CHAPTER 3
RECOGNITION OF ARABIC NUMERALS WITH GROUPING AND UNGROUPING USING BACK PROPAGATION NEURAL NETWORK

3.1 INTRODUCTION

Optical character recognition [132] is the process of conversion of printed texts or images into a computer readable form. The document is converted to a form that is ease for editing, translation, processing and displaying the desired data in human readable form. It is a field of research used in broad range of applications. The document may be machine printed or handwritten. Techniques are also available for converting scanned document, images, PDF files, etc. The OCR systems are available for wide and all varieties of languages and applications.

A new concept of grouping and ungrouping was proposed to recognize both printed and handwritten Arabic numerals. The proposed method first analyzes the quality of the input image. It performs preprocessing steps to set the image in desired format after removing noises, if any. Segmentation and extraction steps are performed to extract individual characters. Recognition is carried out with Backpropagation neural network.

3.2 RECOGNITION OF ARABIC NUMERALS WITH GROUPING AND UNGROUPING USING BACK PROPAGATION NEURAL NETWORK

The general background information about image processing and soft computing techniques was elaborated in the following section.

3.2.1 Character recognition

Wide varieties of algorithms are available for the recognition of images. Among them matrix matching compares the image to the target image of a stored glyph on a pixel by pixel basis. In this the desired image is segmented from the image background. The approach is suitable for printed texts but doesn’t prove better when used for images with different fonts. The desired patterns or features from the image are to be extracted before recognition. The features for extraction may include lines, curves, loops, intersections present in the image.

Arabic is becoming a popular language in most of the western countries. Arabic numerals were chosen in the proposed work for recognition. Arabic numerals traditionally, are read with smallest element first, whereas in modern Arabic, their order is thousand- hundred-unit-decimal. Generally, recognition of handwritten character faces more problem than printed texts. Arabic numerals are the ten digits that were descended
from the Indian numeral system. Although the pattern of 0-9 is the same as in Indian numeral system, the glyphs vary for each numeral.

3.2.2 Image Processing

The input image [142] is to be converted to a form that is suitable for various processing operations. Various Image processing techniques that are utilized in the proposed method are discussed in the below section.

3.2.3 Filters

Filtering [46] is a technique to remove any noise present in the image and to improve the quality of the image. The algorithmic steps used in every filtering technique either reduce some information in the image or add certain features to enhance its quality. Technically, it is termed as suppressing either high or low frequency in the image. In certain cases, the image may be so dark and not so pleasant to the view. During such situations the image pixels can be smoothed. There are also situations in which the location of the pixels may not be visible to the eye. In such situations, the image pixels need to be sharpened by adding some extra information; hence the final retrieved image is at the desired level. In other cases, the curves, loops or the edges may not be clear or it may be deformed. In these cases, the edges are to be enhanced. These are all some common problems faced in recognition. In the proposed method Laplacian filter is used for removing the noise present in the image.

Laplacian Filter

The Laplacian filter [47] was mainly used to detect the edges in an image. The approach highlights the region of rapid change in intensity. It takes gray scale image as input and produces another gray scale image as output. The operator used takes a gray level image as input and produces another gray level image as output.

The filter works as follows,

The Laplacian \( L(x, y) \) of an image with pixel intensity values \( I(x, y) \) is given by the following expression:

\[
L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}
\]  

3.1

The expression specified in equation 3.1 is computed by using a convolution filter. A convolution on kernel is then found. Two commonly used kernels were as shown in figure 3(a).
The Laplacian is calculated by any one of the standard convolution methods. The image is Gaussian smoothed before applying the Laplacian filter to remove any noise. The Gaussian smoothing filter is now convolved with the laplacian filter. The 2-D LoG function centered on zero and the Gaussian standard deviation “$\sigma$” has the form as shown below:

$$\text{LoG}(x, y) = -\frac{1}{\pi \sigma^4} \left[ 1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{\frac{-x^2 + y^2}{2\sigma^2}}$$  \hspace{1cm} (3.2)

**3.2.4 Convolution Technique**

Convolution [113] is the process in which the desired output pixel is computed as the weighted sum of the corresponding neighborhood input pixels. The weight matrix is termed as convolution kernel and is also termed as filter.

As a simple example, suppose the image is

$$A = \begin{bmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{bmatrix}$$

and the convolution kernel is

$$h = \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}$$

The following steps specify how to compute the (2, 4) output pixel:

1. The convolution kernel is rotated 180 degrees about its center element.
2. Slide the center element of the convolution kernel so that it lies on top of the (2, 4) element of A.
3. Multiply each weight in the rotated convolution kernel by the pixel of A underneath.
4. Sum all the individual products from step 3.

Hence the (2, 4) output pixel is

\[ 1.2 + 8.9 + 15.4 + 7.7 + 14.5 + 16.3 + 13.6 + 20.1 + 22.8 = 575 \]

Computing the (2, 4) Output of Convolution

![Figure 3(b): Convolution operation](image)

### 3.2.5 Labeling and Segmentation

Labeling [142] is an approach in which the subsets of connected components are labeled based on heuristics. They are used to detect the connected regions of a binary image. The vertices in the image usually contain the information as specified by the comparison heuristics. The edges in the image indicate the connected neighbors. The algorithm used for labeling, label the vertices corresponding to connectivity and the values of neighbors.

In order to improve the clarity of the image, the objects within the image need to be meaningful. The image is to be in such a way, it must be clear with sharp boundaries. Hence if the input image is not in appropriate format or the image is deformed, it needs to be simplified. Segmentation is one such process to deal with such a problem. It is also the process of assigning label to pixels in an image. Its main goal is to partition the given image into multiple segments. In this case, those pixels which share similar properties were segmented and the boundaries were also made distinct. Hence, the final image is more eminent and meaningful for further processing. Different varieties of segmentation
methods were available. Some of them are threshold based, edge based, region based and histogram based segmentation methods. The major problem in segmentation is, pixels which belong to the same object may be classified as belonging to different segment. The other case is, the pixels from different segments may be classified as belonging to the object.

Among various segmentation methods, histogram based methods proved to be very efficient. A histogram was computed from all the pixels in the image. Peaks and valleys in the histogram denote the clusters in the image. Color and intensity are the two basic measures commonly used to specify clusters. The process continues until no more clusters were found. The disadvantage is the difficulty in unique identification of peaks and valleys in the image.

3.2.6 Extraction Techniques

Various extraction techniques are available in image processing. The type of extraction technique to be used depends on need and area of application. The proposed method uses the corner extraction technique.

A corner is defined as the point of intersection between two edges. They are used to extract the corner features from the image. In an image, an interest point is a point in an image, which may be an isolated point, line ending or can be a corner. In order to detect corners, the interest points are also to be analyzed. The efficiency of the corner detector can be well identified, if it has the ability to detect the same corner with multiple images.

The SUSAN corner Extractor

SUSAN [178] is an acronym for smallest unvalue segment assimilating nucleus. The operation of Susan Corner detector is as follows. It places a circular mask over the pixel to be tested, in order to extract the desired features. The pixel is nothing but the nucleus. The mask region is represented by \( M \) and a pixel within this mask is given by \( \mathbf{m} \in M \). The nucleus point is denoted by \( \mathbf{m}_0 \). Each and every pixel in the surrounding region is then compared with the nucleus using the comparison function as,

\[
c(\mathbf{m}) = e^{-\left(\frac{(I(\mathbf{m})-I(\mathbf{m}_0))^2}{t}\right)^6}
\]  

(3.3)

where \( t \) denotes the radius, \( I \) the brightness of the pixel. The area of the SUSAN is given by the following expression:

\[
n(m) = \sum_{\mathbf{m} \in M} c(\mathbf{m})
\]  

(3.4)
If $c$ is the rectangular function, then $n$ is the number of pixels in the mask which are within $t$ of the nucleus. The response of the SUSAN operator is given by,

$$R(M) = \begin{cases} 
g - n(m) & \text{if } n(m) - g \\
0 & \text{otherwise} \end{cases}$$

(3.5)

where $g$ is the 'geometric threshold'. SUSAN operator has a positive score if the area is small enough. The value of $t$ determines how similar the points have to be to the nucleus before they are considered to be part of the univalue segment. The value of $g$ determines the minimum size of the univalue segment. If $g$ is large enough, it becomes an edge detector.

Normally there are two cases to be considered in corner detection. Initially, centroid of the SUSAN is found. In the first case, retrieved corner is the desired one. It may have the center far from the nucleus. In the second case, it must satisfy that all the points on the line from the nucleus through the centroid of the mask are in the SUSAN. It is best illustrated with the following example.

In Figure 3(c), a dark rectangle is placed on a white background. A circular mask with a centre pixel, also called nucleus was shown at five image positions a, b, c, d, e.

![Figure 3(c): Four circular masks at different places on a simple image](image)
Figure 3(d): Four circular masks with similar coloring; SUSANs are shown as the white parts of the masks

The brightness of every pixel in a mask is compared with the center of nucleus. The area in the region which has the same brightness as the nucleus is then found. This area is termed as the SUSAN. With this area, it is possible to obtain needed information about the image structure. It is even possible with these features to extract the edges. This extraction approach is totally different from other techniques as it is free from any image derivatives or noise reduction methodologies.

From the Figures 3(c) and 3(d), it is evident that the SUSAN area is maximum when the nucleus lies in a flat region of the image surface. If it is near a straight edge, it falls to half of this maximum. It may fall even further when it is inside a corner.

3.2.7 Back Propagation Neural Network

The broad area of neural network [136] is related to Artificial intelligence, machine learning, parallel processing and various other fields. The efficiency of learning in neural network is its ability to solve multiple problems efficiently over other techniques.

As a simple example, consider an object projected against the background of other objects. A human eye can easily identify the object. When considering a machine, it is not possible to identify the object in a single glance. It is so complex and it needs to undergo various processing stages. Different techniques are available to solve this complex problem. In the present days, image processing techniques, soft computing techniques, etc, were utilized.
One of the soft computing technologies used is the Back Propagation Neural Network. The back propagation algorithm [118] was introduced during the early 1970s. Several different neural networks were introduced, among which back propagation works faster than other learning approaches. From the early days and till now, it is the base and standard learning algorithm for training in neural networks.

Backpropagation neural networks [98] employ one of the most popular neural network learning algorithms, the Backpropagation algorithm. The algorithm is used for a wide variety of applications such as, pattern recognition, image analysis, etc. With this algorithm the network overcomes the limitations of existing techniques.

A trained backpropagation neural network [99] has the capability to detect and classify an input pattern even if the input was not trained. The network has the capability to adapt itself to the given set of data and deliver the output as desired. This feature is called as generalization. Neural networks were normally good at classifying the noisy input patterns.

**The Feed-Forward Neural Network Model**

The approach to construct a device with functionalities similar to that of a human brain has various limitations. It is not a simple task. The internal functionalities of the device must be similar to the functionalities of a brain. Initially, neurons were constructed and a connection is established between them. Certain networks were used to imitate the functions of dendrites, axons and synapses. The connection between the neurons is established in such a way that, each neuron in one layer is connected to every neuron in the next layer. With these functionalities, a feed forward network is constructed and used as needed. Finally the researchers met with a success in achieving the goal.

The structure of a feed forward neural network consists of three layers: the input layer, hidden layer and the output layer. Each neuron in a layer, receives its signal from the neurons of the previous layer. The input signals in each and every neuron get multiplied by its corresponding weight value in the neuron. The weighted inputs in a neuron were then summed and passed through a limiting function to obtain a range of values in the output as shown in figure 3(f). The obtained output is then passed as input to the next layer. Finally, the output will be obtained in the output layer.

To solve a problem using this network, inputs are to be applied to the input layer. This input propagates through the neurons in subsequent hidden layers and finally produce the desired output in the output layer.
The better performance of the network relies on the values of the weight between the neurons. A method is needed to adjust the weights to solve a problem. The most common type of learning algorithm used is the Back Propagation learning algorithm. Because of the generalization ability of Back propagation algorithm it allows the network to adapt itself for the given set of input data.

The working of the algorithm is as follows: the network is initially trained with a given set of sample inputs. Inputs are then presented to the network. The inputs pass through every neuron in the corresponding layers and finally produce output in the output layer. The obtained output is compared with the trained data and error value is computed. This error value propagates backwards and the weight values are adjusted in the neurons. The process continues until the error value falls below the determined threshold value. Neural networks are employed in various image processing applications like pattern recognition, character recognition, etc.
Backpropagation Processing Unit

The activation function used in Back propagation neural network [97] is the sigmoid function. The sigmoidal function; retrieve the output and the threshold point using the slope of the sigmoid function. The slope can be derived as follows:

$$\frac{d}{dx} f(x) = \frac{\exp(-x)}{(1 + \exp(-x))^2}$$

$$= \frac{1}{1 + \exp(-x)} \frac{\exp(-x)}{1 + \exp(-x)}$$

$$= \frac{1}{1 + \exp(-x)} \left[ 1 - \frac{1}{1 + \exp(-1)} \right]$$

$$= f(x)[1 - f(x)]$$

Backpropagation Learning Algorithm

Figure 3(g): Back propagation Neural Network

The following is the outline of the backpropagation learning algorithm [95]:

1. The connection weights in the neurons are first initialized into small random values.

2. The $P^{th}$ sample input vector is presented
\[ X_p = \left( X_{p1}, X_{p2}, \ldots, X_{pN} \right) \]  
(3.10)

and the corresponding output target to the network is given as,
\[ T_p = \left( T_{p1}, T_{p2}, \ldots, T_{pM} \right) \]  
(3.11)

3. The input is now passed to the first layer, layer 1. The following operation is performed, for every input node "i" in layer 0,
\[ Y_{0i} = X_{pi} \]  
(3.12)

4. For every neuron "i" in each layer, from input to output layer, the output is calculated by multiplying the input with the corresponding neurons weight value as,
\[ Y_{ji} = f\left( \sum_{k=1}^{N_{i-1}} Y_{j-1}k W_{jk} \right) \]  
(3.13)

where
\[ f(x) = \frac{1}{1 + \exp(-x)} \]  
(3.14)

5. Final output is obtained from the output layer. For every output node, "i" in layer "M", perform the following operation as in below equation 3.16:
\[ O_{pi} = Y_{Mi} \]  
(3.16)

6. The error value \( \delta_{ji} \) is then computed for every neuron "i" in backward order
\( j = M, M-1, \ldots, 2, 1 \), from output to input layer, followed by weight adjustments.

For the output layer, the error value is computed as,
\[ \delta_{Mi} = Y_{Mi} \left( 1 - Y_{Mi} \right) \left( T_{pi} - Y_{Mi} \right) \]  
(3.17)

and for hidden layers:
\[ \delta_{ji} = Y_{ji} \left(1 - Y_{ji}\right) \sum_{k=1}^{N_{i+1}} \delta_{(j+1)k} W_{(j+1)k} \]  
(3.18)

7. The weight adjustment is done for every connection from neuron "k" in layer \( i-1 \) to every neuron "j" in every layer "i",
\[ W_{ijk}^* = W_{ijk} + \beta \delta_j Y_{ji} \]  

Where \( \beta \) represents the weight adjustment factor that is normalized between 0 and 1.

8. The actions in steps 2 through 6 will be repeated for every training sample pattern "P". The process continues until the root mean square of the output error is minimized. The Root Mean Square (RMS) of the errors in the output layer is computed as,

\[ E_p = \frac{1}{2} \sum_{j=1}^{N_k} (T_{pj} - O_{pj})^2 \]  

for the "P"th sample pattern. In generalized delta rule, the error value "\( \delta_{ji} \)" associated with the "i"th neuron in layer "j" is the rate of change in the RMS error "\( E_p \)" with respect to the sum-of-product of the neuron:

\[ \delta_{ji} = -\frac{\partial E_p}{\partial \text{net}_{ji}} \]  

where "\( \text{net}_{ji} \)" represents the sum-of-product value. With the chain rule, the rate of change in the RMS error "\( E_p \)" in response to weight change is obtained as,

\[ \frac{\partial E_p}{\partial W_{ijk}} = \frac{\partial E_p}{\partial \text{net}_{ji}} \frac{\partial \text{net}_{ji}}{\partial W_{ijk}} \]  

\[ = -\delta_{ji} \frac{\partial}{\partial W_{ijk}} \left[ Y_{(j-1)0}W_{(j-1)0} + \ldots + Y_{(j-1)k}W_{(j-1)k} + \ldots \right] \]  

\[ = -\delta_{ji} Y_{(j-1)k} W_{(j-1)k} \]  

The weight change is proportional to the equation 3.25 and is given by,

\[ \Delta W_{ijk} = \beta \delta_{ji} Y_{(j-1)k} \]  

where "\( \beta \)" is a constant.
Thus, weight change can be computed as in equation 3.27, and it should match equation 3.19

\[ W_{ji}^{+} = W_{ji} + \beta \Delta W_{ji} \]  

(3.27)

Now to find an error value associated with the neuron again, use the following chain rule,

\[ \delta_{ji} = - \frac{\partial E_{p}}{\partial Y_{ji}} \frac{\partial Y_{ji}}{\partial \text{net}_{ji}} \]  

(3.28)

For output layer, \( j = M \) and \( Y_{M} = O_{pi} \). Thus,

\[ \delta_{Mi} = - \frac{\partial E_{p}}{\partial O_{pi}} \frac{\partial Y_{Mi}}{\partial \text{net}_{Mi}} \]  

(3.29)

\[ = - \frac{\partial}{\partial O_{pi}} \left[ \frac{1}{2} \left( T_{pi} - O_{pi} \right)^{2} + ... \right] \frac{\partial}{\partial \text{net}_{Mi}} f(\text{net}_{Mi}) \]  

(3.30)

\[ = - \frac{\partial}{\partial O_{pi}} \left[ \frac{1}{2} \left( T_{pi} - O_{pi} \right)^{2} \right] f'(\text{net}_{Mi}) \]  

(3.31)

Using equation (3.9),

\[ \delta_{M0} = (T_{pi} - O_{pi}) f'(\text{net}_{Mi}) (1 - f(\text{net}_{Mi})) \]  

(3.32)

\[ = (T_{pi} - O_{pi}) (O_{pi}) (1 - O_{pi}) \]  

(3.33)

This should correspond with equation 3.17. For error values associated with the hidden layer neurons, target values can’t be used. For this reason, the part \( \frac{\partial E_{p}}{\partial Y_{ji}} \) in equation 3.28 needs to be found using a different approach. The chain rule is applied to the sum-of-product values of neurons in the front layer \((j + 1)\).

\[ \frac{\partial E_{p}}{\partial Y_{ji}} = \frac{\partial E_{p}}{\partial \text{net}_{(j+1)i}} \frac{\partial \text{net}_{(j+1)i}}{\partial Y_{ji}} + \frac{\partial E_{p}}{\partial \text{net}_{(j+1)2}} \frac{\partial \text{net}_{(j+1)2}}{\partial Y_{ji}} + ... \]  

(3.34)
The, combined with \( \frac{\partial Y_{ji}}{\partial net_{ji}} \) is given by:

\[
\delta_{ji} = -\sum_{a=1}^{N_{j+1}} \left[ -\delta_{(j+1)a} W_{(j+1)ai} \right] \frac{\partial Y_{ji}}{\partial net_{ji}} 
= Y_{ji} \left( 1 - Y_{ji} \right) \sum_{a=1}^{N_{j+1}} \left[ \delta_{(j+1)a} W_{(j+1)ai} \right] 
\] (3.40)

This should concur with equation 3.18.

### 3.3 THE NETWORK STRUCTURE

The Backpropagation neural network is used for training and recognition. The structure used here, works in an iterative fashion. The network is initially trained with desired set of data, and then sample sets of data are provided for recognition. The network consists of three layers, the input layer, hidden layer and the output layer. A set of inputs pass through the input layer, hidden layer and then to the output layer. Weight value is calculated for each layer. The network then produces the output, on the output layer. The obtained output is compared with the desired output, and the mean square error value is computed. The calculated error value is then propagated backwards through the network. The weight values are adjusted for each corresponding layer in the network, to reduce the error value. The output is then calculated again. The entire process continues until the
error value drops behind the pre-determined threshold. The network structure used is as follows.

![Network Structure Diagram](image)

**Figure 3(h): The Network structure**

### 3.3.1 Computational Procedure for the Proposed Method

**Step 1:** Read the image file.

**Step 2:** Remove any noise in the image using Laplacian filter.

The laplacian value is found by,

\[
L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}
\]

Where \( I \) specifies the intensity of the pixels, and \( L(x,y) \) specifies the laplacian of the input image.

**Step 3:** Extract the images by using Susan Corner Extractor.

The output is generally calculated by,

\[
R(M) = \begin{cases} 
    g - a(M) & \text{if } a(M) < g \\
    0 & \text{otherwise} 
\end{cases}
\]

where 'M' is the mask, 'g' geometric threshold and 'n' the number of pixels.

**Step 4:** Recognition of the image is carried out with following operations in Back-propagation neural network.

\[
X_p = (X_{p1}, X_{p2}, \ldots, X_{pN})
\]

\[
T_p = (T_{p1}, T_{p2}, \ldots, T_{pNM})
\]

\[
Y_{0i} = X_{pi}
\]
\[ Y_{ji} = f\left(\sum_{k=1}^{N} Y(j-1)kW_{jik}\right) \]

\[ f(x) = \frac{1}{1 + \exp(-x)} \]

\[ O_{pi} = Y_{Mi} \]

\[ \delta_{Mi} = Y_{Mi}(1 - Y_{Mi})(T_{pi} - Y_{Mi}) \]

\[ W_{jik}^{+} = W_{jik} + \beta \delta_{j} Y_{ji} \]

\[ E_{p} = \frac{1}{2} \sum_{j=1}^{N} (T_{pj} - O_{pj})^2 \]

**Step 5:** Accuracy value is computed using Receiver Operating Characteristic curve and Confusion Matrix. True positive rate and false positive rate are computed by using,

\[ TPR = \frac{d}{c + d} \]

\[ FPR = \frac{b}{a + b} \]

**Step 1** reads the given image. Part of the image containing digits is selected for the recognition process.

**Step 2** removes any noise present in the image with laplacian filter. It also detects edges. Step 2 next performs binarization of the image. After removing any noise in the given image, it is converted into binary images with black and white pixels, using gray threshold method. Thresholding normally separates the pixels based on their intensity values into foreground and background. The `graythresh` function uses Otsu's method, which chooses the threshold to minimize the intraclass variance of the black and white pixels. Further, the selected group of numerals was separated individually after step 3 and labeling, segmentation and Normalization are performed for each separated digit. A label number is provided for each binary connected component. The labeled binary images are then segmented and normalized.

**Step 3** performs the extraction process. Susan corner detector places a circular mask over pixels that are to be tested. Every other pixel is compared to the nucleus with the help of comparison function.
Step 4 carries out the recognition process. The extracted numerals were fed into the Back Propagation neural network for the recognition process. In the final output Arabic numeral name and its English equivalent are determined.

Step 5 calculates accuracy value. The accuracy value is calculated based on receiver operating characteristics and the confusion matrix. The confusion matrix contains all the information about the actual and predicted classifications that are done with classification system. Their performance is evaluated with the data in the matrix alone. Meaning of the four entries in confusion matrix are:

- ‘a’ the number of correct predictions that an instance is negative,
- ‘b’ the number of incorrect predictions that an instance is positive,
- ‘c’ the number of incorrect predictions that an instance is negative, and
- ‘d’ the number of correct predictions that an instance is positive(TP).

From the above four cases, True positive rate and false positive rate are considered. The true positive rate is the proportion of positive cases that are correctly identified, and are calculated using,

\[
TPR = \frac{d}{c + d}
\]  

(3.41)

The false positive rate is the proportion of negative cases that are incorrectly classified as positive. They are calculated by,

\[
FPR = \frac{b}{a + b}
\]  

(3.42)
SUMMARY

The proposed method is implemented with Matlab coding. Sample images are tested with the proposed method and the results are plotted. In the proposed methodology, a new approach has been introduced for the recognition of Arabic numerals. Initially the network is trained with a standard training dataset. Sample images are fed into the neural network, which first undergoes preprocessing followed by recognition phase. A group of numerals were selected for the recognition process initially. They are separated individually after the binarization process. Labeling, segmentation and extraction are done for each of the separated numerals. The main reason for separation is that, there are certain possibilities for the pixels to get mislocated. To avoid that, a selected group of numerals is separated individually. Finally they are obtained as a whole in the output. In the final output Arabic numeral name and its English equivalent are determined. Sample handwritten images are tested with the proposed method and the results are plotted. The accuracy value is calculated based on receiver operating characteristics and the confusion matrix.

The method proposed can be implemented for real time datasets. The method can also be extended for the numerals of other languages, so that it is easy for those people who were not familiar with different languages. It can also be extended to recognize characters from various languages. The results obtained from the proposed method are discussed elaborately in chapter 6. As a further extension of the work, the next chapter 4 presents a Multiclass SVM classifier with 36 classes, to recognize characters from vehicle number plate images.