel_size=(3,3), activation='relu'),
    keras.layers.MaxPool2D(pool_size=(2, 2)),
    keras.layers.BatchNormalization(axis=-1),

    keras.layers.Conv2D(filters=64, kernel_size=(3,3), activation='relu'),
    keras.layers.Conv2D(filters=128, kernel_size=(3,3), activation='relu'),
    keras.layers.MaxPool2D(pool_size=(2, 2)),
    keras.layers.BatchNormalization(axis=-1),

    keras.layers.Flatten(),
    keras.layers.Dense(512, activation='relu'),
    keras.layers.BatchNormalization(),
    keras.layers.Dropout(rate=0.5),

    keras.layers.Dense(43, activation='softmax')
])
```

Figure 45 ML Model development

The most frequent model type in Python programming is the Sequential model. In Keras, this is the quickest approach to construct a CNN model. Model building is made easier using this tool. Layers may be added to the model using the 'add()' method. The model has four Convolution and Pooling pairs, followed by batch normalization, which normalizes the layer of the neural network, and a Flatten layer, which connects the Convolution layer to the Dense layer, as stated above.

Most of the output layers are made up of Dense layers. Each class has a 'Softmax' chance of activation, which adds up to a total probability of 1. The model's forecast will be based on the most likely class.

Here’s the summary of the model:

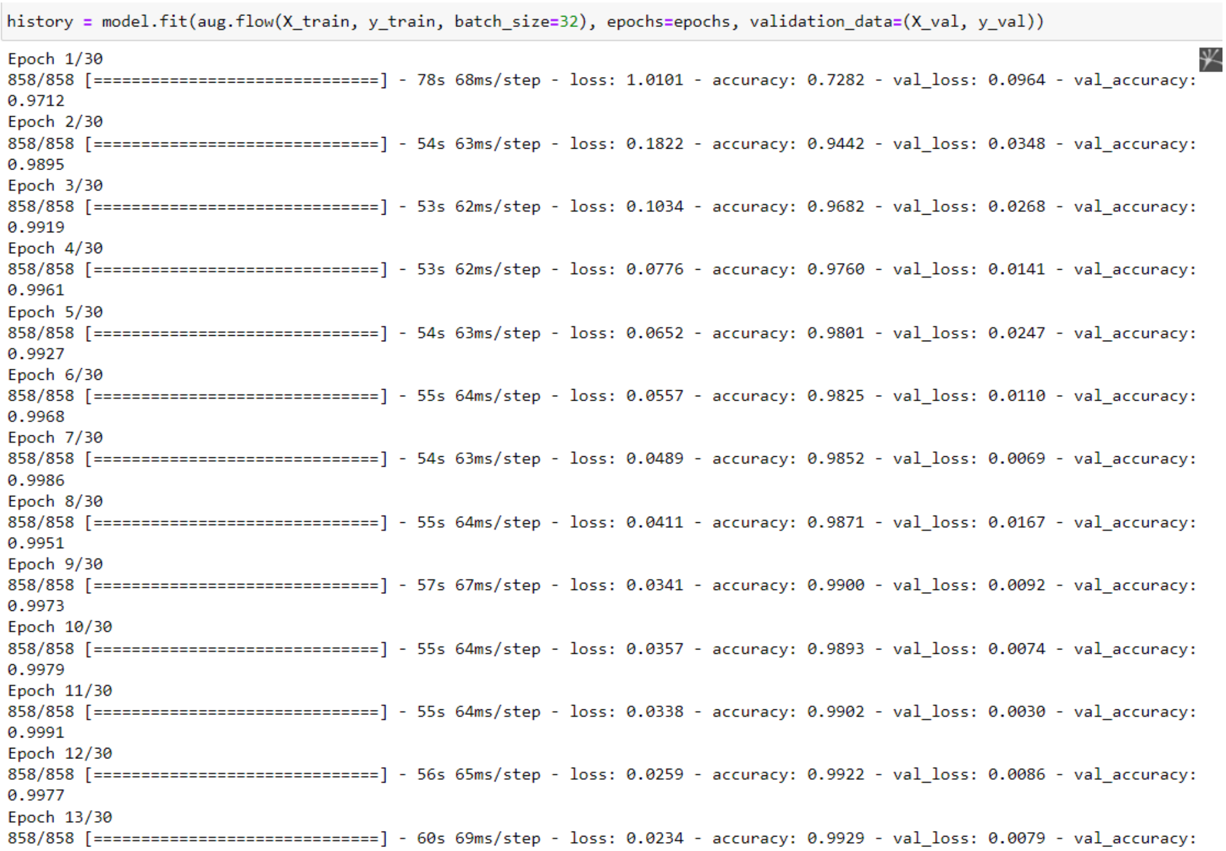


```
In [16]: model.summary()

Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d (Conv2D)              (None, 28, 28, 16)        448       
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 26, 26, 32)        4640      
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 13, 13, 32)        0         
_________________________________________________________________
batch_normalization (BatchNo (None, 13, 13, 32)        128       
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 11, 11, 64)        18496     
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 9, 9, 128)         73856     
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 4, 4, 128)         0         
_________________________________________________________________
batch_normalization_1 (Batch (None, 4, 4, 128)         512       
_________________________________________________________________
flatten (Flatten)            (None, 2048)              0         
_________________________________________________________________
dense (Dense)                (None, 512)               1049088   
_________________________________________________________________
batch_normalization_2 (Batch (None, 512)               2048      
_________________________________________________________________
dropout (Dropout)            (None, 512)               0         
_________________________________________________________________
dense_1 (Dense)              (None, 43)                22059     
=================================================================
Total params: 1,171,275
Trainable params: 1,169,931
Non-trainable params: 1,344
_________________________________________________________________
```

Figure 46 Model summary

5.5.5. Model training and validation

I used the model.fit() method to train our model because it works well after the model architecture is successfully built. After 30 epochs, I was able to achieve 98 percent accuracy on training sets with the help of 32 batch sizes.



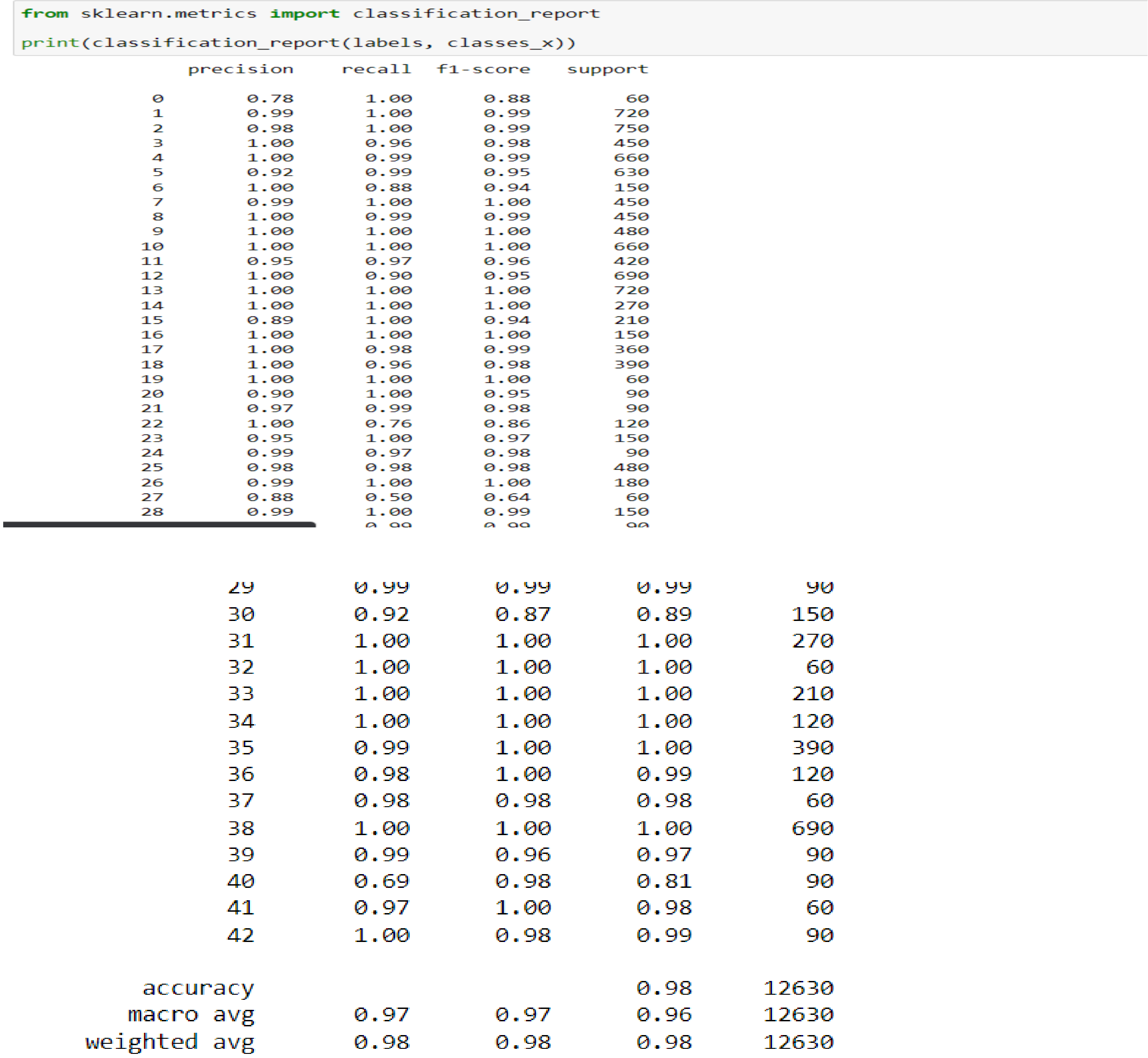
```
history = model.fit(aug.flow(X_train, y_train, batch_size=32), epochs=epochs, validation_data=(X_val, y_val))
```

```
Epoch 1/30
858/858 [==============================] - 78s 68ms/step - loss: 1.0101 - accuracy: 0.7282 - val_loss: 0.0964 - val_accuracy:
0.9712
Epoch 2/30
858/858 [==============================] - 54s 63ms/step - loss: 0.1822 - accuracy: 0.9442 - val_loss: 0.0348 - val_accuracy:
0.9895
Epoch 3/30
858/858 [==============================] - 53s 62ms/step - loss: 0.1034 - accuracy: 0.9682 - val_loss: 0.0268 - val_accuracy:
0.9919
Epoch 4/30
858/858 [==============================] - 53s 62ms/step - loss: 0.0776 - accuracy: 0.9760 - val_loss: 0.0141 - val_accuracy:
0.9961
Epoch 5/30
858/858 [==============================] - 54s 63ms/step - loss: 0.0652 - accuracy: 0.9801 - val_loss: 0.0247 - val_accuracy:
0.9927
Epoch 6/30
858/858 [==============================] - 55s 64ms/step - loss: 0.0557 - accuracy: 0.9825 - val_loss: 0.0110 - val_accuracy:
0.9968
Epoch 7/30
858/858 [==============================] - 54s 63ms/step - loss: 0.0489 - accuracy: 0.9852 - val_loss: 0.0069 - val_accuracy:
0.9986
Epoch 8/30
858/858 [==============================] - 55s 64ms/step - loss: 0.0411 - accuracy: 0.9871 - val_loss: 0.0167 - val_accuracy:
0.9951
Epoch 9/30
858/858 [==============================] - 57s 67ms/step - loss: 0.0341 - accuracy: 0.9900 - val_loss: 0.0092 - val_accuracy:
0.9973
Epoch 10/30
858/858 [==============================] - 55s 64ms/step - loss: 0.0357 - accuracy: 0.9893 - val_loss: 0.0074 - val_accuracy:
0.9979
Epoch 11/30
858/858 [==============================] - 55s 64ms/step - loss: 0.0338 - accuracy: 0.9902 - val_loss: 0.0030 - val_accuracy:
0.9991
Epoch 12/30
858/858 [==============================] - 56s 65ms/step - loss: 0.0259 - accuracy: 0.9922 - val_loss: 0.0086 - val_accuracy:
0.9977
Epoch 13/30
858/858 [==============================] - 60s 69ms/step - loss: 0.0234 - accuracy: 0.9929 - val_loss: 0.0079 - val_accuracy:
```

Figure 47 model training and validation

5.5.6. Optimization Evaluation

In this classification report, we can see the performance of the model. The precision, recall, and f1-score of overall classes of the dataset. The accuracy of this model is around 98 percent.



```
from sklearn.metrics import classification_report
print(classification_report(labels, classes_x))
```

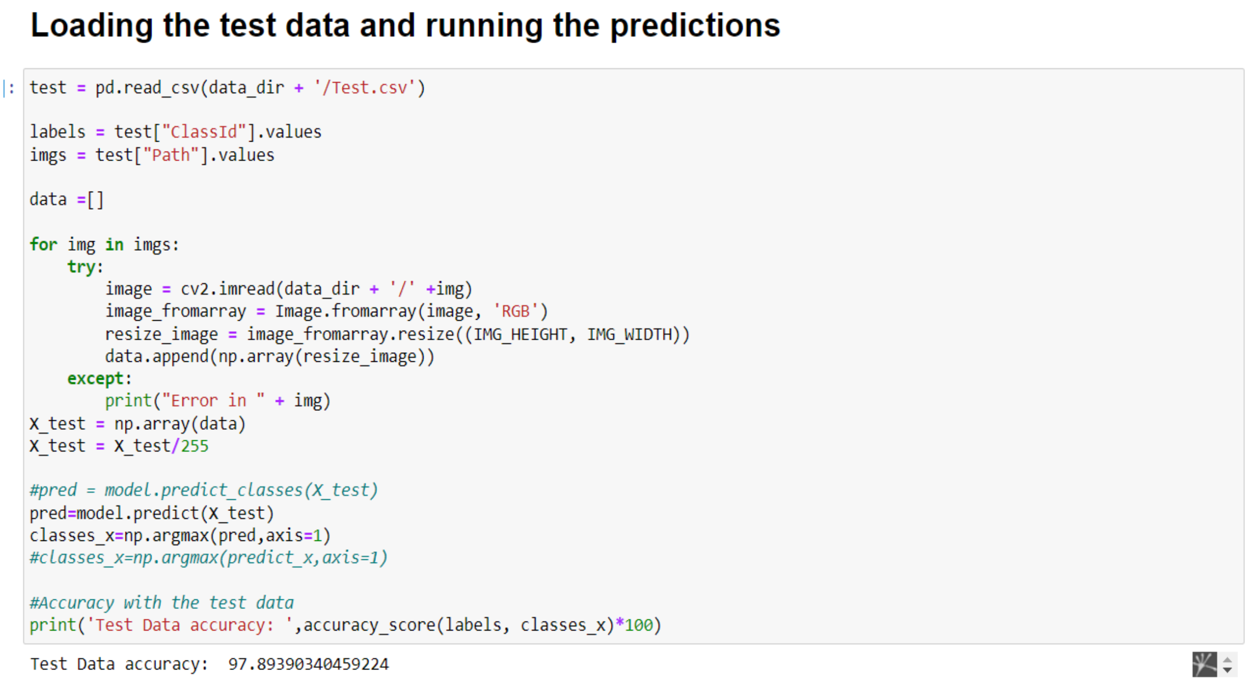
```
              precision    recall  f1-score   support

           0       0.78      1.00      0.88        60
           1       0.99      1.00      0.99       720
           2       0.98      1.00      0.99       750
           3       1.00      0.96      0.98       450
           4       1.00      0.99      0.99       660
           5       0.92      0.99      0.95       630
           6       1.00      0.88      0.94       150
           7       0.99      1.00      1.00       450
           8       1.00      0.99      0.99       450
           9       1.00      1.00      1.00       480
          10       1.00      1.00      1.00       660
          11       0.95      0.97      0.96       420
          12       1.00      0.90      0.95       690
          13       1.00      1.00      1.00       720
          14       1.00      1.00      1.00       270
          15       0.89      1.00      0.94       210
          16       1.00      1.00      1.00       150
          17       1.00      0.98      0.99       360
          18       1.00      0.96      0.98       390
          19       1.00      1.00      1.00        60
          20       0.90      1.00      0.95        90
          21       0.97      0.99      0.98        90
          22       1.00      0.76      0.86       120
          23       0.95      1.00      0.97       150
          24       0.99      0.97      0.98        90
          25       0.98      0.98      0.98       480
          26       0.99      1.00      1.00       180
          27       0.88      0.50      0.64        60
          28       0.99      1.00      0.99       150
          29       0.99      0.99      0.99        90
          30       0.92      0.87      0.89       150
          31       1.00      1.00      1.00       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       0.99      1.00      1.00       390
          36       0.98      1.00      0.99       120
          37       0.98      0.98      0.98        60
          38       1.00      1.00      1.00       690
          39       0.99      0.96      0.97        90
          40       0.69      0.98      0.81        90
          41       0.97      1.00      0.98        60
          42       1.00      0.98      0.99        90

    accuracy                           0.98     12630
   macro avg       0.97      0.97      0.96     12630
weighted avg       0.98      0.98      0.98     12630
```

Figure 48 classification report

5.5.7. Comparing the AI performance (with Test Data)

Model Testing

In our dataset, we have a folder titled "test" and a comma-separated file called "test.csv." Both the picture paths and their corresponding class names are included. To get the picture path and labels, we may use the panda's python package. To forecast the model and build a NumPy array packed with picture data, we need to shrink our photographs to 30x30 pixels. We need to import the accuracy score from sklearn.metrics to determine how the model predicts the real labels. Keras model.save() is finally used to save our trained model.

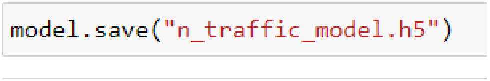


Loading the test data and running the predictions

```
test = pd.read_csv(data_dir + '/Test.csv')

labels = test["ClassId"].values
imgs = test["Path"].values

data =[]

for img in imgs:
    try:
        image = cv2.imread(data_dir + '/' +img)
        image_fromarray = Image.fromarray(image, 'RGB')
        resize_image = image_fromarray.resize((IMG_HEIGHT, IMG_WIDTH))
        data.append(np.array(resize_image))
    except:
        print("Error in " + img)
X_test = np.array(data)
X_test = X_test/255

#pred = model.predict_classes(X_test)
pred=model.predict(X_test)
classes_x=np.argmax(pred,axis=1)
#classes_x=np.argmax(predict_x,axis=1)

#Accuracy with the test data
print('Test Data accuracy: ',accuracy_score(labels, classes_x)*100)
```

```
Test Data accuracy:  97.89390340459224
```

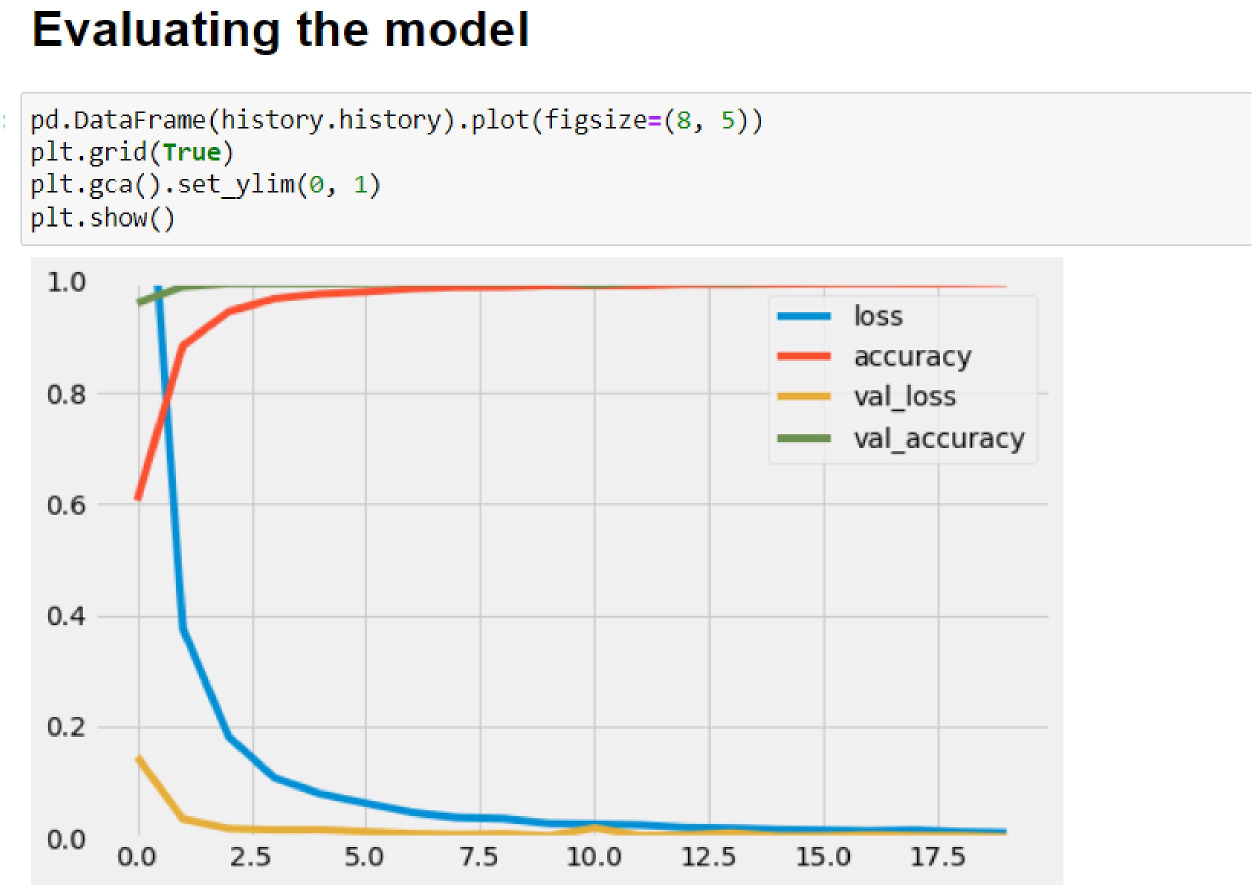
Figure 49 Prediction of test data



```
model.save("n_traffic_model.h5")
```

Figure 50 Saving Model

5.5.8. AI Testing plotting Accuracy



Evaluating the model

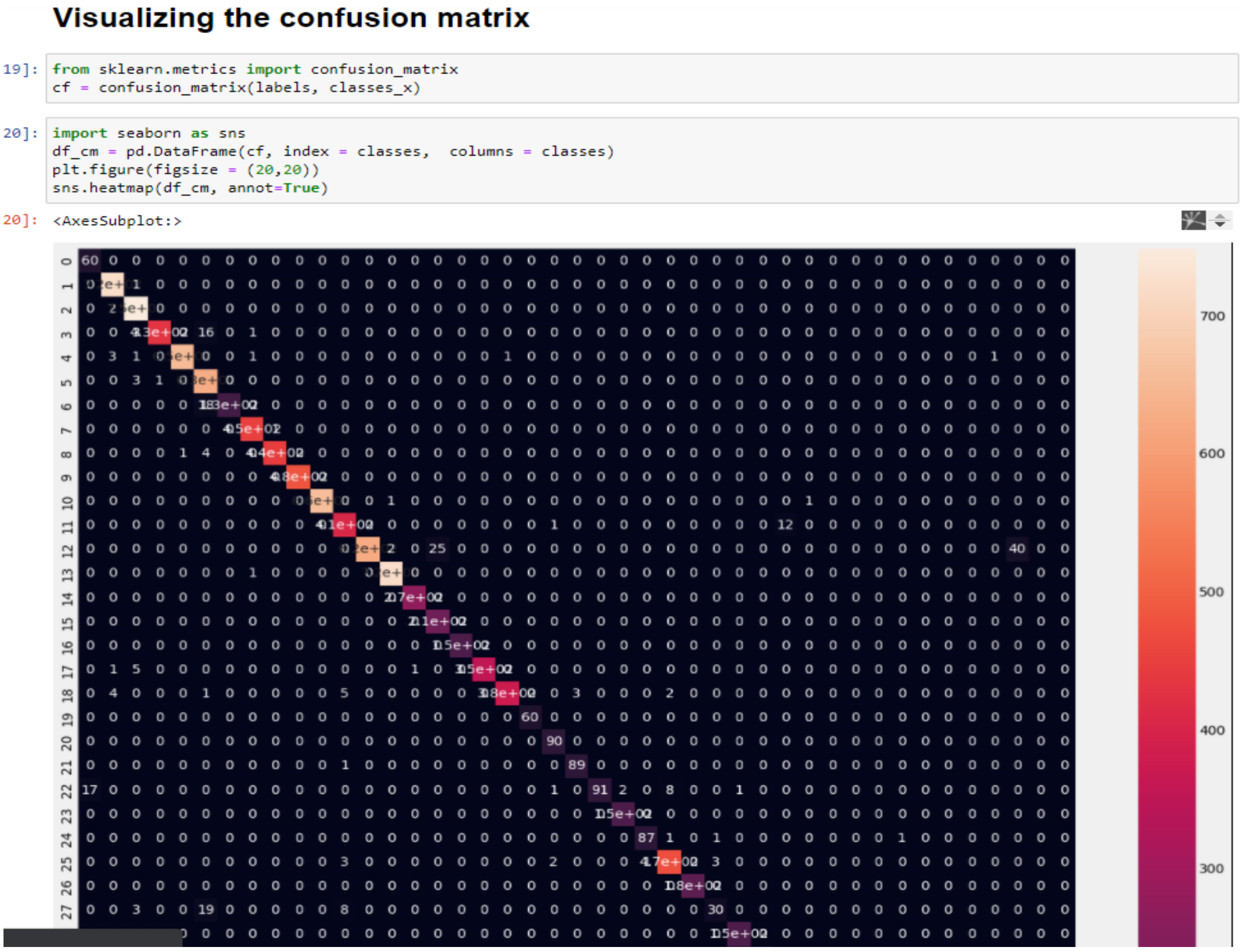
```
pd.DataFrame(history.history).plot(figsize=(8, 5))
plt.grid(True)
plt.gca().set_ylim(0, 1)
plt.show()
```

Figure 51 Evaluating the model

In the above graphs, we can observe that after 10 epochs, the accuracy of the test has nearly reached a standstill and has only seldom risen. It was first linearly growing inaccuracy with loss, but subsequently, it didn't rise at all.

5.5.9. Confusion Matrix

A confusion matrix is used to visualize the performance of the model .i.e precision, recall, accuracy, f1 score, etc.



Visualizing the confusion matrix

```
19]: from sklearn.metrics import confusion_matrix
     cf = confusion_matrix(labels, classes_x)
```

```
20]: import seaborn as sns
     df_cm = pd.DataFrame(cf, index = classes,  columns = classes)
     plt.figure(figsize = (20,20))
     sns.heatmap(df_cm, annot=True)
```

```
20]: <AxesSubplot:>
```

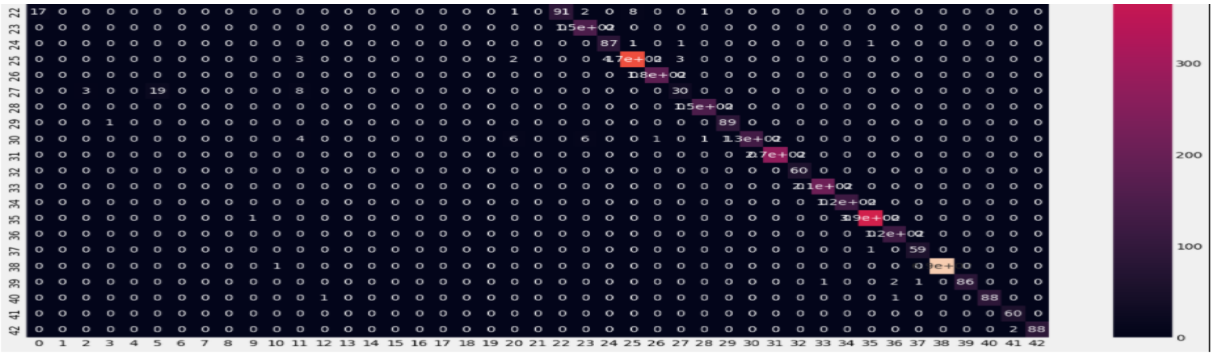


Figure 52 Confusion Matrix

6. Conclusion

6.1. Conclusion of the system

In this project, a traffic and road sign recognition system can help in recognizing the traffic sign and which may be helpful for students also to get to know about the name of traffic signs and also help road sign inventory in maintaining and updating the traffic sign according to the user's review. This system involves several kinds of computer vision and image recognition patterns which able to extract information from a particular image and provide a useful output according to its labeled classification class name. In computer vision and image recognition, algorithms were used to extract information from images by colour and resize it into particular sizes. This aim has been achieved and the system shows high robustness according to the classification report. Many libraries like TensorFlow, Keras, etc are used to complete the system.

The main purpose of this system is to help in safe driving practices by helping users/people to get the information/name of traffic signs through the system by uploading an image. Let’s assume that if a driver goes driving and sees a certain traffic sign that he/she doesn’t about it. At that time, they can click the photo clearly and upload it to this system. By uploading the image, the system recognizes that image and gives output as a particular name of that traffic sign. They can even google it after knowing the name of the traffic sign and know more about it. In this way, they can avoid such obstacles and maintain safe driving practices.

6.2. Future Escalation

As I have previously stated in limitations that the system even classifies the images of anything and prints their name according to its label, this can be solved by using object detection, and only traffic signs will only be detected and recognized. Besides that in the future, the system can be developed in a way so that it recognizes **only traffic signs**, the system can be upgraded so that traffic signs will be recognized in real-time. It can be used in autonomous vehicles.

7. Critical Evaluation of Project

7.1. Final Report

Word limit always plays an important role in writing a report. There comes a certain problem while writing this report. Some of the problems are explanations algorithms are not perfectly explained due to word limits, arranging of some topics into subtopics, etc. if I have the freedom to write a report of the unlimited word, I am pretty sure that the report will be fantastic.

7.2. Finding and process

One of the requirements to develop a machine learning model is to find a dataset and work on that dataset. Dataset sometimes seems to be fine but it is not. So, the finding of the proper dataset is always problematic. I have searched for the traffic sign dataset used in Nepal but the proper dataset is not available. Then I found the dataset from Kaggle and the dataset is about the german traffic sign Benchmark.

While working on datasets, I face difficulty while building the CNN model because this was the first time I was working in CNN.

7.3. System

After developing the machine learning model, the next target is to integrate that model into the system. While integrating the model, it take more time for me to integrate it because I was facing difficulty while working on the flask to integrate the AI model.

The next task that I faced problem is connecting the database. But after going through some documents of python and Database connection with python, It was easy for me to connect with the database.

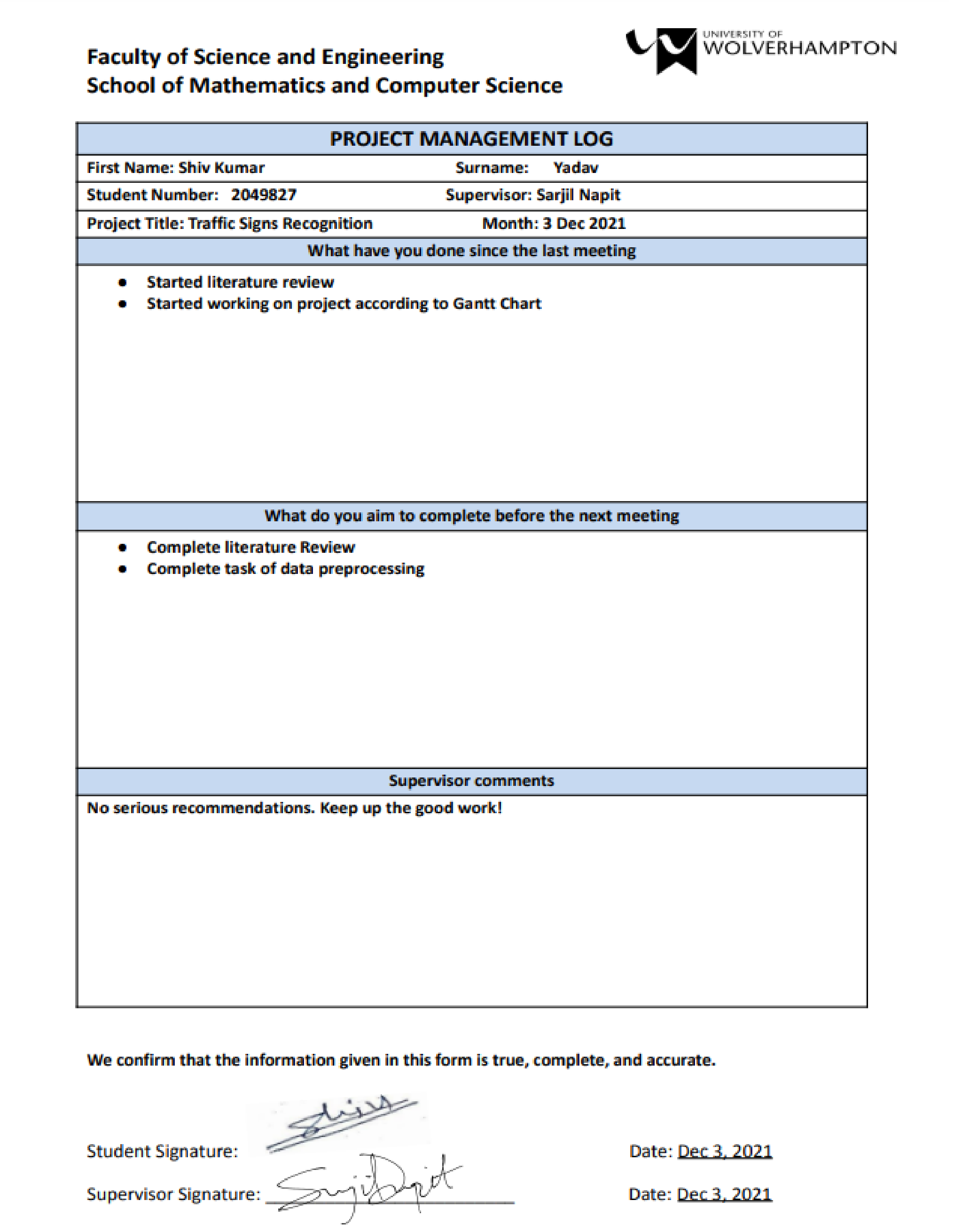
7.4. Self-reflection

Everyone learns something new after working on something. From doing this project, I encountered several new things such as CNN, SQLAlchemy, and many more. I have learned how to work with image datasets, how to build a perfectly working CNN model and familiar with CNN architecture, learned the connection of database with python in the flask, etc.

8. Evidence of Project Management

8.1. LogSheet

This section includes the proof of working on my project parallelly with the support of a supervisor.



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG	
First Name: Shiv Kumar	Surname: Yadav
Student Number: 2049827	Supervisor: Sarjil Napit
Project Title: Traffic Signs Recognition	Month: 3 Dec 2021

What have you done since the last meeting

- Started literature review
- Started working on project according to Gantt Chart

What do you aim to complete before the next meeting

- Complete literature Review
- Complete task of data preprocessing

Supervisor comments

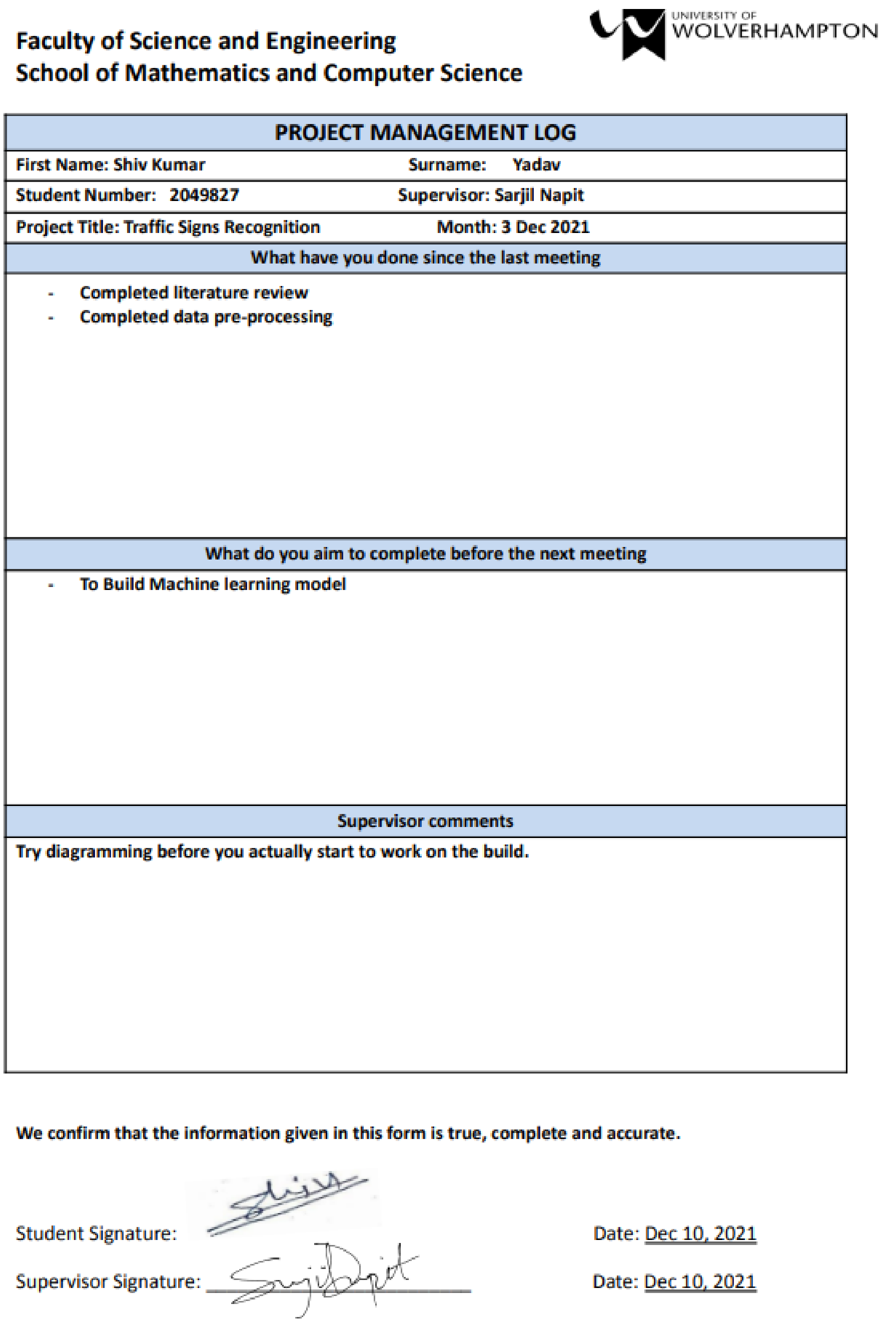
No serious recommendations. Keep up the good work!

We confirm that the information given in this form is true, complete, and accurate.

Student Signature: Date: Dec 3, 2021

Supervisor Signature: Date: Dec 3, 2021

Figure 53 Logsheet -1 Dec 3, 2021



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG	
First Name: Shiv Kumar	**Surname:** Yadav
Student Number: 2049827	**Supervisor:** Sarjil Napit
Project Title: Traffic Signs Recognition	**Month:** 3 Dec 2021

What have you done since the last meeting

- Completed literature review
- Completed data pre-processing

What do you aim to complete before the next meeting

- To Build Machine learning model

Supervisor comments

Try diagramming before you actually start to work on the build.

We confirm that the information given in this form is true, complete and accurate.

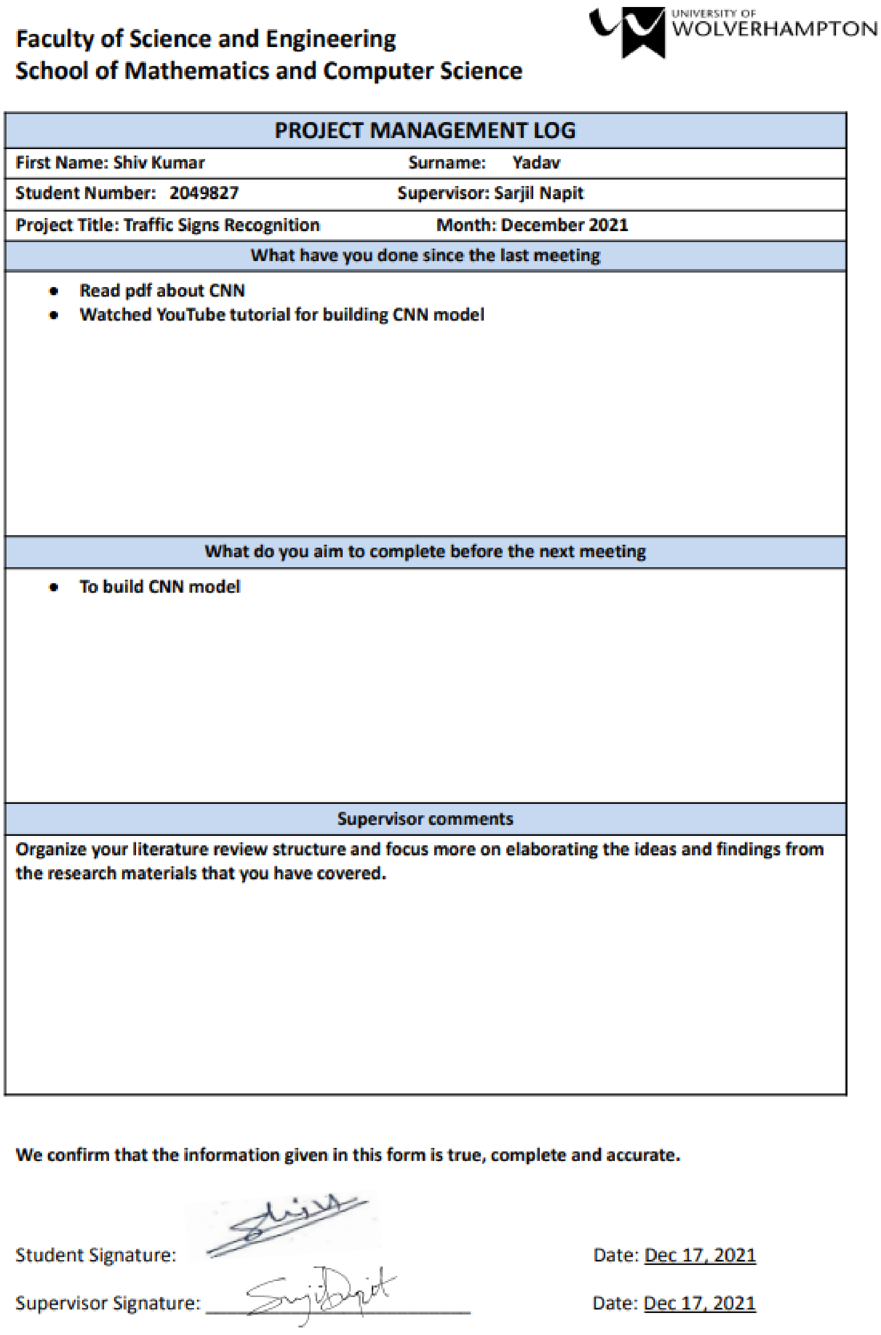
Student Signature: Shiv

Date: Dec 10, 2021

Supervisor Signature: Sarjil Napit

Date: Dec 10, 2021

Figure 54 LogSheet-2 Dec10, 2021



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG	
First Name: Shiv Kumar	**Surname: Yadav**
Student Number: 2049827	**Supervisor: Sarjil Napit**
Project Title: Traffic Signs Recognition	**Month: December 2021**

What have you done since the last meeting

- **Read pdf about CNN**
- **Watched YouTube tutorial for building CNN model**

What do you aim to complete before the next meeting

- **To build CNN model**

Supervisor comments

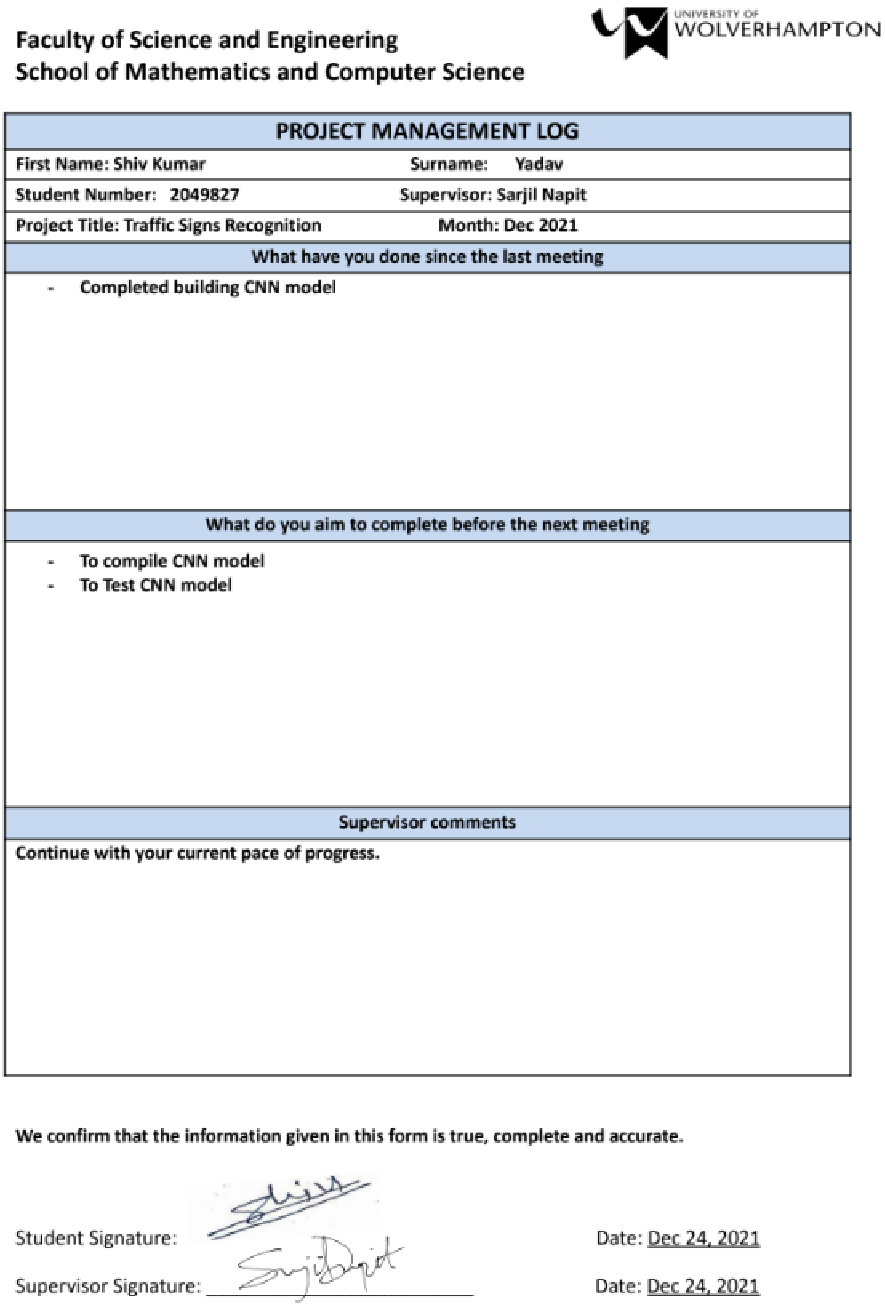
Organize your literature review structure and focus more on elaborating the ideas and findings from the research materials that you have covered.

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Date: Dec 17, 2021

Supervisor Signature: ____________________ Date: Dec 17, 2021

Figure 55 Logsheet-3 Dec17, 2021



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

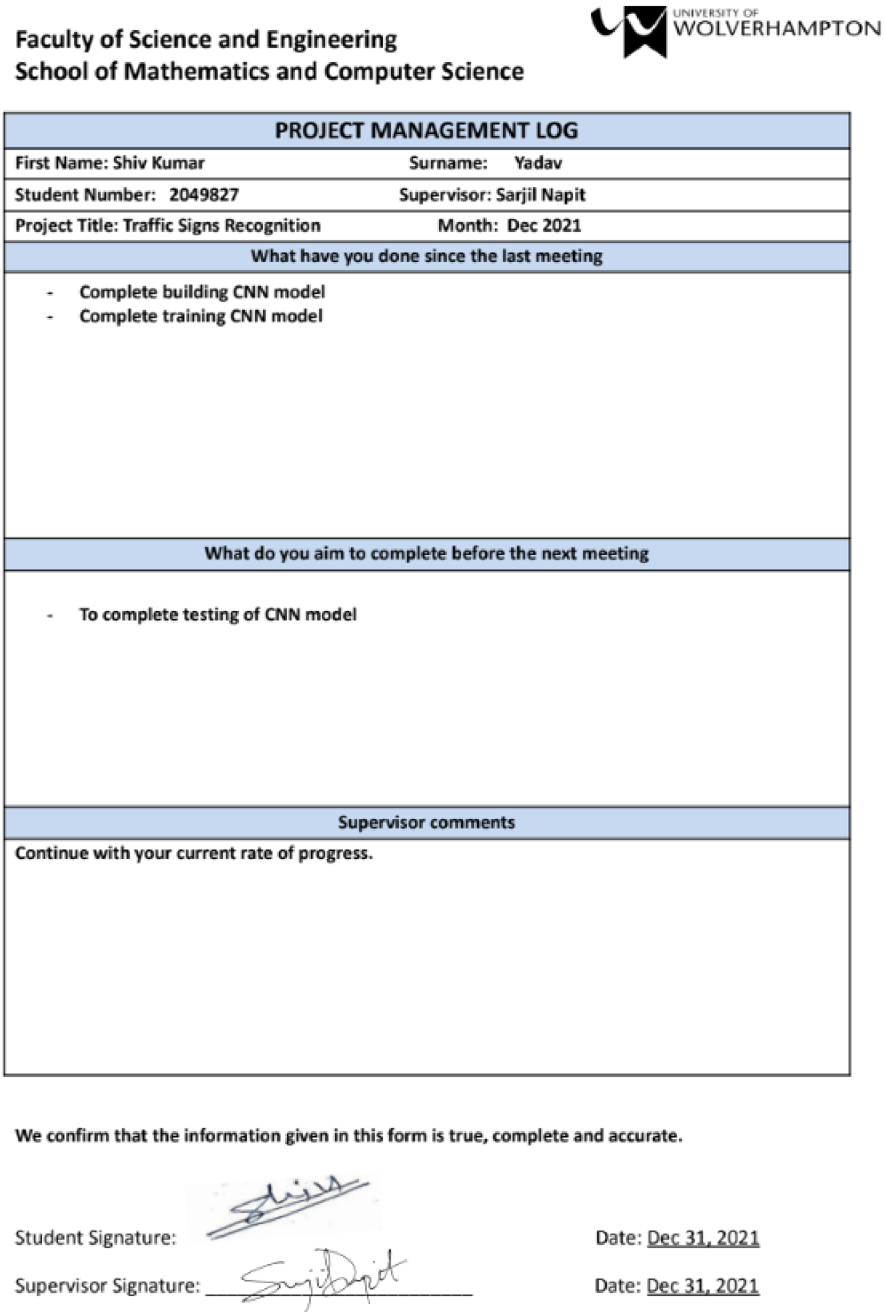
PROJECT MANAGEMENT LOG	
First Name: Shiv Kumar	Surname: Yadav
Student Number: 2049827	Supervisor: Sarjil Napit
Project Title: Traffic Signs Recognition	Month: Dec 2021
What have you done since the last meeting	
- Completed building CNN model	
What do you aim to complete before the next meeting	
- To compile CNN model - To Test CNN model	
Supervisor comments	
Continue with your current pace of progress.	

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Date: Dec 24, 2021

Supervisor Signature: ______________________ Date: Dec 24, 2021

Figure 56 Logsheet-4 Dec 24, 2021



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG	
First Name: Shiv Kumar	Surname: Yadav
Student Number: 2049827	Supervisor: Sarjil Napit
Project Title: Traffic Signs Recognition	Month: Dec 2021

What have you done since the last meeting

- Complete building CNN model
- Complete training CNN model

What do you aim to complete before the next meeting

- To complete testing of CNN model

Supervisor comments

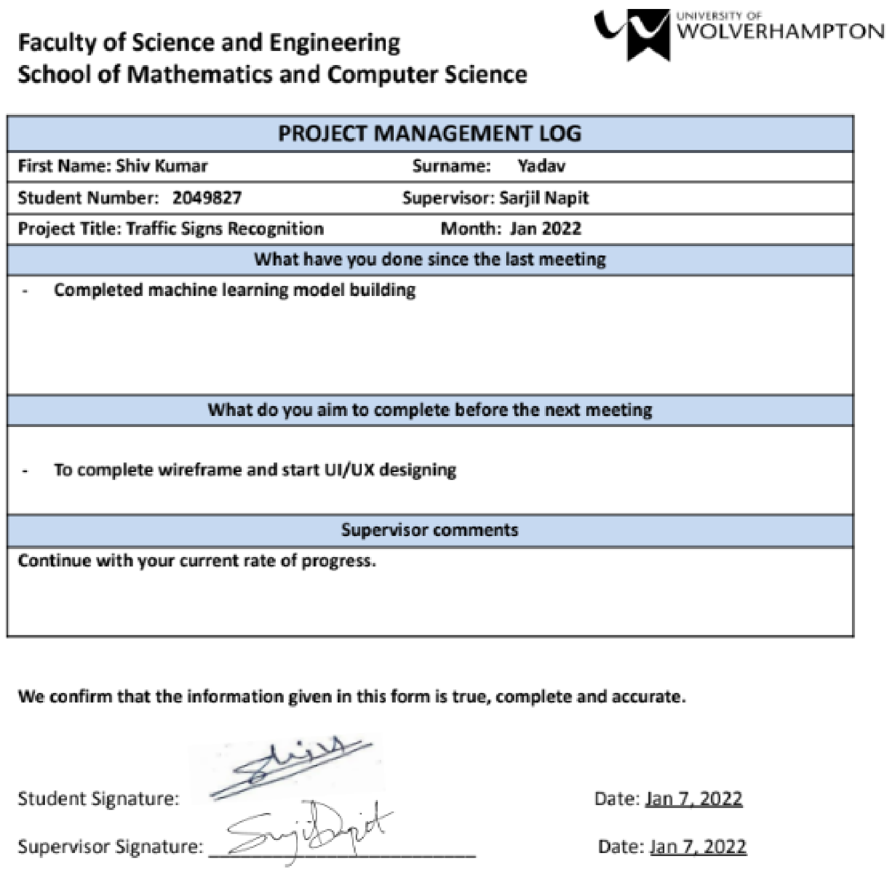
Continue with your current rate of progress.

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Date: Dec 31, 2021

Supervisor Signature: ____________________ Date: Dec 31, 2021

Figure 57 Logsheet-5 Dec31, 2021



Faculty of Science and Engineering
School of Mathematics and Computer Science

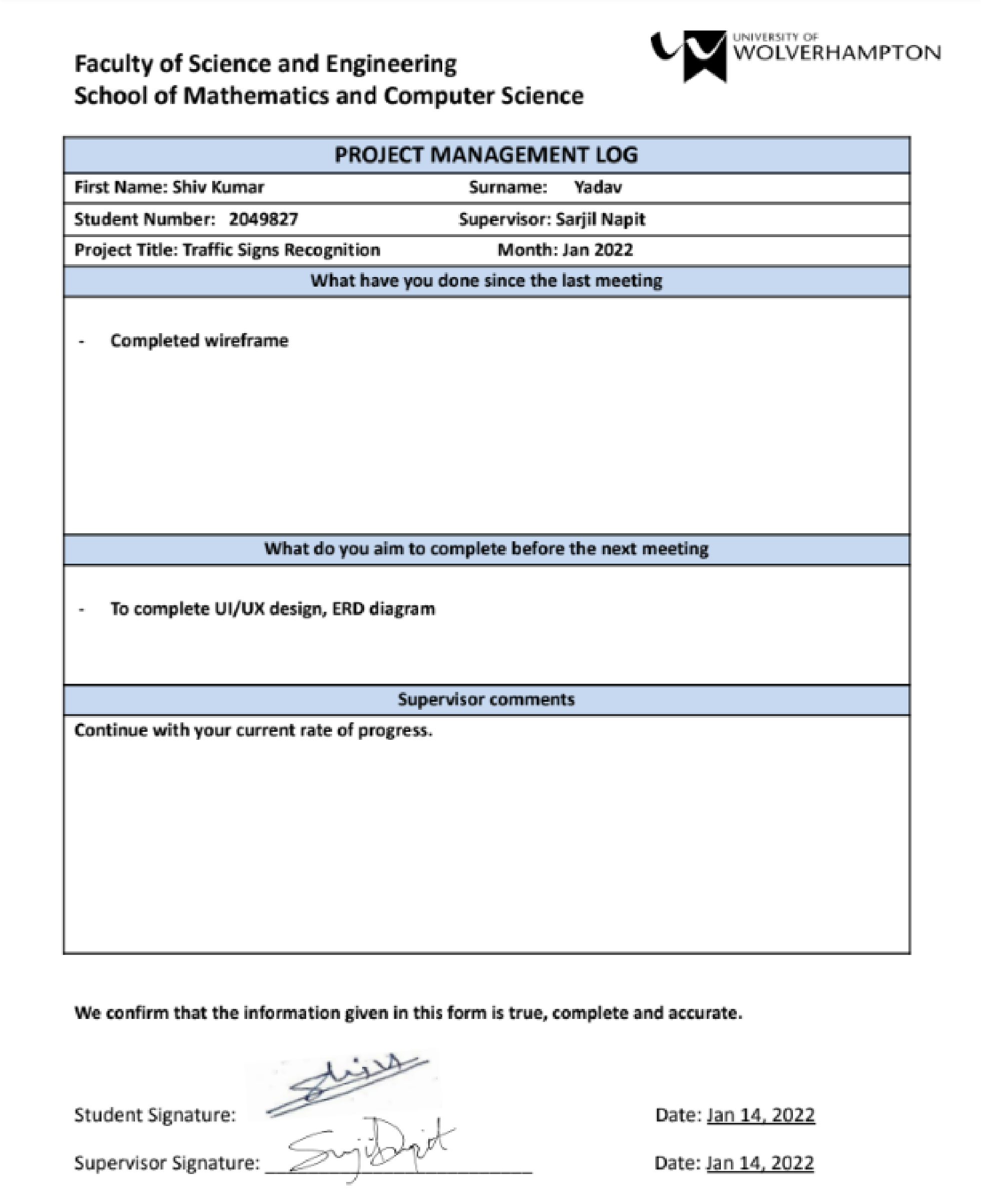
PROJECT MANAGEMENT LOG	
First Name: Shiv Kumar	Surname: Yadav
Student Number: 2049827	Supervisor: Sarjil Napit
Project Title: Traffic Signs Recognition	Month: Jan 2022
What have you done since the last meeting	
- Completed machine learning model building	
What do you aim to complete before the next meeting	
- To complete wireframe and start UI/UX designing	
Supervisor comments	
Continue with your current rate of progress.	

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Date: Jan 7, 2022

Supervisor Signature: ____________________ Date: Jan 7, 2022

Figure 58 Logsheet-6 Jan 7, 2022



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

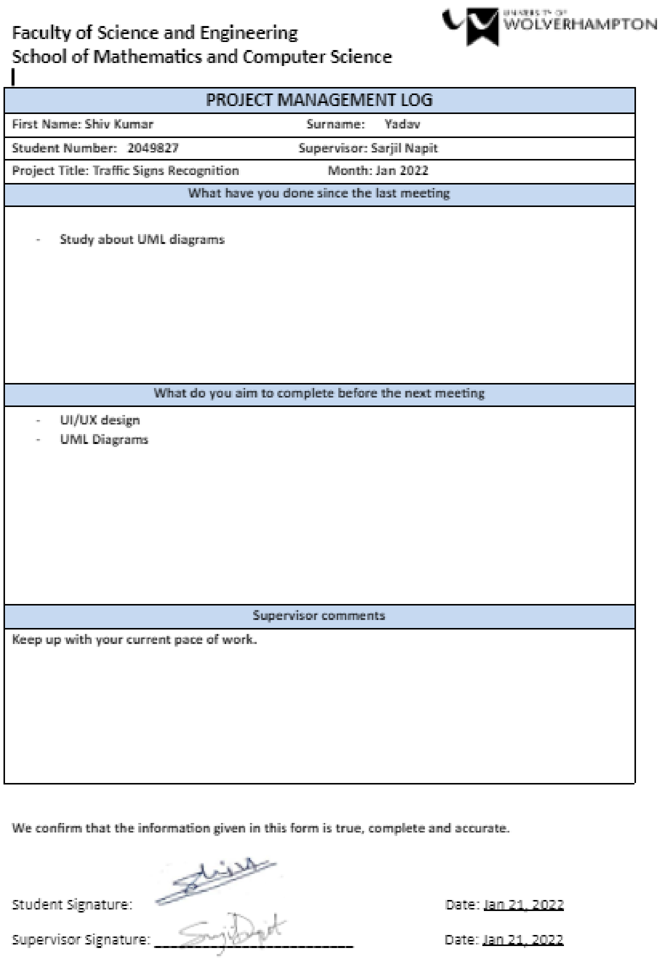
PROJECT MANAGEMENT LOG	
First Name: Shiv Kumar	Surname: Yadav
Student Number: 2049827	Supervisor: Sarjil Napit
Project Title: Traffic Signs Recognition	Month: Jan 2022
What have you done since the last meeting	
- Completed wireframe	
What do you aim to complete before the next meeting	
- To complete UI/UX design, ERD diagram	
Supervisor comments	
Continue with your current rate of progress.	

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Date: Jan 14, 2022

Supervisor Signature: ______________________ Date: Jan 14, 2022

Figure 59 Logsheet-7 Jan 14, 2022



Faculty of Science and Engineering
School of Mathematics and Computer Science

PROJECT MANAGEMENT LOG	
First Name: Shiv Kumar	Surname: Yadav
Student Number: 2049827	Supervisor: Sarjil Napit
Project Title: Traffic Signs Recognition	Month: Jan 2022

What have you done since the last meeting

- Study about UML diagrams

What do you aim to complete before the next meeting

- UI/UX design
- UML Diagrams

Supervisor comments

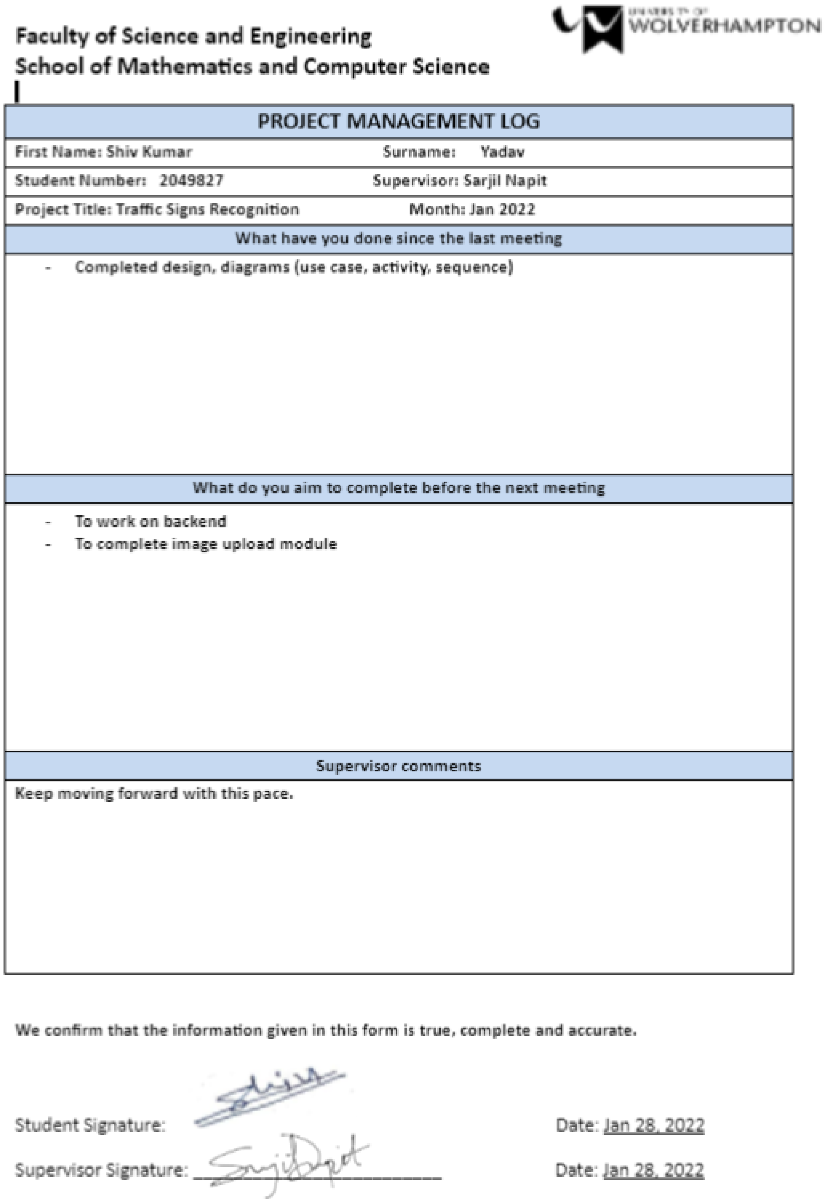
Keep up with your current pace of work.

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Date: Jan 21, 2022

Supervisor Signature: __________________________ Date: Jan 21, 2022

Figure 60 Logsheet-8 Jan 21, 2022



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG

First Name: Shiv Kumar	Surname: Yadav
Student Number: 2049827	Supervisor: Sarjil Napit
Project Title: Traffic Signs Recognition	Month: Jan 2022

What have you done since the last meeting

- Completed design, diagrams (use case, activity, sequence)

What do you aim to complete before the next meeting

- To work on backend
- To complete image upload module

Supervisor comments

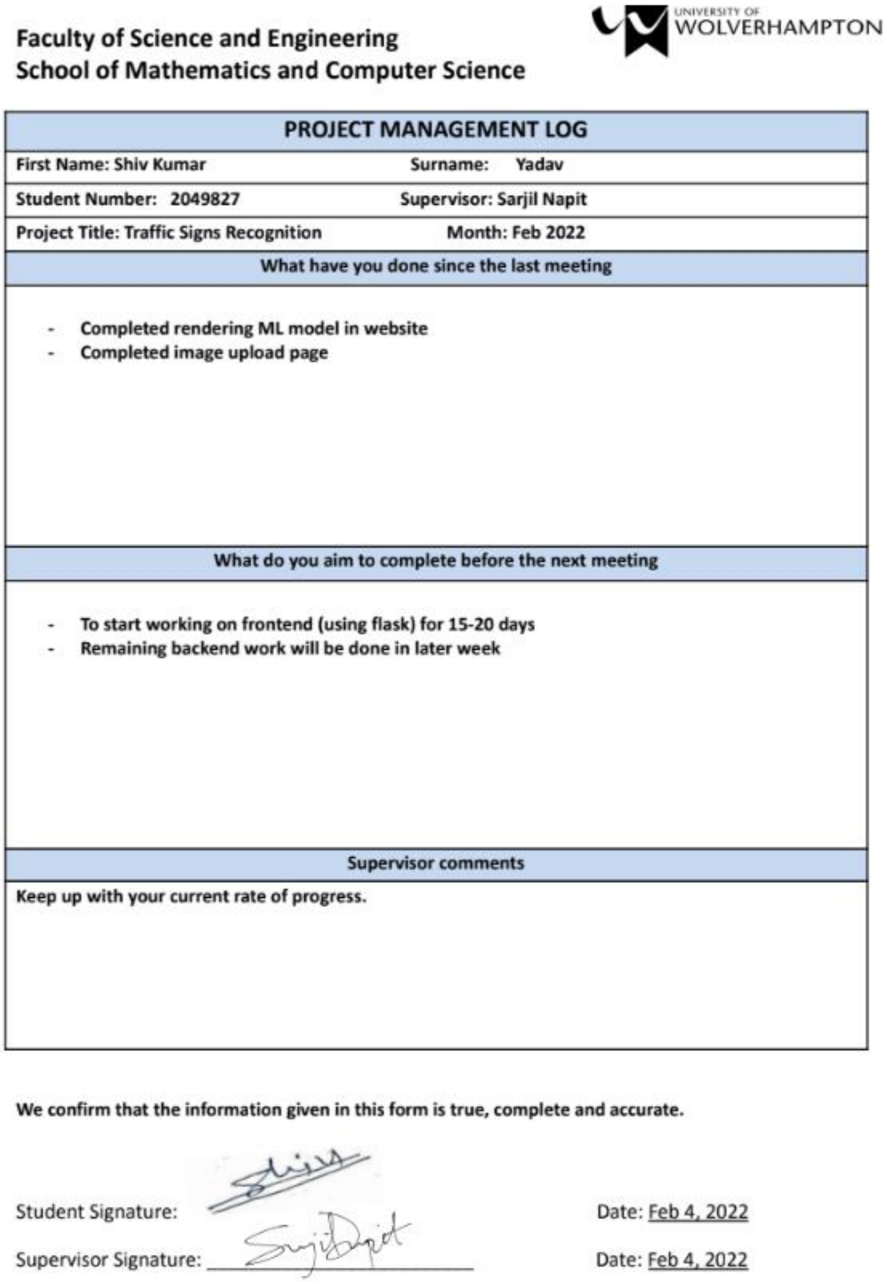
Keep moving forward with this pace.

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Date: Jan 28, 2022

Supervisor Signature: ____________________ Date: Jan 28, 2022

Figure 61 Logsheet-9 Jan 28, 2022



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

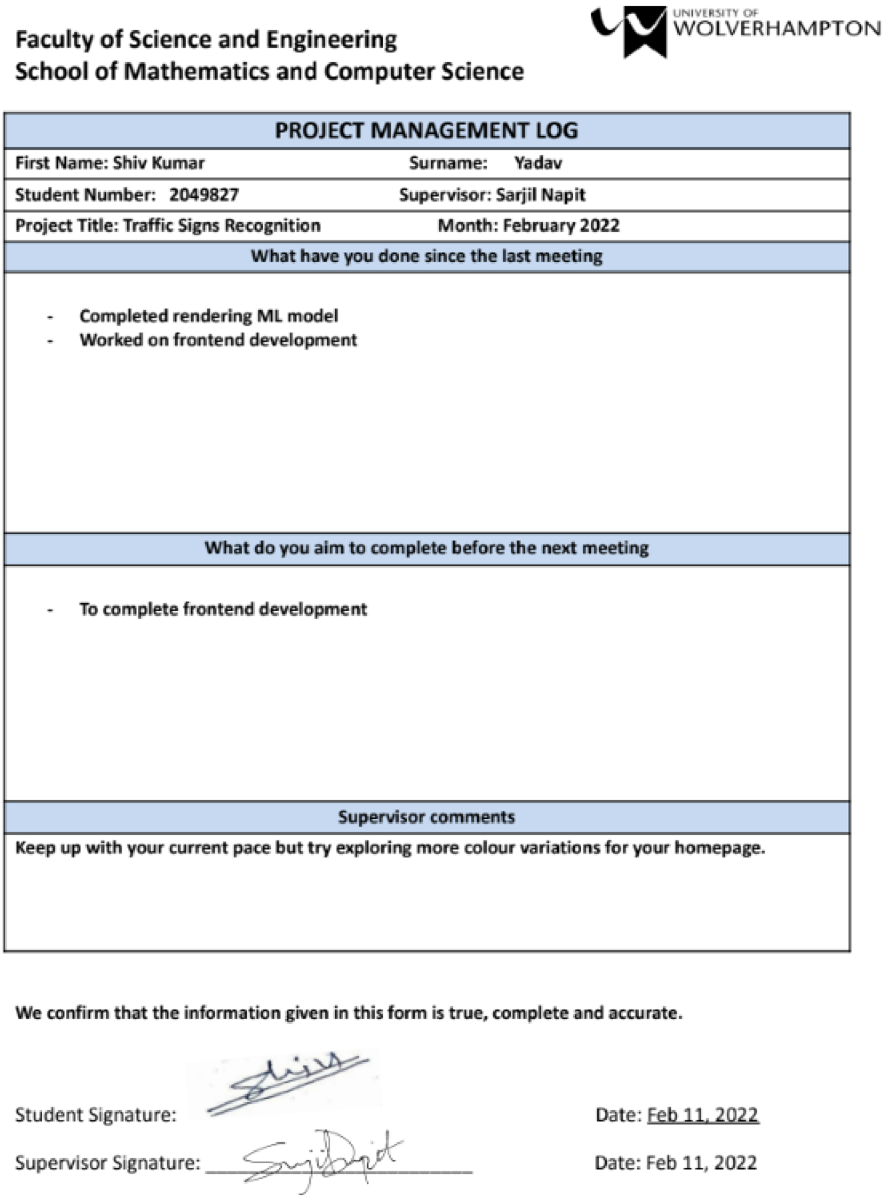
PROJECT MANAGEMENT LOG	
First Name: Shiv Kumar	**Surname: Yadav**
Student Number: 2049827	**Supervisor: Sarjil Napit**
Project Title: Traffic Signs Recognition	**Month: Feb 2022**
What have you done since the last meeting	
- **Completed rendering ML model in website** - **Completed image upload page**	
What do you aim to complete before the next meeting	
- **To start working on frontend (using flask) for 15-20 days** - **Remaining backend work will be done in later week**	
Supervisor comments	
Keep up with your current rate of progress.	

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Date: Feb 4, 2022

Supervisor Signature: ______________________ Date: Feb 4, 2022

Figure 62 Logsheet-10 Feb 4, 2022



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG	
First Name: Shiv Kumar	**Surname: Yadav**
Student Number: 2049827	**Supervisor: Sarjil Napit**
Project Title: Traffic Signs Recognition	**Month: February 2022**

What have you done since the last meeting

- **Completed rendering ML model**
- **Worked on frontend development**

What do you aim to complete before the next meeting

- **To complete frontend development**

Supervisor comments

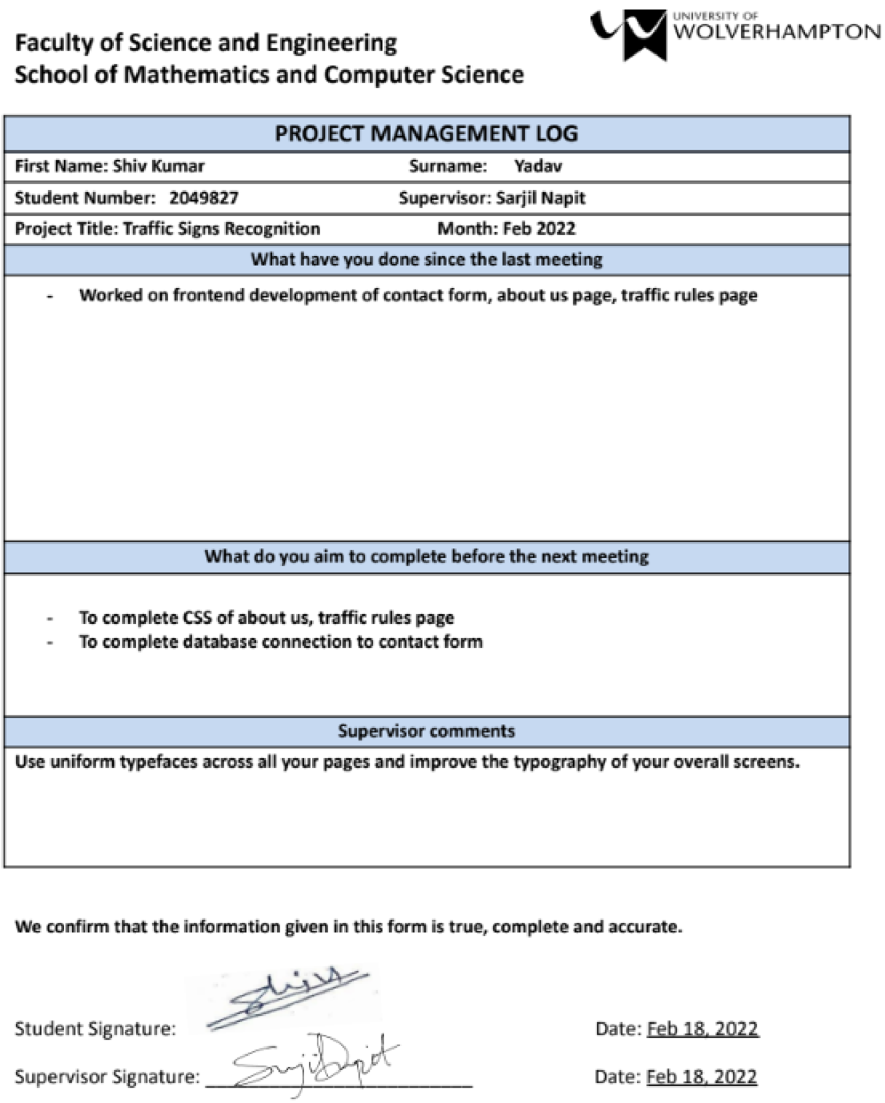
Keep up with your current pace but try exploring more colour variations for your homepage.

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Shiv Date: Feb 11, 2022

Supervisor Signature: ________________ Date: Feb 11, 2022

Figure 63 Logsheet-11 Feb 11, 2022



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG

First Name: Shiv Kumar	Surname: Yadav
Student Number: 2049827	Supervisor: Sarjil Napit
Project Title: Traffic Signs Recognition	Month: Feb 2022

What have you done since the last meeting

- Worked on frontend development of contact form, about us page, traffic rules page

What do you aim to complete before the next meeting

- To complete CSS of about us, traffic rules page
- To complete database connection to contact form

Supervisor comments

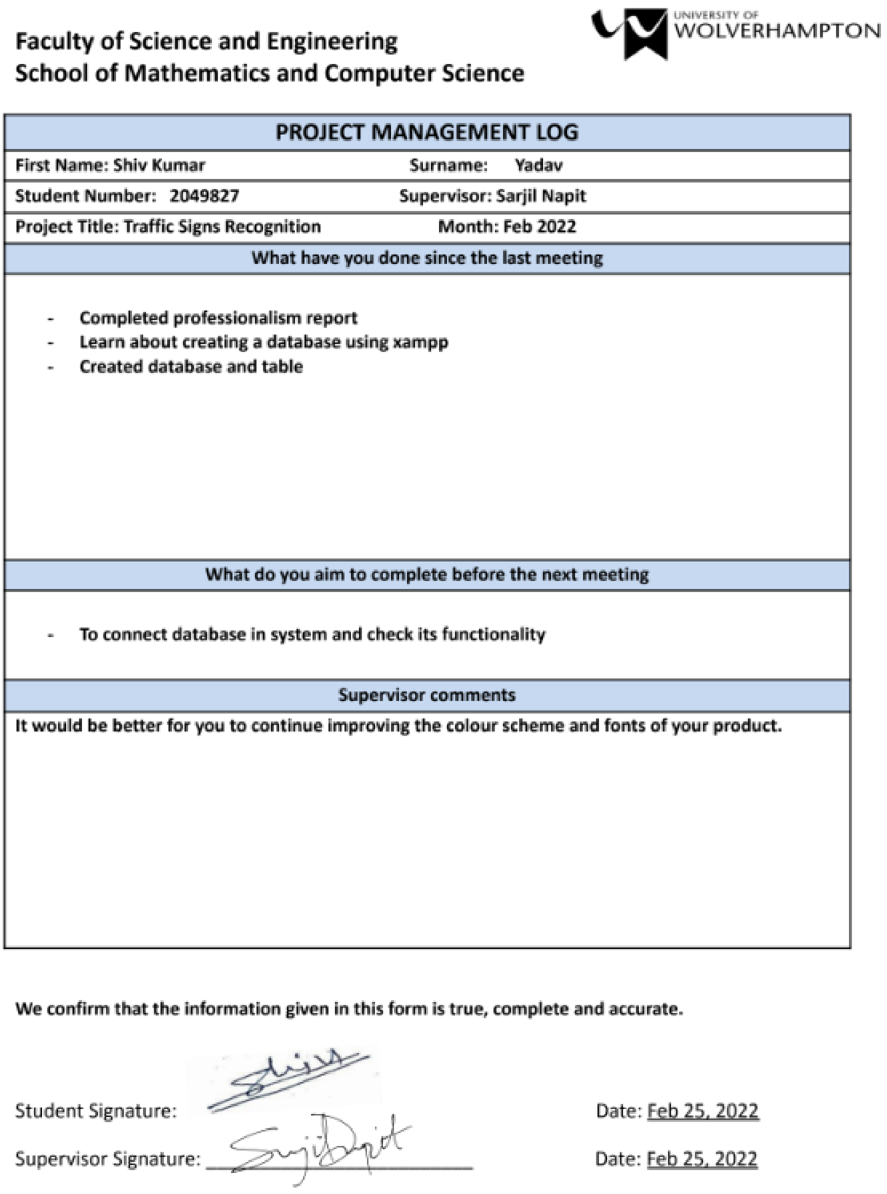
Use uniform typefaces across all your pages and improve the typography of your overall screens.

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Date: Feb 18, 2022

Supervisor Signature: ______________________ Date: Feb 18, 2022

Figure 64 Logsheet-12, Feb 18,2022



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG	
First Name: Shiv Kumar	**Surname: Yadav**
Student Number: 2049827	**Supervisor: Sarjil Napit**
Project Title: Traffic Signs Recognition	**Month: Feb 2022**

What have you done since the last meeting

- **Completed professionalism report**
- **Learn about creating a database using xampp**
- **Created database and table**

What do you aim to complete before the next meeting

- **To connect database in system and check its functionality**

Supervisor comments

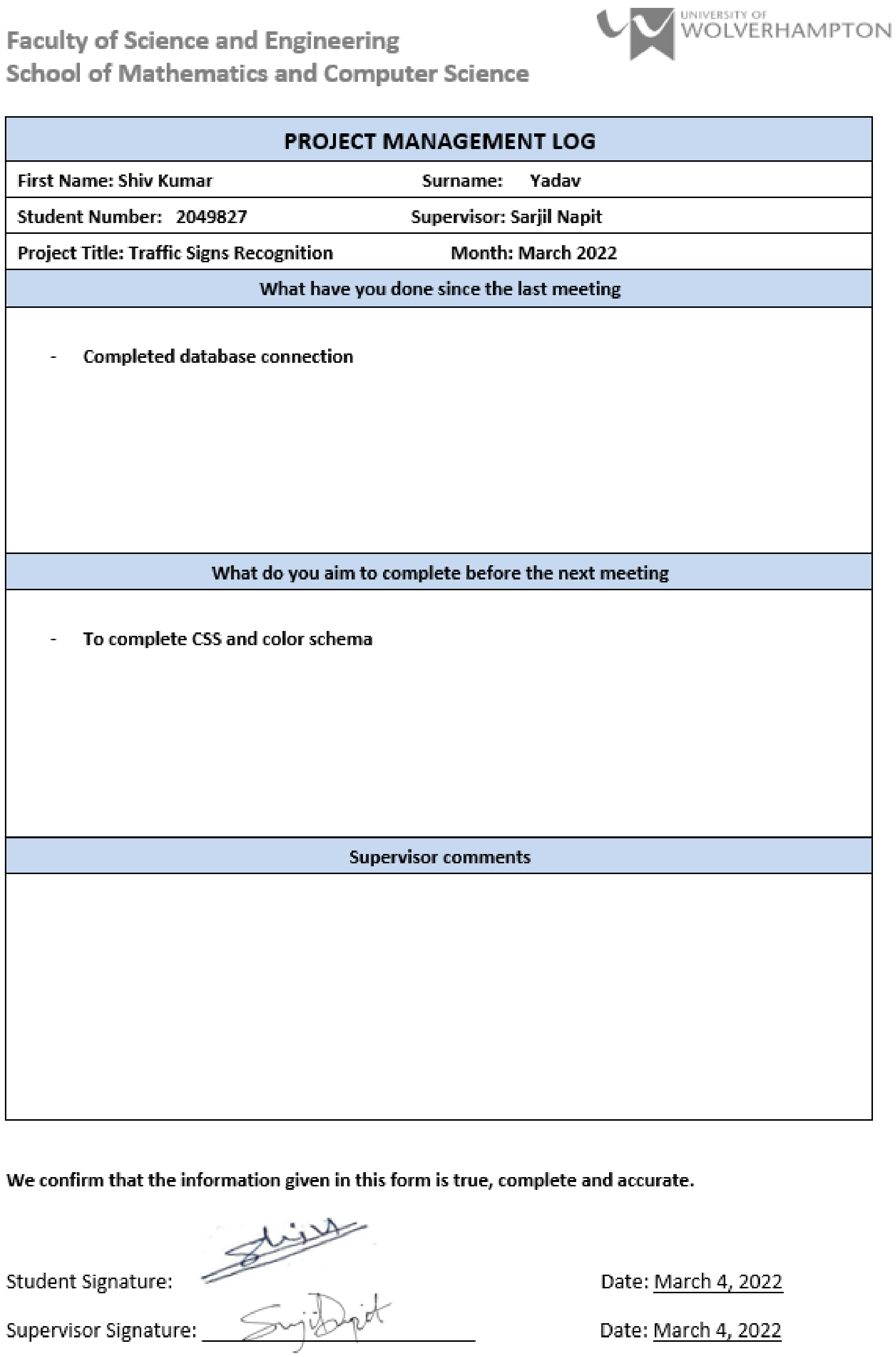
It would be better for you to continue improving the colour scheme and fonts of your product.

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Shiv Date: Feb 25, 2022

Supervisor Signature: ________________ Date: Feb 25, 2022

Figure 65 Logsheet-13 Feb 25,2022



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

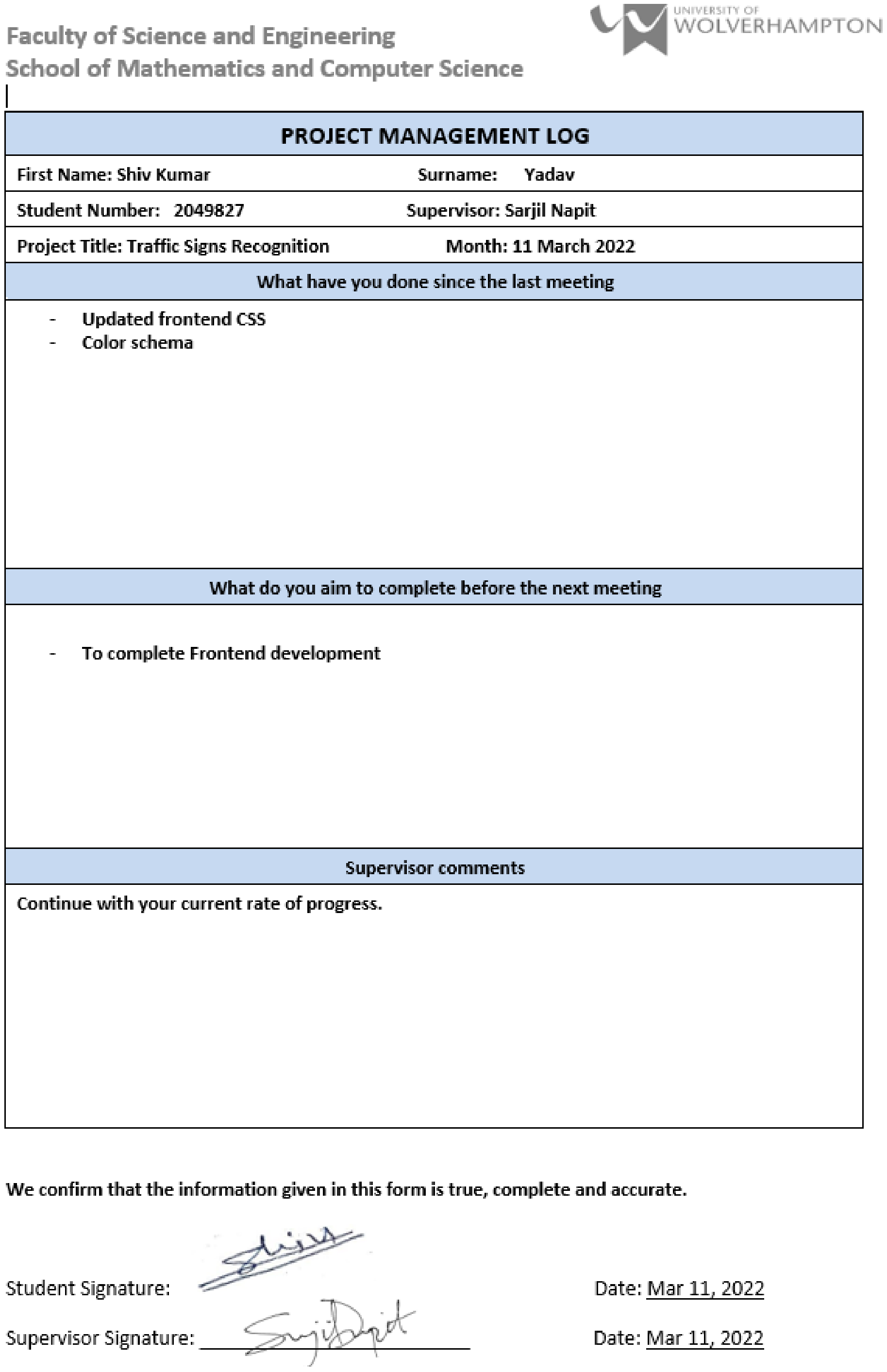
PROJECT MANAGEMENT LOG	
First Name: Shiv Kumar	Surname: Yadav
Student Number: 2049827	Supervisor: Sarjil Napit
Project Title: Traffic Signs Recognition	Month: March 2022
What have you done since the last meeting	
- Completed database connection	
What do you aim to complete before the next meeting	
- To complete CSS and color schema	
Supervisor comments	

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Date: March 4, 2022

Supervisor Signature: ____________________ Date: March 4, 2022

Figure 66 Logsheet-14 March 4, 2022



Faculty of Science and Engineering
School of Mathematics and Computer Science

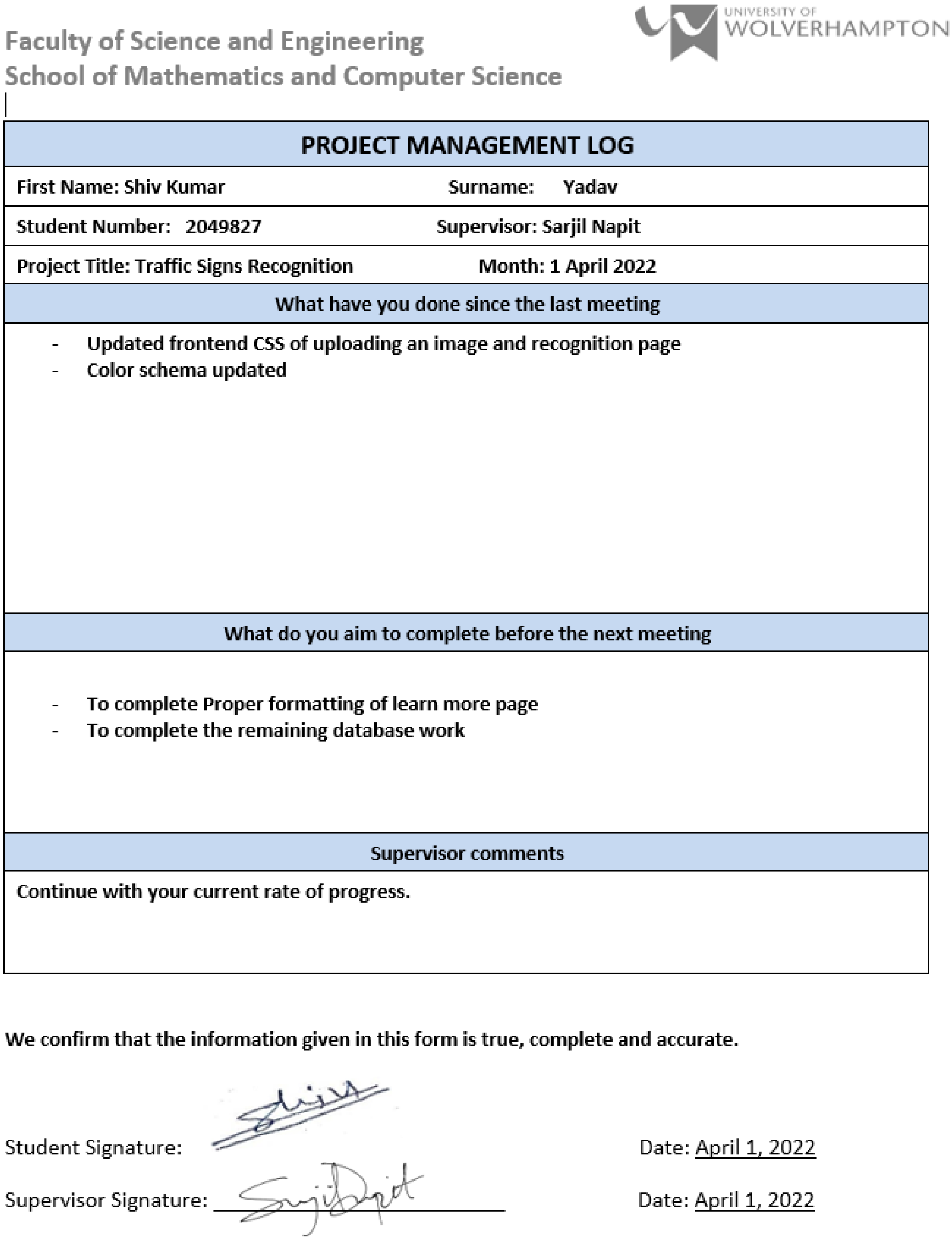
PROJECT MANAGEMENT LOG	
First Name: Shiv Kumar	**Surname: Yadav**
Student Number: 2049827	**Supervisor: Sarjil Napit**
Project Title: Traffic Signs Recognition	**Month: 11 March 2022**
What have you done since the last meeting	
- Updated frontend CSS **- Color schema**	
What do you aim to complete before the next meeting	
- To complete Frontend development	
Supervisor comments	
Continue with your current rate of progress.	

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Shiv Date: Mar 11, 2022

Supervisor Signature: ________________ Date: Mar 11, 2022

Figure 67 Logsheet-15 March 11, 2022



Faculty of Science and Engineering
School of Mathematics and Computer Science

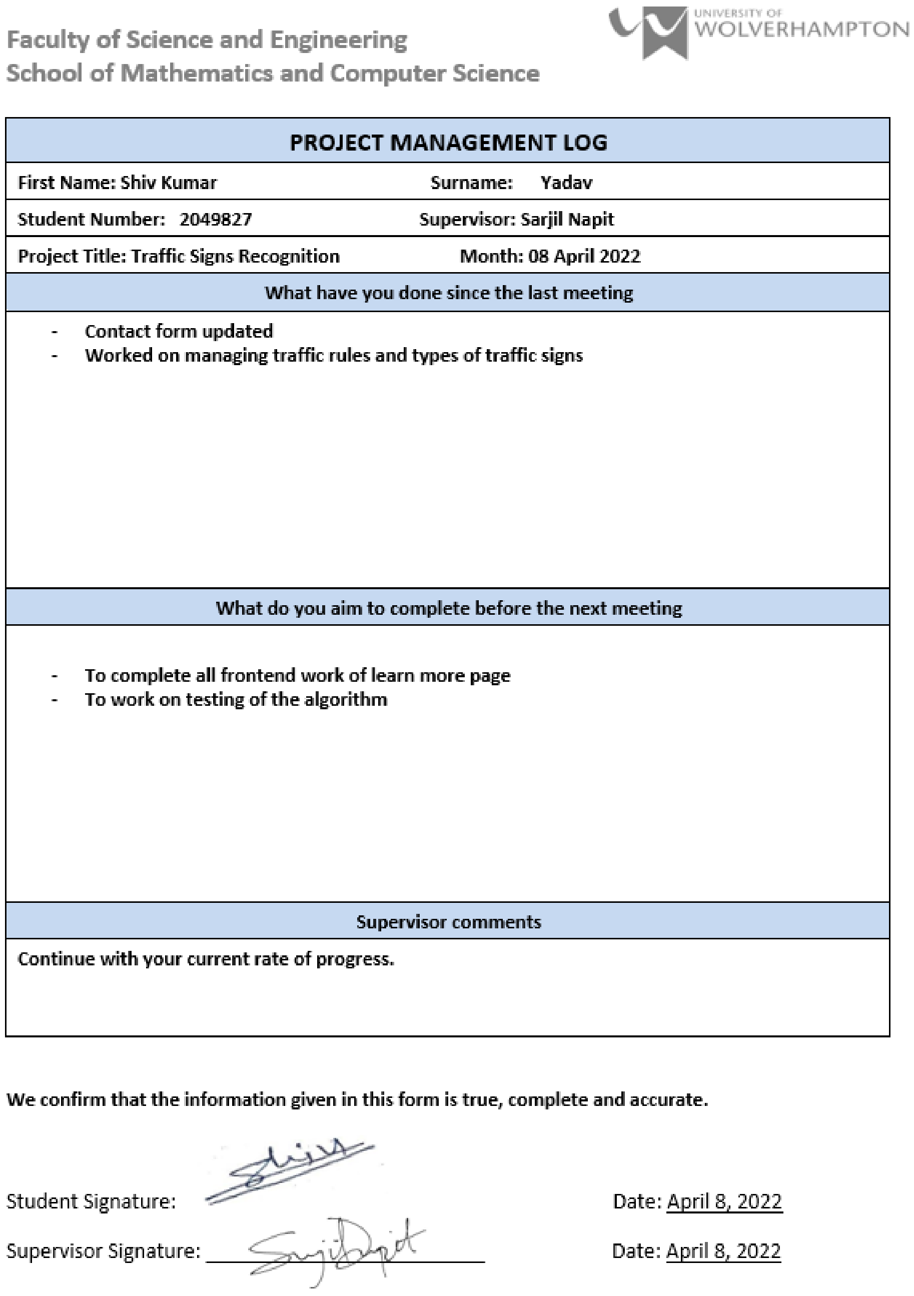
PROJECT MANAGEMENT LOG	
First Name: Shiv Kumar	**Surname: Yadav**
Student Number: 2049827	**Supervisor: Sarjil Napit**
Project Title: Traffic Signs Recognition	**Month: 1 April 2022**
What have you done since the last meeting	
- **Updated frontend CSS of uploading an image and recognition page** - **Color schema updated**	
What do you aim to complete before the next meeting	
- **To complete Proper formatting of learn more page** - **To complete the remaining database work**	
Supervisor comments	
Continue with your current rate of progress.	

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Date: April 1, 2022

Supervisor Signature: ______________ Date: April 1, 2022

Figure 68 Logsheet-16 April 1, 2022



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

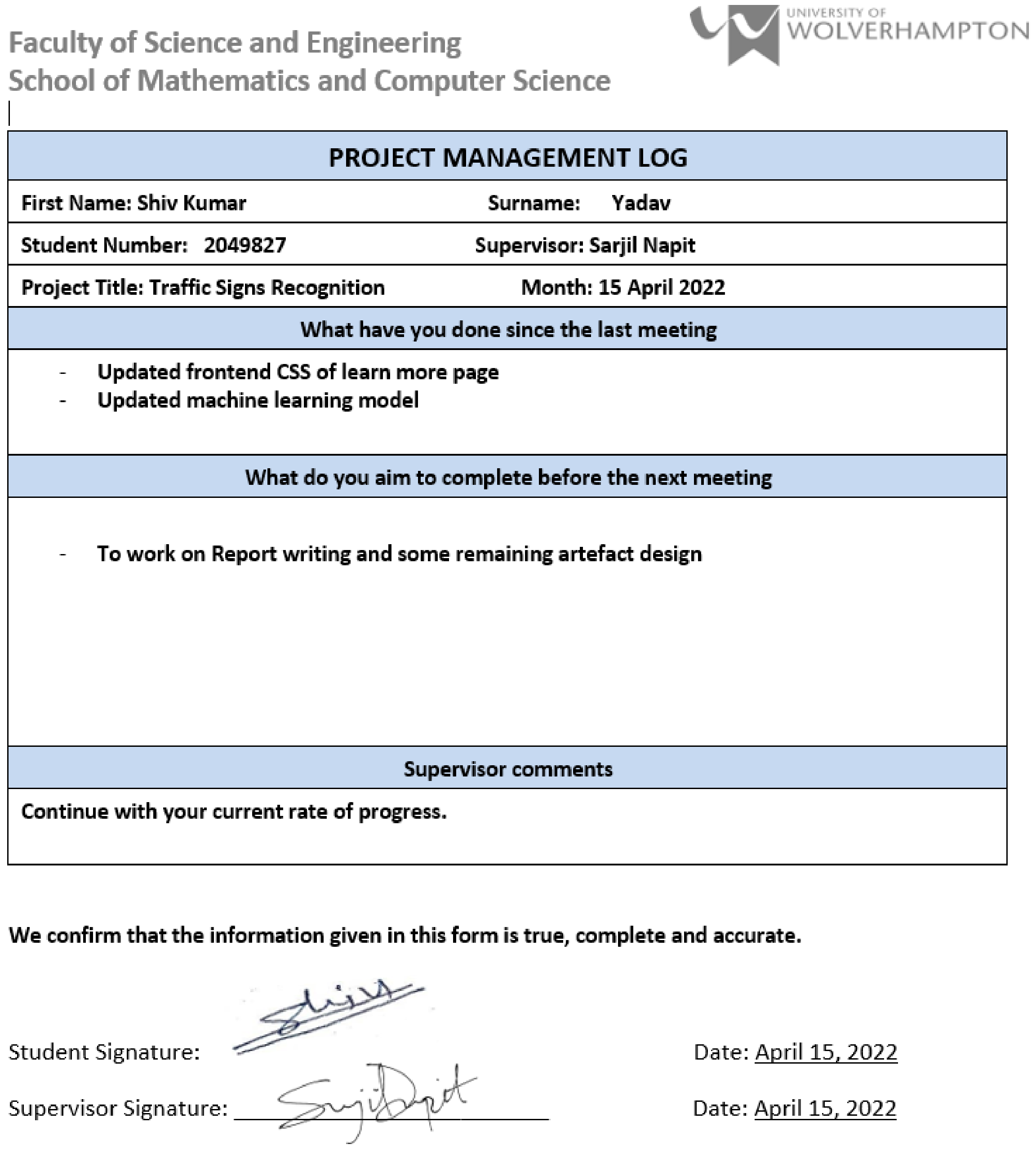
PROJECT MANAGEMENT LOG	
First Name: Shiv Kumar	Surname: Yadav
Student Number: 2049827	Supervisor: Sarjil Napit
Project Title: Traffic Signs Recognition	Month: 08 April 2022
What have you done since the last meeting	
- Contact form updated - Worked on managing traffic rules and types of traffic signs	
What do you aim to complete before the next meeting	
- To complete all frontend work of learn more page - To work on testing of the algorithm	
Supervisor comments	
Continue with your current rate of progress.	

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Date: April 8, 2022

Supervisor Signature: Date: April 8, 2022

Figure 69 Logsheet-17 April 8, 2022



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG

First Name: Shiv Kumar	**Surname: Yadav**
Student Number: 2049827	**Supervisor: Sarjil Napit**
Project Title: Traffic Signs Recognition	**Month: 15 April 2022**

What have you done since the last meeting

- **Updated frontend CSS of learn more page**
- **Updated machine learning model**

What do you aim to complete before the next meeting

- **To work on Report writing and some remaining artefact design**

Supervisor comments

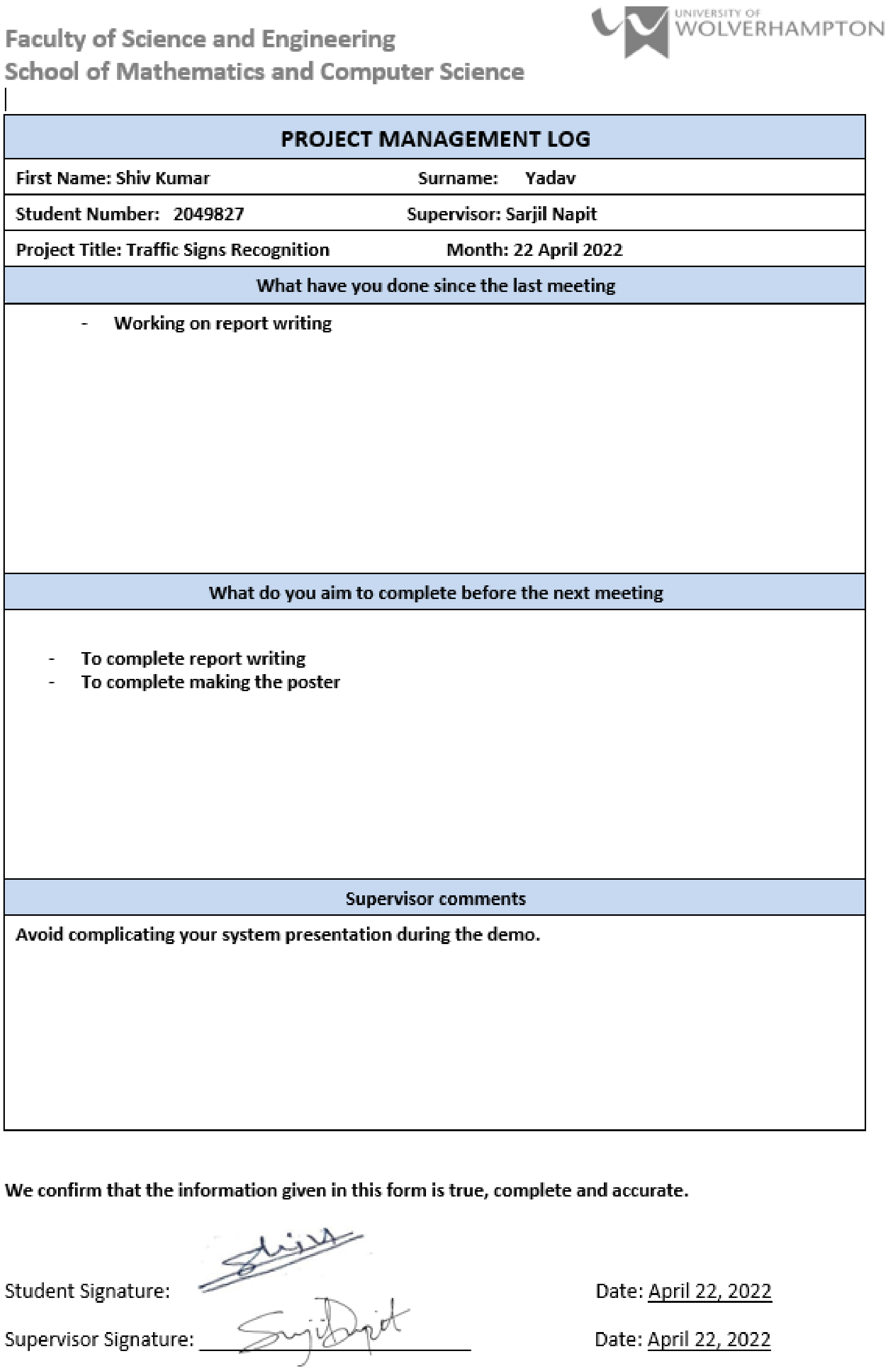
Continue with your current rate of progress.

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Date: April 15, 2022

Supervisor Signature: ______ Date: April 15, 2022

Figure 70 Logsheet-18 April 15, 2022



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG	
First Name: Shiv Kumar	**Surname: Yadav**
Student Number: 2049827	**Supervisor: Sarjil Napit**
Project Title: Traffic Signs Recognition	**Month: 22 April 2022**
What have you done since the last meeting	
- **Working on report writing**	
What do you aim to complete before the next meeting	
- **To complete report writing** - **To complete making the poster**	
Supervisor comments	
Avoid complicating your system presentation during the demo.	

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Date: April 22, 2022

Supervisor Signature: Date: April 22, 2022

Figure 71 Logsheet-19 April 22, 2022

8.2. Gantt Chart

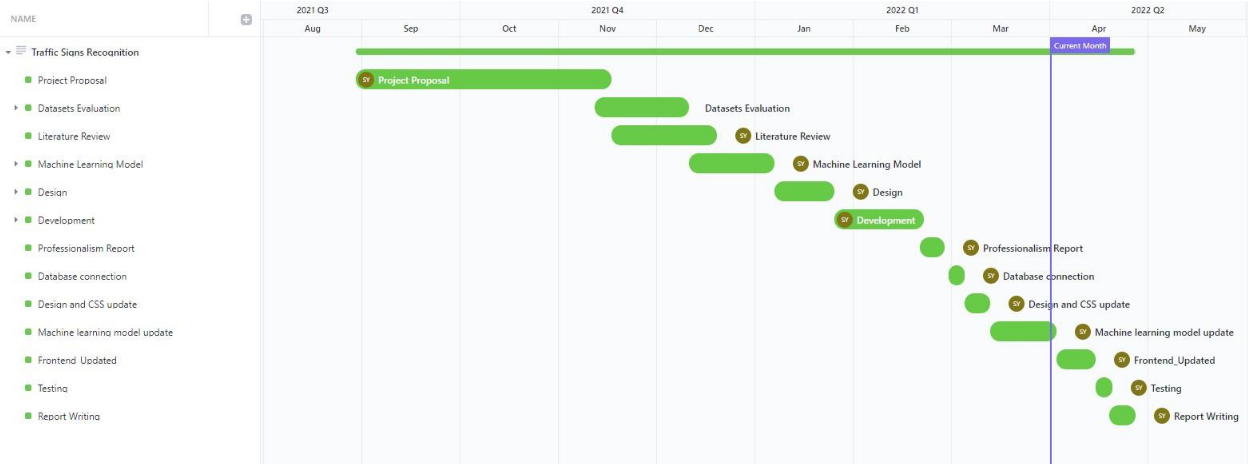
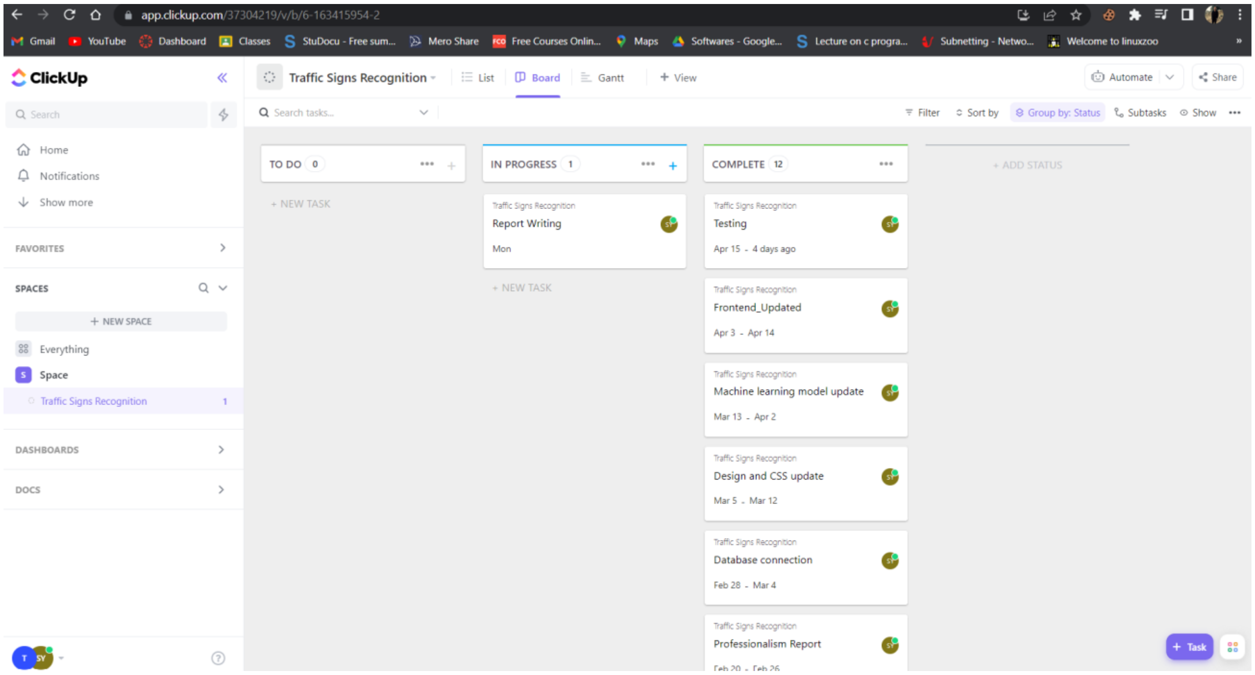


Figure 72 Final Gantt Chart

8.3. Project Management tool

I have used Clickup for project management. As I have previously familiar with this tool and it is easy to use.



TO DO 0

+ NEW TASK

IN PROGRESS 1

Traffic Signs Recognition
Report Writing
Mon

+ NEW TASK

COMPLETE 12

Traffic Signs Recognition
Testing
Apr 15 - 4 days ago

Traffic Signs Recognition
Frontend_Updated
Apr 3 - Apr 14

Traffic Signs Recognition
Machine learning model update
Mar 13 - Apr 2

Traffic Signs Recognition
Design and CSS update
Mar 5 - Mar 12

Traffic Signs Recognition
Database connection
Feb 28 - Mar 4

Traffic Signs Recognition
Professionalism Report
Feb 20 - Feb 26

+ ADD STATUS

Figure 73 Project management tool (ClickUp)

References

¶ Anand, A., 2021. *medium.* [Online] Available at: https://medium.com/data-science-community-srm/introduction-to-cnn-f3aaa8d79891

[Accessed 17 April 2022].

¶ Botekar, N. P. & Mahalakshmi, M., 2017. Development of road sign recognition for ADAS using OpenCV. *International Conference on Intelligent Computing and Control (I2C2).*

¶ Cao, J. et al., 2019. Improved Traffic Sign Detection and Recognition Algorithm for Intelligent Vehicles. *Sensors,* 19(18), pp. 1-21.

¶ Karthika, R. & Murugan, S., 2020. TSD-TSR: Traffic Sign Detection Using Various Techniques Based on Different Conditions. *International Journal of Advanced Research in Engineering and Technology,* 11(9), pp. 1011-1015.

¶ KhabiriKhatiri, R., Latiff, I. A. & Mohamad, A. S., 2021. Road Traffic Sign Detection and Recognition using Adaptive Color Segmentation and Deep Learning. *IEEE International Conference on Signal and Image Processing Applications (ICSIPA),* 5(19), pp. 134-139.

¶ Liao, Y., Yu, N., Li, Z. & Tian, D., 2019. Low power CNN Accelerator for a mobile lensless imaging system. *2019 IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC).*

¶ Li, S. et al., 2019. Deep Learning for Hyperspectral Image Classification: An Overview. *IEEE Transactions on Geoscience and Remote Sensing,* 57(9), pp. 6690-6709.

¶ Moeur's, R. C., 2019. *Manual of traffic signs.* [Online] Available at: http://www.trafficsign.us/signcolor.html [Accessed 16 April 2022].

¶ Rachmadi, R. F., Koutaki, G. & Ogata, K., 2018. Lightweight Spatial Pyramid Convolutional Neural Network for Traffic Sign Classification. *Indonesian Association for Pattern Recognition International Conference (INAPR),* pp. 1-6.

¶ Stojcic, M. et al., 2020. Adaptive neuro-fuzzy model for traffic signs recognition. *19th International Symposium INFOTEH-JAHORINA (INFOTEH).*

¶ Stojcic, M. et al., 2020. Adaptive neuro-fuzzy model for traffic signs recognition. *19th International Symposium INFOTEH-JAHORINA (INFOTEH).*

¶ Swapna, Y. et al., 2021. Deep Learning-Based Road Traffic Sign Detection and Recognition. *Third International Conference on Inventive Research in Computing Applications (ICIRCA),* pp. 1-4.

¶ Tang, Q., Kurnianggoro, L. & Jo, K.-H., 2018. Traffic sign classification with dataset augmentation and convolutional neural network. *Ninth International Conference on Graphic and Image Processing (ICGIP 2017).*

¶ Traffic-Rules, 2022. *Traffic Rules.* [Online] Available at: https://traffic-rules.com/en/nepal/traffic-signs/ [Accessed 16 April 2022].

¶ Wan, H. et al., 2021. A Novel Neural Network Model for Traffic Sign Detection and Recognition under Extreme Conditions. *Journal of Sensors,* Volume 2021, pp. 1-16.

¶ Zhigang, L., Juan, D., Feng, T. & Jiazheng, W., 2021. Traffic Sign Recognition Using an Attentive Context Region-Based Detection Framework. *Chinese Journal of Electronics*, 30(6), pp. 1080-1086.