Rohan
Abhinav
Kamakshi

Project 6

ENPM 673
July 31, 2019

Traffic Sign Recognition

Traffic Sign Recognition can be staged into two sections, Traffic Sign Detection and Traffic Sign Classification.
In the Detection stage,our aim is to extract possible candidates/regions which contain some traffic sign.Whereas in the Classification stage we go over each Region of Interest (RoI) and identify what sign it represents.

Traffic Sign Detection

Traffic Signs can be broadly classified into two categories, RED (Danger Prohibitory) and BLUE (Mandatory) based on color. In this section, the objective is to extract the Region of Interest (i.e., around a traffic sign). This can be accomplished with simple thresholding in an appropriate color space. But, this method is light sensitive. In HSV color space, though it is light insensitive, this method selects many regions which are not of interest (i.e., not a traffic sign). Another method is using MSER regions, a more robust method, to identify regions of similar intensity. This method, though robust, is not very productive when directly applied to the input image. So, images are pre-processed before extracting MSER regions.

In computer vision, maximally stable extremal regions (MSER) are used as a method of blob detection in images. A trivial intuition is that MSER gives regions of similar intensity given a grayscale image. This method of extracting a comprehensive number of corresponding image elements contributes to the wide-baseline matching. Thus, it is a good solution for this project as a traffic sign is mostly a uniform intensity region. But, as stated before, MSER regions does not produce a very effective solution when applied to the input image. So, the first step in the algorithm is Noise Removal. There is no unique way to accomplish this task. For this project, several filters were used before finalizing the algorithm. The next pre-processing step is Normalization. In this step, the image intensity is normalized.

This step is significant as at this location the two color segments are separated. The

Figure 1: Image after Pre-processing

equations used to normalize the image intensity are

$$C_r = max(0, \frac{min(R-B, R-G)}{R+G+B}) \tag{1}$$

$$C_b = max(0, \frac{min(B-R, B-G)}{R+G+B}) \tag{2}$$

Now, the image is ready to extract MSER regions. But, there are several parameters in MSER that need to be tuned for the two sets of signs, red and blue. First parameter is min and max. It is the step size between intensity threshold levels which has a range (0,100].It was kept at the default value of 2. Second parameter is RegionAreaRange. It is the size of the region in pixels. This resulted to be an important parameter as it helped to remove very small and very big regions. Third parameter is MaxAreaVariation. It is the maximum area variation between extremal regions at varying intensity thresholds. This parameter is based on the knowledge that stable regions are very similar in size over varying intensity thresholds. The last parameter is ROI (region of interest) and was the key to eliminate false positives. From the data set, it was observed that the blue signs only occur in the top-half of the image, and the red signs only occur in the top-right-quarter of the image. So, the ROI were set accordingly for the two sets. The outputs from MSER are shown in Figure(2) respectively

Figure 2: a) Blue based Traffic Signal detected and , b) Red based Traffic Signal detected

Traffic Sign Classification

Convolutional Neural Network

This is an initial deep learning model. It is a simple neural network with 4 convolution layers followed by 2 fully connected layers and a softmax output layer. The architecture is shown in fig(3). This architecture did not use any data augmentation while training and also no standardization technique was used. The network reached about 80% training accuracy in about 45 epochs as can be seen in the fig. The important thing to note here is that even though the training accuracy is pretty high, the test accuracy reaches a maximum of 67.33

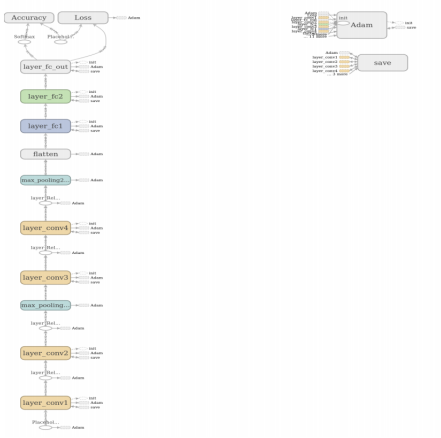

Figure 3: Alex Net - Simple CNN architecture

for the same number of epochs as the train set.

The final result after the completion of the pipeline is shown in Figure (4 and 5)

Figure 4: Recognising both red and blue signs

Figure 5: Recognising both red and blue signs with label number

NOTE

To run the code follow the README file. The videos are available on our drive. Follow this link