

Investigation on road-sign recognition

Sam Gilvine Samuvel
Department of Software Engineering
Kaunas University of Technology
Kaunas, Lithuania
e-mail: gilvine21@gmail.com

Praywin Moses Dass Alex
EinNel Technologies
India

Abstract— Traffic sign recognition is an important research topic for enabling autonomous vehicle driving system. This paper presents an investigation of the road and traffic sign detection and recognition system. Road sign detection systems is kind of system which uses computer based application related to vision. It also allows the drivers to follow the rules by imposing restriction and makes the driver follow and obey the rules and regulations through the use of image processing technique. The aim is to review systems of road sign detection, to present implementations of road sign recognition algorithms, and compare four distinct algorithms (Scale invariant feature transform, Support vector machine and Histogram of oriented gradients, Gabor, Blob analysis) to find which algorithm works the best to detect road sign in terms of speed and robustness of the method. The results show the blob analysis algorithm works better for these both speed and robustness of the methods.

Keywords—Recognition; detection; SIFT, SVM, HOG, GABOR, BLOB analysis

I. INTRODUCTION (HEADING I)

The shapes of the road signs, such as triangle, circle and rectangle, are generally difficult for the observer in nature. Nowadays, there is a demand for road safety, where the safety of driving should not only rely on the driver, but also on the computer system, which the vehicle uses. It can be said that, there are many reasons on how a road accident may occur. As per general knowledge, over speeding is one of the main reasons on why the road accidents occur.

To stop an accident from occurring, there are different sets of road sign placed on the side of the roads. This can pose warnings to the driver at the intended location. Sometimes, because of lack of concentration, lack of visibility or lack of knowledge on the road sign by the driver may cause his to override the traffic rules and drive the vehicles based on their need, because of which, the driver puts himself in a life-threatening situation. There are many researches made, like the obstacle detection and path detection, which can give very useful information to the driver and help him avoid danger, but the road sign detection has still been on the research phase. It is known that the road signs play a vital role while driving on any roads, playing an important role on road conditions and

scenarios. Although there are many number road signs, there is a possibility that the driver cannot perceive the road sign. In such places, if driver supportive systems are used, the chances of accidents occurring could be minimized.

In this type system, the driver will be user of the system and will acquire visual warnings when a traffic sign is identified in front of the user. As per the research, it is said that the road sign detection is still at wide research that needs to be worked on. Here, we input a road sign to the system, this road sign is then compared with the other road signs already available in the database. The image is converted from RGB to gray scale, in order for better processing of the image. The comparison of these images starts with, first detection of the key features. Once the key features are detected, the next step that the system does is by extracting the key features. Here, we use 4 different sets of algorithm for the extraction of key features. The 4 sets are:

- 1) SIFT Algorithm [13]
- 2) Histogram of oriented gradients(HOG) and Support vector machines(SVM) [3]
- 3) GABOR and NEURAL NETWORK [11]
- 4) BLOB ANALYSIS NEURAL NETWORK [5]

II. RELATED WORK

A. Traffic Sign and Recognition

Comprehension and detection of traffic signs have been quite a recurring area of research [1], have reported that public datasets have reached to only a fairly good number and a proportion to studies relating to empirical basis.



Figure 1: Traffic Sign and Recognition [1]

The above figure (fig. [1]) depicts traffic signs that were typically used. The bounding box that surrounds the traffic

sign was used to crop the image as for data acquisition. There were a set of three main categories otherwise referred to as super classes with respect to their shape and colour designated by the letters M which stood for mandatory, D which denoted danger and P which depicted prohibitory. Under the subject of datasets there were subject relating to traffic sign detection which had subsets such as Integral Channel with Features Classifier, Aspect Ratio Detection technique, Basic Training System and Experimental Detection. The features picked up as in the training of channels for the unsafe zone had channel divisions as depicted in the fig [1]. The authors [1] depict modern techniques that performed on large scale detection and classification datasets taken into account from countries Belgium and Germany. Without the use of application and just by virtue of specific modification, existing methods for pedestrian sign detection and face & digit classification, the noted efficiencies were in the range of 95-99%.

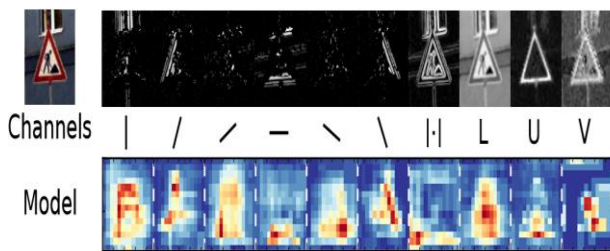


Figure 2: Training of Channels [1]

The acquired image was scaled to size that pertained to the best-fit for general analysis. The technique that the authors [2] implemented had four steps in total which were as follows. The primary step as previously mentioned was to notice the traffic signage and depending upon the net contrast, Sobel edge detection technique in addition to morphological action. In the secondary step the detected sign was worked upon with the concept of row and column count. The third step which in another way was quite imperative was data extraction using discrete cosine transform(DCT), discrete wavelet transform(DWT) and Hybrid DCT-DWT. These are subjected to feature extractions, 20 highest energy coefficients as in training phase as in DCT, 300 features from traffic signs as in DWT and 20 features in Hybrid DWT-DCT. The final step involved the recognition through SVM. [2]

B. Traffic Sign Recognition for Intelligent Vehicle Using Neural Network on Open Source Computer Vision(Open CV)

Traffic Sign Detection and Recognition contains many features to help the driver in proliferating rate of safety. This could be noticed because of immense significance in the automotive industry. This in another way means improving comfort and safety in general. This method [5], provides a neat detection and recognition system which would be of ready assistance to the drivers. The approach fundamentally consisted of two imperative modules. Primarily there is a detection module which has its basis on colour segmentation

and the secondary factor being edge detection to typically make note of road signage. There was a multilayer receptor that functioned as a recognition module. The basic idea behind using this is to relate to the road signage patterns using the software C/C++. The tests were conducted on real images of traffic signage to gauge the efficiency and performance.



Figure 3: Panels Considered in this Work [5]

The system used a traffic signs detection module and by virtue of chain codes the components were operated and freeman method was used to depict the contours. The other step was to make simple a piecewise linear curve and for this Ramer-Douglas algorithm was used. A BGR colour image is same as RGB except the order of areas is reversed. Red occupies the least significant area, Green the second(still), and Blue the third. colour image, threshold masks blue and red in colour are depicted below fig [4]. The traffic sign recognition here was based on the principle of neural networks technique. The traffic sign was initially processed by image acquisition. The Image processing methodologies such as Threshold technique, Ramer-Douglas, Contours and Fit-Ellipse were primarily used. The model serves a low computational cost object which is quite a merit. And thus, real-time implementation is possible as the authors suggest.



Figure 4: BGR Color Image [5]

C. Detection and Recognition of Alert Traffic Signs

Traffic signs provide critical information about safety and navigation. Automatic detection of traffic signage and signals is quite imperative. The method Histogram of Gradient (HOG) [8] to draw out images and complete 1680 features. A cascade classifier built with a Support Vector Machine (SVM) was used. To process the information the colour information from different layers are placed onto a single vector. This functions

a feature descriptor. Colour segmentation is carried out to reduce regions of search which might seem time consuming otherwise.

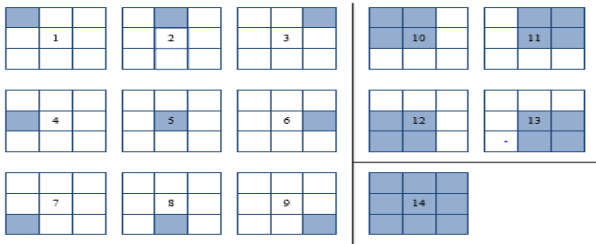


Figure 5: Sub-Blocks of the Image [8]

The features that were imbibed in this were colour encoding, HOG features and learning methods. There were sign and non-sign images in the data sheet which proved to be quite useful with regards to discretion purposes. The important features of the method were stop and do not enter signage. Data set training was carried out in addition to cascade sign detecting. Speed and colour segmentation were other imperative features used to define the methodology. The authors [8] have put down an efficient alert traffic signage recognition scheme. The detection rate was found to be 92 % -100 % and the false positive rate ranged from 0.19 % -5 %. The authors [8] recommend that this system can take into account illumination and pose variance. The processing time in Matlab was found to be 7-9 seconds and the real-time detection could be achieved [8]. The traffic sign with evaluation procedures in the outline of the signage. The observation was basically done with subject to outdoor systems. Since the information of the traffic signage is quite imperative with regard to safety, there were cases with poor illumination of traffic signage which affected the vision in general and this was quite unpredictable considering the position and orientation of the signage. The authors [4] had developed an artificial vision system which recognized traffic signs considering geometric shapes in general even though there was poor illumination in general.

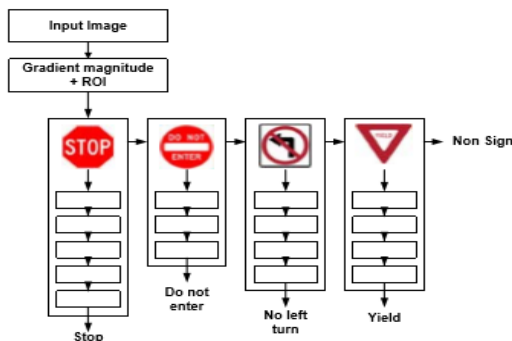


Figure 6: Cascade Sign Detectors [8]

D. Multi-feature and Multi-classifier fusion

A swift and robust method for traffic sign recognition which typically used a range from coarse to fine strategy is presented in this case. The Gabor analysis with subject to traffic signage

that were divided into two categories which basically were main and sub-divisions. There were speed limit signs corresponding to speeds ranging from 20-120 km/hr. The other signs were prohibitory signs, Derestriction signs, Mandatory signs, Danger signs and unique signs. Real and imaginary parts of Gabor filters used for orientations that were four in number and on two scales. Under the process of classification, the pre-processing, complementariness with local features, local binary pattern and multi-feature fusion were used. The classifier accuracy ranged from 98.51 % to 98.76 % under conditions of SVM and SVM+RF respectively. The RGB accuracy was found to be 99.79 % with gray scale % at 98.95. The Gabor filter had an accuracy of 97.09 %. The recognition rate was found to be 98.76 % which the authors claim is very close to human recognition range. The fusion of the features of makes this method a sturdy hierarchical method with a high degree of accuracy. [9]

III. METHODS

A. Scale Invariant Feature Transform

Scale invariant feature transform [13] is constant to image scale, distortion of affine, rotation and revolution, variation in 3-dimension view point, disturbance in the signal and changes in illumination, which are best match for the matching of the image. Here, the method is carried out in two various stages, detection of key point and description of key point. Each stage has their own sub stages and they are given below:

Key point detection:

A) Scale-space detection

B) Key point

Key point Description

A) Orientation assignment

B) Key point descriptor.

In the first stage, the search is made in every scales and image location. This is done by using the function „detect SURF Features“ which then identifies the potential point of interest that stays constant to both the scale and orientation. At every intended location, the feature detection model is employed to determine the exact location and the scale of the points. These key points are based on the estimated values of their stability. The image gradients at every local point are estimated at the intended scale in the region around every key points. These are then converted to depict that allows for all significant levels of variation in local illumination and distortion of shape. Finally, bounding box method is used to display the region of importance. The bounding box is a feasible and a known method, which is used by various existing interactive image segmentation frameworks. Here, we discuss how this method is used to represent a topology, which prevents the solution from enormous amount of compression and guarantees the user provided box bounds segmentation is a needed tight way [13]. The interpolation is done using

quadratic Taylor expansion of the difference of Gaussian scale space function Taylor expansion is given by,

$$D(x) = D + \frac{\partial D}{\partial x} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial x^2} X \text{-----(1)}$$

B. Support Vector Machine Classification and Watershed

The identification of a traffic sign from various set of traffic signs requires knowledge of their structure and brightness or intensity value. Traffic sign detection technique is a tedious tasks and it needs various steps to get the needed result. In the beginning stage, the original input image is converted to gray scale image and the entire work in this method is done with the converted gray scale image. Therefore, the detection will emphasis on edge information, shape and structure of the sign, morphological operations and filtering. The input is taken in as RGB, but soon it has to be converted to gray scale image. The formula used to identify the intensity value is given by:

$$I = 0.2989 * R + 0.5870 * G + 0.1140 * B \text{----- (2)}$$

Where

„I“= image intensity value ranging from 0,255.

„R, G and B“ = red, green and blue from RGB color model.

In this phase, the variation in intensity is detected from the input image. If the traffic sign has some needed contrast with background, then it uses edge detection. The canny method is used to extract traffic sign with background. Filling the holes of the binary images, the dilated gradient mask shows the outline of the image quite nicely, but the holes (noise) in the interior of the image are still appears. To reduce such noises from occurring, the use hole filling algorithm is used. The connected borders are removed to a needed portion as an output.

Smoothen the traffic sign board is then obtained as that the segmented object should look natural as compared with the original image. Let us consider the topographic surface, water would collect in one of the two catchment basins. Water falling on the watershed ridge line separating the two basins would be equally likely to collect into either of the two catchment basins. Watershed algorithm then finds the catchment basins and the ridge lines in the image. The watershed algorithm is implemented in MATLAB image processing toolbox as $L = \text{watershed}(f)$, where f is the input image and L is labelled matrix image having positive integer values at different regions and at the watershed ridge lines. The key behind using watershed transform for segmentation is this changing image into another image whose catchment basins are the objects you want to identify.

Histogram of Oriented Gradient (HOG) is feature descriptors used for the purpose of object detection. HOG is used to capture color and shape as one feature. The gradient at each pixel is the gradient with the greatest magnitude among the gradients computed on each of the channels. The bin is increased from 3 to 9 and the binning of the unsigned gradient orientation from 0-180. Then rescaling each region in to $3*3$ pixels and describe it by $9*9$ blocks of $8*8$ cells with 8 pixels. Extracted output from HOG is given to SVM classifier to analyse and recognize patterns, used for classification and regression analysis. SVM classified as either sign of interest or background [3].

C. Gabor Filter and Feed Forward Propagation(NEWFF)

The basic function of filters in the Gabor family [11] are two dimensional and can be represented as a Gaussian function modulated by the oriented complex sinusoidal signal. These types of filters generally act as band pass filters. In this paper, we use Gabor filter and also create Gabor filter. Once the filter bank is created, Gabor features are then extracted. During the implementation, we generally input the image, which is selected manually. In the beginning, the sub images are scaled to a fixed size. It can be of any fixed dimension. Soon after the scaling, the image is the sub-divided into desired overlapping windows. Later, assumptions on patches contained in the sub image are made, where each patch consists of its sub windows. The Gabor windows are then subjected to each and every sub windows separately.

„newff“ is a tool used in MATLAB for access neural network and it is elaborated as Feed Forward Neural Network, which is an artificial neural network, where the links are established between two units that do not form a cycle. As per the research, this type of network is first of its kind also the simplest type of artificial neural network invented. This operates in a half-duplex mode, where the information flows in only one direction, and in forward direction from the input node, through the other hidden nodes to the output node. In this network, there are no loops or cycles. These neural networks are of two layers.

A) Single layer perceptron

B) Multi-layer perceptron

In case of single layer perceptron, the data flows from input node to single layer of output nodes. But in case of multi-layer perceptron, there are multiple layers of computational units, usually interconnected in a feed-forward direction [11].

D. Blob Analysis

In this method, [5] the main aim is to detect the regions in any digital image, which has a varying property, such as intensity or color that camped be compared to the other surrounding regions. It is said the blob region is the one where the properties of the image lying on them usually have the properties, which do not change at all. Therefore, giving us idea that all the points in blob to be similar to each other.

In the first method, the derivative function is based on the position and in the later method, it is based on identifying the maxima and minima of the local region.

In beginning, the blob detection was used gather regions of interest for further processing. These regions could pose the presence of some part or an object in the selected image domain, with respect to recognition of some object or tracking of object. It can also be related to histogram analysis, where the blob descriptors can be used to detect the peaks with an application to segmentation. The other general use of blob descriptor is mainly for texture analysis and texture recognition.

Soon after the blob analysis on an image, we then process further in neural network. Here, the data are trained, their performances are evaluated. Some part of the needed information is stored to „net“ file and are further processed. The obtained performance of the input image is then compared with the estimated performances of all the images, prior to the experiment. If the condition is found to be satisfied, a small image box is opened where the selected input will be viewed along with some desired message that we enter into our code [5].

IV. IMPLEMENTATION

A. Sift Features

In the implementation phase begins with the selection of image which will be loaded to the program. The filename of the image file is selected along with its destination path and is stored to the temporary file created in the program. The stored image file is then read and then stored to the file. After storing, the image is then converted from RGB to gray scale and further stored with the same name. This image can be later viewed and can be renamed to desired title. Later, the image from the database is been read into the program. This image consists of all the possible traffic signs needed to carry of the experiment. The image is later resized and is converted as gray scale image. After the loading of main image and reading the reference image from the database, the next stages will follow the comparison between these two images, as the features for these images will be detected, later extracted and then matched. Once the images are loaded and read, the next stage for the comparison begins with the feature detection phase.

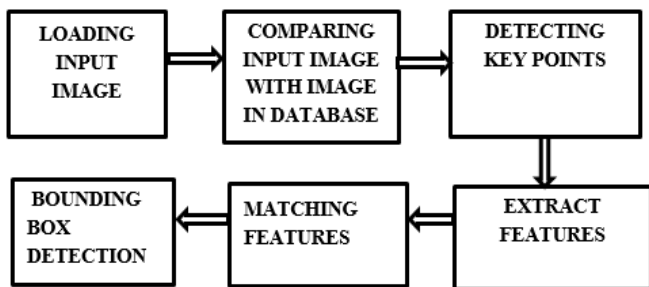


Figure 7: Block Diagram for SIFT Features

The features are first detected for the input image and stored as „Box Points“. It is programmed in such a way to display only 100 key points for the input image and is then plotted under the title „100 strongest key points from the Box Image“. Then later, we detect the key points for the reference image and it is stored under the name „Scene Points“. Since, this reference image has got more traffic sign images into it. It is programmed in a way to display 300 strongest key points, which is 3:1 ratio with the input image. This image is later plotted as „300 Strongest Feature Points from Scene Image“. In both the cases, a special function called „select Strongest“ is used to select the strongest key points. Once the key

features are detected, the next stage will be extraction of these detected features. The function „extract Features“ is used to extract the features from both the images. The needed information for this function to operate is the filename in which the gray scale image is stored and then their respective detected feature points stored in a file. For example, the key features for the input image are extracted and the detected key features stored in box Points“. The extracted features are then stored to the file „box Features“. The same technique is followed for the reference image. where the image is compared with its detected feature points stored under the file name „scene Features“ to obtain its respective feature points and it is later stored under the name „scene Features“. In this phase, the extracted features of the input image and the extracted features of the reference image are matched and the values are made to store. The matched features are viewed using the function „show Matched Features“, where the matching points of the image are obtained and are moved to the default file named „montage“. Now this matched image is then displayed and entitled as „matched points (including outliers)“. In next step, we apply geometric transform on to the matched box points and on matched scene points. Later this image is viewed as „Matched points (inliers only)“.

A box polygon values are created to each rows and columns. The forward transform points are applied on the estimated box polygons and are renamed as new box polygons. This image is then displayed as binary image output. The threshold level of the gray image is obtained and is saved in „level“. The image is then convert to binary using thresholding the size of the binary image is the estimated. To make sure all the needed information to be covered inside the frame. The highest values of 9 are allotted to both the row and column. Once the image is identified, a frame is made to be marked around it. Later, a bandwidth label is estimated for the binary image. We then use the function „region props“, which then estimates the bounding box for the image. The size for the bounding box s again altered based on the needs. Then we shoe the figure of the input image along with a frame around it and rename this as „bounding box image“.



Figure 8: Matched Points

B. Hog Features and Svm Classification

We begin the implementation generally by loading the desired image file into MATLAB program destination. It's

programmed in a way to accept JPG files as the input, and once acquired, it can be stored to any path and into any file name. This file is then read and it is stored. This image is then converted from RGB into gray scale. The converted file is later resized to 512 X 512 and is stored. We detect the border of the dominant part of the image. After the edge detection is phase, the detected image is then dilated.

This dilation process is done to enhance the detected region. The outline of the image can be viewed clearly due to dilated gradient mask. To dilate the image, first we call the function „strel“, which creates as non-flat ball shaped image, whose height in this case is selected as 5 and its radius is also selected as 5. Then, dilate the image along with the elements in the flat structure and this dilated image is moved into the file called „lobrd“. After the image is dilated, there will be noise occurring in the signal, in the form of „holes“. In order to reduce this noise, we use an algorithm, which is especially dedicated for filling the holes. This algorithm is called into the program and is followed by the dilated image stored in the file named „Lobrd“ and the „holes“ themselves. The image is then moved to the file, named „BW3“. This image is then display and is titled as „Hole Filled Image“. The obtained image has to be smoothened in order to make the resulting image look natural when compared to the input image. This can be done by eroding the traffic board object. The smoothened image is then stored to the file „BW4“ and displayed with the title „Smooth Image“. It is an approach which is exclusively used for segmentation of the image, where the objects in contact with one another are segmented clearly. In order to do this, the function, „watershed“ is called on the smoothed image in the file „BW4“ to identify the objects in contact with one another. These values are then stored to label matrix „X“. Then we use the function, „label2rgb“ which converts the labels to the RGB according to the color map specified in the program. Here, the map is a string which contains the color map function „jet“. Since the color map color selected is shuffled, the color map colors will be pseudo randomly shuffled. This image file is then copied to the filenameL1“and entitled and Watershedimage“.

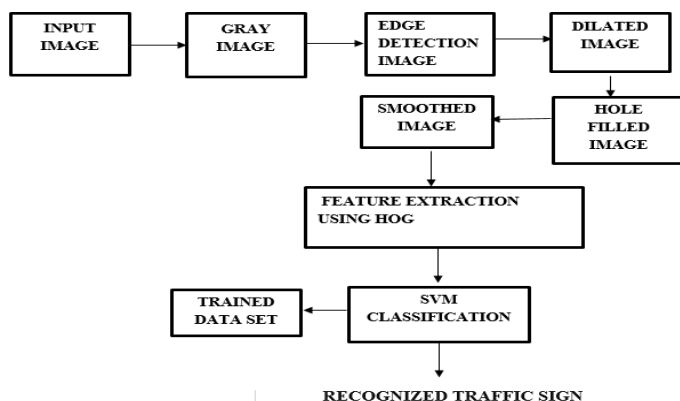


Figure 9: Block Diagram for SVM Classification

HOG is a feature extraction algorithm with is used for the purpose of object detection. This algorithm is mainly used to capture the color and shape of a particular feature. In the

beginning, we set the HOG window value to „3“ in both X and Y coordinates. The Bin value is then set to the value „9“. Then the size of the watershed image is the estimated with the function „size“ and the values of lines and columns are stored into an array. Then, the column vectors are initialized to zero using the functions „zeros“. Later, square root is performed on the estimated values and stored. Later, a condition is checked, and if it is satisfied, a verification is done to estimate the size of the image. We then estimate the image which is present. In the images, any objected placed, will be the key information, where the background of this information will be made to black, therefore, the object present in the image will be only viewed in front and this object will provide a peak value at its location, based on the image will be detected. The obtained histograms are then assembled with 9 bins and it can vary till 20 bins. We then create a bin with the name „H“ and the needed information’s obtained or calculated will be moved into the bin named „H“. In this phase, we use the data’s used in the excel sheet to recognize the data present in the bin „H“. We also sum the values with the estimated values in the bin and store them to the excel file. finally, we use the condition, if the values in the E column matches with the already calculated values of the various images. If this condition satisfies, a small message box is made to open, where the input image entered would have been selected and displayed with the message of our choosing, which we use in the program.

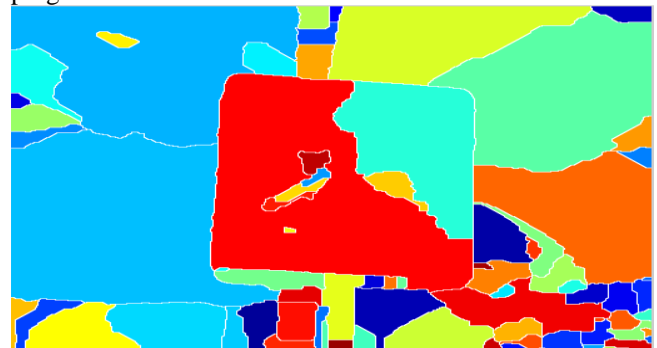


Figure 10: Watershed Image Output

C. Gabor Neural Network

The first stage is where an input image is loaded into the program. The loaded image is of JPG format. The loaded image I then read on to the program and stored. The image is the resized to the desired resolution. The resized image is then converted from RGB to gray scale. This image is stored as gray. We then create a function called filter bank and keep it ready to store variables or key points. The filter bank is then called upon using create filter bank function. We then use Gabor filter, where we use the filter bank and perform a convolution operation using Fast Fourier Transform. This transform is carried on the image, which has been converted to gray scale. The values are the entered to „filterParams“. We the view response of the filter by calling Filter Bank. Show Responses, where we see the parameters of the gray scale image. After the transform is applied to the image, the next

step is to obtain the feature of the image. Here, the Gabor kernel is called using $@(x)$, which then extracts the feature of the image. The values are moved to the file „feature Extractor“. Again, a convolution is done when we apply Fast Fourier Transform on the gray scale image and the extracted features. Later, the response of this filter performance is checked.

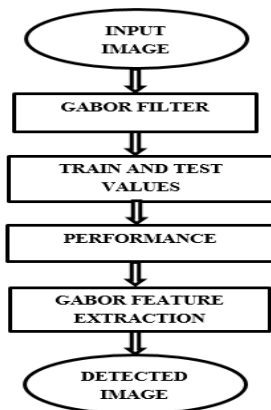


Figure 11: Block Diagram for Gabor Neural Network

In this stage, we create a neural network function training tool, which is used to train the data and compare them with the input image. First, we create a target T and the input P. we keep the database for the images, where their values are computed and are placed in an excel sheet. We compute the size of these targets and in input image. Later we call the newff function, which allows the data to flow in only one direction from the node to output nodes in a single later through various hidden nodes. We then set goal, epoch and other function to train the data and test their performance. A maximum function is taken and the values of P are calculated. We then obtain the error for the code using the target1 and the output files by applying subtraction operation over the mentioned physical parameters. The performance between net, target 1 and output is estimated using the function „perform“. The performance is the confirmed and sealed for the multiplication. Now we compare the performance „per“ with the image is folder. These values are obtained for every image in the data base priory. Here close to 35 to 40 images are being used in the database for comparison value. Once the value of „per“ is matched with any of the images of the files. The image which is imputed will be estimated. Once the condition is satisfied, the image which is selected will be correctly match the image in the database, and we get the exact value. The same condition is used for all the images and their values are satisfying conditions.



Figure 12: Gabor Transform Output

D. Blob Analysis Neural Network

The image is loaded to the system similar to how it was done in other algorithms. The image is then read into the code. The image is the resized to desired dimensions. Later this image s converted from RGB to gray scale. The first step in this phase is to set the level. Then the image is compared with the level and is then stored to „bw“. These stored file is the viewed and entitled as „Binary Image processing“. In this phase, the binary image undergoes morphological operation and the value is moved on to the file „bw2“. This figure is then viewed as „morphological operation“. The image is the compared to 100 and is stored as bw3. The size bw3 is later estimated. Here, the value I is considered as row and j is considered as column. Here, Y2 and X2 are selected as 9 and 10 respectively, is because to frame the outline of the image correctly, if in case the image is a large one. If the selected values are less, the bounding of image might happen in between the sign. The region property is then estimated for bounding box and label. The bounding box values are then converted from the structure to cell. The image is then viewed and the position and edge color are determined and renamed as shape measurement. The label value for the resampled image is estimated at the beginning of this phase. The region properties of these phases are then estimated and saved as stats. we find the indication area, the net stage emphases on estimation of perimeter. The size of the re-scaled image is determined and its perimeter is estimated. This image is later viewed and tiled as perimeter image. We find the perimeter values in an image in both the axis of the image. The perimeter of the image is then summed. And in other lines, the values are subtracted.

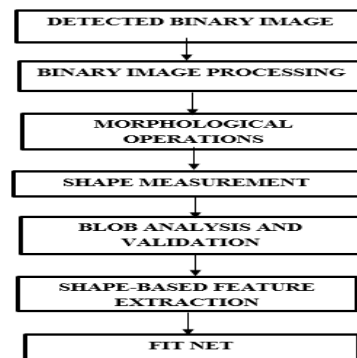


Figure 13: Block Diagram for Blob Analysis Neural Network

Finally, the estimated value of perimeter is the taken square root and is saved as D. In this stage, to find the normalized of radial distance, the standard deviation is divided to estimate it. This value is the viewed as the normalized derivation of radial distance. We then estimate compactness, where the mathematical formula is used to determine compactness, where the estimated area and perimeter are used in the equation. We then write the data to the excel sheet. Once written, the values are now read for all data's and is stored into the location „num“. we then use the neural network tool to train the set of data's, test their performance and check out for errors and store all the values under the file net. Once the performance is measure, we use the measure performance with the performances already measured for the images in the database and check for which performance, the estimated value of performance matched. If a condition is satisfied, the image in that condition is then viewed in a small message box. In this stage, there are more than 50 images used.

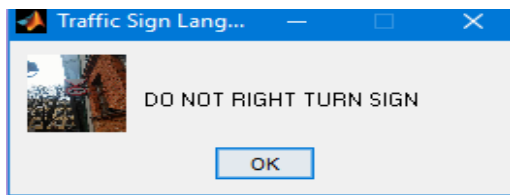


Figure 14: Recognized Output of Blob Analysis

V. COMPARISON OF METHODS

In the beginning, we had to estimate image acquisition speed. After estimating, we compared with method hand an upper hand over other method. It was found out that the Blob method had the acquisition rate of 3.78 images/seconds and it was the highest when compared to other methods. The GABOR method was determined to be the second-best method with the acquisition rate of 4.92 images per second. The SIFT method was the third best method as it had the acquisition rate of 9.56 images per sec. And finally, SVM method was the fourth best method as it had the acquisition rate of 14.71 images per sec. From this result, the Blob analysis method was chosen to be the best in terms of maximum data acquisition rate.

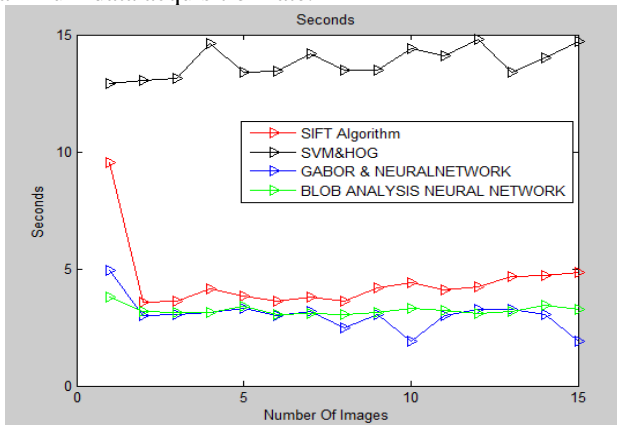


Figure 15: Graph for Speed

In figure 16, the quality of the all the 4 algorithms are estimated. There were 30 images used in this process and the algorithms were used on them to identify how these algorithms were able to recognize these images. After the process, the qualities of these methods were estimated and are listed below:

- 1) Blob Analysis and Neural Network = 96.6%
- 2) HOG and SVM = 93.3%
- 3) GABOR and Neural Network = 93.3%
- 4) SIFT Algorithm = 83.3%

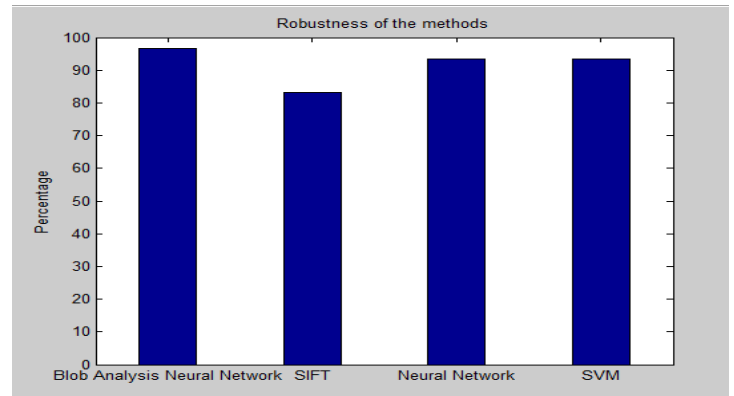


Figure 16: Robustness of the Methods

TABLE I. COMPARISON OF METHODS

Methods	Robustness (%)	Image Detection Time (sec)	Average
SIFT	83.3	9.56	4.46
HOG & SVM	93.3	14.71	13.81
(Newff) Gabor Neural Network	93.3	4.92	3.03
(Fitnet) Blob Analysis Neural Network	96.6	3.78	3.22

VI. CoNCLUSION

In this paper, we implemented a four distinct algorithms for traffic sign recognition. And we calculated speed and robustness of the methods. The robustness is the accuracy of the recognition. The algorithms are as follows: scale invariant feature transform, support vector machine and histogram of oriented gradients, Gabor, Blob analysis. We have proposed a FITNET function in algorithm of blob analysis. The blob analysis has higher detection rate than the other algorithm. The image detection speed was in the precedence order of BLOB (3.78 images/sec) > GABOR (4.92 images/sec) > SIFT

(9.56 images/sec) > SVM & HOG (14.71 images/sec). This indicated that BLOB algorithm had the maximum data acquisition rate. Comparing the above conclusions, it is observed that BLOB algorithm was the best-fit followed by SIFT algorithm. The quality of algorithms had precedence of BLOB was found to be the highest, i.e., BLOB (96.6 %) > SVM (93.3%) = GABOR (93.3%) > SIFT (83.3%).

REFERENCES

- [1] Mathias, M., Timofte, R., Benenson, R., Gool, L. V., "Traffic sign recognition — How far are we from the solution?" *The 2013 International Joint Conference on Neural Networks (IJCNN)*, 2013. doi:10.1109/ijcnn.2013.6707049
- [2] S.Sathiya, M. Balasubramanian, S. Palanivel, "Pattern Recognition Based Detection Recognition of Traffic Sign Using SVM," *International Journal Of Engineering and Technology (IJET 2014)*, vol. 6(2), pp. 1147-1157, 2014.
- [3] M. Blauth, E. Kraft, F. Hirschenberger, M. Bohm, "Large-Scale Traffic Sign Recognition Based on Local Features and Color Segmentation," 2012
- [4] S. Lafuente-Arroyo, P. Gil-Jimenez, R. Maldonado-Bascon, "Traffic Sign Shape Classification Evaluation I: SVM Using Distance to Borders," *Conference: intelligent vehicles symposium*, 2005.
- [5] A. Salhi, B. Minaoui, M. Fakir, "Robust Automatic Traffic Signs Recognition Using Fast Polugonal Approximation of Digital Curves and Neural Network," *International Journal of Advanced Computer Science and Applications*, pp. 1-7, 2014. 10.14569/SpecialIssue.2014.040201.
- [6] A. Lorsakul, J. Suthakorn, "Traffic Sign Recognition Using Neural Network on Opencv: Toward Intelligent Vehicle/Driver Assitance System, Center for Biomedical and Robotics Technology (BART LAB) Department of Biomedical Engineering, Faculty of Engineering," *Conference on Ubiquitous Robots and Ambient*, 2007.
- [7] C.-H. Chen, M. Chen, T. Gao, "Detection and Recognition of Alert Traffic Signs," *Publication of Stanford university*, 2008.
- [8] S. Yin, P. Ouyang, L. Liu, Y. Guo, "Fast Traffic Sign Recognition with a Rotation Invariant Binary Pattern Based Feature," 2015. ISSN 1424-8220 *Journal/Sensors*.
- [9] Y. Ma, L. Huang, "Hierarchical Traffic Sign Recognition Based on Multi-Feature and Multi-Classifer Fusion," *First International Conference on Information Science and Electronic Technology (ISET 2015)*.
- [10] Z. Huang, Y. Yu, J. Gu, H. Liu, "An Efficient Method for Traffic Sign Recognition Based on Extreme Learning Machine," *IEEE Transactions on Cybernetics*, 2016.
- [11] Z. Sun, G. Bebis, R. Miller, "ON-Road Vehicle Detection Using Gabor Filters and Support Vector Machines," *Computer Vision Laboratory Department of Computer Science, University of Nevada, Ford Motor Company, Dearborn, MI, Digital signal processing*, 2002.
- [12] M. A. N. Al-Azawi, "Neural Network Based Automatic Traffic Signs Recognition," *International Journal of Digital Information and Wireless Communications (IJDWC)*, pp.753-766, 2011.
- [13] X. Hu, X. Zhu, D. Li, H. Li, "Traffic Sign Recognition Using Scale Invariant Feature Transform and SVM," *A Special Joint Symposium of ISPRS Technical Commission IV & AutoCarto in conjunction with ASPRS/CaGIS 2010 Fall Specialty Conference November 15-19, 2010 Orlando, Florida*.