KAUNAS UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATICS

PRAYWIN MOSES DASS ALEX AND SAM GILVINE SAMUVEL

INVESTIGATION ON ROAD-SIDE RECOGNITION

Master’s Degree Final Project

Supervisor
Assoc. prof. dr. ARMANTAS OSTREIKA

KAUNAS, 2016

KAUNAS UNIVERSITY OF TECHNOLOGY

FACULTY OF INFORMATICS

INVESTIGATION ON ROAD-SIGN RECOGNITION

Master's Degree Final Project

DEPARTMENT OF MULTIMEDIA ENGINEERING (P000M106)

Supervisor
Assoc. prof. dr. ARMANTAS OSTREIKA

Reviewer
Assoc. prof. dr. SIGITAS DRASUTIS

Project made by
PRAYWIN MOSES DASS ALEX
SAM GILVINE SAMUVEL

KAUNAS, 2016

KAUNAS UNIVERSITY OF TECHNOLOGY

INFORMATICS

(Faculty)

PRAYWIN MOSES DASS ALEX

SAMUVEL SAM GILVINE

(Student's name, surname)

DEPARTMENT OF MULTIMEDIA ENGINEERING P000M106

(Title and code of study programme)

"INVESTIGATION ON ROAD SIGN RECOGNITION"

DECLARATION OF ACADEMIC INTEGRITY

____ ________________ 20 _____

Kaunas

We confirm that the final project of ours, **PRAYWIN MOSES DASS ALEX and SAMUVEL SAM GILVINE**, on the subject "INVESTIGATION ON ROAD-SIGN RECOGNITION" is written completely by our self; all the provided data and research results are correct and have been obtained honestly. None of the parts of this thesis have been plagiarized from any printed, Internet-based or otherwise recorded sources. All direct and indirect quotations from external resources are indicated in the list of references. No monetary funds (unless required by law) have been paid to anyone for any contribution to this thesis.

I fully and completely understand that any discovery of any manifestations/case/facts of dishonesty inevitably results in me incurring a penalty according to the procedure(s) effective at Kaunas University of Technology

________________________________ ________________

(name and surname filled in by hand) *(signature)*

________________________________ ________________

(name and surname filled in by hand) *(signature)*

Surname, Name: PRAYWIN MOSES DASS ALEX AND SAMUVEL SAM GILVINE.

Title of the project: INVESTIGATION ON ROAD-SIGN RECOGNITION.

Master's Final Project

Supervisor assoc. prof. ARMANTAS OSTREIKA

Faculty of informatics

Kaunas University of Technology.

Kaunas, 2016. 05

SUMMARY

In SIFT Algorithm, SVM and HOG method, after estimating the key features; we match the key features of the image and the images from the database. In case of SIFT Algorithm, we go for bounding technique where, the matched image is converted to binary image type, we then estimate the boundary of the image. In case of HOG and SVM, we extract features using HOG feature extractor and then we compared the train of data's which are delivered to the SVM Classifier along with the feature extracted image from HOG extractor to recognize a traffic signal in the given input image. Both the Gabor filter and blob analysis uses neural network in it part of the because, we train data sets, evaluate its performance and then compare these values with other values that are kept in the data base. If the comparison matches, we select the image for the as the output image and these image will be displayed in a small message box. This comparison and recognition is done through 4 distinct algorithms and to find out which algorithm works the best to achieve the objective of road sign detection subject to speed and Robustness of the methods.

SANTRAUKA

Kaip pagrindinė priemonė darbe naudojamas SIFT algoritmas bei SVM ir HOG metodai. Įvertinama ir apdorojama vaizdų atributinė informacija, kuri vėliau lyginama su pagrindiniais vaizdų atributiniais duomenimis prieinamais atvirose duomenų bazėse. SIFT algoritmo atveju, parametrus atitinkantis vaizdas, vėliau konvertuojamas į dvejetainį, tai leidžia įvertinti paveikslėlio ribas. Toliau, panaudojant HOG funkciją, išgaunamos vaizdų savybes ir gauti duomenys palyginami naudojant SVM klasifikatorių, - taip atpažįstamas kelio eismo ženklas konkrečiame įvesties vaizdų sraute. Tiek Gabor filtras tiek „blob“ analizė paremta neuroniniais tinklais, - palyginimo parametrai gaunami apmokymo principu, pateikiant lyginamus vaizdus, vėliau duomenų rinkiniai įvertinami palyginti jų reikšmes su saugomomis duomenų bazėje. Jei palyginimas atitinka pasirinktą įvertį, tai vaizdas laikomas atpažintu ir parodomas ekrane. Vaizdų palyginimas ir atpažinimas yra realizuoti naudojant 4 skirtingus algoritmus siekiant išsiaiškinti, kuris algoritmas veikia geriausiai siekiant aptikti kelio ženklus.

Contents

LIST OF FIGURES:

LIST OF ABBREVIATIONS

S.NO	ABBREVIATED WORD	ABBREVIATION
1.	Scale Invariant Feature Transform	SIFT
2.	Support Vector Machine	SVM
3.	Speeded Up Robust Features	SURF
4.	Histogram Of Oriented Gradient	HOG
5.	Discrete Wavelet Transform	DWT
6.	Discrete Cosine Transform	DCT
7.	Red Green Blue	RGB
8.	Direct Current	DC
9.	Alternating Current	AC
10.	Cellular Automata	CA
11.	Coefficient Vertical	CV
12.	Coefficient Horizontal	CH
13.	Coefficient Diagonal	CD
14.	Range Perturbation Function	RPF

1 INTRODUCTION

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 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
2) SVM and HOG
3) GABOR and NEURAL NETWORK
4) BLOB ANALYSIS NEURAL NETWORK

2 AIM AND TASKS

The aim is to compare four distinct algorithms SIFT, SVM and HOG, GABOR, BLOB ANALYSIS to find which algorithm works the best to detect road sign in terms of speed and robustness of the method. The objectives are formulated in the following manner:

- To review similar systems of road sign detection.

- To make implementations of chosen algorithms.

- Compare four distinct algorithms; SIFT, SVM & HOG, GABOR and BLOB. And compare with foreseen parameters.

3 ANALYSIS OF EXISTING SYSTEM

3.1 LITERATURE SURVEY

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

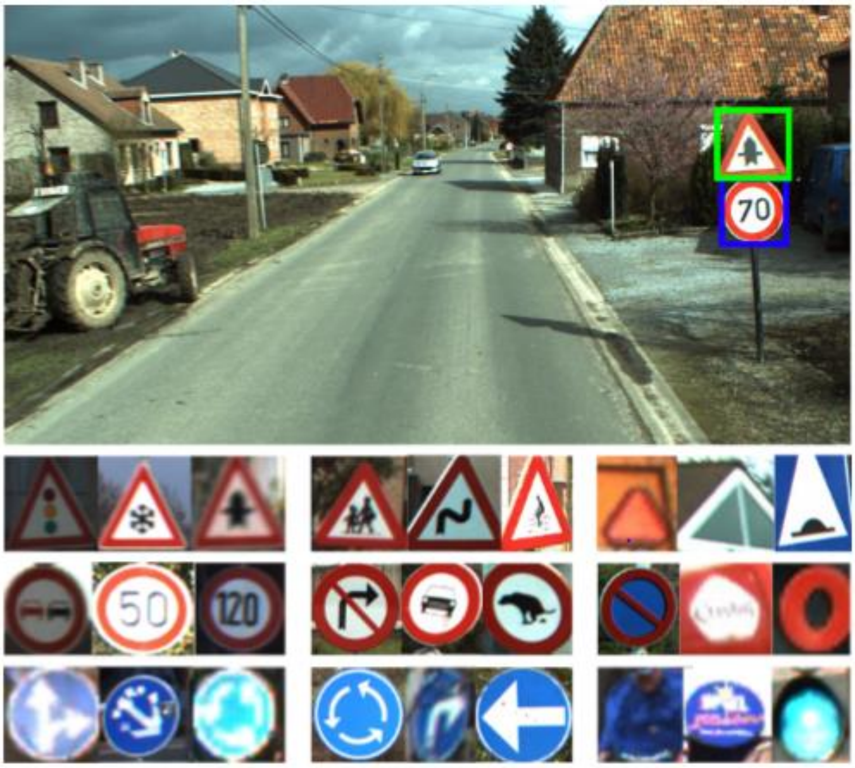


Figure 1: Traffic Sign and Recognition [1]

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.

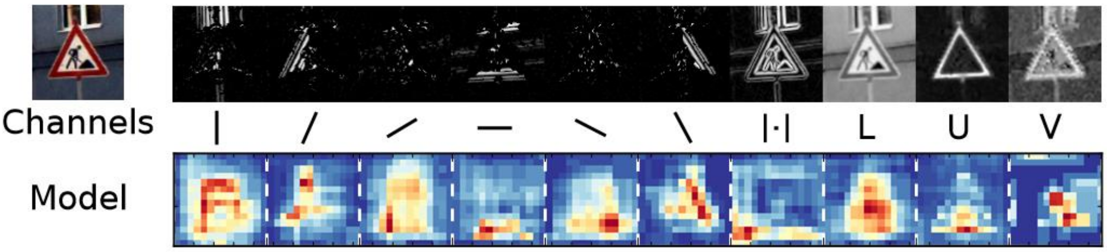


Figure 2: Training Of Channels [1]

The features picked up as in the training of channels for the unsafe zone had channel divisions as depicted in the fig. The authors 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%. The experimental that was implemented was the following:

S. Sathiya et al. [2], had studied a methodology for traffic sign detection which primarily involved recognition from an Indian signage. The acquired image was scaled to size that pertained to the best-fit for general analysis. The technique that the authors 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 dilation. In the secondary step the detected sign was worked upon with the concept of row and column count. The third step which in other words was quite imperative was data extraction using DCT, DWT and Hybrid DCT-DWT. These are subjected to feature extractions, 20 highest energy coefficient 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.

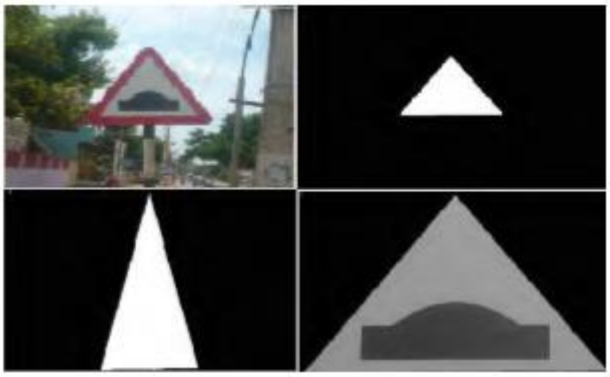


Figure 3: Detection Module [2]

The above figure depicts a traffic sign board extraction with original image which in the next step is perceived as a road sign and this is worked upon the concept of row-wise segmented imaging which is followed by column wise segmented imaging. The basic desideratum was to proliferate improve efficiency of transportation as for differently abled home. The algorithm was defined as pertaining to state-of-the-art subject to various levels of the interruptions. The general efficiency was estimated as 86 % in DCT, 90% in DWT and 96% in Hybrid DWT-DCT and SVM.

M. Blauth et al. [3], studied the traffic signage recognition with capacity of detecting and classifying German traffic signs that were 43 in number. A two stage approach which distinguishes distinctly between detection and classification that were separated. This methodology was a high performance system and neat scalability. The authors used a fast shape detection method which was fundamentally based on multi block binary patterns (MB-LBP) which displayed good results for face detection schemes. In addition to the shape detection, colour segmentation was used in this methodology. This proliferated the affine invariance properties and the whole-inclusive detection rate of the system. For the purpose of classification HOG features and non-linear support vector machines were put in practice.

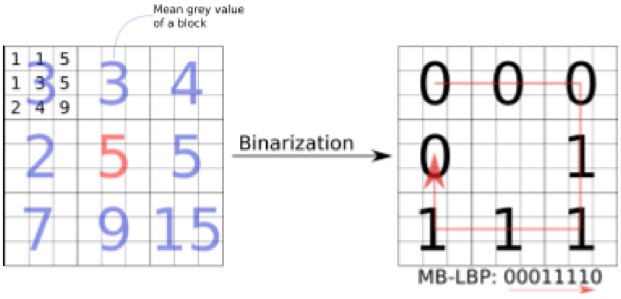


Figure 4: Shape Based Detection [3]

The shape based detection used the process of binarization which converted image outlines in the figures into binary digits 0's and 1's. Typically in the previous step the affine normalization of segmented imaging is done which is basically carried out to precisely classify an object. In other words the region that was tilted geometrically could be read as an image that was nominal. The binarization process typically surrounds the mean grey values which is in relation to the middle block of the image. A compact descriptor is the result with threshold and parity functions. The classifier was rated at 96.93% on German Traffic Sign Recognition Benchmark, an apex-scoring result. And 108 traffic images given by a street inspection company were evaluated by virtue of bounding boxes and centred signs and 97% of traffic signs were recognized accurately with low intensity of false positive rate. The authors recommend quite highly of its application in real-time.

S.L. Arroyo [4] studied 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 had developed an artificial vision system which recognized traffic signs considering geometric shapes in general even though there was poor illumination in general.

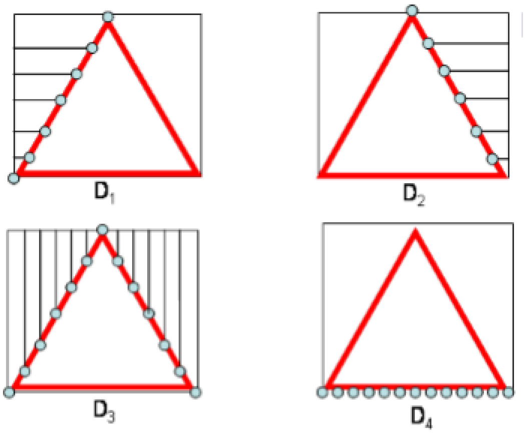


Figure 5: Extraction of Vectors [4]

The technique proposed and developed was termed as a Support Vector Machine abbreviated as an SVM which involved distance to border technique (DtB). The principle of SVM was that it generated vector patterns which defined the distance to the borders of the traffic signage. It has been shown that the methodology and implementation is quite effective. There were basically six types of configurations that were involved in the study and they were different signs, different positions, partial occlusions, rotational angle, and different signs and deteriorated signage.

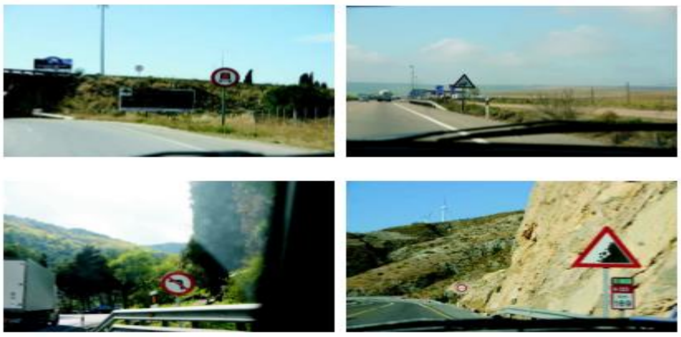


Figure 6: Different Signs and Different positions [4]

It was concluded that by virtue of this literature a new methodology had been defined and was particularly useful in defining and classifying shapes of traffic signage. Four vector signs are used in pattern generation thus reducing overhead costs. The system was found be robust tested under various conditions of reference. But there was one drawback, which were false alarms. This is reported to under further investigation for feasible results.

3.2 TRAFFIC SIGN RECOGNITION FOR INTELLIGENT VEHICLE USING NEURAL NETWORK ON 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 other words means improving comfort and safety in general. This method proposed by A. Salhi et al [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.

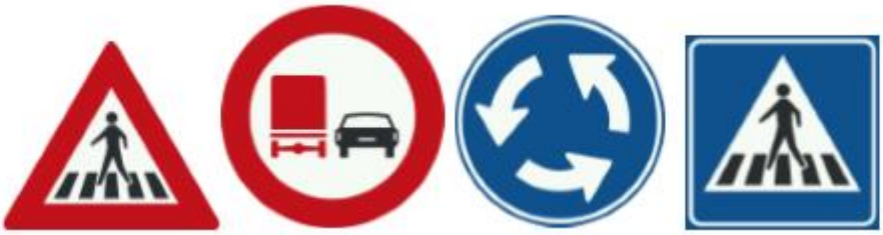


Figure 7: Panels Considered In This Work [5]

The system used a traffic signs detection module and by virtue of chain codes the components were digitally 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, thresholder masks blue and red in colour are depicted below.

Figure 8: BGR Color Image [5]

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 well possible as the authors suggest.

3.3 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) was used by C.C. Chen et al. [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.

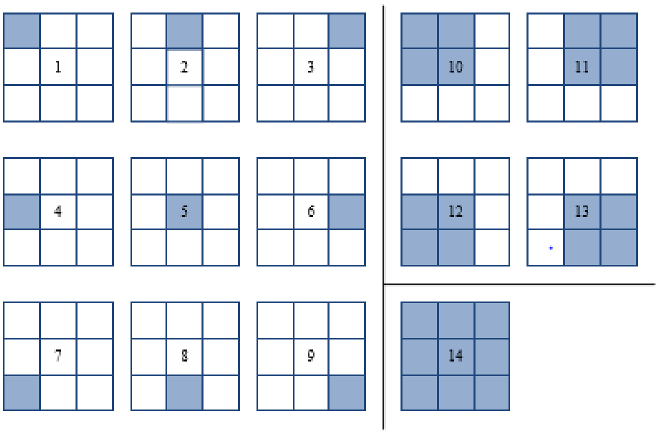


Figure 9: Sub-blocks of the Image [8]

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.

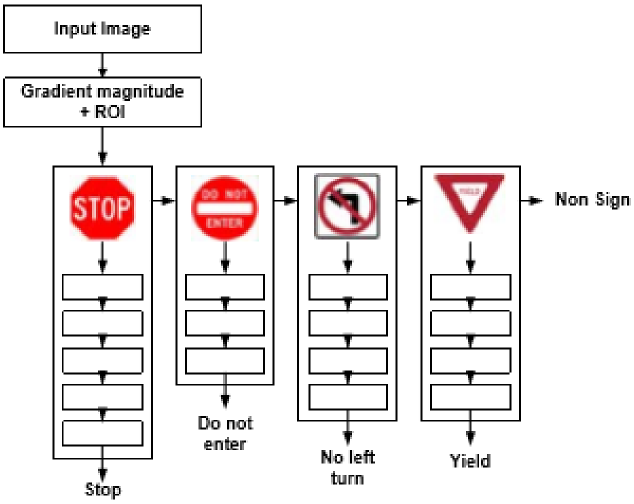
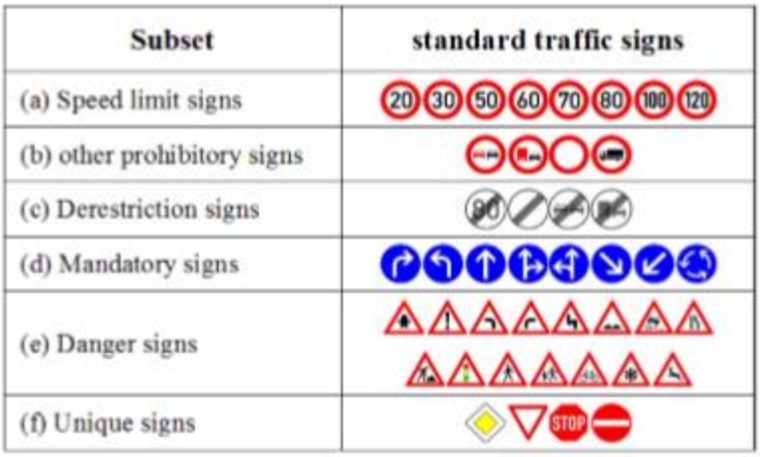


Figure 10: Cascade Sign Detectors [8]

The authors 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 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]

3.4 MULTI-FEATURE AND MULTI-CLASSIFIER FUSION:

A swift and robust method for traffic sign recognition which typically used a range from course to fine strategy is presented in this case. Y. Ma and L [9]. Huang studied the Gaber 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. An illustration of this is shown below in fig.



Subset	standard traffic signs
(a) Speed limit signs	20 30 50 60 70 80 100 120
(b) other prohibitory signs	
(c) Derestriction signs	80
(d) Mandatory signs	
(e) Danger signs	
(f) Unique signs	STOP

Figure 11: German Traffic Signs [9]

Real and imaginary parts of Gabor filters used for orientations that were four in number and on two scales are depicted below. Further subdividing the case, the part on the left represents the real part and one on the right depicts the imaginary part. Under the process of classification the pre-processing, complementariness with local features, local binary pattern and multi-feature fusion were used.

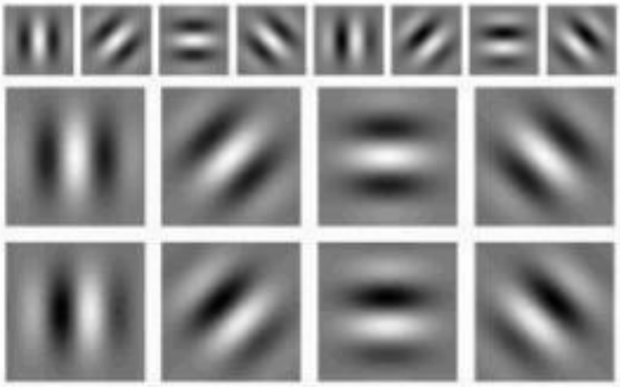


Figure 12: Gabor Filters with 4 Orientations and 2 Scales [9]

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]

4 METHODOLOGY

4.1 SCALE INVARIANT FEATURE TRANSFORM

. Scale invariant feature transform 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's 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.

4.1.1 SCALE SPACE DETECTION

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 points of interest that stays constant to both the scale and orientation.

4.1.2 KEYPOINT LOCALIZATION

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.

4.1.3 ORIENTATION ASSIGNMENT

Multiple orientations will be allotted to every key point locations which is related to the local image gradient directions. Then, all the further operations are performed on image data, which are converted related to their orientation, actual location of every key points, scale, hence giving out difference to all the other transformations.

4.1.4 KEYPOINT DESCRIPTOR

The image gradients at every local points are estimated at the intended scale in the region around every key points. These are then converted to a depict that allows for all significant levels of variation in local illumination and distortion of shape.

Finally bounding box method is used to showcase 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]

4.2 SUPPORT VECTOR MACHINE CLASSIFICATION AND WATERSHED

The identification of a traffic sign from various set of traffic sign with the needed knowledge like their structure and brightness or intensity value. Traffic sign detection technique was found for tedious task 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:

$$I=0.2989* R + 0.5870* G + 0.1140* B\text{-----------} \quad (1)$$

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.

The canny will view the lines in the high contrast image. These lines will not quite delineate the outline of the object of interest. When compared to the original signal, the gaps in these lines surrounding the object in canny can be viewed. The linear gaps, later disappears only if the image is dilated using vertical line.

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 is still appears. To reduce such noises from occurring, the use hole filling algorithm is implemented. 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.

The watershed segmentation is an extremely useful technique that is able to divide regions that is able to divide regions that are generated by overlapped object. The main issue associated with the use of watershed transform is the identification of the local minima in the distance map, which is required to guide the segmentation process. An efficient implementation of this segmentation approach involves the sorting of values in the distance map and the generation of disjoints regions using marker-controlled flood-fill operations. The candidate traffic signs are detached by computing the logical AND operation between the watershed lines and the filled candidate b Morphological watersheds provide a complementary approach to the segmentation of objects. It is especially useful segmentation object that are touching one another. To understand the watershed transform view a grayscale images as a topological surface, where the values of f(x, y) correspond to height.

Consider the topographic surface the 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 algorithms then find the catchment basins and the ridge lines in the image.

The algorithm works in this way suppose a hole is punched at each regional local minimum and the entire topography is flooded from below by letting the water rise through the holes at a uniform rate. Pixel below the water levels incrementally, the flooded. When we raise the water level incrementally, the flooded regions will grow in size. Eventually, the water will rise to level where two flooded regions from separate catchment basin will merge. When this occurs, the algorithm constructs a one-pixel thick dam that separates the two regions. The flooding continues until the entire image is segmented into separate catchment basins divided by watershed ridge lines. So for this using inbuilt mat lab command bwdist this computes the distance transforms a supposed to have an images like 1’s and 0’s basically works on binary images it try to find out distance transform of binary images is the distance from every pixel to the nearest nonzero-valued pixel.

The watershed algorithm is implemented in MATLAB image processing toolbox as L=watershed (f), where f is the input image and L is labeled 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 analyses and recognizes patterns, used for classification and regression analysis. SVM classified as either sign of interest or background. [3]

4.3 GABOR FILTER AND NEWFF

The basic function of filters in Gabor family 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 banks. 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 patches consists of its sub windows. The Gabor windows are then subjected to each and every sub windows separately. The magnitudes of this filter response are later obtained from every sub window and are imposed by three moments namely, mean, standard deviation and skewness. Then the filter bank is applied to every sub windows individually, yielding certain size of feature vectors.

‘newff’ is a tool used in MATLAB for accessing neural network and 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. This function can be triggered by using ‘nsstart’ in the command window. The syntax for this function in MATLAB is given by:

```
feedforwardnet(hiddensizes,trainFcn)
```

where Hidden sizes = Row vector of one or more hidden layer sizes

trainFn = train function (by default = ‘trainlm’)

And then returns a feedforward neural network [11]

4.3 BLOB ANALYSIS:

In this method, the main aim is to detect the regions in any digital image, which has a varying properties, 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 doesn’t change at all. Therefore giving us ide that all the points in blob to be similar to each other.

When some property of liking is expressed as a function on the image, it is classified under two min classes of blob detection methods. They are:

A) Differential methods

B) Methods based on local extreme.

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

As per the study, there is a good reason about the regions, and they are bot obtained y edge detectors or corner detectors. 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's are trained, there performances are evaluated. Some of the needed information are 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 out code. [5]

5 IMPLEMENTATION OF PROPOSED TECHNOLOGY

5.1 SIFT FEATURES

The implementation of this technique consists of 6 stages and the stages are given below:

A) Loading input image

B) Comparing input image with the reference image in the database.

C) Detection of sift features (key points)

D) Extraction of key features.

E) Matching the extracted features.

F) Bounding box detection.

These above mentioned stages are given below as the block diagram. The detailed description of these stages will be described in this chapter.

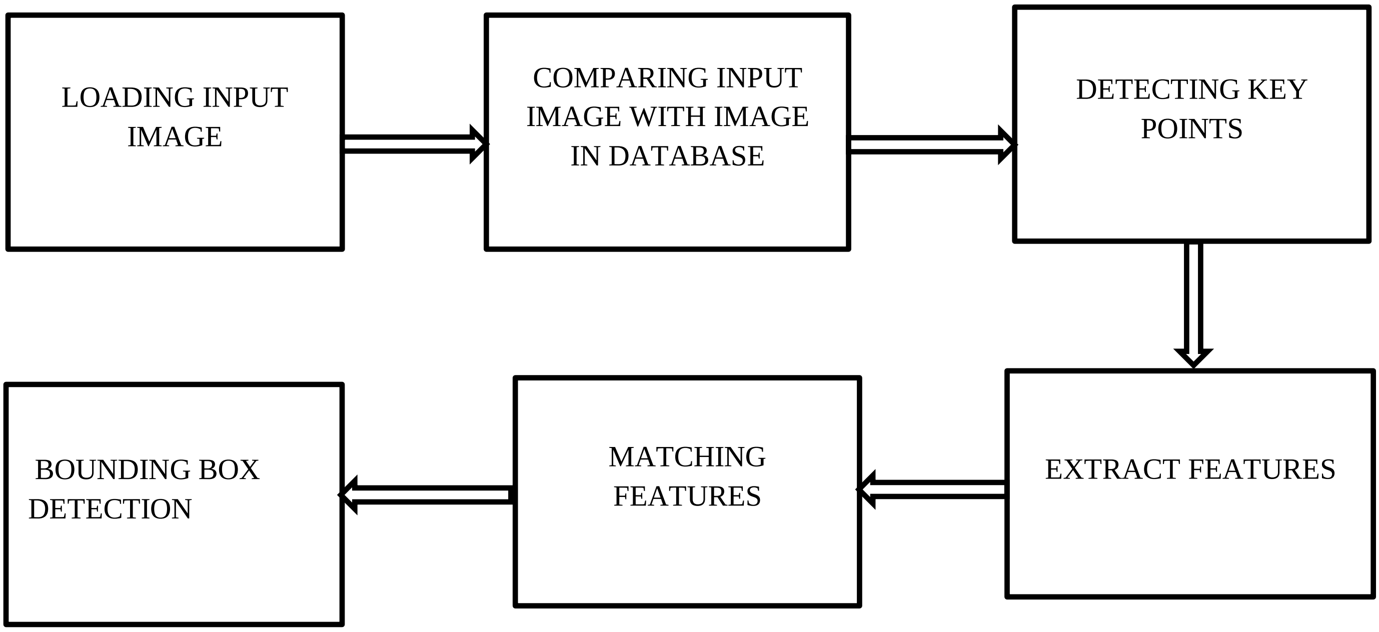


FIG 13: BLOCK DIAGRAM FOR SIFT FEATURES

5.1.1 LOADING INPUT IMAGE FILE

The first step in the implementation phase begins with the selection of image which will be loaded to the program using the function 'uigetfile'. 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 using the function 'imread' and then stored to the file as 'img'. 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 using the function 'imshow' and can be renamed to desired title using the function 'title'.

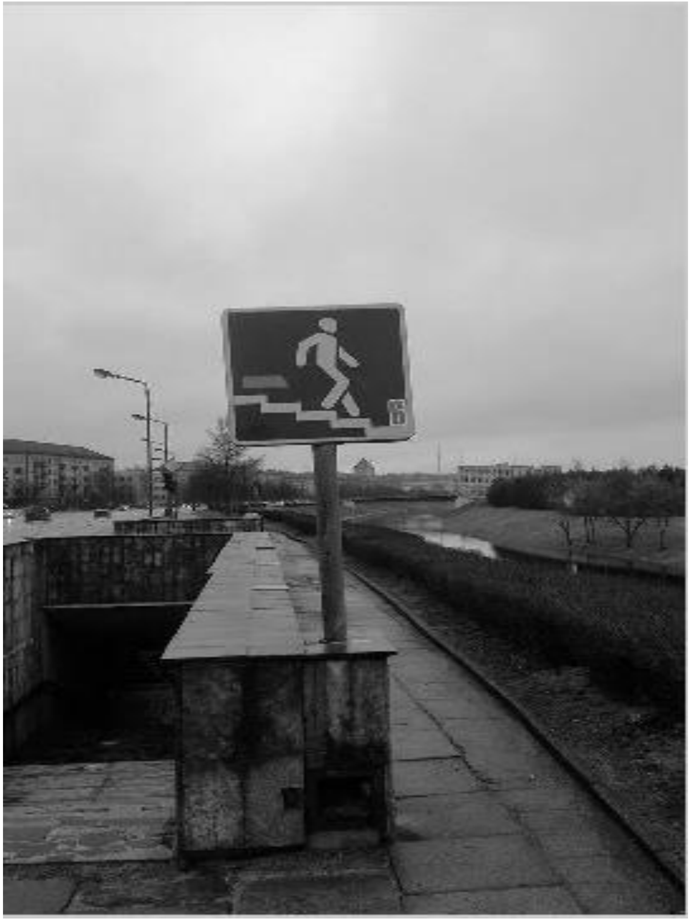


Figure 14: INPUT IMAGE OF SIFT FEATURES

Later, the image from the database is been read into the program and saved to the file as img3. 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. This image is later viewed and is renamed as 'Image Scene'.

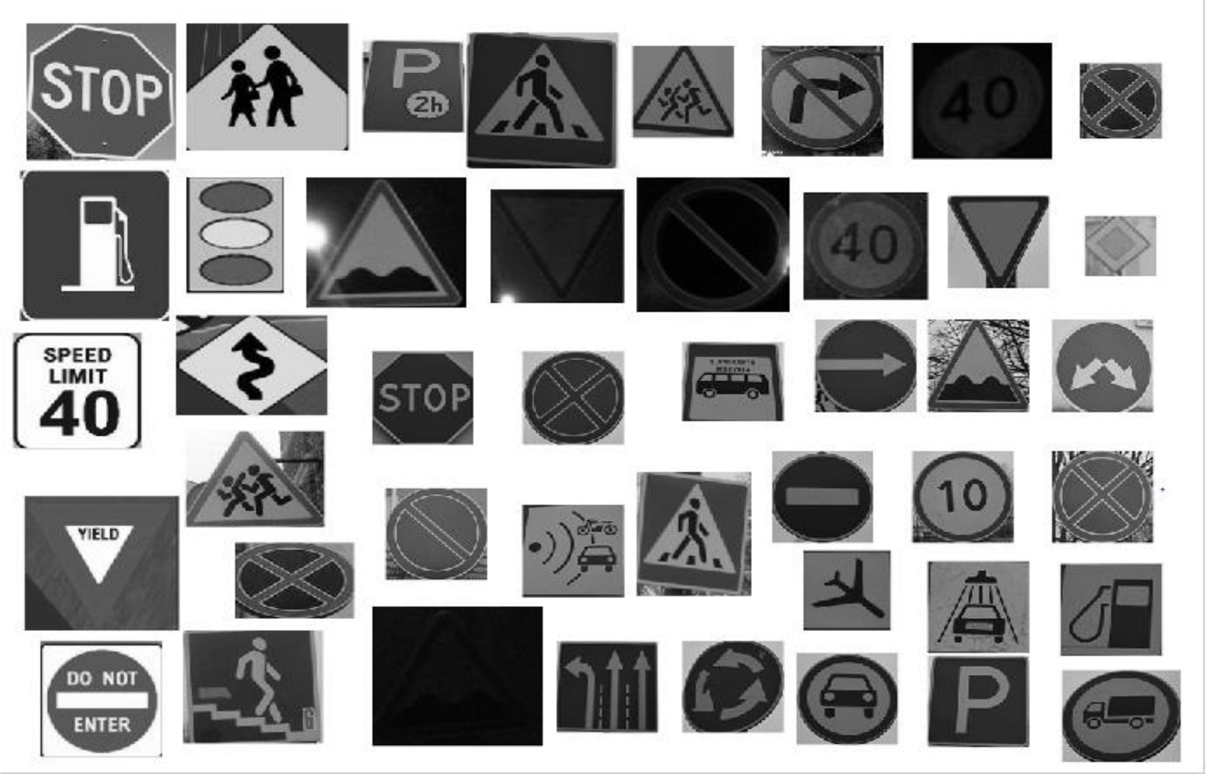


Figure 15: IMAGE SCENE

5.1.2 COMPARING THE INPUT IMAGE WITH IMAGE IN DATABASE

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.

5.1.3 DETECTING SIFT FEATURES

Once the images are loaded and read, the next stage for the comparison begins with the feature detection phase. 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’

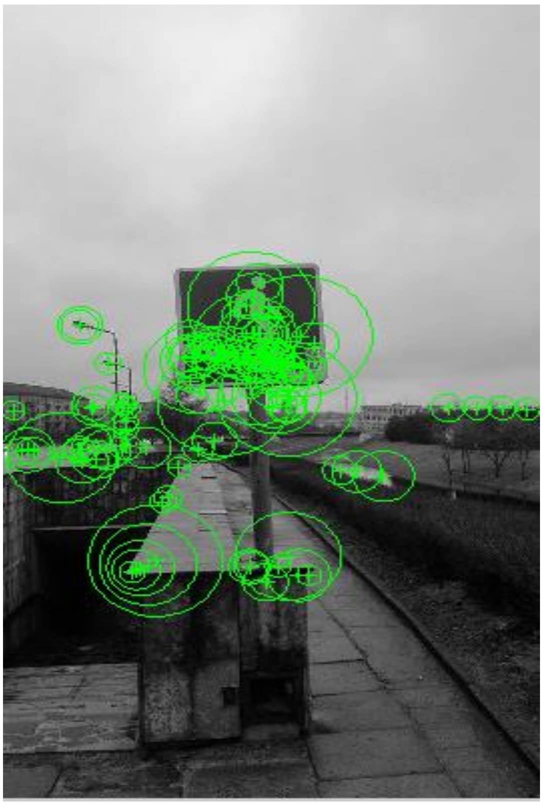


Figure 16:100 STRONGEST FEATURE FROM BOX IMAGE

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.

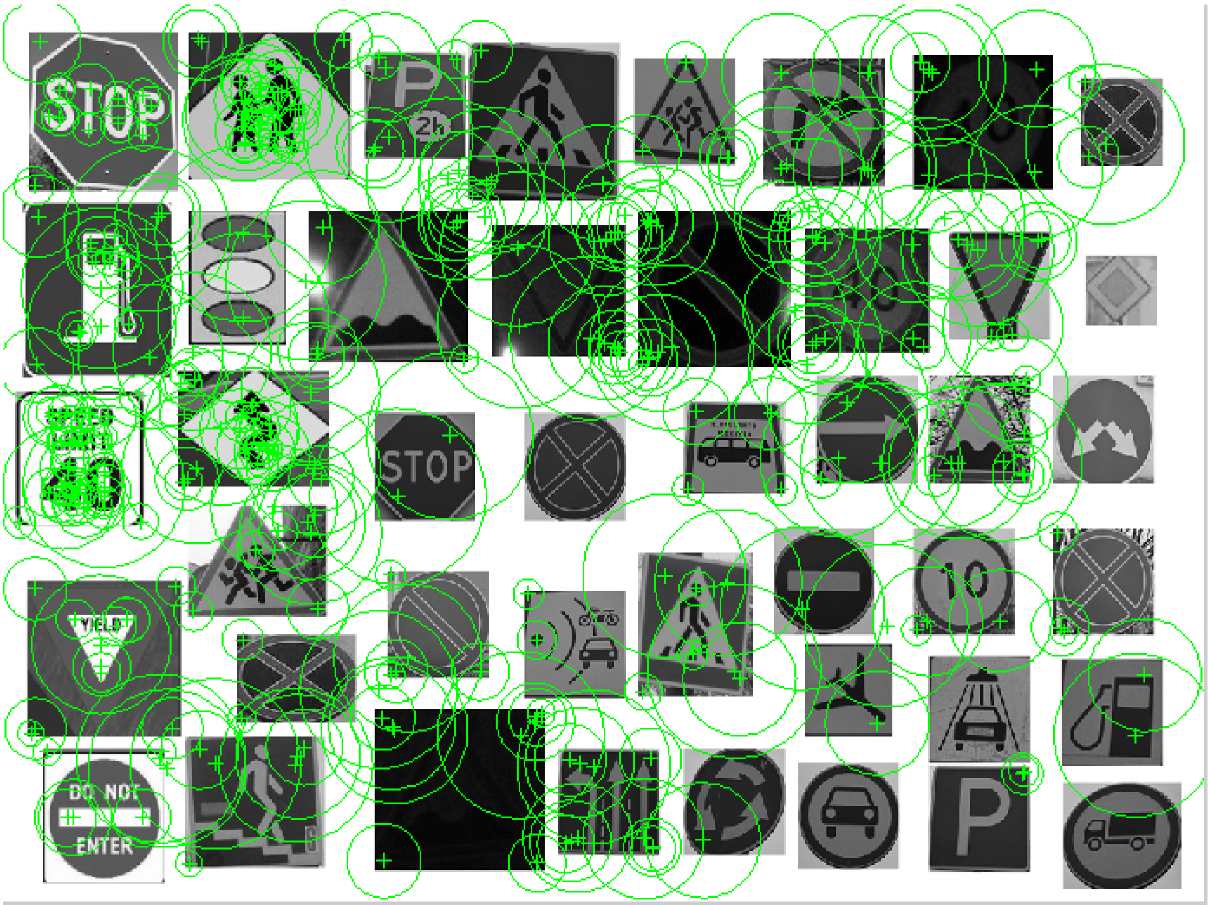


Figure 17:300 STRONGEST FEATURE POINTS FROM SCENE IMAGE

5.1.4 FEATURE EXTRACTION

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 with the file name ‘img’ 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 ‘img1’, where the ‘img1’ 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’.

5.1.5 MATCH 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 in ‘boxPairs’. The matched features are viewed using the function ‘showMatchedFeatures’, 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).

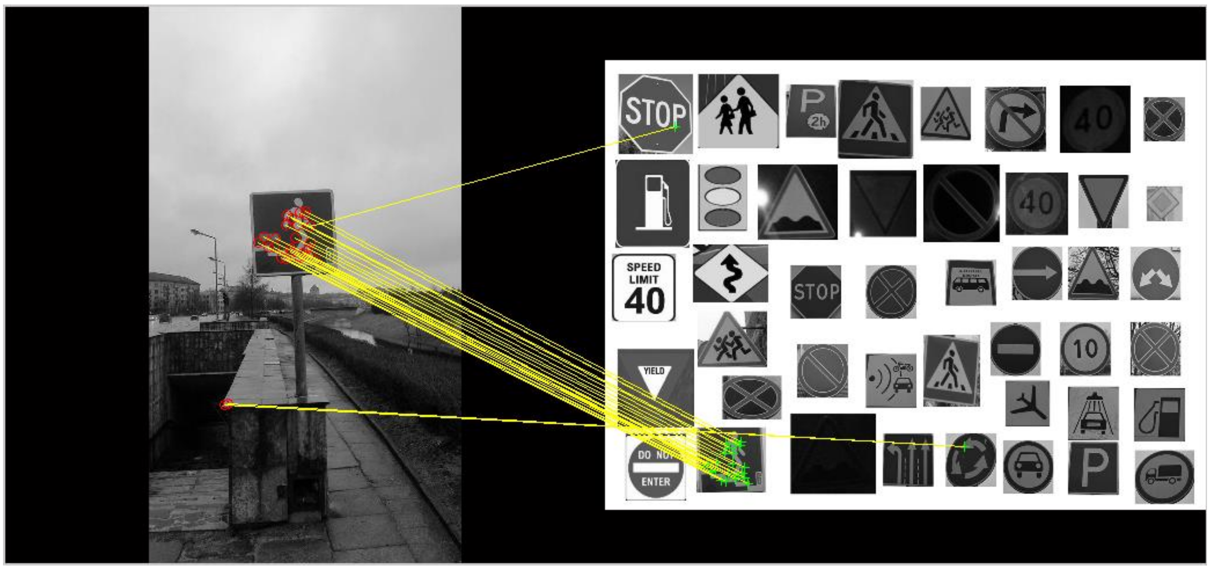


Figure 18: MATCHED POINTS (INCLUDING OUTLIERS)

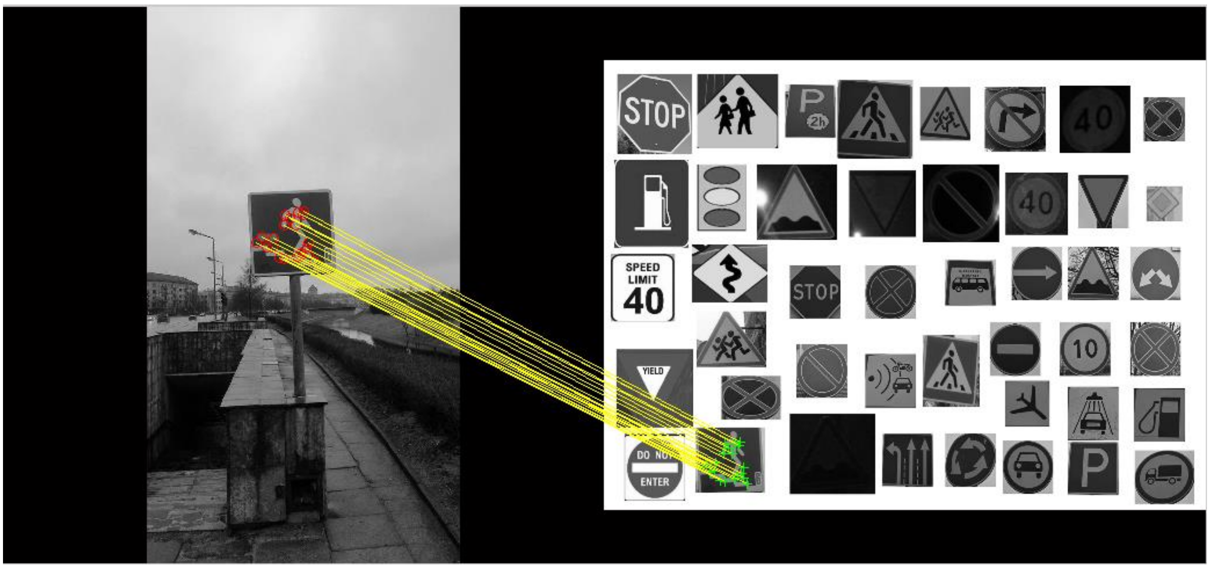


Figure 19: MATCHED POINTS (INLIERS ONLY)

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.

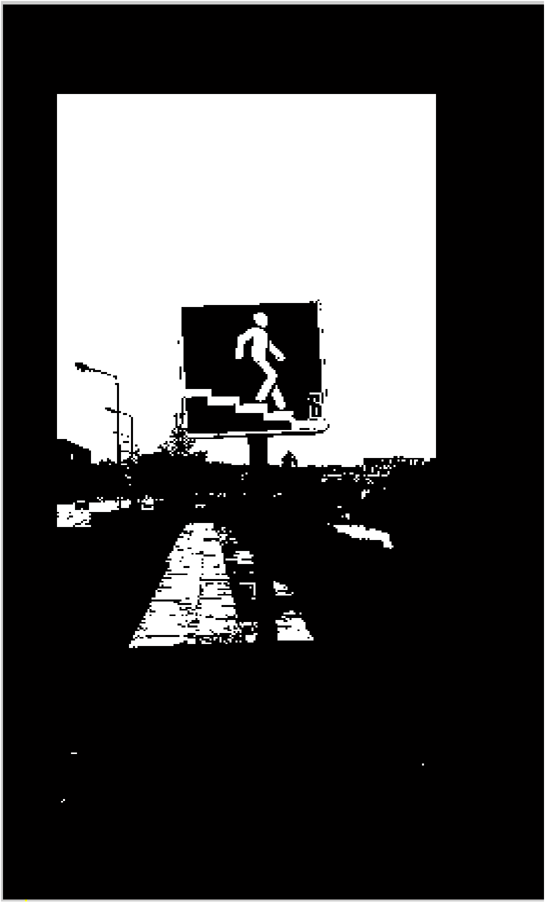


Figure 20: BINARY IMAGE

5.1.6 BOUNDING BOX DETECTION

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 ‘regionprops’, 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’.

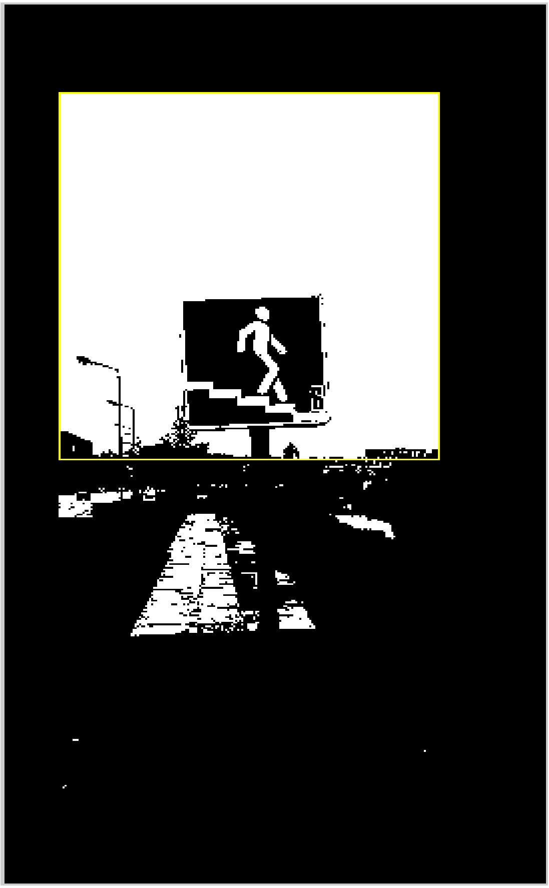


Figure 21: BOUNDING BOX IMAGE

5.2 DETECTION AND EXTRACTION OF TRAFFIC SIGN USING HOG FEATURE EXTRACTION AND SVM CLASSIFICATION

In this method, the implementation phase consists of 3 stages namely:

A) Detection module

B) Feature extraction using HOG

C) SVM classification

The detection modules have 6 sub-stages and they are enclosed below:

A.1) loading input image

A.2) conversion to gray scale image

A.3) Edge detection of the image

A.4) Dilated image

A.5) Hole filled image

A.6) Smoothening of image

Detection Module

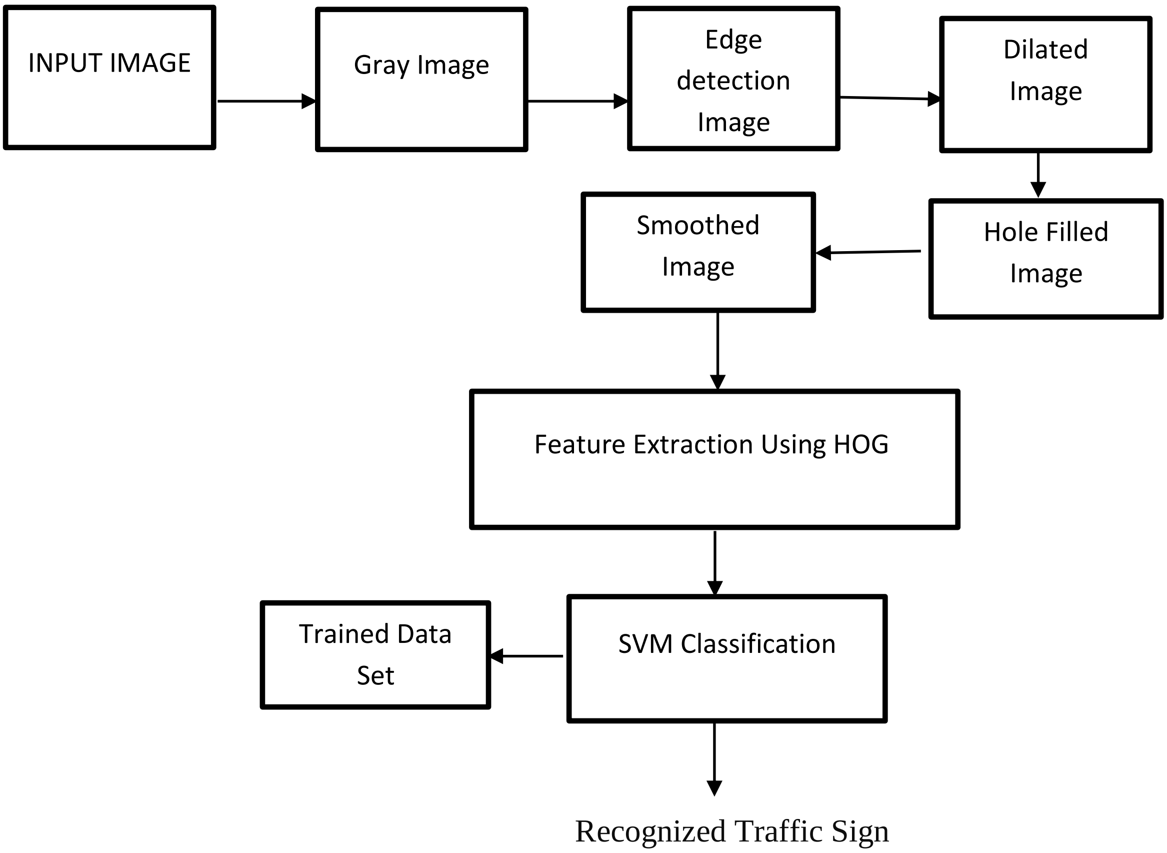


Figure22: Block diagram SVM CLASSIFICATION AND HOG FEATURES

The description about these phases will be discussed in this chapter, based on the implementation using MATLAB.

5.2.1 DETECTION MODULE

5.2.1.1 LOADING INPUT IMAGE

We begin the implementation generally by loading the desired image file into MATLAB program destination. This can be done with the function ‘uigetfile’ and 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 in the program using the function ‘imread’ and it is stored to the name ‘img’. This image is then converted from RGB into gray scale and saved as ‘img1’. The converted file is later resized to 512 X 512 using function, ‘imresize’ and is stored as ‘img2’. This image is the viewed using the function ‘imshow’ and is titled as ‘input image’ of the program.

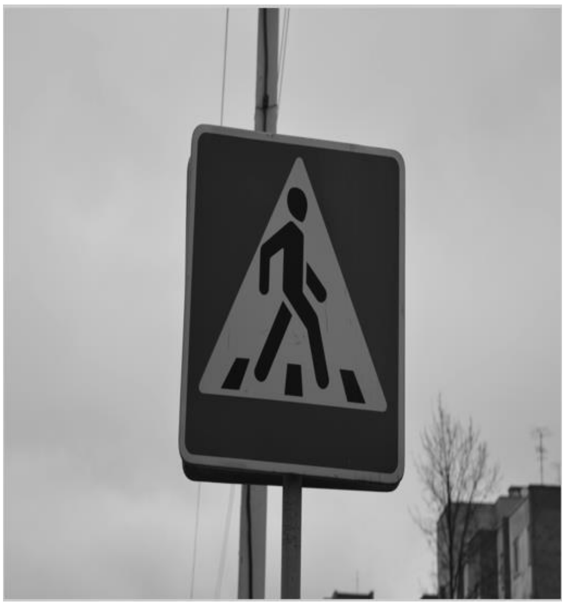


Figure 22: Input Image of SVM classification

5.2.1.2 EDGE IMPLEMENTATION

In this phase, we detect the border of the dominant part of the image. This means, the online of any object, which is present in the image. This is done my calling the function ‘edge’ on the image ‘img2’. This image is put into the file named ‘BW2’. Later, this image is displayed and renamed as ‘Edge detection’ of the selected image.

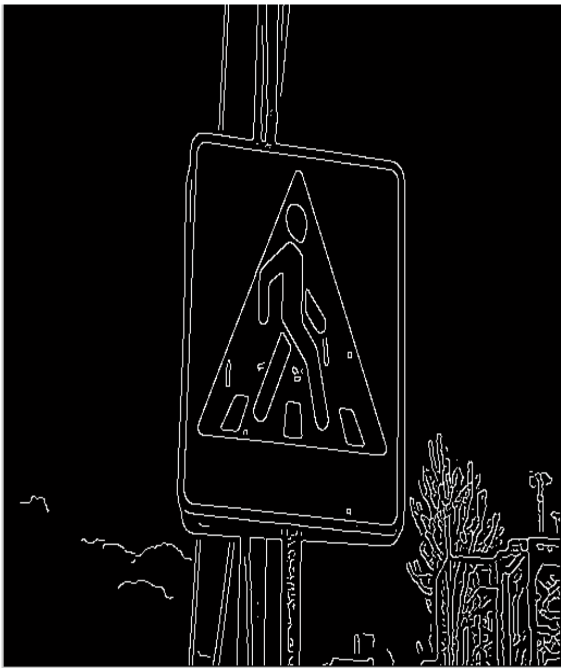


Figure 23: Edge Detection

5.2.1.3 DILATED 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, the function ‘imdilate’ is called to dilate the image ‘img2’ along with the elements in the flat structure and this dilated image is moved into the file called ‘Iobrd’. This image is then viewed and is entitled as ‘Image dilation’ in the source code. The picture of the dilated image is given below.

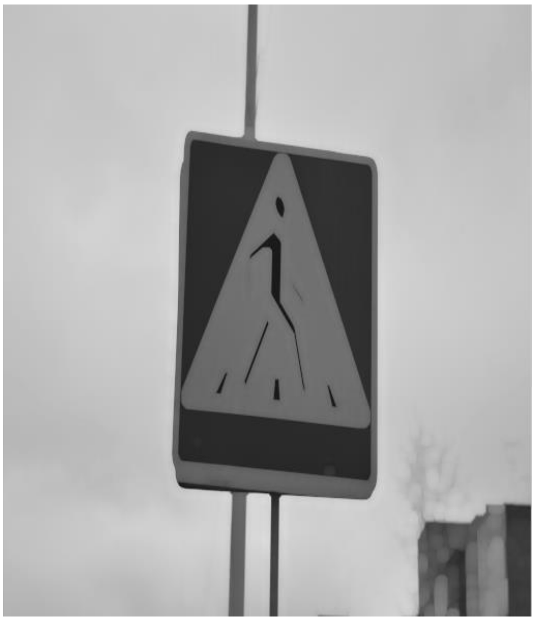


Figure 24: Dilated Image

5.2.1.4 IMAGE HOLE

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 using the function ‘imfill’ 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’.

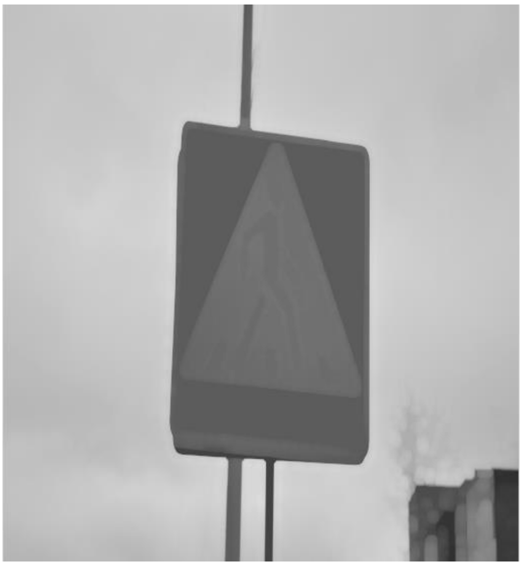


Figure 25: Image Hole

5.2.1.4 SMOOTH 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. This can be implemented using the function ‘medfilt2’on the file ‘BW3’. The smoothened image is then stored to the file ‘BW4’ and displayed with the title ‘Smooth Image’

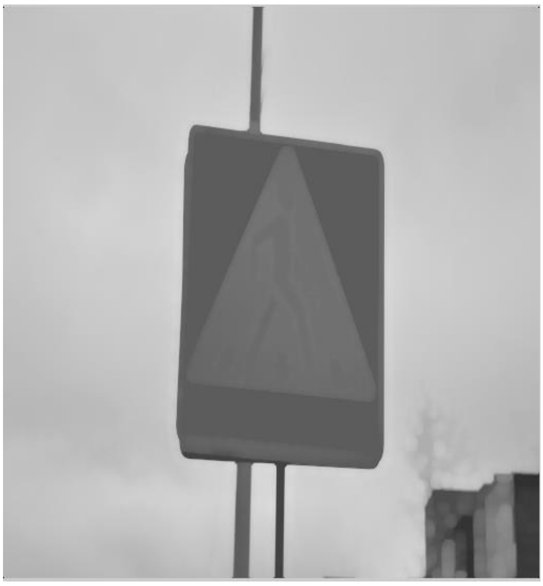


Figure 26: Smooth image

5.2.1.5 WATERSHED 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 file name ‘L1’ and entitled and ‘Watershed image’.

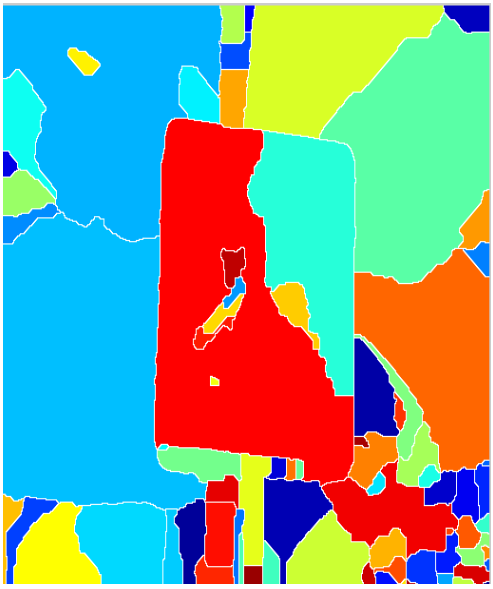


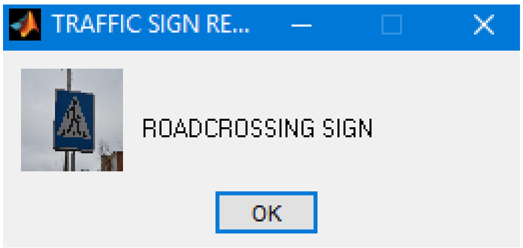
Figure 27: Watershed Image

5.2.2 HOG FEATURES EXTRACTION

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 to the file ‘m’. 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’.

4.2.3 SVM CLASSIFICATION

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 under the column E. 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.



ROADCROSSING SIGN

Figure 28: Result of HOG and SVM classification

5.3 GABOR NEURAL NETWORK ALGORITHM:

The implementation of this algorithm is carried out in 4 stages and they are listed below:

A) Loading input image
B) Creating Gabor filter Bank
C) Extraction of Gabor features
D) Creation of Feed forward propagation network

5.3.1 LOADING INPUT IMAGE

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 under the name 'img'. The image is the resized to the desired resolution. The image is then displayed under the title 'input image'. 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/

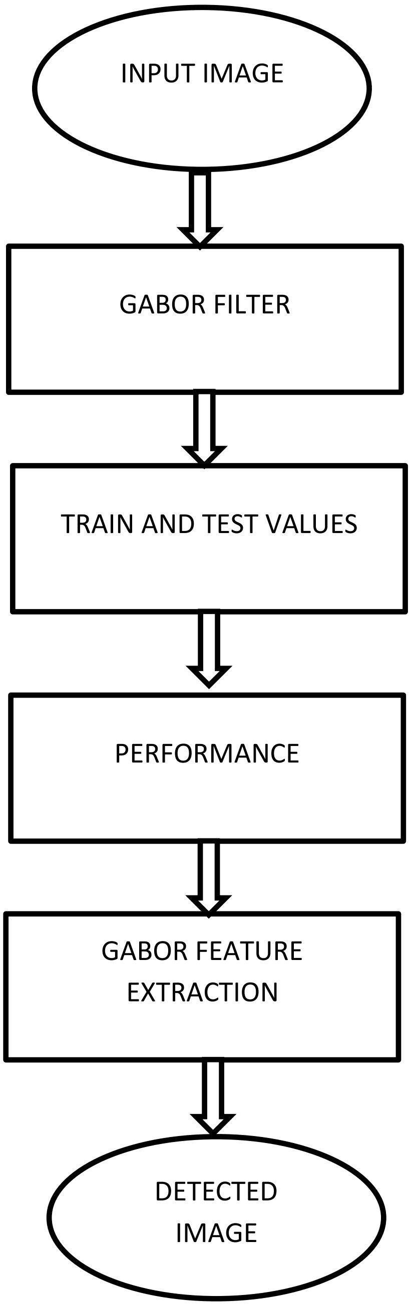


Fig 29: BLOCK DIAGRAM FOR GABOR FILTER

5.3.2 CREATING GABOR FILTER

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.

5.3.3 EXTRACTION OF GABOR FEATURE

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 ‘featureExtractor’. 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.

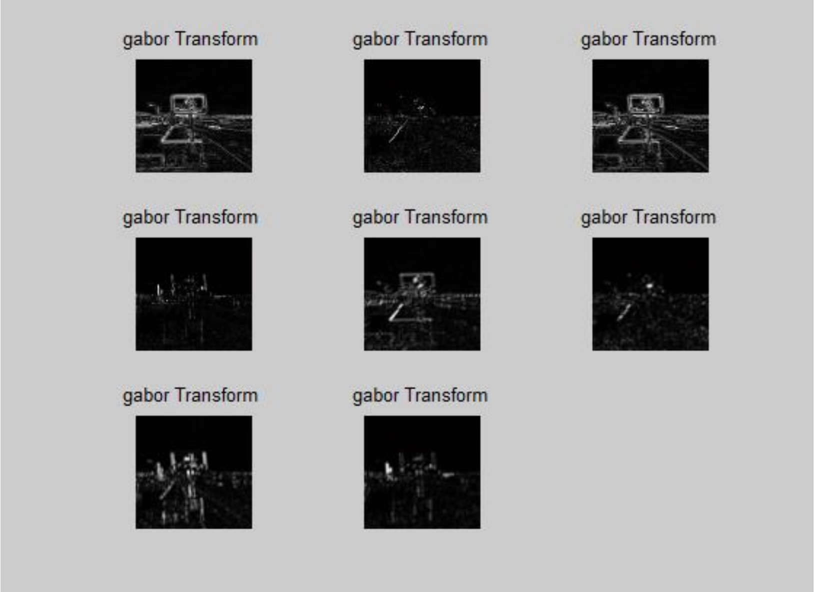
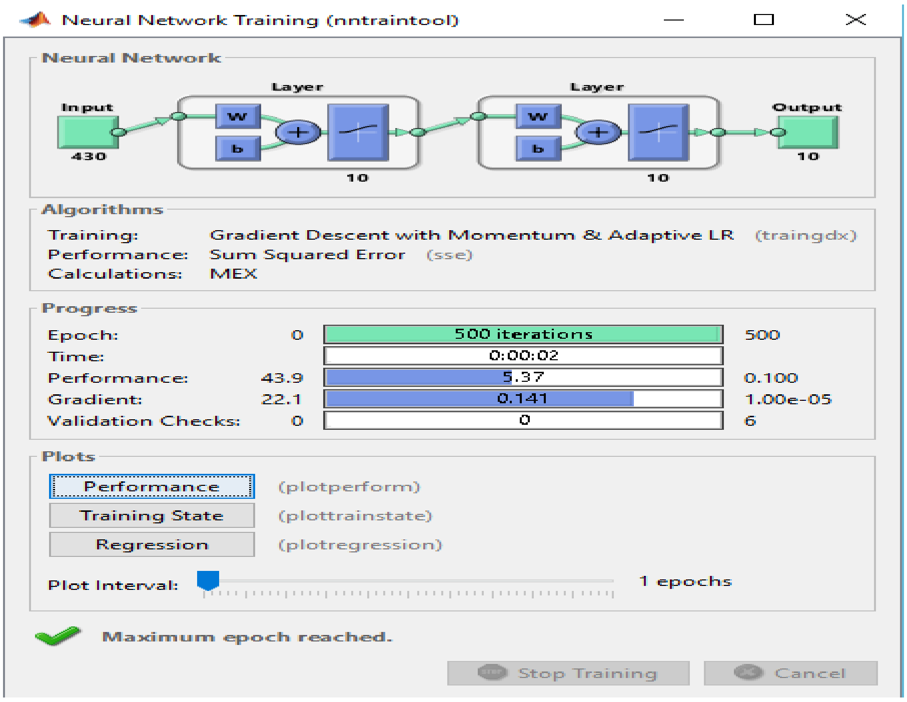


Figure 30: Gabor Transform

5.3.3 NEURAL NETWORK TRAINING TOOL

In this stage, we create a neural network function training tool, which is used to train the data's and the 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 place in an excel sheet under the name Book1.xlxs. 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. All these are stored under the file name of variable name net. A maximum function is taken and the values of P, are later outputted through the code. 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 multiplation.



Algorithms

Training: Gradient Descent with Momentum & Adaptive LR (traingdx)
Performance: Sum Squared Error (sse)
Calculations: MEX

Progress

```
Epoch:              0     500 iterations     500
Time:                     0:00:02
Performance:        43.9  5.37               0.100
Gradient:           22.1  0.141              1.00e-05
Validation Checks:  0     0                  6
```

Plots

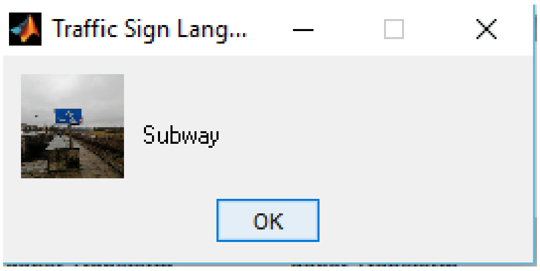
Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)

Plot Interval: 1 epochs

Maximum epoch reached.

Figure 31: Neural Network Training Tool

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 used for all the images and their values pasted in the satisfying condition.



Traffic Sign Lang...

Subway

Figure 32: Recognized Output

5.4 IMPLEMENTATION OF BLOB ANALYSIS AND NUERAL NETWORK

The implementation of this algorithm generally consists of 8 stages, namely:

A) Loading of input image.
B) Binary image processing
C) Morphological operator.
D) Area calculation
E) Perimeter
F) standard deviation
G) Normalized deviation of radial distance.
H) Compactness

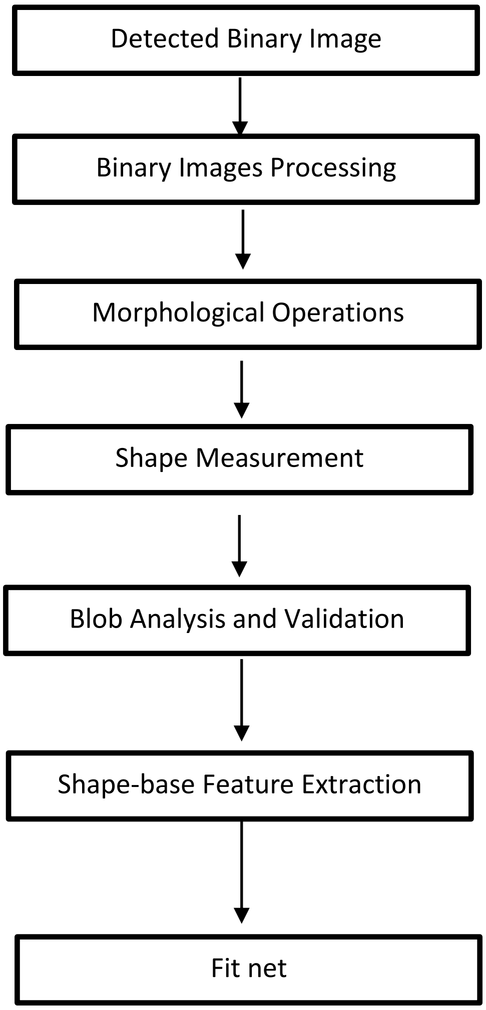


FIG 33: BLOB ANALYSIS AND NUERAL NETWORK

5.4.1 LOADING OF INPUT IMAGE.

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 image is later viewed and titled as input image.

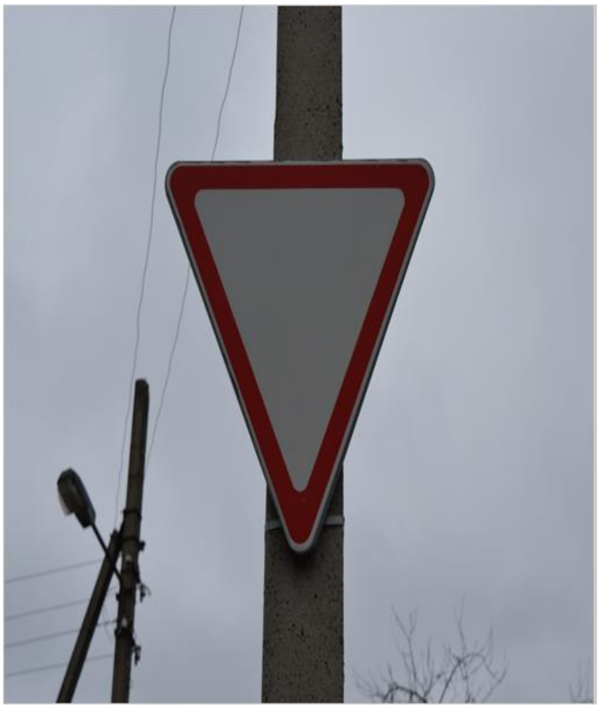


Fig: 34 Input Image of Blob Analysis

5.4.2 BINARY IMAGE PROCESSING

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’.

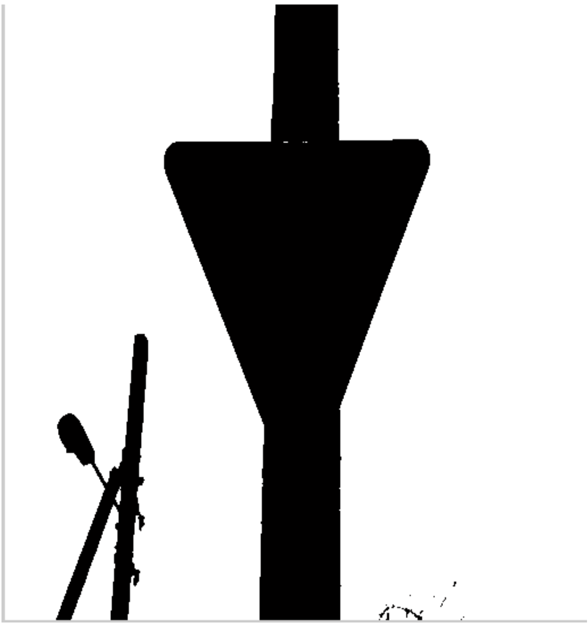


Fig: 35 Binary image processing

5.4.3 MORPHOLOGICAL OPERATOR

In this phase, the binary image undergoes morphological operation, using the function 'bwmorph' and the value is moved on to the file 'bw2'. This figure is then viewed as 'morphological operation'. The image 'img1' 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.

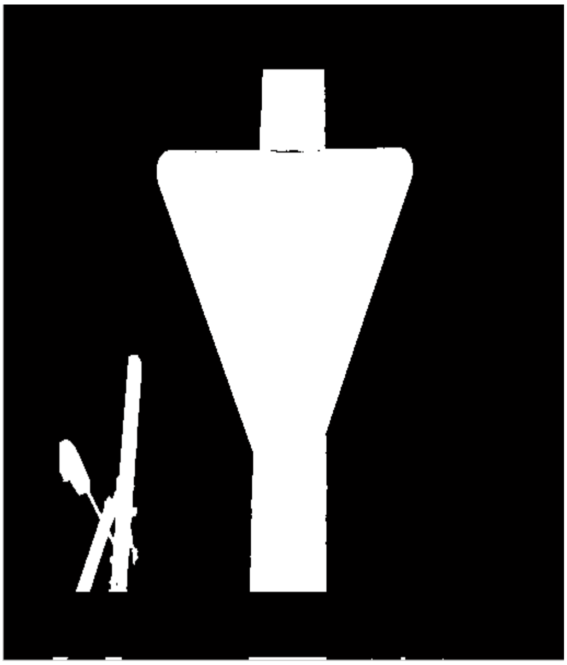


Fig: 36 Morphological operator

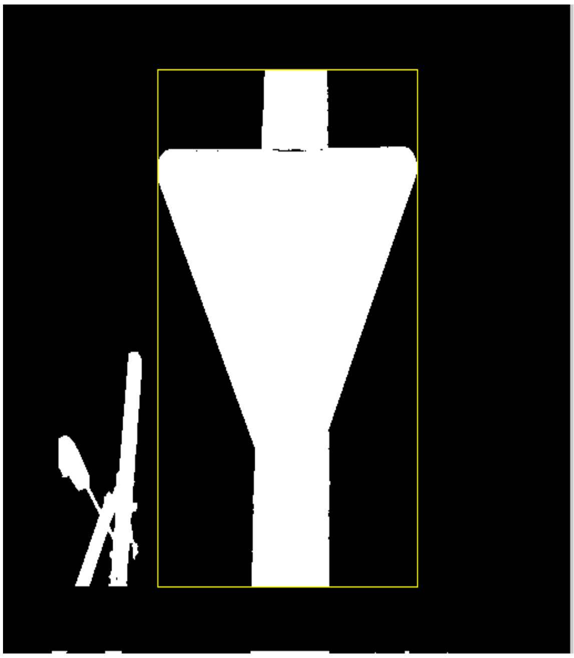


Fig: 37 Shape Measurements

5.4.4 AREA CALCULATION

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. Then the area is estimated using the function ‘cat’ and we find the indication area.

5.4.5 ESTIMATIING PERIMETER

The net stage emphases on estimation of perimeter. The size of the re-scaled image is determined and its perimeter is estimated using the function ‘bwperim’. 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. Finally, the estimated value of perimeter is the taken square root and is saved as D.

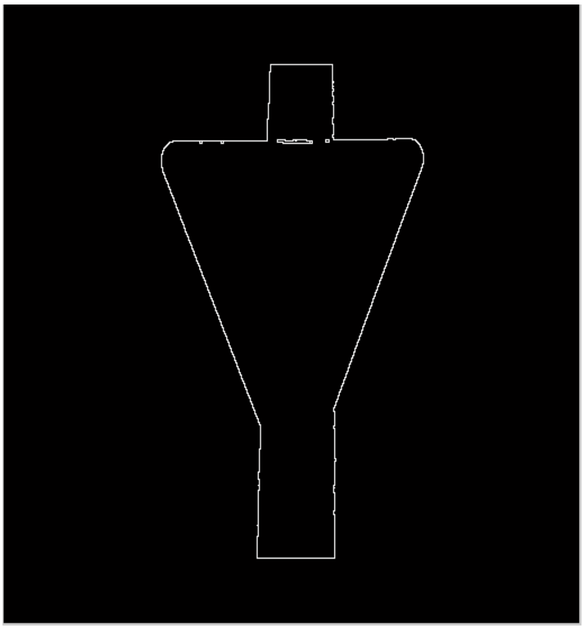


Figure 38.Perimeter

5.4.6 ESTIMATING AREA

The area of the image is obtained using the function ‘bwarea’ and the values are stored to ‘A’. The value of the area function is then viewed using disp.

5.4.7 ESTIMATING PERIMETER

The area is divided by diameter to obtain the perimeter value. This values is stored as ‘S’ and is displayed as perimeter.

5.4.8 STANDARD DEVIATION

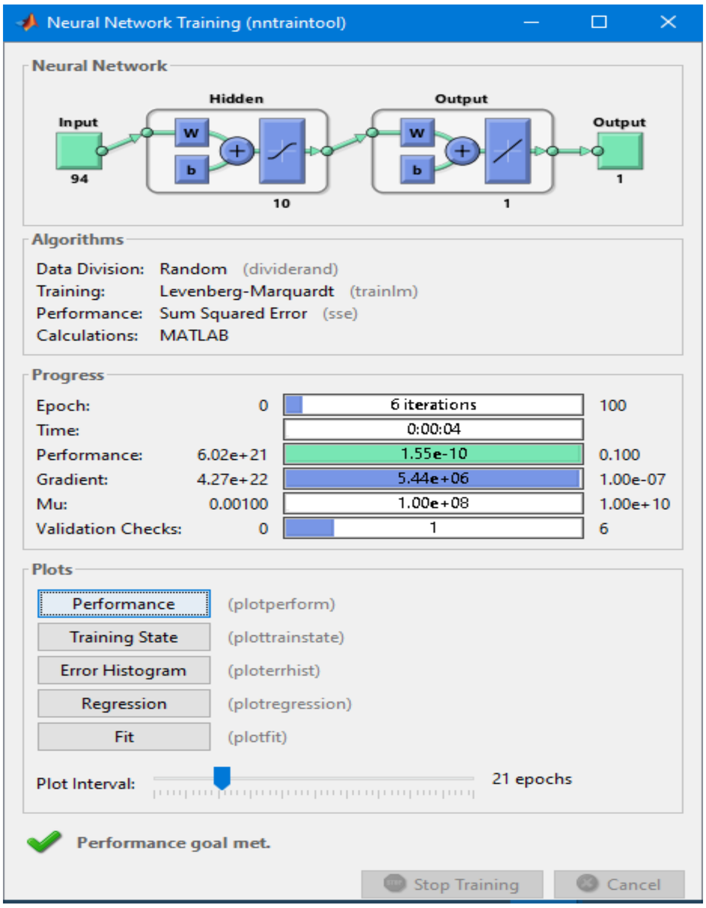
The standard deviation is then estimated at this phase, where the function ‘std2’ is called on the scaled image ‘bw3’. The standard deviation is then displayed.

5.4.9 NORMALIZED DERIVATION OF RADIAL DISTANCE

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.

5.4.10 COMPACTNESS

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.


Neural Network Training (nntraintool)
Neural Network
Input
94
Hidden
W
b
10
Output
W
b
1
Output
1

Algorithms

Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Sum Squared Error (sse)
Calculations: MATLAB

Progress

Epoch:	0	6 iterations	100
Time:		0:00:04	
Performance:	6.02e+21	1.55e-10	0.100
Gradient:	4.27e+22	5.44e+06	1.00e-07
Mu:	0.00100	1.00e+08	1.00e+10
Validation Checks:	0	1	6

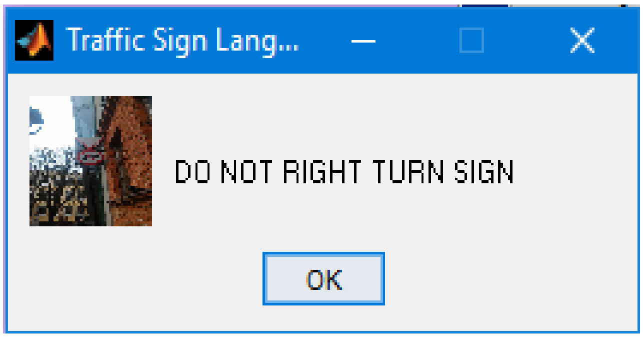
Plots

Performance (plotperform)
Training State (plottrainstate)
Error Histogram (ploterrhist)
Regression (plotregression)
Fit (plotfit)

Plot Interval: 21 epochs

Performance goal met.

Fig: 39 Neural Network Training Tool(newff)



Traffic Sign Lang...

DO NOT RIGHT TURN SIGN

Fig: 40 Recognized Output for Blob Analysis & Fitnet

5.5 COMPARISION 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.

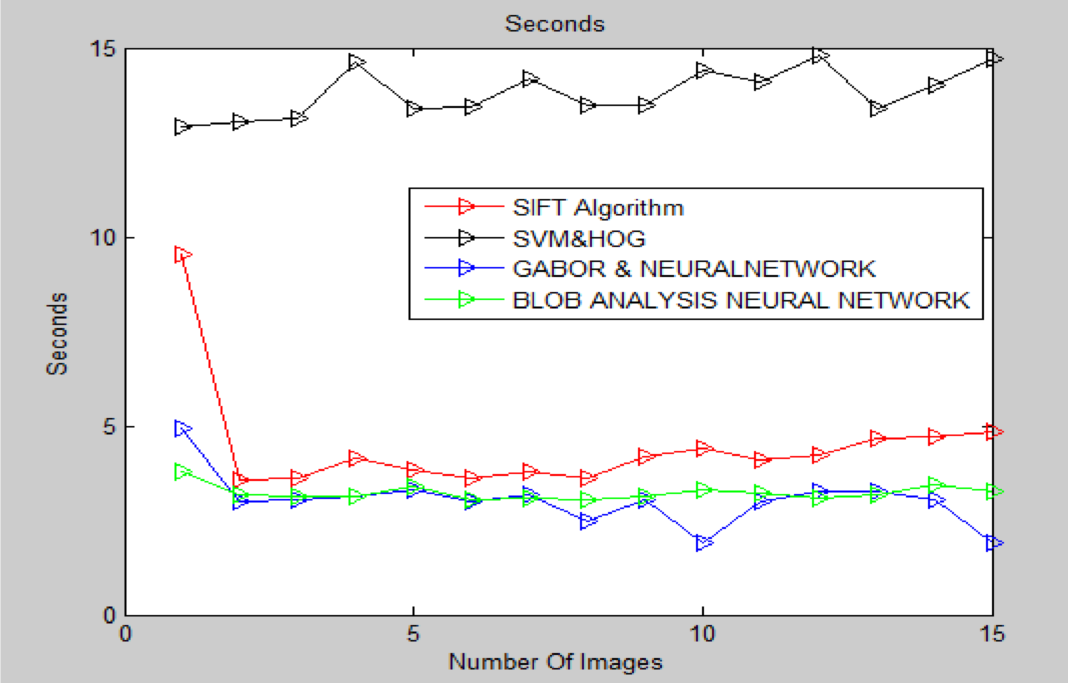


Fig: 41 Graph for speed

In figure 42, 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) SVM and HOG = 93.3%
3) GABOR and Neural Network = 93.3%
4) SIFT Algorithm = 83.3%

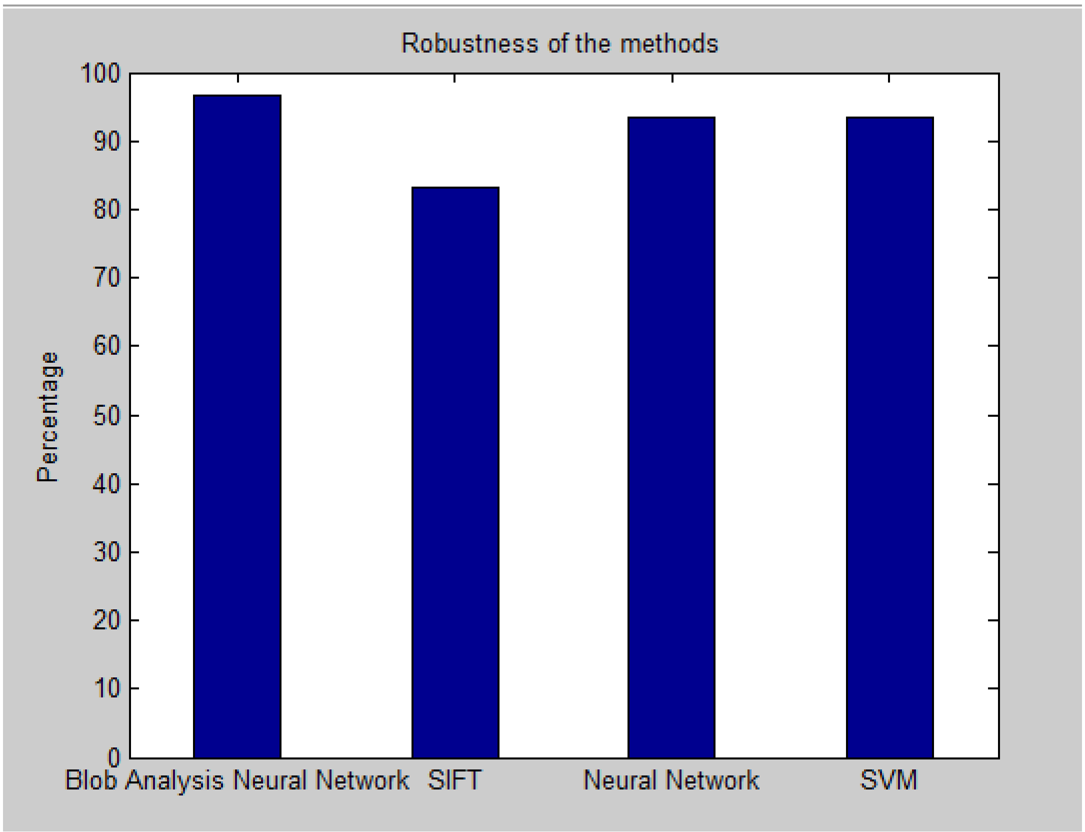


Fig: 42 Robustness of the methods

6. FUTURE PERSPECTIVE AND SCOPE

1. To extend the principle of *performance-efficiency* to algorithms SIFT and SVM & HOG, and thus understand the net performance characteristics of the four algorithms (BLOB, SIFT, GABOR and SVM & HOG) to extend its implications in the future.

2. To study the *accuracy* and *quality characteristics* of the four algorithms (BLOB, SIFT, GABOR and SVM & HOG)

7. CONCLUSION

1. The four distinct algorithms (BLOB, SIFT, GABOR and SVM & HOG) were compared and studied and the following conclusion is derived

1a) 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.

1c) Comparing the above conclusions it is observed that BLOB algorithm was the best-fit followed by SIFT algorithm

2. 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%).

3. In this we proposed FITNET function in blob analysis. So, it gives best result compare to other algorithms.

8. REFERENCE

[1] Markus mathia, Radu Timofte, Rodrigo Benenson and Luc van gool, Traffic Sign Recognition.

[2] S.Sathiya, M. Balasubramanian, S.palanivel, Pattern Recognition Based Detection Recognition Of Traffic Sign Using SVM, International Journal Of Engineering and Technology(IJET)

[3] M.blauth, E. Kraft, F. Hirschenberger, M.bohm, Large-Scale Traffic Sign Recognition Based On Local Features And Color Segmentation.

[4] S. Lafuente-Arroyo, P. Gil-Jimenez, R. Maldonado- Bascon, Traffic Sign Shape Classification Evaluation I: SVM Using Distance To Borders.

[5] Abderrahim SALHI , Brahim MINAOUI, Mohamed FAKIR, Robust Automatic Traffic Signs Recognition Using Fast Polugonal Approximation Of Digital Curves And Neural Network, International Journal Of Advanced Computer Science and Applications.

[6] Auranuch Lorsakul and Jackrit 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.

[7] Chia-Hssiung Chen, Marcus Chen and Tianshi Gao, Detction and Recognition of Alert Traffic Signs.

[8] Shouyi Yin, Peng Ouyang, Leibo Liu, Yike Guo,Fast Traffic Sign Recognition With A Rotation Invariant Binary Pattern Based Feature,ISSN 1424-8220 Jounal/Sensors.

[9] Yunxiang Ma, Linlin Huang, Hierarchical Traffic Sign Recognition Based on Multi-Feature and Multi-Classifier Fusion, First International Conference On Information Science and Electronic Technology (ISET 2015).

[10] Zhiyong Huang, Yuanlong Yu, Jason Gu and Huaping Liu, An Efficient Method For Traffic Sign Recogntion Based on Extreme Learning Machine, IEEE TRANSACTIONS ON CYBERNETICS.

[11] Zehang Sun, George Bebis and Ronald Miller, ON-Road Vehicle Dtection Using Gabor Filetrs And Support Vector Machines, Computer Vision Laboratory Department of Computer Science, University of Nevada, Ford Motor Company, Dearborn, MI.

[12] Mohammad A.N. AL-Azawi,Neural Network Based Automatic Traffic Signs Recognition, International Journal of Digital Information and Wireless Communications(IJDIWC):753-766.

[13] Xiaoguang HU, Xinyan ZH, Deren LI,Hui LI, Traffic Sign Recognition Using Scale Invariant Feature Transform And SVM.