Lane Detection and Traffic Sign Recognition using OpenCV and Deep Learning for Autonomous Vehicles

Birali Prasanthi[1], Kyathi Rao Kantheti[2]

[1] *Student, Bachelors in CSE, Mahatma Gandhi Institute of Technology, Hyderabad, Telangana, India*
[2] *Senior Assistant Professor, Dept. of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Hyderabad, Telangana, India*

***

Abstract –*By the means of automation, number of car crashes on road can be reduced. Testing of autonomous vehicles on public roads can be done on the public roads of the US. Major benefits of automated vehicles include a 90% reduction in traffic deaths, a 60% reduction in harmful emissions, a 40% reduction in travel time, and a 500% increase in lane capacity. Autonomous vehicles are expected to be safer. "Over 90% of accidents today are caused by driver error," said Professor Robert W.Peterson. Autonomous cars are designed for the elimination of traffic created by stop-and-go behavior, according to research done at the University of Illinois. This will be helpful in saving the time of people and as well decreases the time their cars are on the roads, which will reduce the emission of harmful gases from the vehicles. A very close part of driver assistant systems is lane detection. Lane detection refers to the process of tracing white markings on the road, capturing and processing images using a camera mounted in front of the car, and this is done using the OpenCV library. Safety driving also involves recognition of traffic signs as a major part. Promising results have been presented by the CNN (convolutional neural networks).*

Key Words: **CNN, Lane detection, Traffic sign recognition, OpenCV, Autonomous Vehicles.**

1. INTRODUCTION

The white markings on the road parallel to its direction, usually two in number are spoken as lanes. one of the major elements for a vision-based driver assistance system is lane detection and its tracking. The previous method was accustomed to detecting the obstacle on the road and Therefore the distance of the obstacle with regard to vehicles is still as lanes structured road. In an advanced driver assistance system (ADAS), recognition of traffic signs is incredibly important for safe driving. Promising results have been presented by convolutional neural networks (CNNs) recently. During this work, a strong model supported VGG network by adding batch normalization operation is proposed. To reduce the overfitting of the model, dropout is also used. With the help of the dataset imbalance data augmentation is performed. Then, to boost the images, Contrast limited adaptive histogram equalization (CLAHE) and normalization are performed. The performance of the model is evaluated using various performance metrics such as confusion matrix, precision, recall on German traffic sign recognition benchmark (GTSRB) dataset. According to the Experiments results, the proposed model reaches a state-of-art accuracy of 99.33 % and surpasses the best human performance of 98.84 %

1.1 LITERATURE SURVEY FOR LANE DETECTION

During the literature review, it was discovered that the majority of the existing literature has neglected one or more of the following:-

1) According to the survey, the present methods provide good precision for high-quality photographs, although can be a little sloppy at times. Bad outcomes due to poor environmental circumstances such as fog, haze, smog, Noise, dust, and so on

2) The majority of present approaches are better suited to straight lanes. However, they perform poorly on curved roadways.

3) The majority of lane detection systems are based on industry standards. Hough transform can be tweaked to improve the results even more accurate.

1.2 LITERATURE SURVEY FOR TRAFFIC SIGN RECOGNITION

In 1987, Akatsuka and Imai conducted the first traffic sign recognition research, attempting to create a very basic traffic sign recognition system.

A system capable of recognizing traffic signs on its own and providing aid to drivers by informing them of the presence of a certain restriction or danger, such as speeding or construction work. It may be used to detect and recognize specific traffic signs automatically. A traffic sign recognition system's operation is often separated into two parts:

- Detection and
- Classification.

DETECTION:

To find the areas of interest (ROI) where a traffic sign is most likely to be present is the major target of traffic sign detection. To acquire the main sign, you'll need to crop the extra space. ROI stands for Return on Investment. ROI determines the location of the traffic sign based on its shape, dimensions, and other factors. The traffic signs have been clipped, which is advantageous. The background image has been deleted because it isn't relevant. As a result, we assume that a considerable portion of the image can be dismissed as unnecessary. The color and shape of traffic signs are predetermined, making them easier to distinguish and recognize.

CLASSIFICATION:

The sign is classed in this section depending on the kind, shape, and color of the sign, as well as the information it provides.

Features of a histogram of oriented gradients (HOG):

An image is broken down into small cells of squared shape with the help of HOG, calculates oriented gradients in each cell, standardizes the output with a pattern which is block-wise, and for each cell a descriptor is returned.

2. METHODOLOGY

2.1 ARCHITECTURE OF LANE DETECTION:

I. FRAME MASK:

A NumPy array represents a frame mask. Changing the pixel values of the targeted region in an image to 0, 255, or any other number satisfies the need for applying a mask to that image. An example of picture masking is shown below:

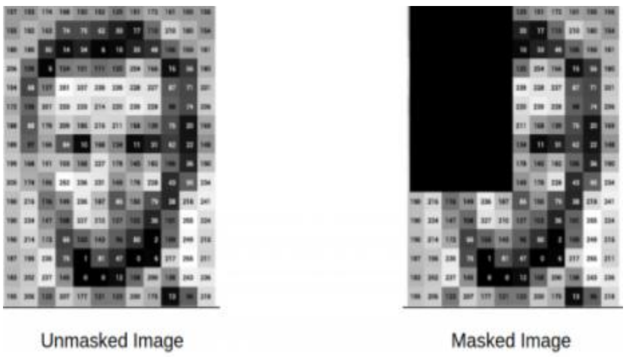

Fig -1: Frame Mask

II. IMAGE PRE-PROCESSING:

To begin, we'll apply a mask to all of the frames in the received input video. Then, to recognize lane markers, we'll use picture thresholding followed by Hough Line transformation.

III. IMAGE THRESHOLDING:

Based on the threshold setting, the grayscale images with pixel values are assigned either black or white values. If a pixel's pixel value is larger than a threshold value, it will be allocated value one; otherwise, value 2 will be assigned.

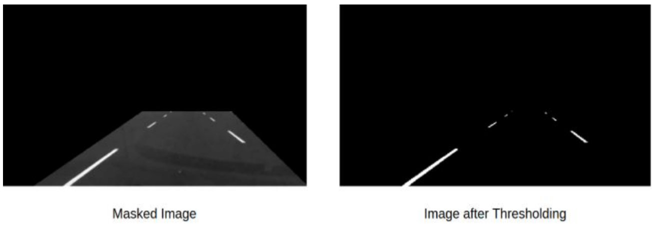

Fig -2: Image Threshold

IV. HOUGH LINE TRANSFORMATION:

A method for detecting any geometry that can be mathematically described is the Hough Transform. It can recognize shapes like rectangles, circles, triangles, lines, and many others. The key goal is to locate the lane marking. After conducting image thresholding, the Hough line transformation is applied to the image, and the result is shown below.

Fig -3: Hough Line Transform

2.2 ARCHITECTURE OF TRAFFIC SIGN RECOGNITION:

I. IMAGE AND DISTRIBUTION:

Selections of photos from the dataset are shown below, with labels provided above the row of associated photographs. We'll seek to boost contrast later because some of them are extremely black. In the training set, there is also a large imbalance among classes, as illustrated in the histogram below. Some courses have less than 200 photos, while others have over 2000. This suggests that our model may favor over-represented groups, especially when its projections are uncertain. We'll look at how data augmentation can help us deal with this disparity later.

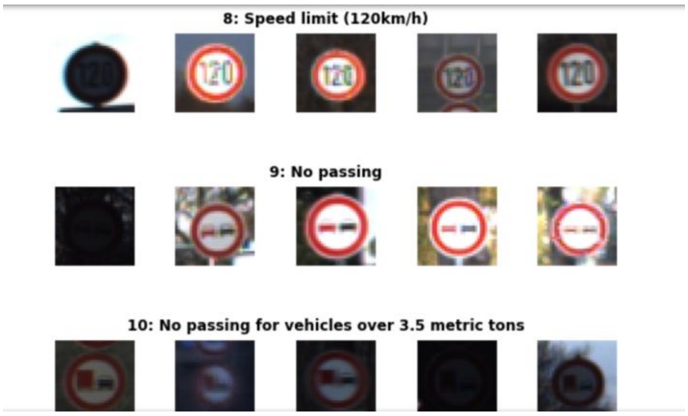

Fig -4: Sample of Training Set Images with Labels Above

II. PRE-PROCESSING STEPS:

GRAYSCALE AND IMAGE NORMALIZATION:

We convert the images to grayscale just like lane detection and by subtracting each image from the dataset mean and dividing by its standard deviation, we can center the distribution of the image collection. This assists our model in treating photos consistently. The images that result are as follows:

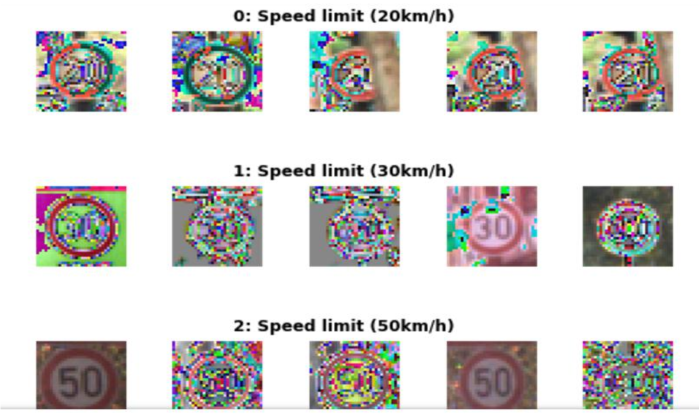

Fig -5: Normalized Images

III. MODEL ARCHITECTURE:

Yann Le Cun's research on traffic sign classification inspired the proposed architecture. We made a few adjustments and built a modular framework that allows us to experiment with varied filter widths, depth, and number of convolution layers, as well as completely linked layer dimensions. We named such a network EdLeNet in honour of Le Cun. For our first convolutional layer, we used 3x3 and 5x5 filter sizes, and a depth of 32 was the initial start. The 3x3 architecture of EdLeNet is shown below:

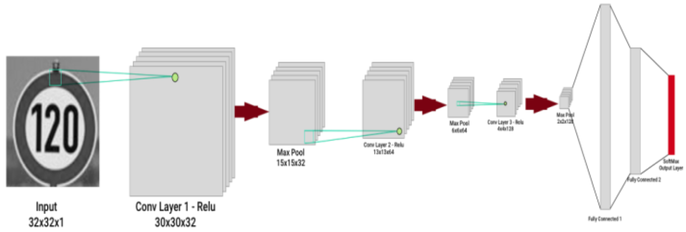

Fig -6: EdLeNet 3x3 Architecture

With the help of three convolutional layers a network is made. We used ReLU as the activation function, each followed by a 2x2 max pooling operation. The kernel size is 3x3, with depth doubling at the successive layer. There are three final layers which are fully connected. When the SoftMax activation function is used, the final layer yields 43 results. The network is trained using the Adam optimizer and mini-batch stochastic gradient descent.

3. CONCLUSION

The OpenCV library was used in the methodology. Then we created a mask with no intensity and used the bitwise technique to map our region of interest. Then we utilized the Hough Transform technique to identify the lane lines and detect the straight lines in the image.

For traffic sign recognition we looked at how deep learning may be used to accurately recognize traffic signs using a number of pre-processing and regularization approaches, as well as a variety of model architectures. On the test set, our model was close to 98 percent accuracy, and on the validation set, it was 99 percent accurate.

REFERENCES

[1] Alexander Shustanov and Pavel Yakimov," CNN Design for Real-Time Traffic Sign Recognition", vol. 201, April. 2017, doi: 10.1016/j.proeng.2017.09.594.

[2] Aditya Singh Rathore, "Lane Detection for Autonomous Vehicles", Jan 2019.

[3] Sanam Narejo, Shahnawaz Talpur, Madeha Memon, Amna Rahoo, "An Automated System for Traffic Sign Recognition using Convolutional Neural", November 2020.

[4] Dhruv Pandey, "Lane Detection for Self-Driving Car using OpenCV," unpublished.