International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 04 Issue: 11 | Nov -2020 ISSN: 2582-3930

Image Enhancement for Street Sign Detection System

Anik Ghosh, Manoj Gowda G

Abstract— In this paper we detect traffic signs and classify them. This algorithm would work for bright light as well as during low light conditions. We will use low light image enhancement in order to improve the visibility of the road ahead, for the driver. We would increase the dynamic range of the image and use an algorithm that would recognize various types of traffic signs on the road and would be able to tell the driver about the traffic sign seen. This would allow driver to see all important road signs during low visibility conditions and ensure that a safe driving is achieved. Also we will train an algorithm using RESNET 50 on various traffic signs and test the trained algorithm using the original dataset and the enhanced images using our algorithm and determine the accuracy and precision in each case.

Index Terms—CNN, Deep learning,Image Enhancement , Image Processing , Machine Learning,

1 INTRODUCTION

Driving in todays world is more challenging than ever. With so many cars on the roads and road signs and marking to look for driving is more complex than ever. It is very important that driving in a manner that is safe for both the people inside the car and pedestrians outside the car. One way our safety is ensured is by using road signs which tells the driver necessary information about the road , the speed which they should maintain and other important information of the road. However driving at night or bad weather can sometimes make it difficult to see the street signs at once and thus it causes the driver to look at the signs for longer period of time which may compromise the safety. So in order to make sure that the street signs are properly visible to the driver an image enhancement algorithm has been developed which can enhance the images of the street signs and thus it can be easily spotted by the driver or can be classified properly by an pre trained classifier algorithm.

2 WHAT IS IMAGE ENHANCEMENT

In digital image processing, image enhancement refers to the transforming the image so that we can extract more information and features from the image. In order to detect or classify any object from an image or read any writing from an image we can enhance the image so the feature of our interest is further highlighted and could be easily identified by our algorithm. Since the project deals with pic-tures of the roads it is essential that our algorithm detects the necessary road signs even during low light and fog

2.1 Enhancing the low light image

We use dehiring algorithm to enhance the image. Our goal is to sharpen the image. Then in order to enhance the low light image we have to remove the haze from the image.

The first step is normalizing the R,G,B values of the pixel. Since we use 8 bits to represent each channel. The range of RBG channels is from 0 to 255. Upon normalizing its range will be 0 to 1. Now to obtain the inverse of the image we subtract the normalized value from 1.

$R' = R/255$; $G' = G/255$; $B' = B/255$;

Where R',G',B' are normalized R,G,B values respectively. To obtain the inverse we use

$R_{inv} = (1-R')$, $G_{inv} = (1-G')$, $B_{inv} = (1-B')$.

Then we apply Laplacian filter to sharpen the edges of the image. This removes blur from the image due to bad light and helps to enhance the features. Thus our primary aim which is to enhance the road markings, road edge and traffic signs from the image is done.

2.2 How to extract the features

I. Resizing the image

Depending on the camera used the resolution of the image obtained can be anything. So we have to convert the image into a standard resolution for our algorithm to detect the road sign from it. We choose 600x600 three channel images as the resolu-tion of our images.

II. Extracting various regions

After enhancing the image, we have to separate various regions from the image. We convert the RGB image to a grayscale image with 0 to 255 gray levels. Then with the help of thresholding we have to convert the grayscale image to a binary image. For a threshold gray level value T_h, if the gray level value it above T_h we set that pixel value to be 1. If the pixel value is below T_h set the pixel value to 0. For a pixel value P, The binary value will be

$$P = 1 , P > T_h$$

$$P = 0 , P < T_h$$

Almost all traffic signs have a clear border and a good contrast between the sign background and the actual sign or text. This is done do that drivers can easily spot the sign from a distance. Further we have used contrast stretching on our image to improve the contrast of the various portions of the traffic sign. As a result when we obtain the binary image we can clearly see the boundary of the traffic sign and the actual text or sign.

III. Morphological Operation

However the sign or text may appear to be connected and might reduce the accuracy of your algorithm which will classify the various signs. To solve this issue we use thinning operation to remove any connected components from our binary image. For thinning we require a structuring element containing 1s or 0s. The thinning operation for an image A and structuring element B is defined by,

Thin(A,B) = A - hit-and-miss(A,B) or

Thin(A,B) = A union hit-and-miss(A,B) where B is a series of 8 structuring elements as shown below.

0		1
0	1	1
0		1

1	1	1
	1	
0	0	0

	1	
1	1	0
	0	0

	1	
0	1	1
0	0	

1		0
1	1	0
1		0

0	0	0
	1	
1	1	1

	0	0
1	1	0
	1	

0	0	
0	1	1
	1	

2.3 Figures

The figure shows the image enhancement and thresholding of a low light image using the algorithm.

The above figure shows the image thresholding of a daylight image. Since the average intensity of the image is more than 100 we did not apply any image enhancement algorithm.

3 WORKING OF FEATURE EXTRACTION

Before the image can be enhanced or process we would determine if the image is taken during day time or night time. If the image is taken during day time them we need not enhance the image further as sufficient light is present in the image for the traffic sign to be detected. If the image is taken during low light or night time where there isn't sufficient illumination we shall enhance the image so as to improve the chances of detection.

To determine whether an image is taken during day or night we convert the image into grayscale and calculate the mean intensity of the image.Intensity refers to the amount of light or the numerical value of a pixel. It is given by the formula

Mean intensity(I) = ∑(intensity of each pixels) / number of pixels.

After finding the mean intensity of 20 samples, 10 of day time road image and 10 of night time image we found that aver-age intensity of an day time image is above 100 and the average intensity of a night time image with poor light conditions is be-low 60. So using this threshold value we first classify if the im-age is daytime image or night time.

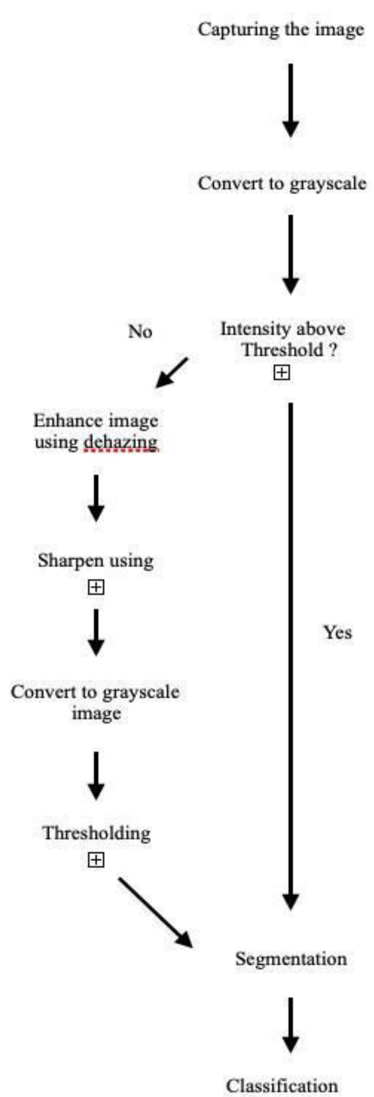
4 MATLAB CODE FOR IMAGE ENHANCEMENT

```
A = imread();
Ag = rgb2gray(A);
A1 = mean2(Ag);
disp(A1);
if (A1 < 100)
    AInv = imcomplement(A);
    BInv = imreducehaze(AInv);
    B = imcomplement(BInv);
    Binvert = imcomplement(B);
    B2 = imreducehaze(Binvert);
    C = imcomplement(B2);
```

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 04 Issue: 11 | Nov -2020 ISSN: 2582-3930

```
    D = imadjust(C,
stretchlim(C, [0.10, 0.90]),[]);
    E = rgb2gray(D);
    F = im2bw(E,0.7);
    G = bwmorph(F,'thin')
    figure,
montage({A,B,C,D,F,G}); else
    Y = im2bw(A,0.5);
    Y = bwmorph(Y,'thin');
    se = strel('disk',5);
    figure, montage({A,Y});
end
```

5 FLOWCHART OF THE ALGORITHM



6 USING CNN TO CLASSIFY THE TRAFFIC SIGNS

For our purpose we will use convolutional neural networks or CNN. A CNN is specially designed for image classification, object detection and image recognization. Convolutional Neural Network (CNN), sometimes referred to as a ConvNet, is the most well-known image recognition and classification algo-rithm. A typical CNN consists of a combination of convolution-al, pooling, and dense layers. CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer.

For our purpose ,we will use Resnet which is a short form of *Residual Network* 50 which is a pre trained neural network which is trained on 1000 different categories of image each hav-ing a million training images from imageNet database. This will result in faster training of the images. The CNN is 50 layers deep.

6.1 RESNET 50 ARCHITECTURE

It consist of the following layers:

- A convoultion with a kernel size of 7 * 7 and 64 dif-ferent kernels all with a stride of size 2 giving us **1 layer**.
- Next we see max pooling with also a stride size of 2.
- In the next convolution there is a 1 * 1,64 kernel fol-lowing this a 3 * 3,64 kernel and at last a 1 * 1,256 kernel, These three layers are repeated in total 3 time so giving us **9 layers** in this step.
- Next we see kernel of 1 * 1,128 after that a kernel of 3 * 3,128 and at last a kernel of 1 * 1,512 this step was repeated 4 time so giving us **12 layers** in this step.
- After that there is a kernal of 1 * 1,256 and two more kernels with 3 * 3,256 and 1 * 1,1024 and this is re-peated 6 time giving us a total of **18 layers**.
- And then again a 1 * 1,512 kernel with two more of 3 * 3,512 and 1 * 1,2048 and this was repeated 3 times giving us a total of **9 layers**.
- After that we do a average pool and end it with a fully connected layer containing 1000 nodes and at the end a softmax function so this gives us **1 layer**

The first layer defines the input dimensions. Each CNN has a different input size requirements. The one used in this example requires image input that is 224x224x3. In out case the classifi-cation labels of the respective categories of the image will be the folder names. The intermediate layers make up the bulk of the CNN. These are a series of convolutional layers, interspersed with rectified linear units (ReLU) and max-pooling layers [2]. We use ReLU function because of its linear nature and it can output a truely zero value. Following the these layers are 3 fully-connected layers. The final layer is the classification layer and its properties depend on the classification task. In this example, the CNN model that was loaded was trained to solve a classify 65 different traffic signs. Thus the classification layer has 65 classes.

Pre Processing for Classification

We first split the dataset images into two portions. The first portion will be used for training our neural network. The second portion will be for testing our trained algorithm. For our purpose we will use 70% of the datasets to train our neural network and remaining for testing. We randomize the split to avoid any bias in our final result. Thus the first 70% pictures will be used for training and the last 30% for testing the algorithm.

As mentioned earlier we preprocess the dark or low light image in our datasets for better accuracy of the network. The enhanced image will be passed into the neural network for training. However the net can only process RGB images that are 224x224. So we have to use augmentedImageDatastore to resize and convert any grayscale images to RGB on-the-fly. Since we do not have any grayscale image this will only resize the image.

6.5 Feature Extraction

The first layer of the CNN will capture the primitive features from the images like edge and blobs. These features are then processed by deeper network layers, which combine the early features to form higher level image features. These higher level features are more suitable for recognition tasks because they combine all the primitive features into a richer image representation.

6.5 Training in CNN

We now use the CNN image features to train a multiclass SVM classifier. A fast Stochastic Gradient Descent solver is used for training.

7 RESULTS

Since our algorithm works both for day time and night time scenarios on the road it will be able to detect and show traffic signs to driver in any condition. Also the enhanced image is able to show other details in the roads like the road edge, road markings and other cars in front of the driver. The enhanced image derived from the datasets are used to train our algorithm. We have trained it for 10 epochs.

Max Epoch: 20
Training Error: 3.61%

Without using Low Light Image enhancement Algorithm:

Mean Accuracy = 61.86%
Mean Error=9.12%

With using Low Light Image enhancement Algorithm

Mean Accuracy = 69.32%
Mean Error=6.78%

7 INFERENCE

We have successfully trained our Image classifier using Resnet 50. Initially for our testing set we did not use the enhanced images to test the algorithm. But after using the low light image enhancement algorithm the accuracy of production increased by around 8%. Also the mean error in validation reduced from 9% to 7%.

We can see that on applying enhancement on the images, their features get enhanced and prominent which makes it easy for our algorithm to classify them.

For real world implementation it is necessary that the algorithm can detect traffic sign in real time using a camera. We can perform our algorithm on the video captured from the camera so that the street signs can be detected.

7 ADVANTAGES

The paper would also identify the advantages of the proposed traffic light detection system.

It has been tested on images of various lighting conditions and the image enhancement works fine as it can differentiate efficiently between daylight and night time conditions and process the image accordingly. Also the low light enhanced image can be directly useful to the driver for better visibility. Since we are not improving the low light image using AI, the overall time for training the algorithm will be less.

The algorithm does not use color channel values to classify the traffic signs, instead it uses contour to detect the shape and binary image array to classify the signs. So it can be used to detect signs in almost any part of the world since most traffic signs have similar shapes and symbols.

Additionally upon classification of the traffic sign instead of displaying it to the driver, the information of the sign can be played in the form of a voice. This will greatly benefit drivers who have poor visibility. Also it helps the drivers to keep focussed on the roads thus reducing the chances of accidents.

CONCLUSION

The image enhancement algorithm is able to successfully differentiate between a daytime image and low light image and is enhancing the low light image so that the machine learning algorithm can easily detect the road signs. Also it is observed that there is a improvement of accuracy when image enhancement is used to test the street sign classifier algorithm

REFERENCES

1. Y. Sun, P. Ge and D. Liu, "Traffic Sign Detection and Recognition Based on Convolutional Neural Network," 2019 Chinese Automation Congress (CAC), Hangzhou, China, 2019, pp. 2851-2854, doi: 10.1109/I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
2. CNN Design for Real-Time Traffic Sign Recognition, Alexander Shustanova, Pavel Yakimova
3. Nithyananda C R, Ramachandra A C and Preethi, "Survey on Histogram Equalization method based Image Enhancement techniques," 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE), Ernakulam, 2016, pp. 150-158, doi: 10.1109/SAPIENCE.2016.7684156.
4. Chan, Yung-Kuan & Huang, Der-Chen & Liu, Kuo-Ching & Chen, Rong-Tai. (2014). An Automatic Indirect Immunofluorescence Cell Segmentation System. Mathematical Problemsin Engineering. 2014. 1-13. 10.1155/2014/501206.
5. Chang, Chin-Chen & Lin, Huei-Yung & Huang, Shu-Chun. (2018). Traffic Sign Detection and Recognition for Driving Assistance System. Advances in Image and Video Process-ing. 6. 10.14738/aivp.63.4603.
6. K. Lu, Z. Ding, and S. Ge. Sparse-representation-based graph embedding for traffic sign recognition. IEEE Transactions on Intelligent Transportation Systems, 13(4):1515–1524, 2012.
7. P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. CoRR, ab- s/1312.6229, 2013.
8. J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. Neural Networks, (0):–, 2012.
9. C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In ECCV. European Conference on Computer Vision, September 2014.