Traffic Sign Detection and Recognition for Driverless Vehicles using Convolutional Neural Network in Deep Learning

[1]Syed Khaleel, [2]Khaled Ibrahim Bayzeed, [3]Syed Osman Khaleel,
[4]Dr. Mohammed Jameel Hashmi
[1]BE Student – Dept. Of CSE, ISL Engineering College, TS, India
[2]BE Student – Dept. Of CSE, ISL Engineering College, TS, India
[3]BE Student – Dept. Of CSE, ISL Engineering College, TS, India
[4]Associate Professor – Dept. Of CSE, ISL Engineering College, TS, India

Article Info

Page Number: 1609 - 1623

Publication Issue:

Vol 72 No. 1 (2023)

Article History

Article Received: *15 October 2022*

Revised: *24 November 2022*

Accepted: *18 December 2022*

Abstract

Traffic Signs are essential for assisting drivers in the world to ensure smooth flow of traffic avoiding accidents. Road Symbols are the symbolic representations for communicating the drivers. Traffic signs convey a variety of messages about the road and what you should anticipate as a driver. They control traffic by guiding them to their destinations and informing them in advance of entry, exit, and turn spots. Drivers sometimes overlook traffic signs on the road in order to concentrate on driving, or due to bad weather conditions (e.g., fog, rain, etc.), which can be hazardous for both drivers and pedestrians. These days we are seeing many advancements in automobile technologies which is leading to replacement of human labour. Similarly, in this project we are trying to design a system which helps in detecting traffic signs on roads by ML (Machine Learning) algorithms replacing the drivers in vehicles which can be called as autonomous or driverless vehicles. These driverless vehicles can detect and recognise traffic signs and follow them respectively. Our software system would assist in identifying and detecting traffic signs without causing drivers to lose concentration while travelling. We create a CNN (Convolution Neural Network) model to classify images into their corresponding divisions. For image categorization, CNN is the most effective algorithm and is implemented using TensorFlow. The German Traffic Sign Recognition Benchmark (GTSRB), which consists of about 50,000 images captured by camera, was used to train our suggested CNN model. This system focuses to develop a CNN model that can successfully detect and recognize traffic signs in real-time using OpenCV. The results using our suggested system demonstrate 95 to 100% accuracy.

Keywords: Traffic Signs, Autonomous or Driverless vehicles, Deep Learning, CNN, GTSRB, OpenCV.

---

1. INTRODUCTION

Traffic signs are important for managing traffic, preventing accidents, and ensuring safety because they communicate various standards or regulations that must be adhered to on the road. Speed restrictions, no entry, traffic signals, turns to the left or right, children crossing, no passing of heavy vehicles, etc. are just a few examples of the various kinds of traffic signs. Traffic signs are frequently made of a specific shape and colour with symbols inside so that there is a clear distinction between the sign and the background in order to make them readily readable and recognisable. Traffic accidents that are fatal may occur if these warnings are ignored. It is a mechanised system for identifying and detecting road signs created in such circumstances, it might be very beneficial for drivers who might become preoccupied while driving. The traffic sign pictures are captured by the camera and fed to the model. The deep learning algorithms should be dependable in all of these circumstances because the vehicles are operating on the road in a variety of settings, including low light, rainy, and foggy ones. Convolutional neural networks are the ideal deep learning algorithms to record and recognise these traffic signs in their various orientations. The visibility of road signs can be affected by a variety of temporary and permanent circumstances, as their performance is greatly influenced by the current environmental conditions. Non-identical traffic signs and poor sign posting are challenges that the traffic sign detection and recognising system must overcome. The input layer, concealed layers, and output layer are the three layers that make up a convolutional neural network. One or more secret layers are possible. There are hidden layers that perform convolutional pooling, completely connected layers, and normalisation layers. CNN works very well when it comes to image operations, but it also has some flaws. CNN may become confused if a posture or image's orientation changes. A max-pooling layer is used in the convolutional neural network to downscale the information and minimise the spatial information related to the data.

2. LITERATURE REVIEW

Identification of traffic signs has become an essential element of our existence in today's world. Given the growing traffic, traffic sign classification is critical to ensuring everyone's protection and the future of automated driving. Significant study has been conducted on traffic and road sign identification. *Akatsuka and Imai [1]* conducted the first study on the subject of "Traffic Sign Recognition" in 1987, where they attempted to create a basic system that could recognise traffic signs, alert drivers, and guarantee their safety. However, this was only used to provide automated identification for a subset of traffic signs.

Richa Jain et al. [2] suggest a hybrid method based on MSER (Maximally Stable Extremal Regions) and OCR (Optical Character Recognition) for identifying and recognising traffic text signs. The sample used is Jaguar Land Rover Research's movement content indicator data. The method is split into two steps: detection of text areas and recognition of text. Following text recognition, the Lucy Richardson algorithm is used to remove noise and de-blur the picture. Following that, the picture's contrast is adjusted, and the image is transformed to a binary grayscale image using the rgb2gray tool. For picture edge detection, the Canny edge detection algorithm is used, and the Maximally Stable Extremal Region (MSER) region detection method is applied to the edge augmented image. Finally, the picture is morphologically segmented and geometrically filtered. OCR recognises text contained within the enclosing frame. The main disadvantage of their suggested system is that it only detects text signs (e.g., Left Turn on Green), which is only a partial answer for traffic sign recognition. Also, in the geometric filtration phase, extremely huge and extremely tiny items are rejected, so if a sign is written in very small text, it may be rejected based on the conditions. Given the shortcomings of existing systems, such as detection of only specific signs, specific colours, specific shapes, and no real-time detection, we aimed to improve Autonomous Driver Assistant System (ADAS), provide a system that can detect traffic signs in real-time, and provide a system that can detect signs that are not limited to a single shape or colour while also providing high accuracy. In the following part, we will go over our suggested system in greater depth.

Pranjali Pandey et al [3] recognise traffic signs using the Template Matching method, particularly CCORR_NORMED (A Template Matching technique). Their suggested method is divided into two phases, namely training and testing. Signs are transferred from IRRS (Indian Regulatory Road Signs) during the training process and used to teach the system. Following that, a template is created by directly choosing the road sign and recording it as a ROI. When a template is chosen, it is assigned a name and stored in the database. Also, if the image recorded during template construction is dull due to lighting conditions, Histogram equalisation is used to colour fix that image. During the trial process, the recorded source image is slid over the template image, matching both images pixel by pixel from left to right and top to bottom. All matched pixel values are recorded in a matrix, and once the maximum intensity matrix is acquired, a speech trigger and a display message are generated. Though their suggested system can identify traffic signs in any illumination situation, it cannot detect traffic signs from low-quality blurred pictures because matching the intensities of blurred images during template matching is challenging. Because it is only Template Matching, false positives (showing the existence of a sign that isn't truly present in the picture) are also recorded, reducing the system's effectiveness.

Md. Abdul Alim Sheikh et al. [4] employ a database that includes four traffic signs: a stop sign, a no-entry sign, a give-way sign, and a speed limit sign. They used 300 pictures for instruction, 75 of which were used for each of the four sign types. The writers tested the network with 200 images, 50 photographs per sign category. The writers developed two modules: traffic sign detection and categorization and recognition. To begin, colour space translation and division are performed to determine the presence of a traffic sign. If the Region of Interest (ROI) is between 100 and 3000 pixels, it is deemed a traffic sign, and all other noise is ignored. If it is visible, the sign will be emphasised and its size will be normalised. The data is then categorised using a Neural Network (Multi-Layer Perceptron (MLP) in this case).It is then tried against the testing dataset and assessed for detection and recognition phase accuracy. Because their suggested method concentrates only on four specific signs (i.e., Give Way, No Entry, Stop, and Speed Limit), the recognition of these signals is highly accurate. Because only 500 pictures are used to train the MLP here, computing power is reduced, allowing it to operate on low-end PC setups. However, their proposed system can only identify four kinds of traffic signs: stop, no entry, give way, and speed limit. As a result, its ability to recognise traffic signs is severely restricted. Furthermore, the suggested system cannot detect traffic signs that are present in a collection, which means that if numerous signs are present in an input picture, the system will fail to recognise all of them and will detect only one of the four kinds of traffic signs mentioned above.

A. Gudigar et al. [5] describe a system for Automatic Traffic Sign Detection and classification (ATSDR) that consists of three modules: segmentation, detection, and classification. A novel environmental selection approach is used to extract the Region of Interest (ROI) using numerous thresholding methods. The identification of traffic signs is accomplished through correlation computation between log-polar mapped interior areas and the reference template. Finally, the classifier Support Vector Machine (SVM) is used for identification. This algorithm obtained 90% recognition accuracy, demonstrating the robustness of traffic sign detection and recognition in a real-world situation.

Hee Seok Lee et al. [6] proposed a traffic sign identifying system that calculates the location and exact boundaries of traffic signs at the same time. (CNN). This study formulates the border estimation of traffic signals as a 2D pose and shape class prediction problem, and a single CNN successfully resolves this issue. Author approximated the actual boundary of the target sign by projecting the boundary of a matching template sign image into the input image plane with the expected 2D posture and the shape class of a target traffic sign in an input image.

Real-time traffic sign identification, or quickly determining what kind of traffic sign is present where in an input picture, was the focus of *Y. Yang et al.'s [7]* research. The writers suggested a detection tool to accomplish this. This detection module is based on the extraction and categorization of traffic sign proposals and is based on a colour likelihood model and a colour HOG. To further categorise the observed signs into their subcategories within each superclass, a convolutional neural network is used.

A Color Global and Local Oriented Edge Magnitude Pattern (Color Global LOEMP). was suggested by *X. Yuan et al. [8]*. It balances the two considerations of uniqueness and resilience by combining color, global spatial structure, global direction structure, and local shape information. Due to the context frame being created by the geometry of the traffic sign, colour angular patterns are suggested to provide colour distinguishing information and a context frame is established to provide global spatial information. The writers also suggest using an LOEMP to symbolise each cell. The HOG feature in each cell describes the distribution of the orientation patterns, and each direction of the HOG is then specifically depicted by the frequency of the local binary pattern histogram in that direction.

A recent method done by *A. Ellahyani, I. E. Jaafari, and S. Charfi [9]* using color, shape, and ML algorithm-based methods for traffic sign detection and recognition intelligent transportation systems that inform drivers about safety precautions and sign-related information, and provided comparative information on the same. The words taken into account in the evaluation of the detection module include colour space, segmentation method, features, and form detection method. The comparison between these techniques using datasets from various nations is presented in the article.

3. PROPOSED SYSTEM

Our proposed system employs a convolutional neural network approach/algorithm, which improves recognition and detection accuracy, also it allows for the extraction of additional information. The goal of this system is to train and evaluate a cost-sensitive CNN model using the GTSRB dataset.

ISSN: 2094-0343
2326-9865

Later, the model is used to categorise pictures put in front of a camera which is detected using OpenCV.

The initial goal is to divide the image-filled dataset into train and test data, then we use grayscale and histogram equalisation algorithms to pre-process the images in the dataset. We then move ahead to create a CNN model that is cost-sensitive. Afterwards we train and test the model using the appropriate datasets. After training and testing, we put the model to the test with images that should be detected and recognized by OpenCV.

4. SYSTEM ARCHITECTURE

This project will be built in Python. We are working on developing a model that detects traffic signs. We created a deep neural network model to determine which traffic sign is visible in the image. Our dataset has a train folder, which contains additional folders, each of which represents a distinct class.

As we walk through the system architecture, we will find that there are several processes that are followed to create this project. These procedures are separated into two phases, which are Phase 1 and Phase 2 respectively, as shown in the figure below.

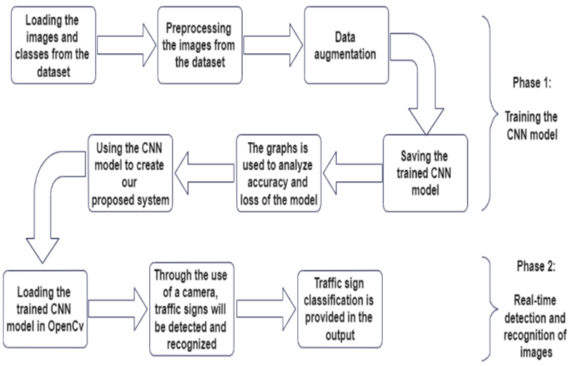

Fig 1: SYSTEM ARCHITECTURE

5. IMPLEMENTATION

We created a model that efficiently detects and recognizes traffic sign images and learns to select the most appropriate characteristics for this challenge on its own by applying deep learning to it. In this paper, we constructed a deep learning architecture capable of detecting and recognising traffic signs.

When adopting deep neural network approaches, the model will demand a big amount of data and massive matrix multiplication operations, which will necessitate additional computer capacity, so to handle this issue a new algorithm was implemented called Convolutional Neural Network. For computer vision tasks, Convolutional Neural Network has been found to be very much efficient and faster when compared traditional deep neural network.

To train and test the model, we utilised the German Traffic Sign Dataset (GTSRB), which comprises over 50,000 traffic sign pictures classified into 43 classes (for example, Speed Limit

20km/h, Speed Limit 50km/h, Children Crossing, Stop sign, and so on). This dataset is large enough to enable us train the model more precisely and obtain better outcomes.

There are various procedures that are followed which are given below: -

i) Collecting Data

Download the German traffic sign dataset from (https://www.kaggle.com/datasets/meowmeowmeomeowmeowm/gtsrb-german-traffic-sign).To compute summary statistics for the traffic signs dataset, we utilised the NumPy library:

The training set is 34799 in size.

The validation dataset is 4410 in size.

The test set has a total size of 12630.

The data collection contains a total of 43 distinct classifications.

ii) Exploration and visualisation of datasets

To begin, we will examine the dimensions of all the images in the collection so that we can process the images to have identical dimensions. The pictures in this dataset have different dimensions ranging from 16*16*3 to 128*128*3 and hence cannot be fed straight to the CNN model.

Fig 2: Traffic sign images of all the 43 classes

Next, we will investigate our data by creating a distribution plot, which will provide us with more information about our data and the amount of images each class has.

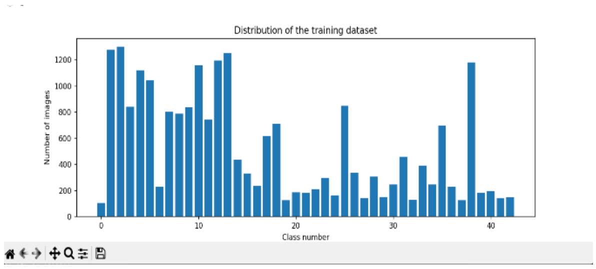

Fig 3: Distribution of images from Training data

As illustrated in the data histogram figure above, there is also a considerable imbalance among classes in the training set. Some classes include fewer than 200 photographs, while others contain more than 2000. This indicates that our model may be biased towards more data in particular classes, especially if its predictions are uncertain.

These images are nothing but samples that are taken from real-world setting. Also our model should deal with all of these circumstances. As a result, it is preferable not to cut short our dataset to achieve balanced data.

iii) Data Pre-processing

We must first compress the traffic sign pictures to a one dimension. To minimize reducing too much details and expanding the picture excessively, we must determine the dimension that lies in between and retain the exact image data. As a result, we chose to adjust each traffic sign image to 32×32x3 dimensions.

Fig 4: Gray scaling of image before and after

The images will then be converted to augmented images, which will allow our algorithm to detect more characteristics in the images. As a result, pre-processing is a critical step since it minimises the number of features and hence the time required for execution.

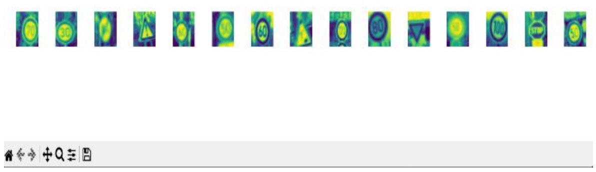

Fig 5: Augmentation of images

iv) Architecture Model

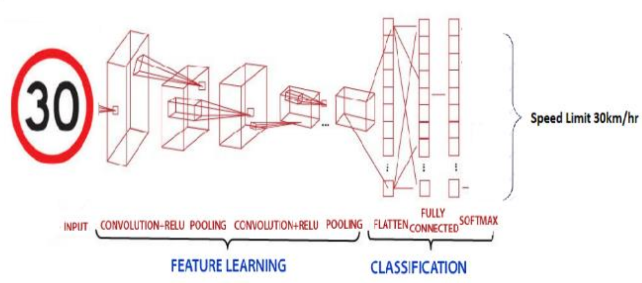

Fig 6: Model of Convolutional Neural Network

A Convolutional Neural Network (CNN) is a form of neural network that is trained in image processing using a grid-like topology. A CNN is build using 3 layers: convolutional layer, pooling layer, and fully connected layer.

Layers	Description
Input Layer	32x32x1 images
Convolution-1	Convolution and rectified linear activation (ReLU).
Pool-1	Max pooling.
Convoution-2	Convolution and rectified linear activation
Pool-2	Max pooling.
Local-3	Fully connected layer with ReLU
Local-4	Fully connected layer with ReLU
softmax	Classification result

Fig 7: Architecture of Convolutional Neural Network

1. The INPUT layer will keep the input picture as a three-dimensional array of pixel values.

2. The CONV layer will determine the dot product of a kernel and a sub-array of an input picture of the same size as a kernel. Then it will combine all of the values returned by the dot product to form the single pixel value of an output image. This technique is continued until the entire input picture has been covered.

3. The RELU layer will apply the activation function max (0, x) to all pixel values in an output picture.

4. The down sampling along the width and height of a image is done in POOL layer, resulting in a smaller image dimension.

5. The class score for each classification category will be calculated by the Fully-Connected layer.

6. Convolutional Neural Network does this by layering the actual image from the initial values of pixels to the resulting class labels. The constant function is carried out by the ReLU and POOL layers. In this layer the parameters are not at all trained. The gradient descent optimizer is used to train the parameters at the Convolutional layer and Fully Connected layer.

v) Training the model

'model.fit()' was used to train the model after the model architecture was built. We utilised an Adam optimizer to train the model. The model performed admirably with a batch size of 30 and an epoch size of 50. Our model obtained a 94% accuracy rate on the training dataset with low loss and we achieved a test accuracy of 99%. The matplotlib library was used to represent the graph. The train data in accuracy and loss graph is shown by the blue line, while the test data in accuracy and loss graph is shown by the orange line.

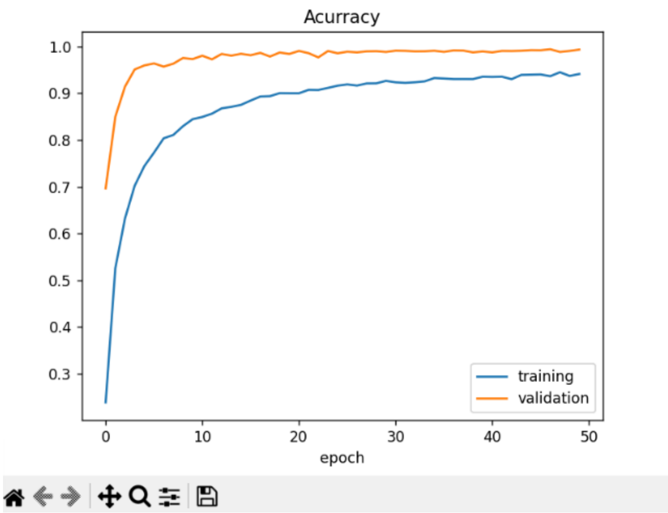

Fig 8: Accuracy Graph

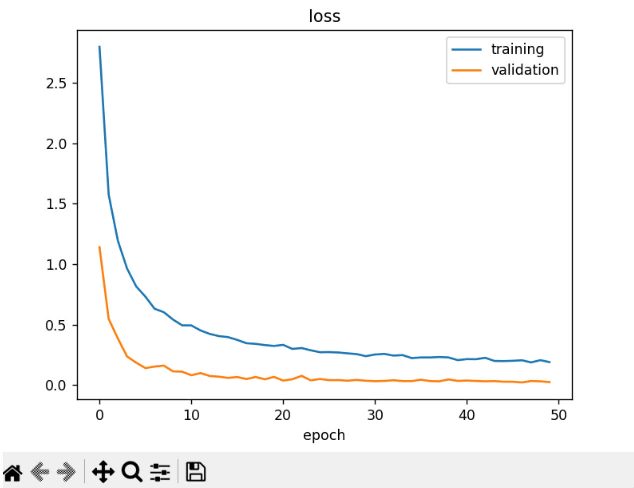

Fig 9: Loss Graph

```
Test Score: 0.02950345352292061
Test Accuracy: 0.9925287365913391
```

Fig 10: Test Accuracy

vi) Testing in Real-time using a camera

In this proposed method, the model is being tested in real time with a camera or an external live feed. The images of traffic signs provided to the camera are entirely fresh and have not been produced from training or testing data, which means that for testing the model new images from each class will be used. If the probability value of image prediction exceeds the threshold, the class label will be displayed with the likelihood. Every traffic sign in each of the 43 categories is tested in real time. Some have probability values of greater than 90%, while others have scores that vary. Here the probability score differs when, for example, the image of the traffic sign 'Speed limit (20 km/h)' is displayed in front of the camera.

6. RESULTS

We tested the model with 43 different traffic sign images, one for each class. As we can see from the resulting images shown below, that our system is accurately recognizing and detecting the images that are displayed in front of the webcam. The probability of accurate recognition and detection of traffic sign images from each class varies from 95% to 100%.

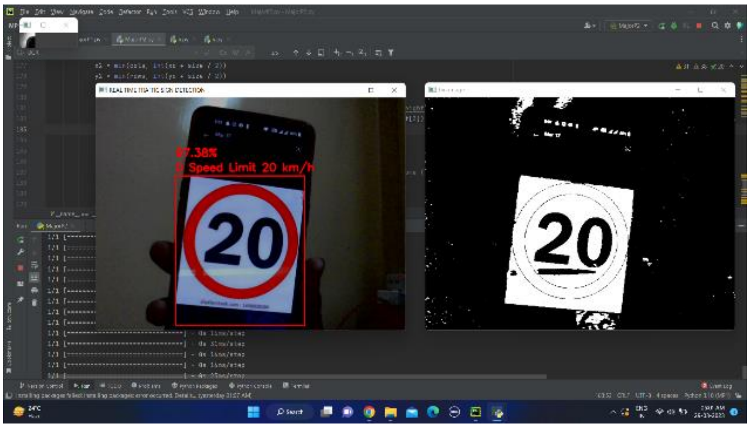

Fig 11: Speed Limit (20 km/h) detected using webcam in real time

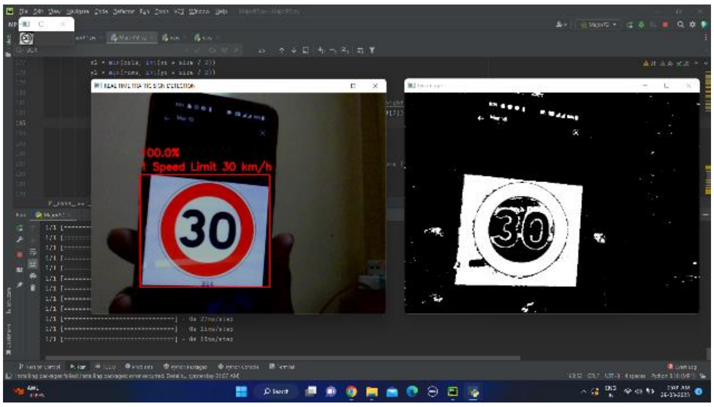

Fig 12: Speed Limit (30 km/h) detected using webcam in real time

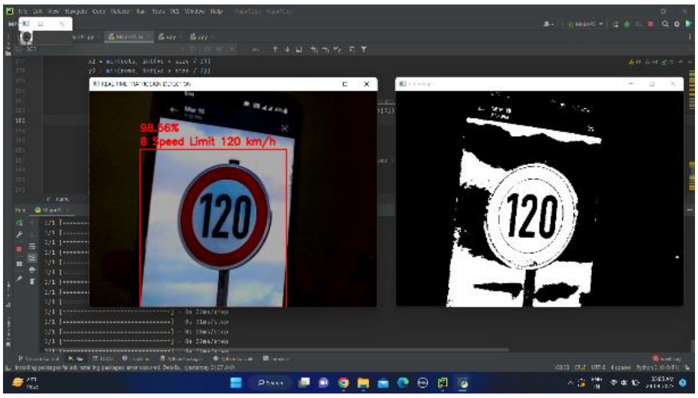

Fig 13: Speed Limit (120 km/h) detected using webcam in real time

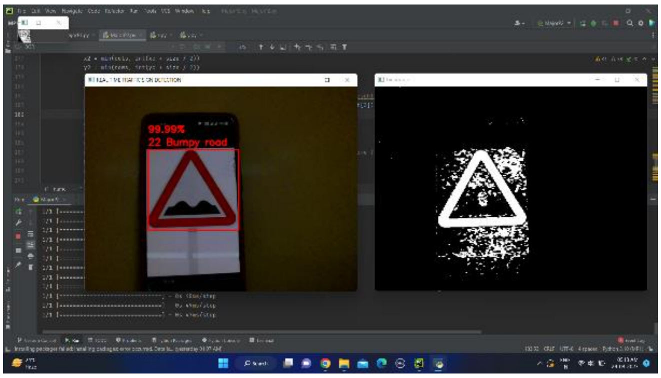

Fig 14: Bumpy road detected using webcam in real time

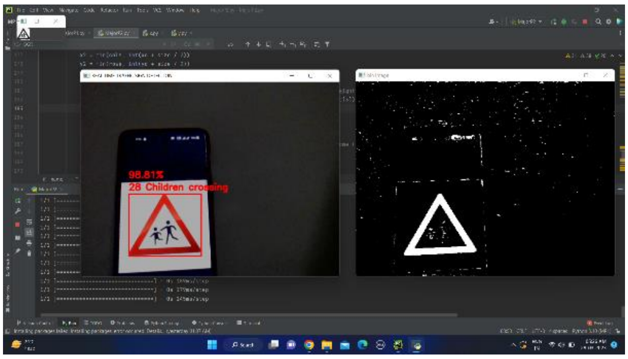

Fig 15: Children crossing detected using webcam in real time

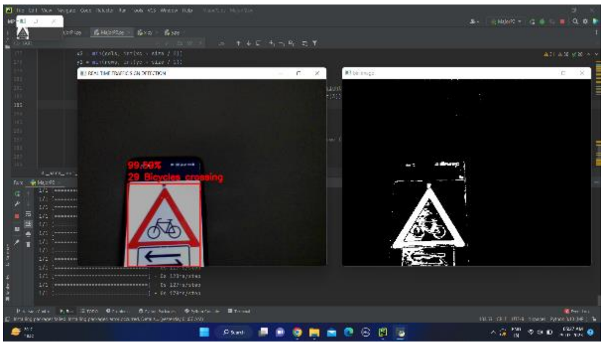

Fig 16: Bicycles crossing detected using webcam in real time

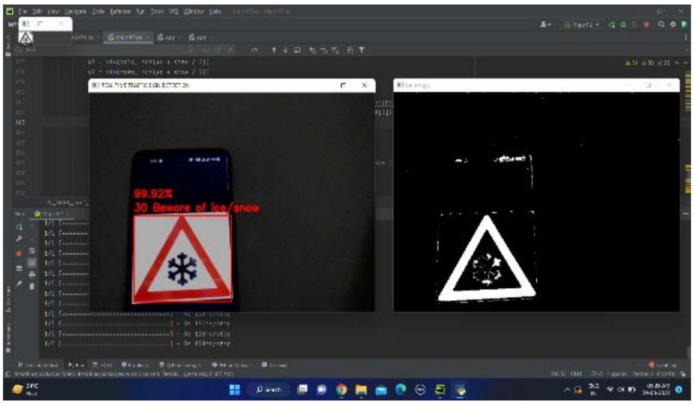

Fig 17: Beware of ice/snow detected using webcam in real time

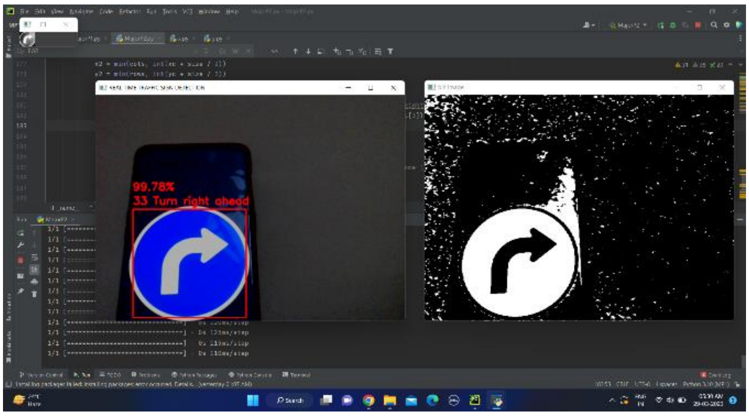

Fig 18: Turn right ahead detected using webcam in real time

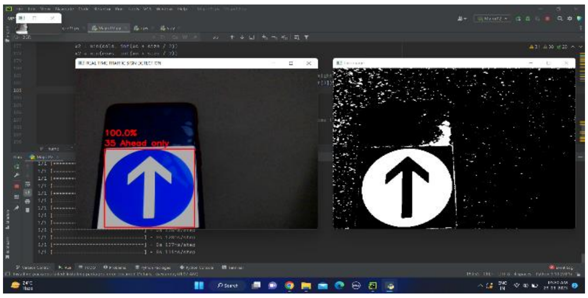

Fig 19: Ahead only detected using webcam in real time

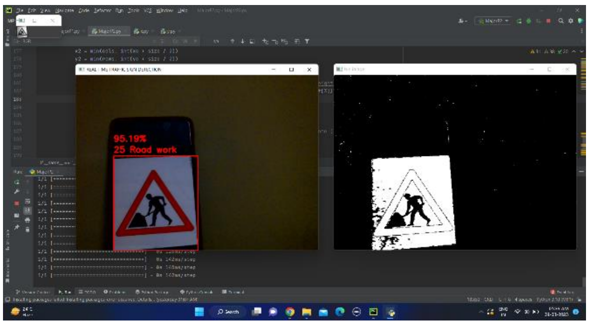

Fig 20: Road work detected using webcam in real time

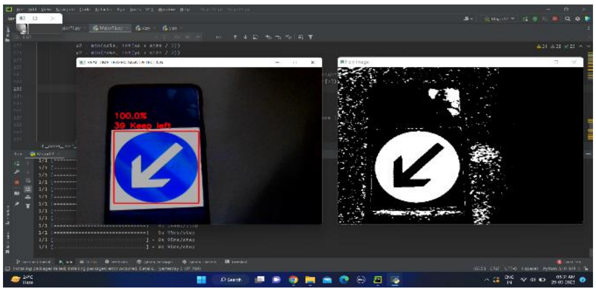

Fig 21: Keep right detected using webcam in real time

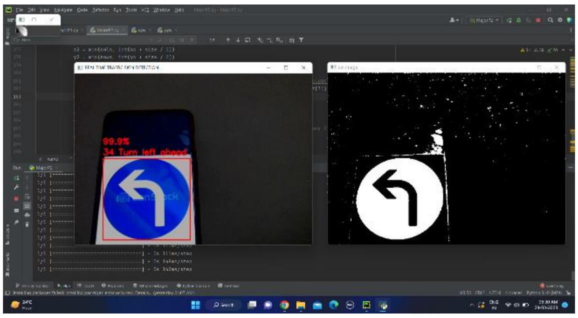

Fig 22: Turn left ahead detected using webcam in real time

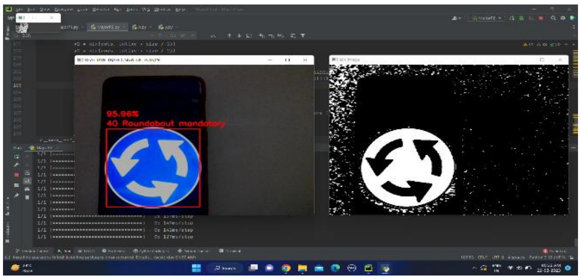

Fig 23: Roundabout mandatory detected using webcam in real time

7.CONCLUSION

A collection of 43 traffic signs can be detected and recognized by our proposed system in real-time using a webcam. We successfully used a CNN to the Traffic Sign Detection and Recognition for Driverless Vehicles challenge, achieving an average accuracy of more than 95%. We've discussed how deep learning may be used to accurately categorise traffic signs using a number of pre-processing and visualisation approaches. We created a basic, easy-to-understand CNN model to properly recognise road traffic signs in real-time. Our proposed system achieves roughly 94% training accuracy on the GTSRB dataset and while testing the images in real-time using a webcam it can obtain a high detection rate of 95-100%. With the aid of deep learning, our proposed system effectively attained the required output and fulfilled all of the vision for which the project was developed.

REFERENCES

[1] Akatsuka, H., & Imai, S. (1987). Road signposts recognition system (No. 870239). SAE Technical Paper.

[2] Richa Jain and Deepa Gianchandani, “A Hybrid Approach for Detection and Recognition of Traffic Text Sign using MSER and OCR”, 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud).

[3] Pranjali Pandey and Ramesh Kulkarni, “Traffic Sign Detection Using Template Matching Technique”, 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA).

[4] Md. Abdul Alim Sheikh, Alok Kole and Tanmoy Maity, “Traffic sign detection and classification using colour feature and neural network”, 2016 International Conference on Intelligent Control Power and Instrumentation (ICICPI).

[5] A. Gudigar, S. Chokkadi, and R. U, “A review on automatic detection and recognition of traffic sign,” Multimedia Tools Appl., vol. 71, pp. 1363– 1380, Oct. 2014.

[6] Hee Seok Lee and Kang Kim, “Simultaneous Traffic Sign Detection and Boundary Estimation Using Convolutional Neural Network”, IEEE transactions on intelligent transportation systems, no. 8, pp. 1524-9050, 2018.

[7] Y. Yang, H. Luo, H. Xu, and F. Wu, “Towards real-time traffic sign detection and classification,” IEEE Trans. Intell. Transp. Syst., Vol. 17, No. 7, pp. 2022–2031, Jul. 2016.

[8] X. Yuan, X. Hao, H. Chen, and X. Wei, “Robust traffic sign recognition based on color global and local oriented edge magnitude patterns,” IEEE Trans. Intell. Transp. Syst., Vol. 15, No. 4, pp. 1466–1477, May 2014.

[9] A. Ellahyani, I. E. Jaafari, and S. Charfi, “Traffic Sign Detection for Intelligent Transportation Systems: A Survey,” E3S Web of Conferences, vol. 229, p. 01006, 2021.

[10] Rubel Biswas, Hasan Fleyeh and MoinMostakim, “Detection and classification of speed limit traffic signs”, 2014 World Congress on Computer Applications and Information Systems (WCCAIS).

[11] Traffic-Sign Detection and Classification Under Challenging Condition, Uday Kamal 2018.

[12] Automatic Traffic Sign Detection and Recognition using CNN Jisha Elizabeth Shaji1, Hari S 2019.

[13] Kaoutar Sefrioui Boujemaa, Afaf Bouhoute, Karim Boubouh and Ismail Berrada, “Traffic sign recognition using convolutional neural networks” in International Conference on Wireless Networks and Mobile Communications (WINCOM) 2017 IEEE.

[14] Faming Shao, Xinqing Wang,* Fanjie Meng, Ting Rui, Dong Wang, and Jian TangReal-Time Traffic Sign Detection and Recognition Method Based on Simplified Gabor Wavelets and CNNs

[15] Amal Bouti, Mohamed Adnane Mahraz, Jamal Riffi, Hamid Tairi, “Road sign recognition with Convolutional Neural Network” in International Conference on Intelligent Systems and Computer Vision (ISCV). 2018 IEEE.

[16] Hung Ngoc Do, Minh-Thanh Vo, Huy Quoc Luong, An Hoang Nguyen, Kien Trang, and Ly T.K, “Speed Limit Traffic Sign Detection and Recognition Based on Support Vector Machines” in 2017 International Conference on Advanced Technologies for Communications.

[17] Yann Lecun, Leon Bottou, Yoshua Bengio, Patrick Haffner, “Gradient-Based Learning Applied to Document Recognition” in PROC, Of the IEEE, November 1998.

[18] A. Ellahyani, I. E. Jaafari, and S. Charfi, ‚Traffic Sign Detection for Intelligent Transportation Systems: A Survey,‛ E3S Web of Conferences, vol. 229, p. 01006, 2021.

[19] U. Kamal, S. Das, A. Abrar, & M. K. Hasan, ‚Traffic-sign detection and classification under challenging conditions: a deep neural network-based approach‛, IEEE video and image processing cup , 2017.

[20] Boujemaa, K.S., Berrada, I., Bouhoute, A., & Boubouh, K. (2017). Traffic Sign Recognition Using Convolutional Neural Networks. In 2017 International Conference on Wireless Networks and Mobile Communications (WINCOM), pp. 1-6.

[21] Dhabliya, D. (2021). Delay-tolerant sensor network (DTN) implementation in cloud computing. Paper presented at the Journal of Physics: Conference Series, , 1979(1) doi:10.1088/1742-6596/1979/1/012031 Retrieved from www.scopus.com

[22] Dhabliya, D. (2019). Security analysis of password schemes using virtual environment. International Journal of Advanced Science and Technology, 28(20), 1334-1339. Retrieved from www.scopus.com