TRAFFIC SIGN CLASSIFICATION PROJECT REPORT

Name: Dishant Ahuja

Introduction:

Nowadays, there is a lot of attention being given to the ability of the car to drive itself. One of the many important aspects for a self driving car is the ability for it to detect traffic signs in order to provide safety and security for the people not only inside the car but also outside of it.

The traffic environment consists of different aspects whose main purpose is to regulate flow of traffic, make sure each driver is adhering to the rules so as to provide a safe and secure environment to all the parties concerned.

We have focused our project on the US traffic signs and a few of the traffic signs which we have in our dataset is as shown in the figure below. We used the LISA traffic sign dataset [3]. The dataset consisted of 48 different types of US traffic sings. About 75% of the frames were in gray scale and the rest in color.

The problem we are trying to solve has some advantages such as traffic signs being unique thereby resulting in object variations being small and traffic signs are clearly visible to the driver/system [1]. The other side of the coin is that we have to contend with lighting and weather conditions [1].

The main objective of our project is to design and construct a computer based system which can automatically detect the road signs so as to provide assistance to the user or the machine so that they can take appropriate actions. The proposed approach consists of building a model using convolutional neural networks by extracting traffic signs from an image using color information. We have used convolutional neural networks (CNN) to classify the traffic signs and we used color based segmentation to extract/crop signs from images.

Background and related work:

Many different techniques have been applied to detect traffic signs. Most of these techniques are based on using HOG and SIFT features.

In our approach we use biologically inspired convolutional neural networks to build a model which can predict the type of traffic sign. One such related work based on convolutional neural networks is published in 'Traffic Sign Recognition with Multi-Scale Convolutional Networks' by Pierre Sermanet and Yann LeCun[4].

Methods:

The problem of traffic sign recognition is twofold:

1) <u>Extracting a potential traffic sign from an image.</u>
   Traffic signs are designed such that they appear unique and easily identifiable to the human eye. Traffic signs in the United States of America are of 3 main colors: Red, White, and Yellow. Other colors like orange and blue are also used. In our approach we concentrate on Red, White, and Yellow traffic signs.

   Since the color of a traffic sign is unique in a background we can use the color information to narrow down our areas of interest (parts potentially containing the traffic sign).

   Since RGB colored images are susceptible to variations in lighting, we use HSV (Hue, Saturation, and Variation) images.

   Once we have the HSV image our next goal is to define our areas of interest (i.e. range of Yellow, Red and White) so that we can segment our HSV image based on these 3 colors. The color ranges used are as follows:

Color	Lower Range (HSV)	Upper Range (HSV)
Yellow	([10,50,50])	([30,255,255])
Red	([170,50,50])	([185,255,255])
White	([0,0,50])	([120,15,255])

   The next step is to use these color ranges and create binary masks for each of the 3 colors. For Example, the red binary mask will have 0 assigned to all the regions which are not in the red range and 1 assigned to all regions which are in the red range. The Red, Yellow and white binary masks for an image are shown below:

As seen from the above example the the original image is segmented based on colors.

We know that traffic signs are usually occur in different closed shaped like rectangles, triangles, diamonds etc. We can use this property to extract closed shaped from each of the 3 binary masks. This can be done by using 'Topological Structural Analysis of Digitized Binary Images by Border' [5]. We used the OpenCV implementation of tis algorithm [6].

The extracted contours from the binary masks are as follows:

As we can see from these images we have narrowed down the areas of interest from the entire image. These areas of interest are further refined based on the size of the contour to reduce the areas of interest.

Once we have refined the set of areas of interest, we use the convolutional neural network which we are going to build in the next step to predict the type of this sign (or if it is not a sign).

2) <u>Predicting the type of Extracted traffic sign.</u>
From the extracted areas of interests in the previous step we want to determine if it is a sign or not and if it is a sign we wish to know what the type of sign it actually is.

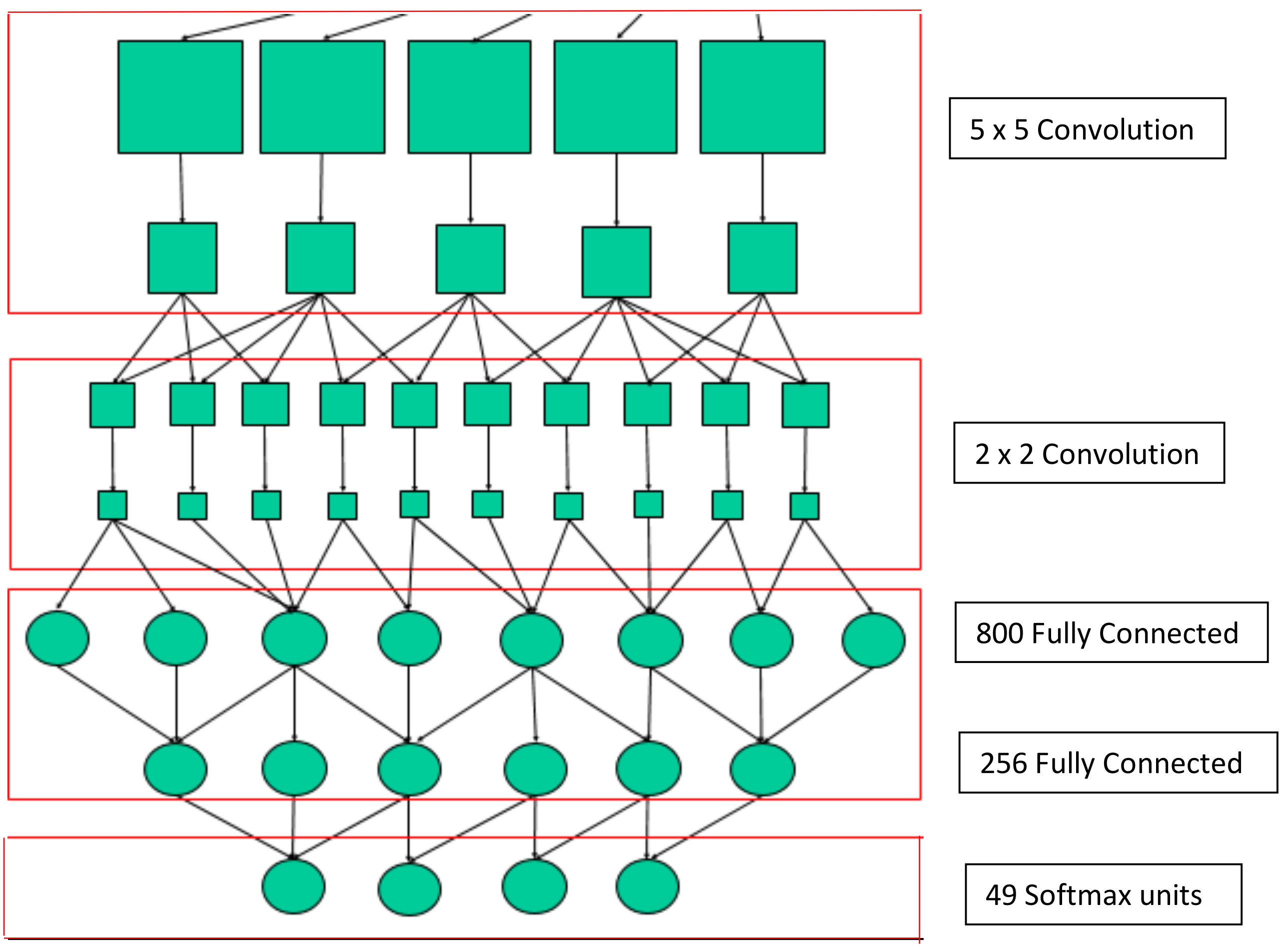
For this purpose, we can train a convolutional neural network. The data used to train and test the CNN was obtained from **http://cvrr.ucsd.edu/LISA/lisa-traffic-sign-dataset.html**. It had about 6000 frames and 49 different types of traffic signs. For each frame, the coordinate positions for the traffic sign in the image was given. From these positions the traffic signs were cropped out to use for training the CNN.

A CNN is basically inspired by the connections between the neurons in the visual cortex of animals. [7]Since traffic signs have unique shapes inside them like arrows, words, circles and so on. It is useful to convert the traffic sign into a more useful form by using a Laplacian operation on the traffic sign. We can apply the Laplacian operation by convolving the following kernel on the input image:

0	-1	0
-1	4	-1
0	-1	0

Consider the following traffic sign and its Laplacian:

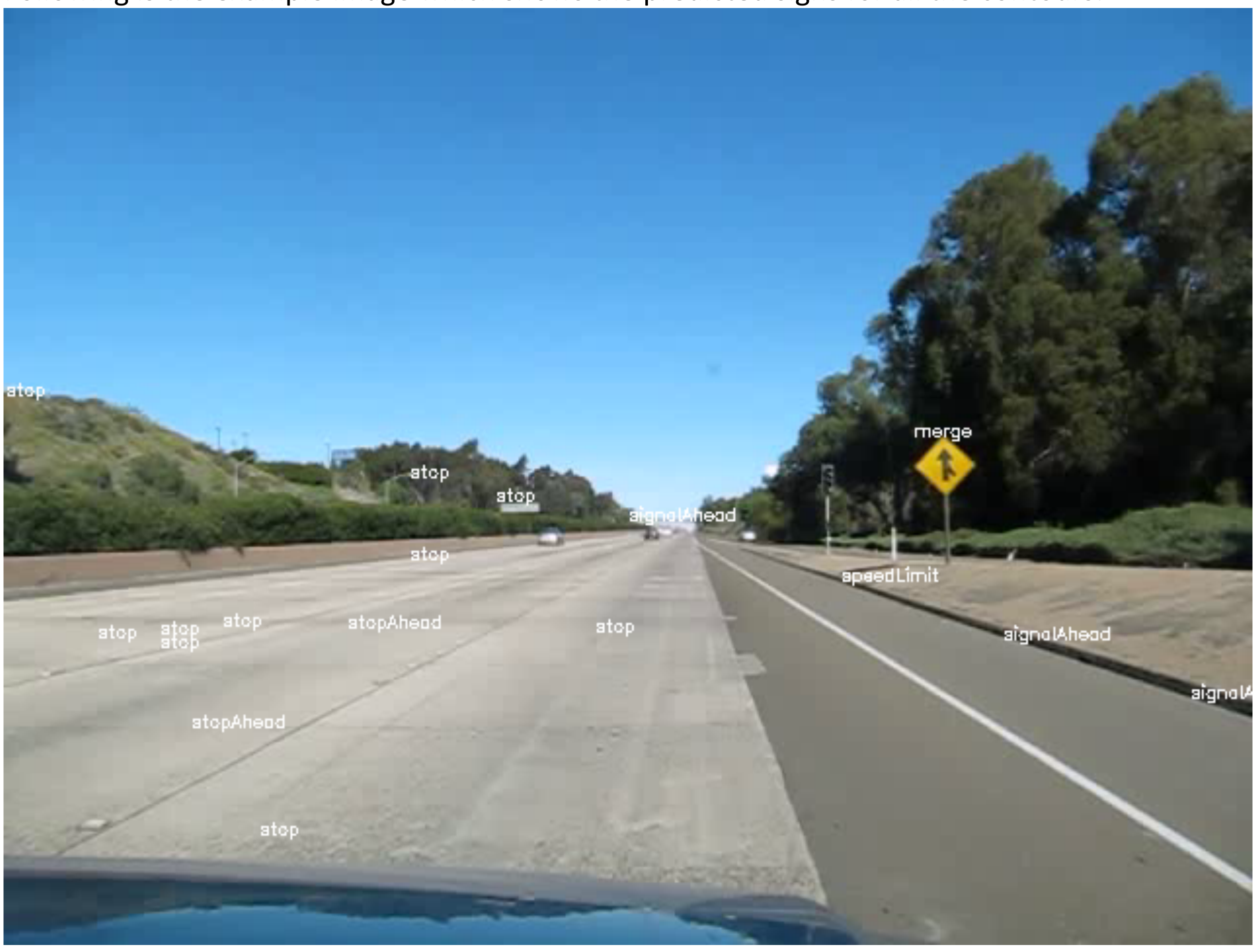
The Laplacian is now fed into the CNN whose architecture in shown below:



The learning rate used to train the CNN was 0.001 and the momentum used was 0.9**.** The CNN was trained for 200 iterations (magic numbers).

Once the CNN has been trained, it is used to predict the sign of the contours obtained in step 1. Each of these contours are assigned the sign with the maximum probability which is the output of the CNN.

Following is the example image which shows the predicted signs for all the contours:



We can also use the trained CNN to get the Accuracy, Precision, Recall and F1 score metrics on the test set. These results are discussed in the next section.

Results:

The following table gives the Accuracy, Precision, Recall and F1 score metrics on the test set. The test set was obtained by splitting the whole dataset into 70% train data and 30 % validation and test data. Out of the 30% validation, 15% was the test data.

The followings results take into consideration the traffic sign that is perfectly cropped from the image. This may not be true when we are extracting traffic signs from the image without the prior knowledge of their position.

Metric	Score
Accuracy	86.9 %
Precision	0.8638
Recall	0.8694
F1 Score	0.8633

Conclusion:

From the following results we can see that the CNN is doing a good job in classifying different types of traffic signs when the extracted signs are cropped perfectly from the image.

Our approach fails to give good results when the extracted signs from test images are cropped incorrectly. Another drawback of our approach is that when the color of the traffic signs vary which may be due to bad weather conditions and poor camera quality, the image masks obtained are not perfect and hence the signs are not detected properly.

Future improvements can be made for extracting signs from test images by using advanced segmentation methods