ISSN (e): 2250 – 3005 || Volume, 11 || Issue, 4|| April – 2021 ||
International Journal of Computational Engineering Research (IJCER)

Traffic Sign Detection and Recognition (TSDR)

Kartik Jain[*1], Mayank Sinha[*1], Eraa[*1], Mayank Vats[*1]
Mr. Pramod Sethy [*2]
[*1]Student, Krishna engineering college, Ghaziabad, India
[*2]Assistant professor, Krishna engineering college, Ghaziabad, India

ABSTRACT

The world is evolving every day, as the people are continuously working to make things simpler and simpler by automating them and one of such task is the Advance Driver Assistance System(ADAS). The application of ADAS is 'Traffic Sign Detection And Recognition" (TSDR). TSDR is the system in which traffic signs are automatically detected and recognized. It plays a crucial role for the one who is driving the vehicle. As the driver needs to stay focused on the road while driving, the drivers might miss some of the road signs which can be dangerous for the driver of the vehicle as well as for other drivers. The TSDR will reduce this risk by automatically detecting the road sign using Computer Vision and machine learning algorithms such as Convolutional Neural Network (CNN) as well as recognizing them. This whole process will reduce the human efforts and the machine will accurately detect the sign without any human error.

Keywords: *Computer Vision, Image Processing, CNN, Tensorflow, Traffic Sign Detection, Traffic Sign Recognition, Advance Driver Assistance System*

Date of Submission: 28-04-2021 Date of acceptance: 12-05-2021

I. INTRODUCTION

Traffic Signs are the road facilities provided to warn, inform, guide or restrict the driver from getting into any kind of accident. But keeping an eye on the traffic signs is not the only task of the driver, they need to focus on the road to prevent accident from other vehicles, keeping balance of their own vehicle and while carrying out such task it may happen that the driver might miss the traffic sign, or may be if he sees the traffic sign but doesn't understand what this sign indicate, which might be dangerous for the everyone on the road.

So, for problems like this Advance Driver Assistance Systems comes into the play. Its application TSDR, can prevent many accidents by detecting the traffic sign by capturing the images from the cameras and informing the driver about the same. This will not only minimize the accident-rate over the road but also allows the drivers to drive with ease as they no more need to check for traffic Sign.

ADAS will become the future of automobiles, as the advancement in automobile technology in the industry is increasing what cars can do.

II. METHODOLOGY

The dataset we have used for the project to train our Traffic Sign Classifier is taken from the kaggle dataset(German Traffic Sign Recognition Benchmark (GTSRB)). This dataset consists of 43 different traffic sign classes and 39,209 images.

Data Size & Shape

- Size of training set: 31,367(60%)
- Size of validation set: 7842 (15%)
- Size of test set: 12,631 (25%)
- Shape of a image: (30, 30, 3)
- Number of unique classes/labels: 43

The proposed System here, works in 3 phases:-

- Image Pre-processing
- Traffic Sign Detection
- Traffic Sign Recognition

Image Pre-processing :-

This phase plays a crucial role in our TSDR system. It is used to remove the background noise from the image and equalize the intensity of light. Moreover, it separates the RGB image into 3 different channels and converts it into an HSV (Hue Saturation Value) color space. Although instead of HSV other color space like YCrCb can also be used, we have used HSV color space here.

At first the input RGB image is separated into 3 different channels and filters are applied on each threshold to convert the RGB color space into HSV color space. This conversion is necessary because the RGB color space describes colors in terms of the amount of red, green, and blue color present whereas HSV color space describes colors similarly to how the human eye tends to perceive color.

After the color space conversion, the light intensity is taken care of. CLAHE (Contrast Limited Adaptive Histogram Equalization) algorithm is used here for over-amplification of the contrast. CLAHE operates on small regions in the image, called tiles, rather than the entire image. The neighbouring tiles are then combined using bilinear interpolation to remove the artificial boundaries.

Traffic Sign Detection :-

Once the pre-processing of the image is done, then comes the part for detecting the traffic signal from the captured image. This detection process is further divided in 2 different phases: **Color Based Detection** and **Shape Based Detection.** Since all objects that are red in color can't be a traffic sign, in order to attain more true positive results we use shape and area for verification of a traffic sign.

1) Color Based Detection

The most important feature of a traffic sign is its color. Whenever we see a red board on the road side we suspect that it could be a traffic sign. So, our detection system works around the same logic. In our proposed algorithm the captured image is processed for the red colour. We apply filters on each channel threshold to extract the part of the image we suspect that can be a traffic sign, following which the contours of the extracted image is found. The threshold of the channel R is in the range of 90-255 and that for channel G and channel B is in the range of 0-70.

Fig 1: (a) Original Image (b) Color Detection (c) With Contour

Fig 2: (a) Original Image (b) Color Detection (c) With Contour

2) Shape Based Detection

The contours that we found in the previous steps will help us further in the detection phase. The contours with a much smaller area are filled up for noise reduction and to better deal with the ROI(Region Of Interest}. The contours with much higher area are not considered for Traffic Sign. Once the area of the contour satisfies the minimum and maximum condition, we pass the image into the SVM.

We use SVM (Support Vector Machine) to classify different shapes of the extracted part of the image. Once the shapes are found, to circular, triangular or octagonal, we can be sure that the ROI contains the traffic sign and we can continue further for the recognition part.

Recognition

Once the sign is detected, we proceed to the recognition part, where we classify the image into different categories. For this part we have used the neural network algorithm. With the help of machine learning frameworks such as keras and TensorFlow, a CNN(convolutional neural network) model is built for the classifications.

The dataset used here does not have uniform distribution. This is the real case scenario because there are certain signs that appear more from others, but it is generally good to have normal distribution so that the model gets equal opportunity to learn every sign.

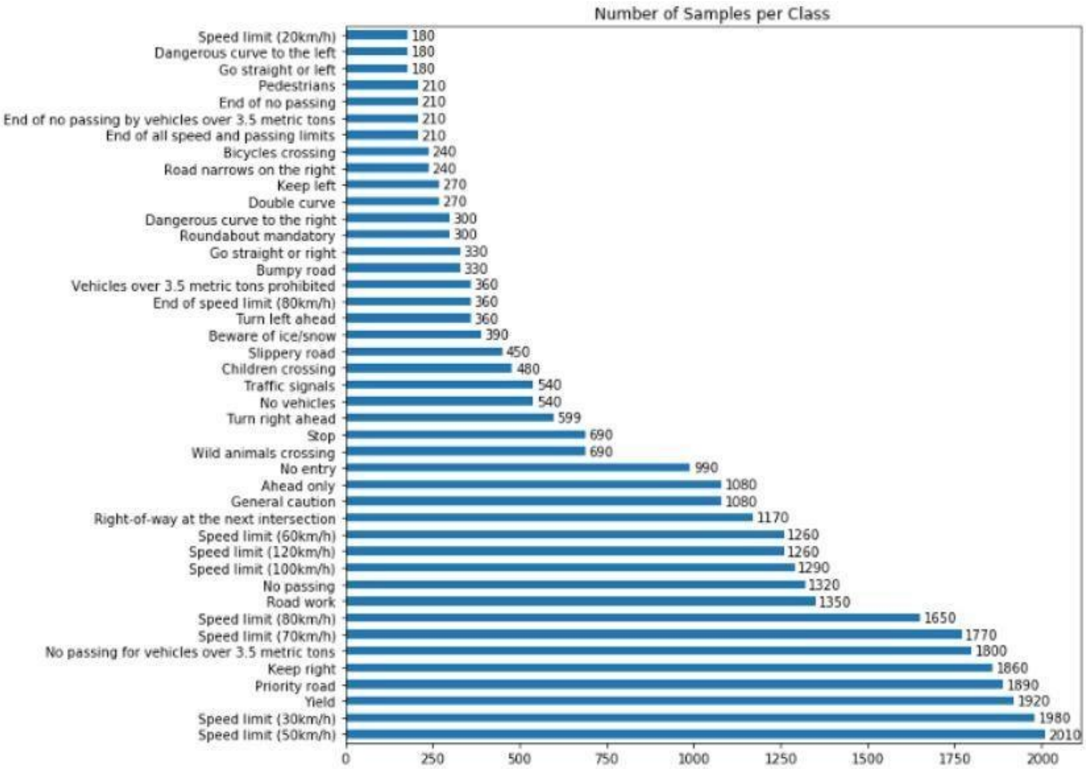

Fig 3: Image Distribution

In Neural Network Algorithm, a model is made and is trained on lots of training images. In particular, we have made a convolutional neural network (CNN) and used 60% of the data in training the model.

The Convolutional Neural Network (CNN) is a multi-layered feed-forward neural network, which is made by assembling hidden layers on top of each other in a definite order. A CNN can have multiple layers, adding more and more layers to the CNN, makes the model complex. The 1st layer is called the Input layer and the last layer is known as the output layer. All the layers between them are called the hidden layers. In CNN, some of them followed by grouping layers and hidden layers are typically convolutional layers followed by activation layers. We have also used dropout layers to prevent over-fitting.

Algorithms used :-

1. **Loss:** categorical cross-entropy
2. **Optimization:** Adam
3. **Activation Function:**
    1. **ReLU:** Rectified Linear Unit => f(x) = max (0,x)
    2. **SoftMax:** The outputs of the Softmax transform are always in the range [0,1] and add up to 1. Hence, they form a **probability distribution.**

Epochs: 15
Batch size: 32
Learning rate: 0.004

Model Details

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 26, 26, 32)        2432
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 22, 22, 32)        25632
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 11, 11, 32)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 11, 11, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 9, 9, 64)          18496
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 7, 7, 64)          36928
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 3, 3, 64)          0
_________________________________________________________________
dropout_2 (Dropout)          (None, 3, 3, 64)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 576)               0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               147712
_________________________________________________________________
dropout_3 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 43)                11051
=================================================================
Total params: 242,251
Trainable params: 242,251
Non-trainable params: 0
```

Fig 4: Model Summary

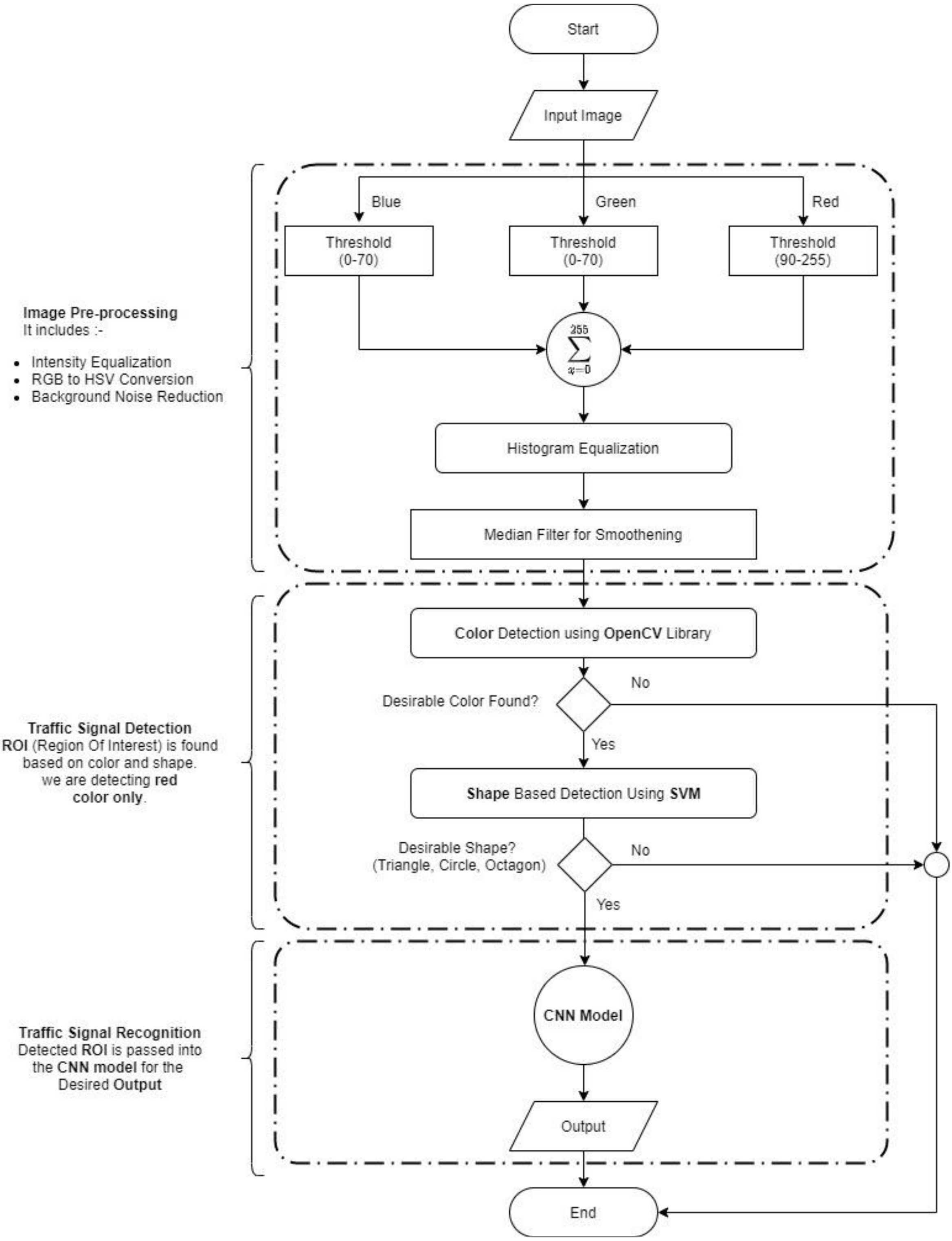

Fig 5: The overall block diagram of the proposed system.

III. RESULTS AND DISCUSSION

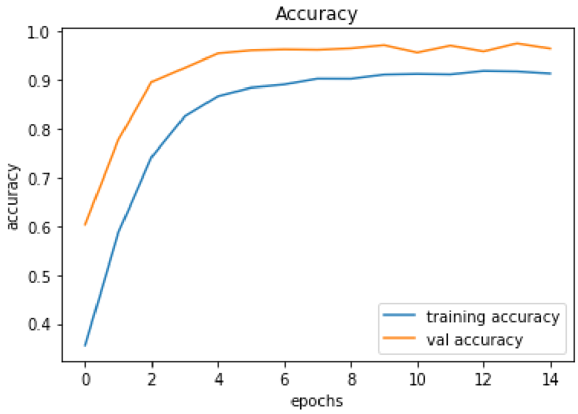

Fig 6: Training Accuracy

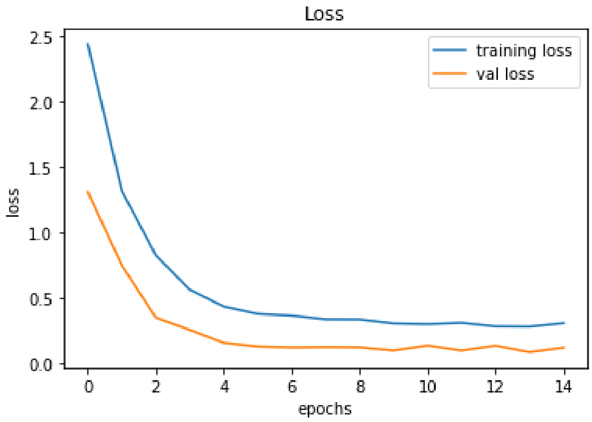

Fig 7: Training Loss

Upon training our CNN model, we achieved an accuracy of 91.24% and 92.61% when tested.

REFERENCES

[1]. Traffic Sign Detection and Recognition using a CNN Ensemble Aashrith Vennelakanti, Smriti Shreya, Resmi Rajendran, Debasis Sarkar, Deepak Muddegowda, Phanish Hanagal Affiliation: Qualcomm India Private Limited, Bangalore

[2]. An Automatic Traffic Sign Detection and Recognition System Based on Colour Segmentation, Shape Matching, and SVM Safat B. Wali, Mahammad A. Hannan, Aini Hussain, and Salina A. Samad Department of Electrical, Electronic & Systems Engineering, Universiti Kebangsaan Malaysia, Jalan Reko, 43600 Bangi, Selangor, Malaysia.

Kartik Jain, et. al. "Traffic Sign Detection and Recognition (TSDR)." *International Journal of Computational Engineering Research (IJCER)*, vol. 11, no.4, 2021, pp 30-35.