**D.K.T.E. Society’s Textile and Engineering Institute, Ichalkaranji**

(An Autonomous Institute, Affiliated to Shivaji University, Kolhapur)

Accredited with ‘A+’ Grade by NAAC

Department of Computer Science &

Engineering

2022-2023



THE PROJECT REPORT ON

**Traffic Signal Recognition**

Under the guidance of

**Miss. A. A. Gat**

|  |  |
| --- | --- |
| Developed By: |  |
| Aditya Kulkarni | 20UCS077 |
| Pranav Mali | 20UCS087 |
| Ajinkya Patil | 20UCS096 |
| Kiran Patil | 20UCS099 |
| Prathamesh Patil | 20UCS102 |

**D.K.T.E. Society’s Textile and Engineering Institute, Ichalkaranji**

(An Autonomous Institute, Affiliated to Shivaji University, Kolhapur)

Accredited with ‘A+’ Grade by NAAC

Department of Computer Science & Engineering

**CERTIFICATE**

This is to certify that,

|  |  |
| --- | --- |
| Name | PRN |
| Aditya Kulkarni | 20UCS077 |
| Pranav Mali | 20UCS087 |
| Ajinkya Patil | 20UCS096 |
| Kiran Patil | 20UCS099 |
| Prathamesh Patil | 20UCS102 |

Have successfully completed the project work, entitled,

**Traffic Signal Recognition**

In partial fulfillment for the award of degree of Bachelor of Technology in Computer Science and Engineering. This is the record of their work carried out during academic year 2022-2023.

**Date: Place:** Ichalkaranji

Guide Name Prof. Dr. D.V.Kodavade

Miss. A.A. Gat [Head of Department]

External Examiner

Prof. Dr. Mrs. L.S. Admuthe

[I/C Director]

**DECLARATION**

We the undersigned students of T.Y.C.S.E. declare that the Project work report entitled **Traffic Signal Recognition** written and submitted under the guidance of **Miss. A.A. Gat** is our original work. The empirical findings in this report are based on the data collected by us. The matter assimilated in this report is not reproduction from any readymade report.

Date: Place: Ichalkaranji

|  |  |
| --- | --- |
| Name | PRN |
| Aditya Kulkarni | 20UCS077 |
| Pranav Mali | 20UCS087 |
| Ajinkya Patil | 20UCS096 |
| Kiran Patil | 20UCS099 |
| Prathamesh Patil | 20UCS102 |
|  |  |

**INDEX**

**CONTENTS PAGE NO.**

1. Abstract 5

1. Introduction 6
2. Problem Statement 7
3. Problem Description 7

1. Objectives 8
2. Requirement Specification 9
3. Requirement Analysis 9
4. Problem Solution 10
5. Snapshot 11

10. Flowcharts 13

1. Conclusion 18
2. References 19

**ABSTRACT**

Computer vision is the branch of science of computer and software system which can recognize as well as understand images and scenes. Computer vision is consisting of object detection and many more. Object detection is widely used for face detection, vehicle detection, security systems and self-driving cars. Object detection is a computer vision technique that allows us to identify and locate objects in an image or video. Object detection is a key technology that enable cars to detect driving lanes or perform pedestrian detection to improve road safety

**INTRODUCTION**

Recently the number of road vehicles has increased enormously thanks to the technological achievements in the motor industry and very precisely the availability of low rates. With this remarkable growth, the number of accidents is as well in an infinite raise year after year, due to different causes, in which the ignorance of traffic signs is considered as a major cause of these lasts. Developing automated traffic sign recognition systems helps assisting the driver in different ways in order to guarantee his/her safety, which preserves as well the safety of other drivers and pedestrians. These systems have one main goal, detecting and recognizing traffic signs during the driving process. With these functionalities the system can guide and alert the drivers to prevent danger.

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

In this Python project example, we will build a deep neural network model that can classify traffic signs present in the image into different categories. With this model, we are able to read and understand traffic signs which are a very important task for all autonomous.

**PROBLEM STATEMENT**

Classifying traffic signs for intelligent autonomous vehicles and advanced driver assistance systems in real-world scenarios.

**PROBLEM DESCRIPTION**

Classifying traffic signs is a very important task for autonomous driving systems as the safety of everyone as well as the passenger depends on it. Depending on the country, traffic signs possess a variety in their visual appearance making it harder for classification systems to succeed.

Nowadays, Intelligent Autonomous Vehicles together with Advanced Driver Assistance Systems (ADAS) deal with the problem of traffic sign recognition. It is a challenging real world computer vision problem due to the different and complex scenarios they are placed into. The proposed system will help understand the problem and provide a systematic way of approaching the problem

All Through this project we aim to make a system which can be used universally for traffic signs and symbols, which in turn would help us learn more about deep learning. The objectives of the system is first to detect and then classify signs. To detect, we will build a deep neural network model that can classify traffic signs present in the image into different categories . Now this working makes the results more accurate and we get a model with high accuracy.

**OBJECTIVE**

The benefits are generally focused around driver convenience, which for many is great news. As mentioned above, it should take the stress off attempting to keep on top of speed signs when driving somewhere new,

Also, for those who spend a lot of time on the motorway where variable speed limits are becoming all the more frequent, many systems, such as Ford's traffic sign recognition software, will identify speed limits displayed on motorway gantries.

More advanced versions, such as that offered by Mercedes, will offer you the chance to jump to the new speed limit (with the press of a button) if you are driving using cruise control. There are even systems which will jump for you should you wish them to, so you never miss a speed change again - or in theory at least.

**INPUT**

The input is given in the form of images input by the user and according to user requirement he/she will give input such as:

* .jpeg
* .jpg
* .png

**OUPUT**

Output will generate according to user input like if he has given input of the image then he will get the image description. In short output is based on users input.

**REQUIREMENT SPECIFICATION**

1. The input must be given only through images.

2. The input must be provided only in .jpeg, .jpg, .png formats.

3. Good and stable Internet connectivity is required.

**REQUIREMENT ANALYSIS**

**Step 1:** First the user must select an image.

**Step 2**: Then the user can upload the image on the website.

**Step 3**: After successful upload he can get the image description.

**Step 4**: In order to perform this activity, user has to give the image suitable to the results.

Hence, in such order system will perform different task for image processing according to user demand.

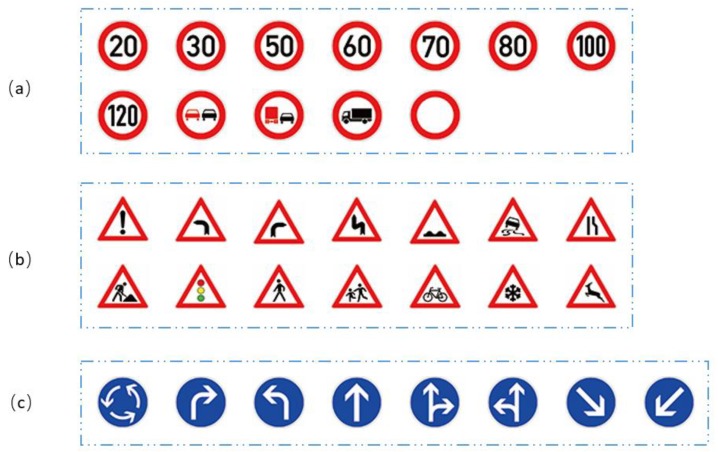
For performing these all step system includes models like Sequential and neural networks. These modules will help the interaction between the user and system.

**PROBLEM SOLUTION**

**ALGORITHM**

1. Begin
2. Input the image from the local directory
3. Submit the Image
4. The image will be processed from a python file
5. Image will be passed to the Sequential model that uses neural networks
6. The model is trained on a large dataset of Traffic Sign Images which are in various forms.
7. The result will be generated.
8. Result will be stored in the text file.
9. The result will get rendered on the webpage.
10. End

**TRAFFIC SIGNS (DATA SET)**



**TRAFFIC SIGNS (DATA SET)**

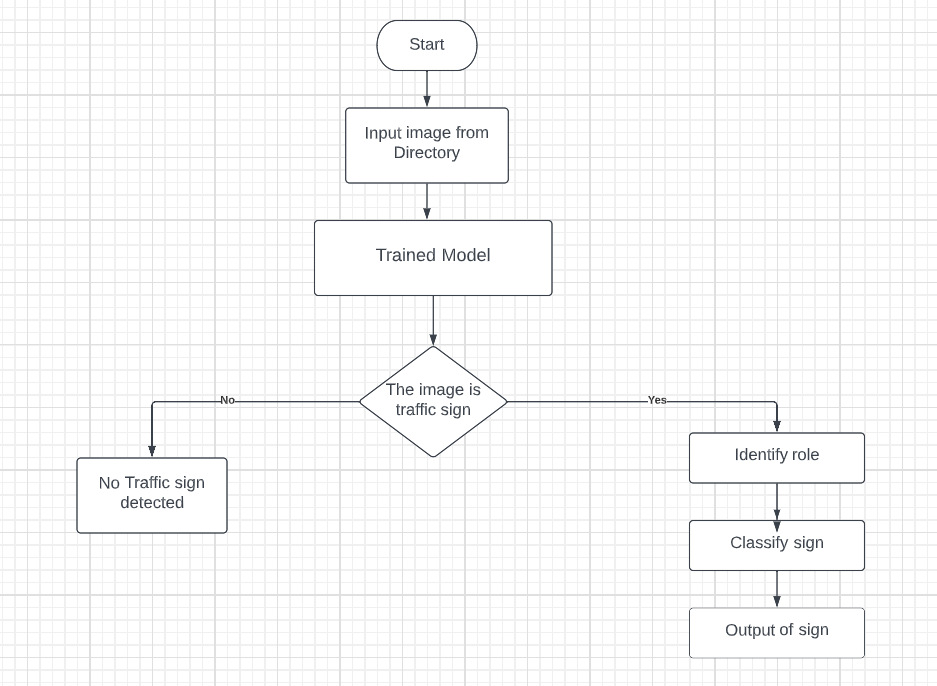


**INFORMATION OF DATABASE**

The input of the neural network needs to be a fixed size, but during data gathering of the traffic signs, the image sizes of the GTSRB varied from “15 × 15” to “223 × 193”, for the varying distances and angles between the camera mounted on a vehicle and the traffic sign. To solve this problem, we resized the images of the dataset to “32 × 32”, before convoluting them using the simplified Gabor filters. In this way, it was ensured that the input of the CNN was fixed to “32 × 32”.

To show the performance of our classification method, we tested the three structures of the CNNs on the dataset of the GTSRB. For training and evaluation, the GTSRB dataset was separated into training datasets and test datasets. For triangular traffic sign recognition in the CNN, we used 8970 images for training and 2790 images for testing, with a total of 15 traffic sign classes. For circular traffic sign recognition in the CNN, we used 22,949 images for training and 7440 images for testing, with a total of 20 traffic sign classes. Finally, for overall traffic recognition in the CNN, we used 39,209 images for training and 12,630 images for testing. For the three CNNs, we trained them for 500 epochs, on the GTSRB dataset with a batch size of 100, respectively.

**FLOW CHART**



**ADVANTAGES AND DISADVANTAGES**

We have now completed detecting moving objects using frame differencing with OpenCV. We saw some advantages and disadvantages. Let us outline them here.

**Advantages**

* The first one is low computation power. As we do not use any neural network or deep learning technique, it is not computationally demanding.
* We can even run it on a CPU. Even a moderately powerful CPU will suffice for employing this detection technique for moving objects.

**Disadvantages**

* First, we can only detect moving objects. If our goal is that, then it is all fine. But we will not be able to detect static objects using this technique. This also means that we cannot use this technique on images but only on videos.
* It cannot be completely real-time as we must wait at least for a certain number of frames to get the background model. We also must get a certain number of frames for differencing and summing and then only we can start detection.
* Using this with static cameras works well. But with moving cameras it will not work at all as the objects will be just everywhere. Therefore, it is best suited for surveillance tasks where the camera is stationary.
* The background model and the moving objects should be distinguishable. The lighting should also be good. Else, we may face issues like double detections for a single object

**APPLICATIONS**

A well-known application of object detection is face detection, that is used in almost all the mobile cameras. A more generalized (multi-class) application can be used in autonomous driving where a variety of objects need to be detected. Also, it has a important role to play in surveillance systems. These systems can be integrated with other tasks such as pose estimation where the rest stage in the pipeline is to detect the object, and then the second stage will be to estimate pose in the detected region. It can be used for tracking objects and thus can be used in robotics and medical applications. Thus, this problem serves a multitude of application.



**HARDWARE AND SOFTWARE SPECIFICATION**

|  |  |
| --- | --- |
| **Hardware and software** | **Characteristics** |
|  |  |
| Memory | 8.0 GB |
|  |  |
| Processor | Intel Core i5 CPU @ 2.40GHZ |
|  |  |
| Graphics | NVIDIA GeForce RTX 2060 6GB GDDR6 |
|  |  |
| Operating system | Windows 10 Home 64 |
|  |  |

**SOFTWARE** **REQUIREMENTS**

|  |  |
| --- | --- |
| Sr. No. | Software Requirements |
| 1. | Keras 2.9.0 |
| 2 | Tensorflow 2.5.0 |

**CONCLUSION**

This project considers an implementation of the classification algorithm for the traffic signs recognition task. Combined with preprocessing and localization steps from previous works, the proposed method for traffic signs classification shows very good results: 95.94% of correctly classified images.

The proposed classification solution is implemented using the TensorFlow framework. The use of our TSR algorithms allows processing of video streams in real-time with high resolution, and therefore at greater distances and with better quality than similar TSR systems have.

FullHD resolution makes it possible to detect and recognize a traffic sign at a distance up to 50 m. Also, we plan to use a CNN not only for classification but for object detection too.

**REFERENCES**

[1] German Traffic Sign Database from Kaggle

[2] “Recognition of Traffic Signs” by Sayed Omar Sadat, Vishal Kumar Pal, Kshitij Jassal, International Journal of Scientific research and Review, Volume 07, Issue 03, March 2019.

<https://en.wikipedia.org/wiki/Traffic-Sign_recognition>