SPEED LIMIT SIGN BOARD DETECTION & EXTRACTION OF DIGITS IN DIFFERENT WEATHER CONDITIONS

BY

KHAN HAFIZUR RAHMAN (09101029)

MD. ZUBAIR ALAM (09101012)

SUPPORTING BODY:

DR. MD. KHALILUR RAHMAN (SUPERVISOR)
RUBEL BISWAS (CO-SUPERVISOR)

DECLARATION

We hereby declare that this thesis is based on the results found by myself. Materials of work found by other researcher are mentioned by reference. This thesis, neither in whole nor in part, has been previously submitted for any degree.

SIGNATURE OF AUTHOR

KHAN HAFIZUR RAHMAN (ID: 09101029, DEPT: CSE)

MD. ZUBAIR ALAM (ID: 09101012, DEPT: CSE)

SIGNATURE OF SUPPORTING BODY:

SUPERVISOR:

Dr. Md. KHALILUR RAHMAN

DOCTOR OF INFORMATION ENGINEERING

ASSISTANT PROFESSOR,

BRAC UNIVERSITY, DHAKA

PHONE: +880-2-8824051, ext: 4066

CO-SUPERVISOR

RUBEL BISWAS

LECTURER-II

BRAC UNIVERSITY, DHAKA

ACKNOWLEDGEMENT

First of all we would like to thank our Almighty "Allah" for everything.

Next we would like to thank from the deepest corner of our heart to those people who helped us at every steps of our thesis with ideas and blessings. From them we would firstly like to thank our supervisor Dr. Md. Khalilur Rahman & co-supervisor Rubel Biswas for all their supports and advices throughout the thesis.

We also owe our deepest gratitude to Md. Zahangir Alam, lecturer of Brac University, whose guidance, encouragement and support from the initial level till the end helped us to complete the thesis project and also inspired us for further research.

We would like to give special thanks to our faculties, families and friends for all their continuous supports.

Index

ABSTRACT

The goal of this paper is to explore how to find the prohibitory speed detection board automatically by doing image processing. After detection, it detects the numerical portion from the board by OCR operation and informs the vehicles about the maximum speed allowed for that particular road. In this paper, we present a new modular traffic signs recognition system, successfully applied to all part of the world including country in Asia as well as Europe. Our sign detection step is based only on shape-detection (circle). We try to extract the red portion from the image then using Hough transformation (HT) we detect the circle from the image avoiding all other unnecessary information from the image. This system able to detect board in all situation like in bad lighting, noisy images, blurred images, dawn and dusk and also in different environment like foggy condition, snow fall area, sunny bright images with a very high detection rate. Speed sign candidates are classified by segmenting potential digits and then applying neural digit recognition.

INTRODUCTION:-

Intelligence transportation system (ITS) is now demand of the time. As the number of vehicles increases, the probability of traffic accident also increases. There are many reasons to occur traffic accident. Among them, over-speed is one of the main issues. To avoid the accident, there are different prohibitory signs are displayed. These are designed to show us some rule or warn us to avoid accidents. Unfortunately, sometimes we miss these signs due to lack of concentration or lack of knowledge about the prohibitory signs. Sometimes we also don't obey the rules; we drive the vehicles according to our own desire. Leaving these signs or to fulfill our desire we put ourselves into a very danger situation, become a participant in car accident. These accidents are well capable enough to take thousand people live in one moment. Therefore, if there is a system which can compel us to follow these prohibitory signs; we may avoid these sorts of accident.

Traffic signs or road signs are signs erected at the side of or above roads to provide information to road users. From 1930, many countries adopted pictorial signs or otherwise simplified and standardized their signs to facilitate international travel where language difference would create barriers, and in general to help enhance traffic safety[1]. These signs are usually based on international protocols. These protocols are well agreed by the different country in the world [2]. According to protocol, regulatory traffic signs are generally circular and have a red border and black symbols or letters on a white background. Speed-limit boards also follow this rule shown in fig1.

(a) AUSTRALIA (b)NEDERLAND (c)MALAYSIA

(d) SWEEDEN (e) GERMANY (f) FRANCE

(g) UNITED KINGDOM (h) SPAIN (i) POLAND

(j) ITALY (k) AUSTRIA

FIG 1: speed limit board in different country

In recent years, traffic sign research recognition has grown rapidly because future vehicle system really needs for such systems. Performance indexes headed by these systems include high recognition rates, real-time implementation, many traffic sign categories as recognition objects, robustness for variant environments, and feasibility under poor visibility condition.

Colors represent an important part of the information provided to the driver to ensure the objectives of the road sign. Therefore, road signs and their colors are selected to be different from the nature or from the surrounding in order to be distinguishable. Detection of these signs from moving vehicle helps drivers to take the right decision in good time.

In this chapter, we are able to detect the red color easily. We innovate a new concept to find out the red colors from an image. We subtract original image from the red color plate. It

helps us to focus the red portion in the images. After finding the red color objects, we detect the circle using HT. From the circle we look for the numerical portion from the circular image.
We segment the circular numerical images to extract the individual number. These individual numbers are trained. Then any number are detected can be recognized with the help of trained numbers.

The second phase of our thesis is OCR that is "Optical Character Recognition". Optical Character Recognition, or OCR, is a technology that enables us to convert different types of documents, such as scanned paper documents, PDF files or images captured by a digital camera into editable and searchable data.

OCR is one of the leading research topics of today. It has become one of the important parts of many IT based research projects. We have also used it because after we have done with character segmentation it's time to recognize the digits. This is because only after we successfully identify the digits from the prohibitory speed limit sign board only then we can take the necessary actions and decisions to control the speed of the vehicles.

In the next chapter, we will try to do with the video streams so that it can be applied into any realistic situation. We will also try to develop a circuit which will take the numbers from the recognized numbers and compare the current speed of the car with the allowable speed and if current speed exceeds the allowable one then there will be a beep sound which will help drivers to reduce the speed.

LITERATURE OVERVIEW:-

Paper [16] worked by using LVQ and window hough transform. Paper [17] worked by using local energy based shape histogram (LESH). Paper [18] worked on American and European speed limit signs. Paper [19] worked by using colour segmentation. Paper [20] worked by using HSV colour space. Paper [21] worked on road sign detection and recognition. Paper [22] worked by using HSV colour space. Paper [23] worked on detection in poor light condition.

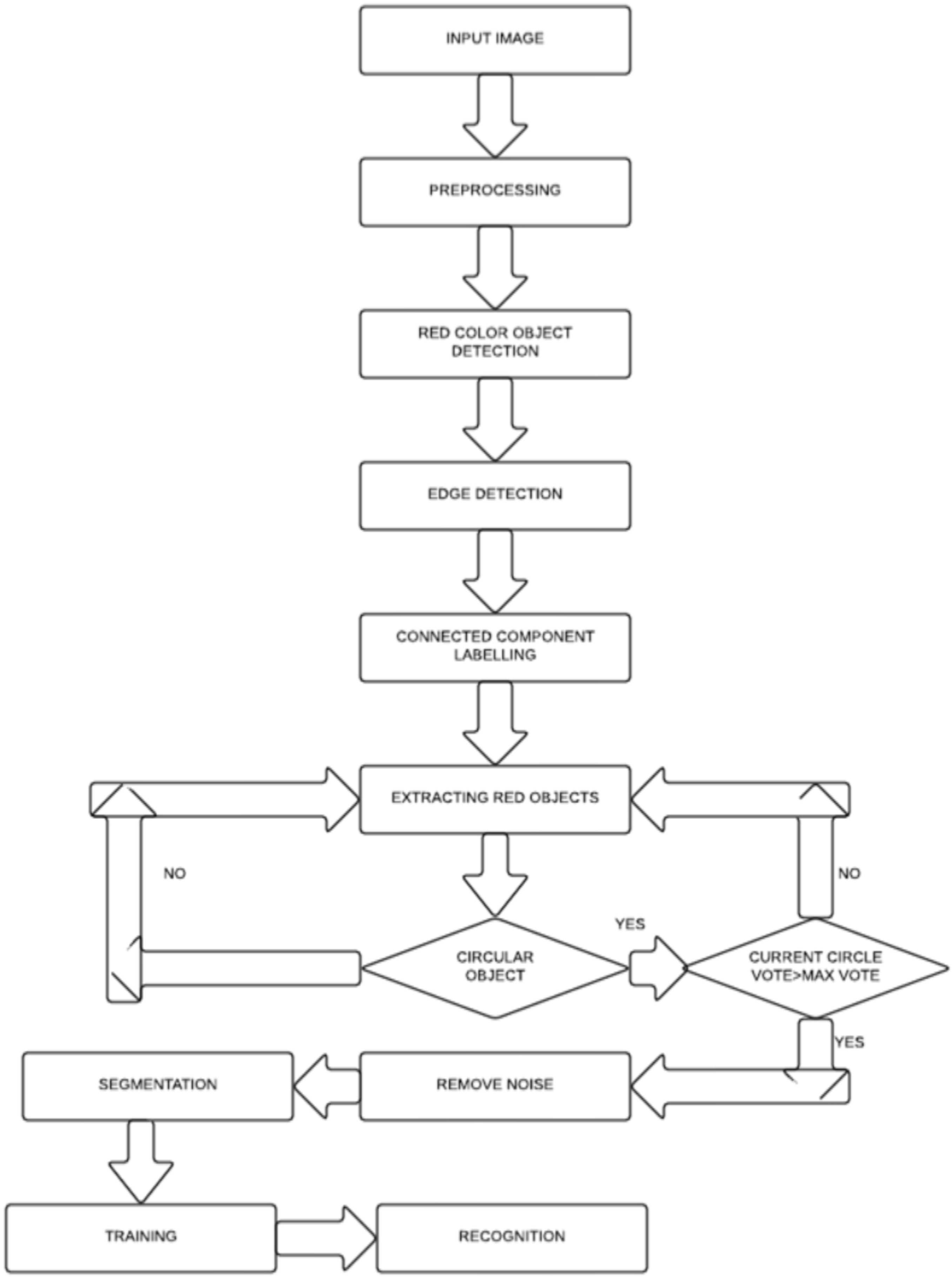
By using the knowledge from all these above papers we made our system.

Paper [24] knowledge on artificial neural network. Paper [25] talked about neural network and backpropagation. Paper [26] talked about implementation of neural network using backpropagation in matlab. Paper [27] talked about fundamental of neural network. Paper [28] talked about training with backpropagation. Paper [29] gave idea about implementation of neural network in matlab. Paper [30] gave idea about implementation of neural network in matlab.

Using knowledge from these papers we created our OCR system.

Overview of my solution:-

The block diagram of the proposed approach is illustrated in figure 2. The image which is under consideration is first color segmented and filtered from the noise and other aversion objects. The aversion of the object is selected by measuring areas of all red objects. Objects having small area are not considered. Canny edge detection is done on the target objects. It will give us the edges of objects which will help us to understand the outer shape of the object. We will look for the connected component objects by labeling the similar property objects. We will pass HT on the labeled objects. With the help of HT, we can come into conclusion whether our object is circular or not. Circular objects will be examined to detect our target circle and the target circle when find is cropped out from an image. At present we have only an image containing the numbers which are marked as black. We will remove all the noises and unnecessary objects. After removing we will segment the images to differentiate the numbers. The numbers are then recognized by the OCR system.


INPUT IMAGE
PREPROCESSING
RED COLOR OBJECT DETECTION
EDGE DETECTION
CONNECTED COMPONENT LABELLING
EXTRACTING RED OBJECTS
NO
NO
YES
CIRCULAR OBJECT
CURRENT CIRCLE VOTE>MAX VOTE
YES
SEGMENTATION
REMOVE NOISE
TRAINING
RECOGNITION

DETAILS OF MY SYSTEM:-

Input Image:

The input image will be an RGB image. RGB color model is an additive color model in which red (R), green(G), and blue (B) light are added together in various ways to reproduce a broad array of colors. Then we will convert the RGB image into Gray image. We have used the simply avearaging the value. The equation is:

GrayScale Image = (R+G+B)/3………………………………….(i)

INPUT IMAGE

RED COLOR OBJECT DETECTION:

We subtract the gray image from the red color plate. Suppose an the gray image is declared as gImage and the red plate is rImage then result will be

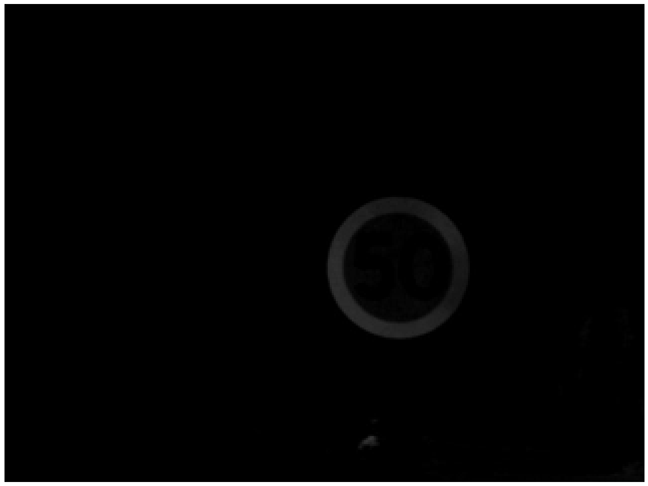
Result= rImage- gImage………………………………………………….(ii)

If the rImage value is for a particular pixel is [255 10 75; 44 225 100] and the gray image value is [50 50 50; 50 50 50] then subtract will yield a result

Z= 205 0 25

0 175 50

That is, the rImage value contain the only red portion of the image where the gImage contain the average value of the original image. When we subtract, the red value object only remains and the rest portion become the zero that means those portions are considered as black i.e. no object is there.



RED COLOR DETECTION

EDGE DETECTION:

Edge basically contains image shape information. It is one of the fundamental tool in image processing. The aim of doing it to identify the points at which image brightness changes sharply or more formally image discontinues. Edges are significant local changes of intensity in an image[3]. Edges typically occur on the boundary between two different regions in an image. The intensity basically change on object boundary, surface boundary, specularity, shadows and inter- reflections etc. Thus, applying an edge detection algorithm to an image significantly reduce the amount of data to be processed furthur by filtering out the less relevant information.

There are different types of edges like step edge, ramp edge, ridge edge, roof edge. There are four steps to find out all these edges.

1. Smoothing
2. Enhancement
3. Detection
4. Localization

We can detect edges using derivatives. Points which lie on an edge can be detected by:

(1) Detected local maxima or minima of the first derivative

(2) Detecting the zero-crossing of the second derivative.

There are three criteria for optimal edge detection[4]. These are:

Good Detection: the optimal detector must minimize the probability of false positives (detecting spurious edges caused by noise), as well as that of false negatives.

Good localization: the edges detected as must be as close as possible to the true edges.

Only one response to a single edge: this is implicitly captured in the first criterion since when there are two responses to the same edge, one of them must be considered false. However, the mathematical form of the first criterion did not capture the multiple response requirement and it had to be made explicit.

EDGE DETECTORS:

In earlier days many scientist gave many equation for edge detection. These equations are described below :

THE ROBERTS EDGE DETECTOR:

$$\partial f/\partial x = f(i,j) - f(i+1,j+1)$$

$$\partial f/\partial y = f(i+1,j) - f(i, j+1)$$

where i and j are pixels value.

THE PREWITT EDGE DETECTOR:

Consider the surrounding pixels arrangement of a pixel(i,j) :

a0	a1	a2
a3	[i,j]	a5
a6	a7	a8

the partial derivatives can be computed by:

$$Mx= (a2+ca3+a4) - (ao+ca7+a6)$$

$$My= (a6+ca5+a4) - (a0+ca1+a2)$$

Here, the constant c implies the emphasis given to pixels closer to the centre of the mask

Setting c=1, we get the Prewitt operator:

$$Mx= \begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix} \quad My= \begin{matrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{matrix}$$

THE SOBEL EDGE DETECTOR:

Setting c=2 we will get the sobel operator:

$$Mx= \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix} \qquad My= \begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix}$$

There is another edge detector which is known as Canny Edge Detector which will be described in next paragraph.

CANNY EDGE DETECTOR:

The edge detection process serves to simplify the analysis of images by drastically reducing the amount of data to be processed, while at the same time preserving useful structural information about object boundaries.All of the edge detector tried to follow this principle; but Canny edge detector is the best one in respect of the error rates.

The canny edge detection algorithm is known as the optimal edge detector. Canny followed some basic criteria to improve the current edge detector. The first and most obvious is low error rate[4]. It is important that edge occuring in images should not be missed and that there be NO responses to non-edges. The second criterion is that the edge points will be localised. In other words, the distance between the edge pixels as found by the detector and actual edge is to be minimum. A third criterion is to have only one response to a single edge. This was implemented because the first 2 were not substantial enough to completely eliminate the possibility of multiple responses to an edge.

Based on these criteria, the canny edge detector first smoothes the image to eliminate and noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along those regions and suppresses any pixels that is not at the maximum. The gradient is now further reduced by Hysteresis. Hystreresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made an nonedge). If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the 2 thresholds, then it is set to zero unless there is a part from this pixel to a pixel with a gradient above T2.

To do the canny edge detection, we have to follow some basic steps. The steps are described below.

1. Filter out any noise in the original image before trying to locate and detect any edges. A suitable Gaussian mask is needed . the larger the width of the Gaussian mask, the lower is the detector’s sensitivity to noise.
2. After smoothing the image and eliminating the noise, the next step is to find the edge strength by taking the gradient of the image. If the magnitude are Gx, Gy for x and y direction respectively then the magnitude G will be,

$$|G| = |Gx| + |Gy|$$

3. Finding the edge direction is trivial once the gradient in the x and y directions are known. The formula for finding the edge direction is just:

$$\text{Theta} = \text{invtan}\ (Gy/Gx)$$

4. Once the edge direction is known, the next step is to relate the edge direction to a direction that can be traced in an image. So if the pixels of a 5*5 image are as follows:

x	x	x	x	x
x	x	x	x	x
x	x	a	x	x
x	x	x	x	x
x	x	x	x	x

Then, it can be seen by looking at pixel “a”, there are only four possible directions when describing the surrounding pixels – 0 degrees (in the horizontal direction), 45 degress (along the positive diagonal), 90 degress (in the vertical direction), or 135 degrees (along the negative diagonal). So now the edge orientation has to be resolved into one of these four directions depending on which direction it is closest to. Think of this as taking a semicircle and dividing it into 5 regions.

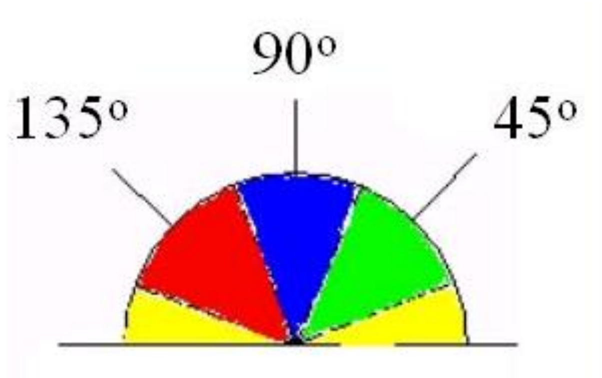


Fig: orientation angle

5. After the edge directions are known, non maximum suppression now has to be applied. Nonmaximum suppression is used to trace along the edge in the edge direction and suppress any pixel value.
6. Finally, hysteresis is used as a means of eliminating streaking. Streaking is the breaking up of an edge contour caused by the operator output fluctuating above and below the threshold.

CANNY EDGE DETECTOR

ENHANCEMENT OF IMAGES:

Hence we are considering images in different environment in different situation, there is a huge possibility of getting the bad quality image. By bad quality we mean image may be blurred, may be noisy, may be foggy, may be distorted and many more. Therefore, it is very much troublesome to find the circles from this situation. To get rid out of this, we have to enhance the image quality. For enhancement, we have to do some certain tasks. The very first thing is to convert the image into a binary image.

Binary image means where the image pixel value is more than the threshold mark those as an object otherwise zero. The equation for converting into binary image is described below:

$$\text{bin}(f)(x) = \begin{cases} 1 \text{ if } f(x) \geq k1 \\ 0 \text{ otherwise} \end{cases}$$

To remove noise and unnecessary objects we pass morphological analysis on binary image. The word morphology commonly denotes a branch of biology that deals with the form and structure of animals and plants. In image processing, it is called mathematical morphology as a tool for extracting image components that are useful in the representation and description of region shape. The basic idea of mathrmatical morphology is its shapes are represent as a set[6]. Binary images is a subset of Z^2 where each pixel is a tuple of coordinates (x,y) to identify whether a pixel is white or black.

There are many operations in mathematical image processing. Among all of them, we used the opening and closing operation. All of the operation is done along with a Structuring Element Decomposition. A structuring element is a shape, used to probe or interact with a given image, with the purpose of drawing conclusions on how this shape dits or misses the shapes in the image[7]. The structuring element can be composed of ones and zeros. It can be on different shape like polygonal,diamond or it may denote a point having a particular distance from the object.

Among all these operation, we used the opening and closing operation. The morphological opening of A by B, denoted $A \circ B$, is defined as the erosion of A by B, followed by a dialation of the result by B:

$$A \circ B = (A \ominus B) \oplus B$$

An equivalent formulation of opening is

$$A \circ B = \bigcup \{ (B)_z \mid (B)_z \subseteq A\}[8]$$

Morphological opening removes completely regions of an object that can not contain the structuring element, smooths object contours, breaks thin conection and removes thin protrusions.

The morphological closing of A by B, denoted A· B, is a dialation followed by the erosion:

$$A \cdot B = (A \oplus B) \ominus B$$

Geometrically, A· B is the complement of the union of all translations of B that do not overlap A.

ENHANCEMENT OF IMAGES

CONNECTED COMPONENT LABELLING:

After image enhancement, now we are looking for the connected component labelling. Connected-component labeling (alternatively connected-component analysis, blob extraction, region labeling, blob discovery, or region extraction) is an algorithmic application of graph theory, where subsets of connected components are uniquely labeled based on a given heuristic. Connected-component labeling is not to be confused with segmentation [9].

A binary image contains two types of pixels: object pixel and background pixel. The connected component labeling problem is to assign a label to each object pixel so that connected (or neighboring) object pixels have the same label. There are two common ways of defining connectedness for a 2-D image: 4-connectedness and 8- connectedness.

8-connected:

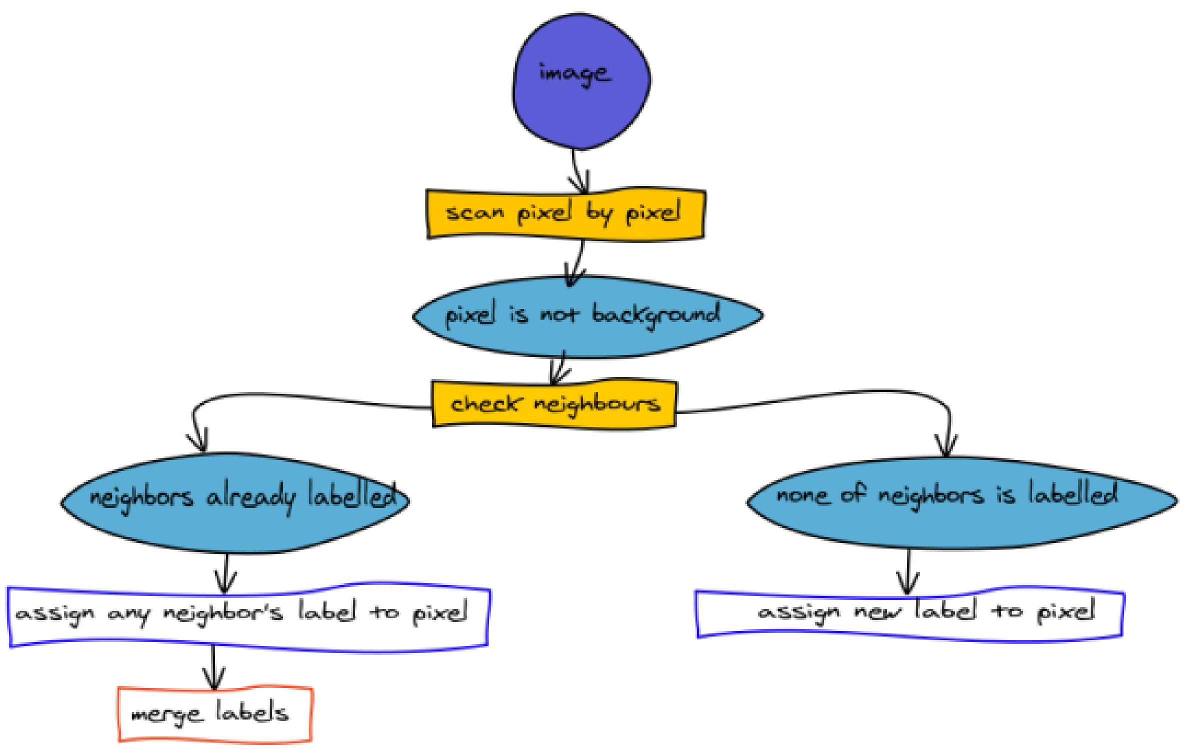
8-connected pixels are neighbors to every pixel that touches one of their edges or corners. These pixels are connected horizontally, vertically, and diagonally. The 8- connected pixels are:

(x+1, y+1) , (x-1, y-1), (x$\pm$1,y±1) or (x$\pm$1,y$\mp$1) are connected to the pixel (x,y)[10].

Connected component labelled alogorithm can be divided into mainly three categories. These are multi-pass algorithm, two-pass algorithm and one- pass algorithm. Among these, we used the two-pass algorithm. The algorithm is described below[11]:

1. Run – through out the input image.
2. Scan the runs, assigning preliminary labels and recording label equivalence in a local equivalence table.
3. Resolve the equivalence classes.
4. Relabel the runs based on the resolved equivalence classes.

The algorithm is shown in the flow- chart in a more concise manner and represent in a very easy way[12]:



Detection of circle using Hough transform:

This process is also known as Circular Hough Transform (CHT) . Circular Hough transform can be invoked to detect the circular shapes. It can be described as a transformation of center point of a circle in x-y plane to the parameter space. The equation of a circle in x-y plane is given by:

$$(x-a)^2 + (y-b)^2 = r^2$$

Where a , b are the center points of the circle in the x and y direction and r is the radius of the circle. The parametric representation of the circle is given by:

$$X = a + r\cos(\theta)$$

$$X = a + r\sin(\theta)$$

To determine the presence of a circle, it is necessary to accumulate votes in the three-dimensional parameter space (a, b, r) [13]. The objective is to find the (a, b) coordinates of the center. The locus of (a,b) points in the parameter space fall on a circle of radius r centered at (x, y). The true center point will be common to all parameter circles and it can be found with a Hough accumulator array. This approach is illustrated in figure:

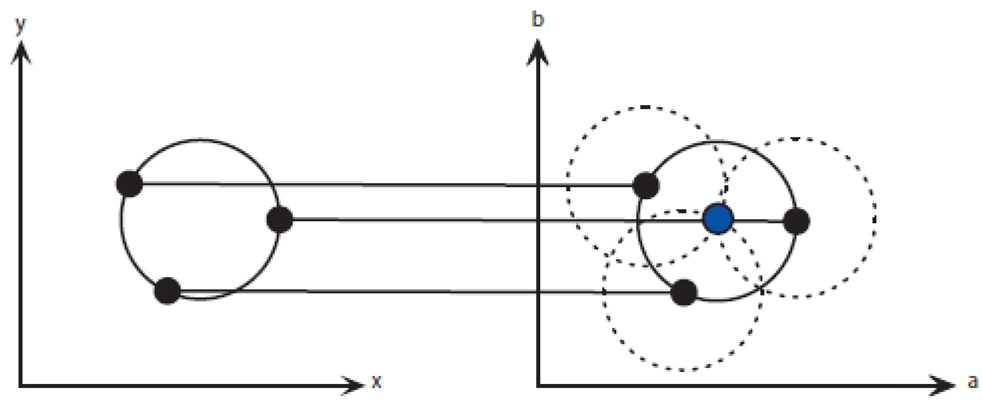


Figure 2: A Circular Hough transform from the x,y-space (left) to the parameter space (right), this example is for a constant radius

HT can be performed for the single circle detection with Fixed radius also it can be performed for multiple circle detection. Figures shown in the above are for single circle. A figure shown below is for detecting multiple circles.

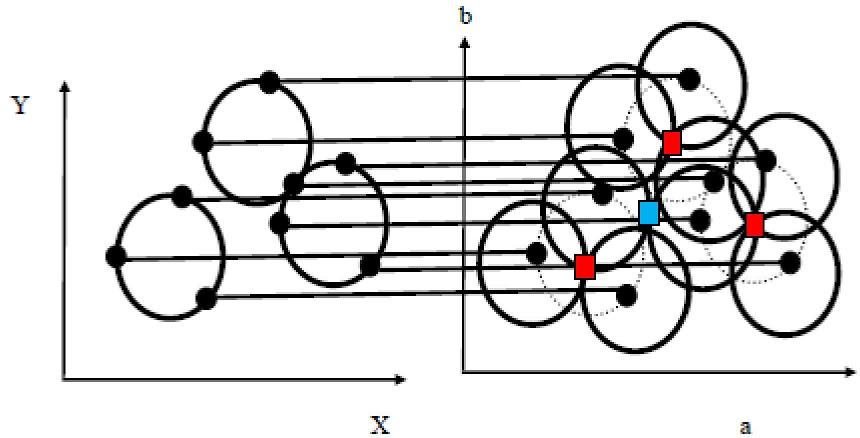


Figure: Mapping of 3 points of each circle on x-y plane to 3 circles on a-b plane

Hence, we are dealing with the dynamic images that mean we don't have any idea about the radius of the circle. With the help of HT we are able to find out the circle with unknown radius.

If the radius of a circle is unknown then the locus of points of circle in parameter Space will fall on the surface of a cone. As the radius is unknown, in this case it has three parameters (a, b and R). Each point (x, y) on the perimeter of a circle which is on x-y space or geometric space will produce a cone surface in parameter space. The triplet (a,b,R) will correspond to the accumulation cell where the largest number of cone surfaces intersect [15]. Figure describes the generation of a conical surface in parameter space for one (x, y) point [15]. A circle with different radius will be generated at each level, r. The search for circles with unknown radius can be found by using a three dimensional matrix.

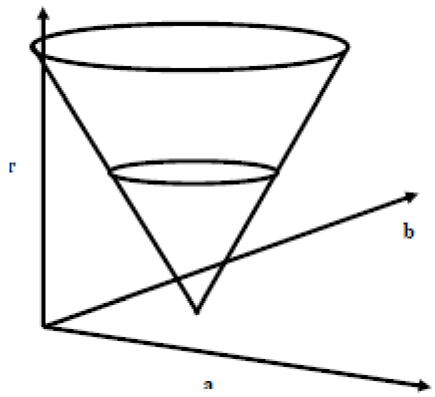


Fig: The parameter space of a circle.

Circle Properties:

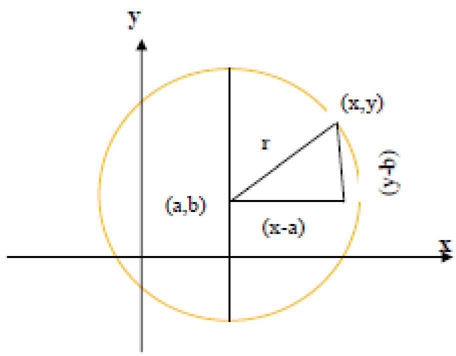


Fig: Circle Properties

The general equation of a circle is $(x-a)^2+(y-b)^2=r^2$. where 'a' and 'b' are the center point and 'r' is the radius of the circle. From this equation we can get 'a 'by-

$$(x-a)^2+(y-b)^2=r^2$$
$$(x-a)^2=r^2-(y-b)^2$$
$$(x-a)=sqrt(r^2-(y-b)^2$$
$$a= x- sqrt(r^2 - (y-b)^2)$$

Following the same way the value of 'b' also can find out.

$$(x-a)^2+(y-b)^2=r^2$$
$$(y-b)^2=r^2-(x-a)^2$$

$$(y-b)=sqrt(r^2-(x-a)^2)$$
$$b = y - sqrt(r^2-(x-a)^2)$$

If any coordinate is known of the center point, rest coordinate can be found out using any equation mentioned above for ‘a’ and ‘b’.

In our case, these properties are used to detect circle from images. In the proposed algorithm ‘b’ is considered as all pixels of height of the image. Then ‘a’ is calculated using each value of ‘b’ on equation (1) with unknown radius (centre ranged). After finding ‘a’ and ‘b’ which is the center point, voting accumulation is carried on the Hough matrix. This voting is done following CHT discussed above. For each edge point of each object in the image this voting accumulation is performed. This voting is continued for all edge points of an object. When voting is finished the maximum vote is found out and compared with a certain threshold. If the maximum vote is greater than or equal to the certain threshold then the object is selecred as a circular shape limit sign.

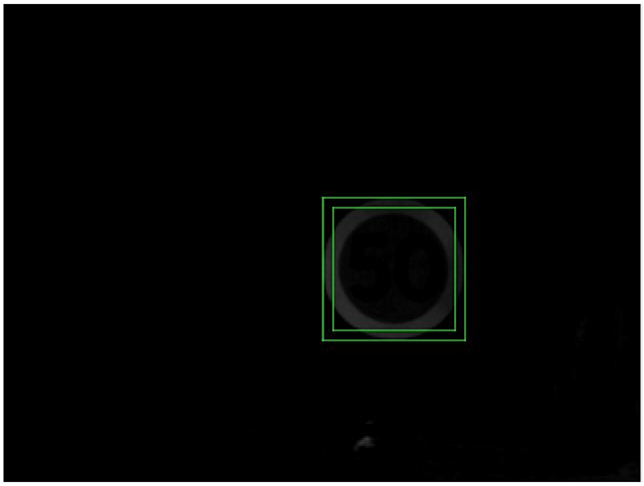


Fig: circle detection

Implementation of OCR:

OCR can be implemented in various ways. The type of implementation depends on what kind of system we are designing it for. As part of our OCR implementation we have used "Artificial Neural Network" (ANN) implementation technique.

Artificial Neural Network (ANN):

One efficient way of solving complex problems is following the lemma "divide and conquer". A complex system may be divided into simpler and smaller systems, in order to be able to understand it. Networks are one approach for achieving this. There are a large number of different types of networks, but they all are characterized by the following components: a set of nodes, and connections between nodes.

Nodes can be seen as computational units. They receive inputs, and process them to obtain an output. This processing might be very simple (such as summing the inputs), or quite complex. The connections determine the information flow between nodes. They can be unidirectional, when the information flows only in one direction, and bidirectional, when the information flows in either direction.

Similarly we are also using a kind network known as "Artificial Neural Network" in short (ANN). This network has been inspired by biological human brain neural structure. They are powerful tools for modeling, especially when the underlying data relationship is unknown. ANN can identify and learn similar patterns between input data sets and corresponding target values. After training ANN can be used to predict the outcome of new independent input data. As we mentioned ANN follows the learning process of the human brain and can process problems involving non-linear and complex data even if the data are imprecise and noisy.

It consists of simple computational units called neurons, which are highly interconnected. A very important feature of these networks is there adaptive nature where, "learning by example" replaces "programming" in solving problems. This feature makes such computational models very appealing in application domains where one has little or incomplete knowledge of the problem to be solved but there is one major condition behind its successfulness is that training data must be available.

There are various types of network to work with like, Multilayer Perceptron, Radial Basis Function and Kohonen networks. From here we are going to use Multilayer Perceptron (MLP) or sometimes called Multilayer Feed Forward neural network. One additional algorithm we need to complete the learning part of the network and for that we would use the most widely used "Backpropagation" algorithm.

Structure of the Network:

The network we have talked about so far must have a structure or model that would help all the learning and computational processes. The kind of model we are implementing is called Multilayer Perceptron (MLP) or sometimes called Multilayer Feed Forward neural network. The name itself defines the structure that is a model that has multiple layers of neurons or nodes. MLP neural network learned by back propagation algorithm is based on supervised procedure, i.e. the network constructs a model based on examples of data with known output. As like any network models this network also contains two basic layers called input layer and output layer. The number of neurons in the input layer depends on the input image size that is each pixel represents each node respectively. The output layer may contain one or more neurons depending on the developer and the system. In our system we kept 10 output neurons which will resembles first ten numeric digits (0 to 9).

In between these two layers there may be one or more additional layers called the hidden layers. We have added one hidden layer in our system because it increases accuracy while the learning process proceeds. There is no fixed rule for the number of neurons that would be in the hidden layer. These bunches of layers are interconnected and as part of MLP the information flows unidirectionally from input layer to output layer through hidden layer. There are some characteristics of MLP:

i. Has any number of inputs.
ii. Has one or more hidden layers with any number of nodes.
iii. Uses linear combination function in the hidden and output layer.
iv. Uses generally sigmoid activation function in the hidden layers.
v. Has any number of outputs with any activation function.
vi. Has connection between the input layer and the first hidden layer, between the hidden layers and between the last hidden layer and the output layer.

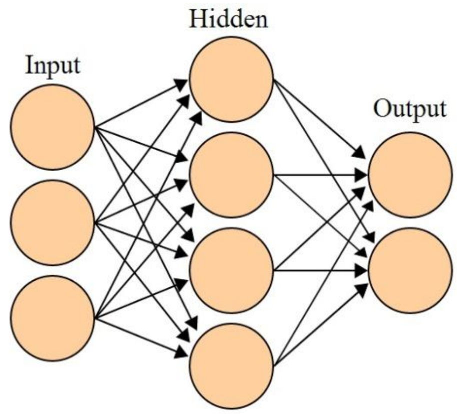


Fig1: Artificial Neural Network with input, hidden and output layers.

Backpropagation and Calculations:

So far we have seen how our network would look like but the most important thing that is still missing is the learning algorithm. We have used the term “Backpropagation” couple of times in our paper now let’s see what this is.

To train our network we need an algorithm that would first train our network so that it can recognize any patterns that we want it to. Backpropagation is that learning algorithm. This training session goes through number of steps with some calculations. The basic calculation that goes is that every input in a neuron is assigned a weight. Initially the input layer neurons are assigned random weight between -1 to +1. Then the output is calculated with that weight (this is called the forward pass). This output is way very different from the output that is required. Then we calculate the error difference between out obtained value with the required value. After that mathematically the weights are updated so that this error difference gets smaller and proceeds towards our target value (this part is called reverse pass). This process continues until the error is minimized. Let's see an example between connection of an output layer and a hidden layer:

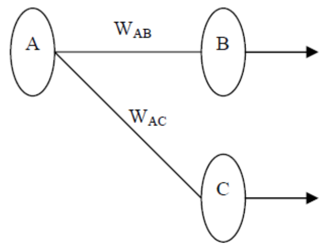


Fig 2: A single connection between the hidden layer and an output layer

The connection in fig. 2 is between neuron A (a hidden layer node) and neuron B (an output layer node) which has the weight W_{AB}.

1) First of all insert inputs to the network and obtain the output. This initial output can be anything because the weight take at first were random.

2) Now we need to find the error for neuron B, this error is "*What we want – What we actually got"*.

$$Error_B = Output_B\,(1-Output_B)(Target_B – Output_B)$$

Here "$Output_B\,(1-Output_B)$" is necessary in the equation because of the sigmoid function.

3) Now we need to change the weight. Let W_{+AB} be the new (trained) weight and W_{AB} be the initial weight.

$$W_{+AB} = W_{AB} + (Error_B \times Output_A)$$

Notice one thing that it is the output from the neuron A to B that we update not B. We now update all the weights in the output layer in the same way.

4) After this we now need to calculate the error for the hidden layer neurons. It's not as easy as we saw in the case of calculating error in the output layer (this is because here we don't have any target to compare with). Here we need to back propagate them from the output layer (as the name of the algorithm signifies). Here we take the error from the output neurons and running them back through the weights to get the hidden layer errors. From the fig 2 we can say that since neuron A is connected to neuron B and C so the errors of B and C are taken to calculate the error for A.

$$Error_A = Output_A (1 - Output_A)(Error_B W_{AB} + Error_C W_{AC})$$

5) After this we obtain the error for the hidden layer as well. This process is followed from step 3 to change the hidden layer weights.

So after all these we can say that "Backpropagation" algorithm is used to train our network in order to reach our target by minimizing the error.

Graphical Representation:

After all these explanation we would like to demonstrate the whole procedure of character recognition step by step graphically.

Step 1) After segmentation we get an image like below which we would like to recognize.

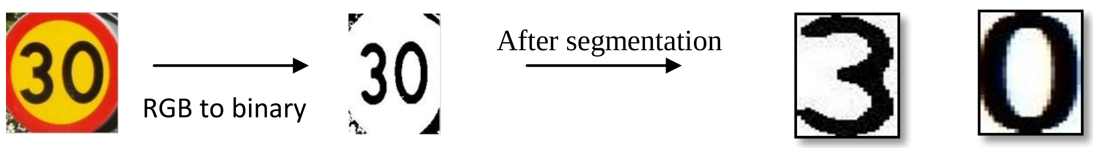


Fig 3: Segmented image

Step 2) Inputs are given in an input layer in a matrix form where each pixel carries a value. Like below.

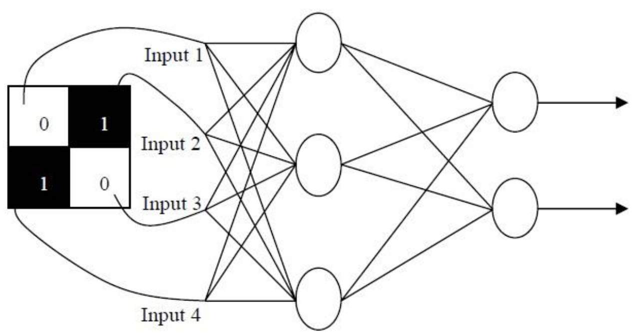


Fig 4: Input mechanism

Step 3) We then create the network which some where look like this.

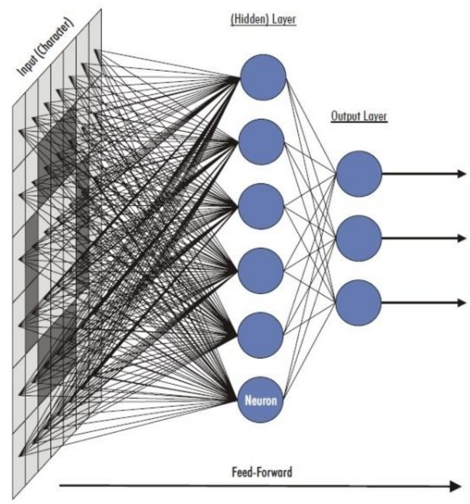


Fig 5: Network looks like

Step 4) Now we need to train the network with backpropagation algorithm. For training we need to create a training set that includes some of the outputs after segmentation. Also we need to create a testing set that would evaluate the generalization ability of our trained network.

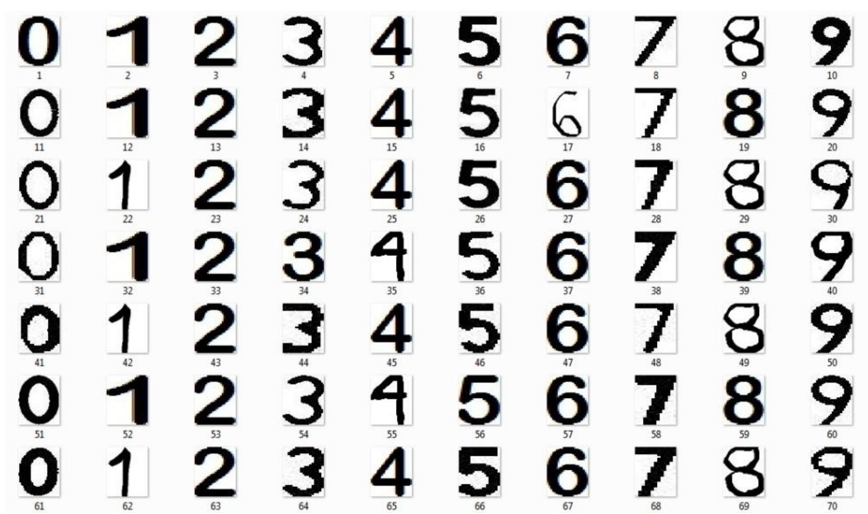


Fig 5: Training set

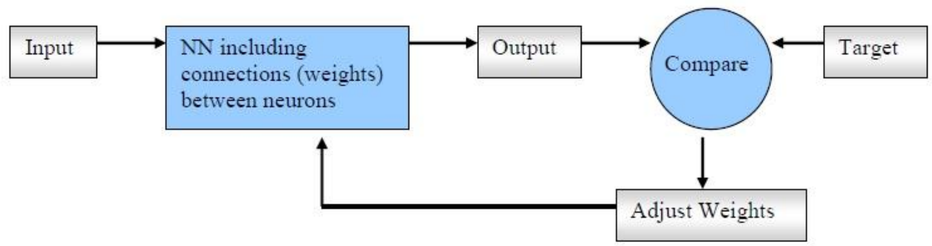


Fig 6: Training with backpropagation algorithm

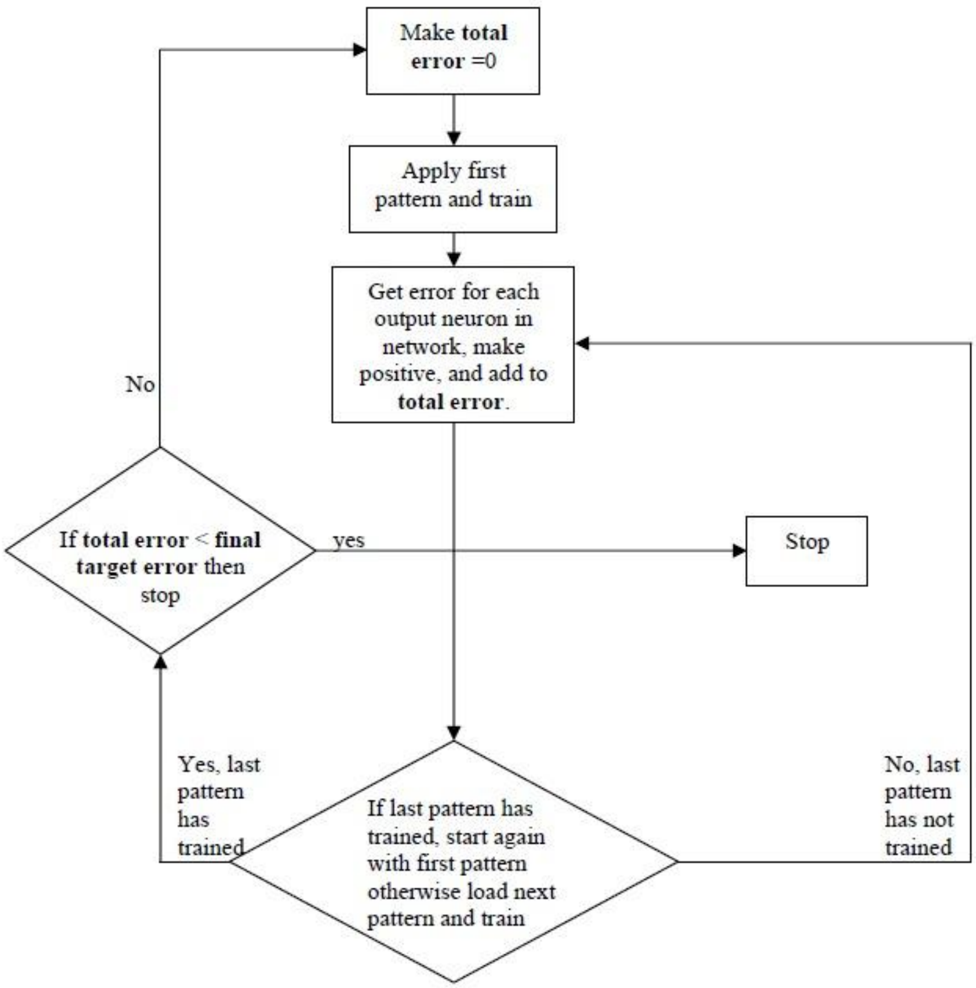


Fig 7: Error Calculation in backpropagation

Step 5) After our training is successful then we will see if out network able to recognize any given pattern.

Fig 8: Input file to the system

Fig 9: Recognized character by the system

We need to remember certain things while training that all input image should be of same size. Both training and testing images should be cropped to the starting pixel of the characters. Also we need to train the network with all available inputs because its recognition depends how well it is trained.

Scalable:

Object detection [16] is one of the most important tasks in the computer vision and pattern recognition. Objects detection has an impact on the systems reliability and effectiveness directly. Object detection falls into two broad categories: irregular object direction and regular object detection. Irregular object doesn’t have a fixed shape (such as leaves, flowers, etc.), in general it is difficult to detect such objects unless proper training is given before. On the other hand regular object has fixed geometric shape or fixed pattern which helps us to detect the object.

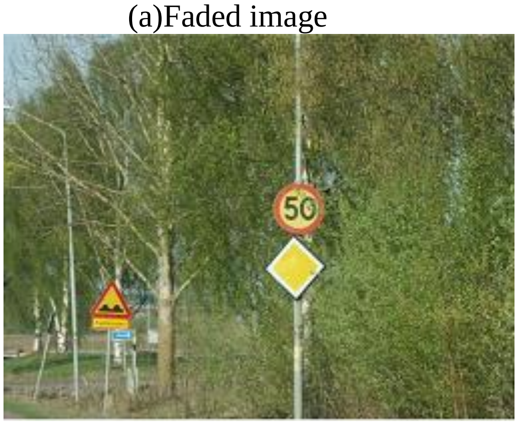
At first glimpse, the traffic sign detection system has some advantageous characteristics. But there are some situations where detection is really very much tough. Some of them are given below:

- As a result of the long exposure to the sun light the color of the sign fades with the passage of time. The boarder may be damaged also.
- Obstacles, such as trees, poles, buildings, and even vehicles may occlude the image

- Lighting conditions are changeable and not controllable. Intensity of light is different on different seasons and different time and also on different weather conditions such as fog, rain, clouds and snow.
- Although shapes and colors of the traffic signs are deterministic, but due to the shape variation size and aspect ratio may vary a lot. This is the toughest challenge to meet up the entire situation. This is really a threat to find out the optimized way for the board detection.
- There may be a presence of same color of object except circle in the image. Moreover, there may be also a circular object containing another prohibitory sign.
- If the image is acquired from a moving car, then it is often suffers from motion blur and car vibration.
- Other parameters like local light variations (the direction of the light, the strength of the light depending on the time of day and the shadows generated by the other objects can also affect visibility.

(a)Faded image

(b) Damaged Image



(c) Partial occlusion

(d) Blurred Image

(e) Bad light

(f) Bad weather

(g) Shape Deformation

(h) Objects having same color

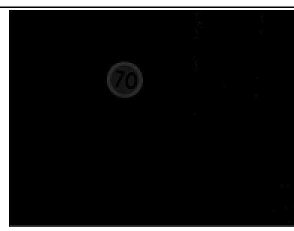
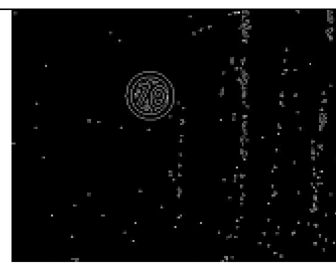
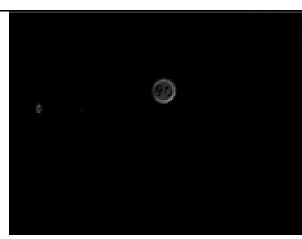
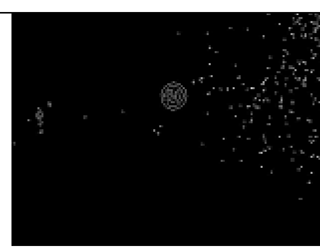
(I) More than one circular object

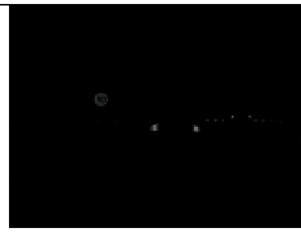
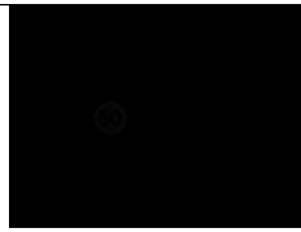
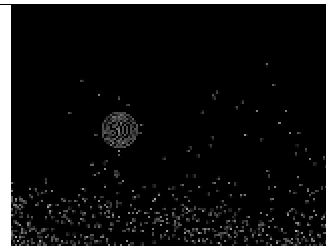
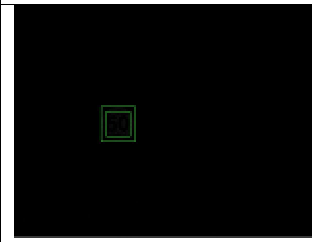
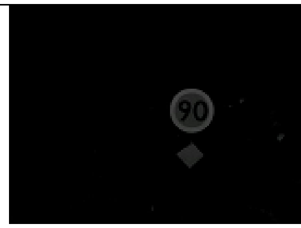
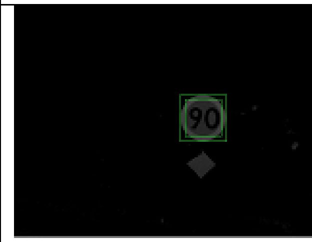
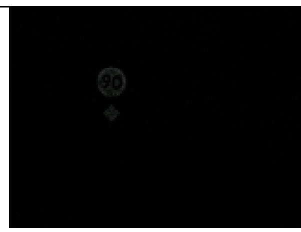
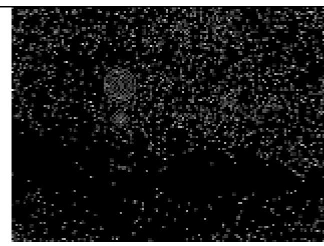
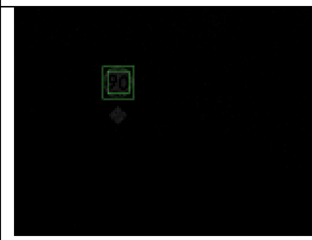
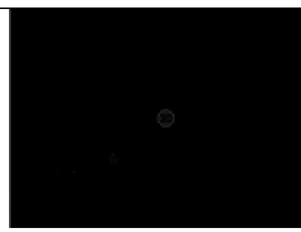
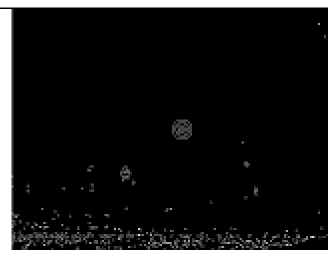
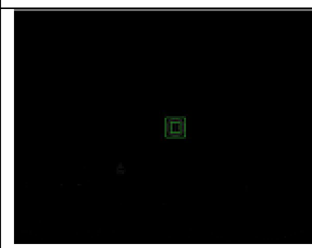
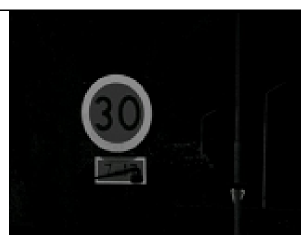
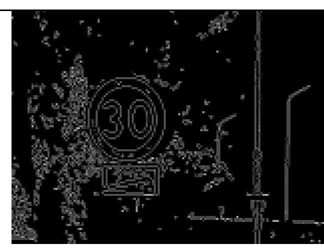
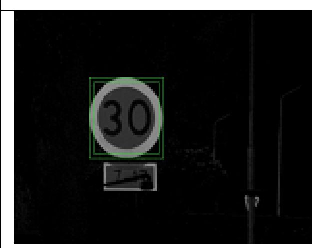
RESULTS:

We detect the circular object in various different conditions. Therefore, the success rate is different on the basis of different situation. These different conditions can be divided into mainly eight categories. These are:

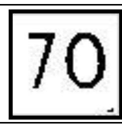
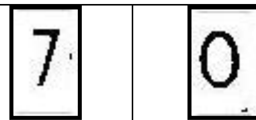
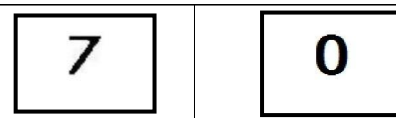
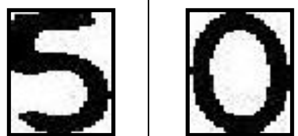
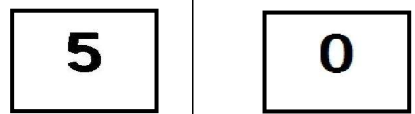
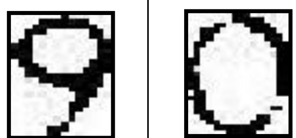
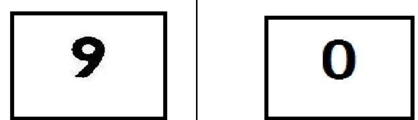
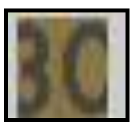
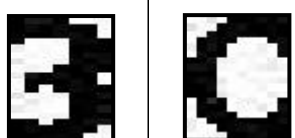
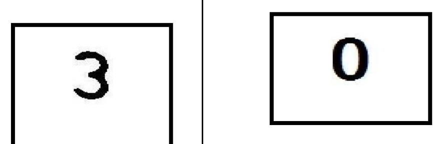
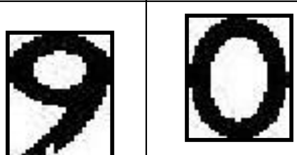
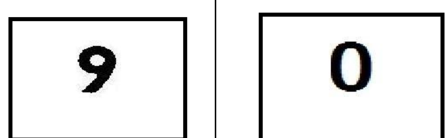
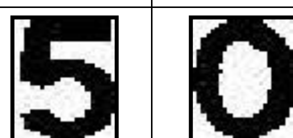
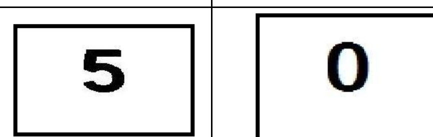
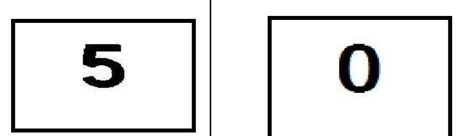
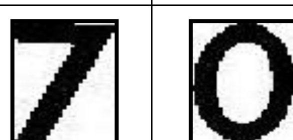
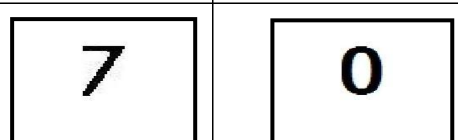
1. Faded
2. Blurred
3. Dawn and dusk
4. Fog
5. Highlight
6. Noisy
7. Snow fall
8. Sunny

For each above mentioned condition, one input image, one image after detection of red color, one after canny edge detection and finally a circled detection image are shown in a tabular form.

condition	Input image	Red color objects	Canny edge detector	Circle detection
Faded				
Blurred				

Dawn & Dusk				
Fog				
highligh t				
Noisy				
Snow fall				
Sunny				

Now we will show that our recognition was successful at all of these conditions.

Conditions	RGB	Binary	Segmented	Recognized
Faded				
Blurred				
Dawn and Dusk				
Fog				
Highlight				
Noisy				
Snowfall				
Sunny				

CONDITION	NO. OF IMAGES	POSITIVE OBJECTS	TRUE POSITIVE OBJECTS	FALSE NEGATIVE OBJECTS	NEGATIVE OBJECTS	TRUE NEGATIVE OBJECTS	FALSE POSITIVE OBJECTS
Faded	14	15	14	0	18	17	0
Blurred	13	13	12	0	17	17	1
Dawn & Dusk	14	16	12	1	19	17	2
Fog	3	3	2	0	5	4	1
Highlight	11	11	1	0	12	11	1
Noisy	21	22	18	0	23	22	3
Snow fall	10	10	9	0	13	12	1
Sunny	12	12	12	1	14	12	0
TOTAL	98	102	89	2	121	112	9

LIMITATION:

In this work we have to consider for both the accuracy and time. Failing in one case ultimately make the work meaningless. To do accuracy, we have to sacrifice time. Moreover, we are doing it on matlab which itself is a slow IDE. So we have to shift it OpenCV to make it more fast and robust one. Moreover, we get only 98 images in Internet in different condition. It will be really nice if we get more images.

Future Work:

We are working here on still images. In future we will try to do it from video streaming. We also try to build up the circuit for our work.

Conclusion:

In this thesis, a new method to detect a group of prohibitory traffic signs namely speed limit signs, is studied. Detection of the above type of road sign is interesting since they play a key role in improving the road traffic safety and reducing the number of road traffic casualties from traffic collisions. Speed has many positive impacts, the most important impact is it allows a reduction in journey time and therefore enhances the mobility. On the otherhand, excessive and inappropriate speed is the number one road safety problem in most of the countries. From previous fatal accidents history it is found that speed is an irritating factor in the brutality of all accidents. Therefore, detecting this group of signs correctly and controlling the road vehicles according to that speed limit may be helpful in preventing fatal accidents. The proposed system is an adaptive and standalone which means that it works without any kind of previous training to detect speed limit signs. The proposed algorithm is tested on static images. For testing 110 images under different weather and lighting conditions are used. The speed limit sign detection rate is about 92%.In terms of detection rate of speed limit signs Hough Transform produces very good results. The experiment is conducted on color segmented images where the detection of false positive is very rare.

After our successful segmentation we were able to train our neural network using backpropagation successfully. Our character recognition hits the accuracy of 83%.

REFERENCES:

1. http://en.wikipedia.org/wiki/Traffic_sign

2. http://www4.worldbank.org/afr/ssatp/Resources/HTML/legal_review/Annexes/Annexes%20II/Annex%20II-19.pdf

3. http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0CEQQFjAC&url=http%3A%2F%2Fwww.cse.unr.edu%2F~bebis%2FCS791E%2FNotes%2FEdgeDetection.pdf&ei=cCu6UN-uDIn4rQfYs4CYAg&usg=AFQjCNGNoWn0qDIdGIjsXWfcFzN3NOTZbg&sig2=3GXL-qcC9FjO24Xoi3S1bA

4. http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CDEQFjAA&url=http%3A%2F%2Fwww.limsi.fr%2FIndividu%2Fvezien%2FPAPIERS_ACS%2Fcanny1986.pdf&ei=3FO6UNLgL4_OrQfFhIHQAw&usg=AFQjCNE-VlCkp86OJEq1FT7L3bsk9-pQMA&sig2=qQQYaht9kBpTIw6jLIYrhg

5. http://dasl.mem.drexel.edu/alumni/bGreen/www.pages.drexel.edu/_weg22/can_tut.html

6. http://www8.cs.umu.se/kurser/5DV015/VT09/handouts/slides/morphology-

7. http://en.wikipedia.org/wiki/Structuring_element

8. Digital image processing using matlab second edition.

9. http://www.codeproject.com/Articles/336915/Connected-Component-Labeling-Algorithm

10. http://en.wikipedia.org/wiki/Pixel_connectivity#8-connected

11. http://blogs.mathworks.com/steve/2007/05/11/connected-component-labeling-part-5/

12. http://www.codeproject.com/Articles/336915/Connected-Component-Labeling-Algorithm

13. http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CDEQFjAA&url=http%3A%2F%2Fopencf.pcg.ull.es%2Fredmine%2Fprojects%2Fopencfr%2Frepository%2Fentry%2Fbranches%2Ftesis%2Fcelulas%2Fdocumentos%2FSimon_Pedersen_CircularHoughTransform.pdf&ei=nnC7UP7WJoLKrAe3g4GoAg&usg=AFQjCNGT1jAt0JT7dwzg-fuZ6Uucql-jZQ&sig2=oIChAzglnfN2-JDmAft86w

14. http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CC4QFjAA&url=http%3A%2F%2Fwww.cis.rit.edu%2Fclass%2Fsimg782%2Flectures%2Flecture_10%2Flec782_05_10.pdf&ei=OXi7UKGIGMzJrAfHpYGwAQ&usg=AFQjCNEbJCkuoponGsRlSQl7qZhYvn_zMg&sig2=2YXPs69Vll_PYoJv3tHD-w

15. http://scholar.google.com/scholar_url?hl=en&q=http://www.merl.com/reports/docs/TR2004-043.pdf&sa=X&scisig=AAGBfm1oGvoqYfKSjVFC5Lqu99-DJy5fyw&oi=scholarr

16. Hasan Fleyeh , Nizam U. Bhuiyan and Rubel Biswas , " PROHIBITORY TRAFFIC SIGNS DETECTION USING LVQ AND WINDOWED HOUGH TRANSFORM"

17. Usman Zakir ,Iffat Zafar and Eran A.Edirisinghe, "Road Sign Detection and Recognition by Using Local Energy based Shape Histogram (LESH)"

18. Fabien Moutarde and Alexandre Bargeton and Anne Herbin and Lowik Chanussot , "MODULAR TRAFFIC SIGNS RECOGNITION APPLIED TO ON-VEHICLE REAL-TIME VISUAL DETECTION OF AMERICAN AND EUROPEAN SPEED LIMIT SIGNS"

19.Wen-Yen Wu, Tsung –Cheng Hsieh and Ching-Sung Lai ,"Extracting Road Signs using the Color Information"

20. Martin FIFIK, “Slovak Traffic Signs and Their Preprocessing in HSV Color Space”

21. Michael Shneier, “Road Sign Detection and Recognition”

22. Hasan Fleyeh, “COLOR DETECTION AND SEGMENTATION FOR ROAD AND TRAFFIC SIGNS”

23. Hasan Fleyeh, “Traffic Signs Color Detection and Segmentation in Poor Light Conditions”

24. Carlos Gershenson, “Artificial Neural Networks for Beginners”.

25. SargurSrihari “Artificial Neural Networks”

26. Jamshid Nazari, Okan K. Ersoy “Implementation of back-propagation neural networks with MatLab”

27. Martin H. Luerssen, “Character Recognition with Neural Networks”

28. Thomas Natschl¨ager, “Optical Character Recognition”

29. prof. Primož Potočnik “CHARACTER RECOGNITION”

30. http://www.csee.umbc.edu/~dpatte3/nn/backprop.html

31. http://www.mathworks.com/help/nnet/ref/network.html

32. http://www.philbrierley.com/main.html?code/matlab.html&code/codeleft.html