A PROJECT REPORT ON

UNIFIED CHARACTER RECOGNITION

SUBMITTED IN PARTIAL FULFILLMENT FOR THE DEGREE OF
BACHELOR IN COMPUTER ENGINEERING

SUBMITTED BY

SNEHA KAILAS GANDEWAR (5219)

NEHA JAIN (5222)

AMEY KANTILAL PATIL (5247)

UNDER THE GUIDANCE OF

MRS. ASHWINI PANSARE

DEPARTMENT OF COMPUTER ENGINEERING

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING

BANDRA(W) MUMBAI 400050

UNIVERSITY OF MUMBAI

2010-2011

CERTIFICATE

This is to certify that, Ms. Sneha Kailas Gandewar, Ms. Neha Jain, Mr. Amey Kantilal Patil have completed their project report on **Unified Character Recognition** satisfactorily in partial fulfillment under the department of Computer Engineering during the academic year **2010-2011**.

_____ _____

Internal Guide Internal Examiner

_____ _____

External Examiner Head of Department

_____

Principal

Mumbai University, Academic Year : **2010-2011**
Fr. Conceicao Rodrigues College of Engineering, Fr. Agnel Ashram, Band Stand,
Bandra (West), Mumbai - **400 050**

AKNOWLEDGEMENT

We impart special gratitude to our Principal **Dr. Srija Unnikrishnan** and **Mr. B.R. Prabhu** the H.O.D of computer department who were a constant source of help and played an important role in the successful execution of the project.

We also appreciate **Mrs. Ashwini Pansare**,our Project Guide who also put in lot of efforts in giving us the right guidance during the development process of the project. We also appreciate her eagerness and enthusiasm in encouraging us to develop our creative and technical ideas, which ultimately led to success of our project.

Our special thanks also to the non-teaching staff for their great support and kind cooperation to provide us with whatever we required for the project. We also thank our family and our friends for their support and good wishes for our project. Never to be forgotten, we thank God for granting us success in our efforts during the formation of the project.

Sneha Kailas Gandewar
Neha Jain
Amey Kantilal Patil

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

This project aims at recognizing characters from a scanned image with the help of neural network.

This project comprises of several stages all integrated towards recognizing the optical characters:*preprocessing,segmentation,feature extraction* and finally *classification* assisted by the trained neural network.

In this presentation, we shall describe the unified character recognition process step by step for each case concluding with the working and complexities of the trained neural network.

After the presentation, it would be evident how a character is recognized and we would discuss the future scope about the evolution of neural network and how we can extend this project for the further applications.

1. INTRODUCTION

1.1. PROBLEM DEFINITION

Optical character recognition (OCR) is one of the most successful applications of automatic pattern recognition. Since mid-1950s, OCR has been an active field in research and development. Today, reasonably good packages of OCR can be easily found in the commercial market. However, they are still limited to recognize high quality printed text documents. Further there are different applications exist for similar pattern recognition systems.

'Unified Character Recognition' aims at building a single application for different character recognition systems. The UCR system may contain different subsystems like

- Printed Character Recognition
- Handwritten Character Recognition
- Optical Mark Recognition
- Magnetic Ink Character Recognition
- Barcode Recognition

The high variability and complexity of handwritten words and numerals, notable deformations during the writing process, and possible noise contamination during the scanning process make HCR as a very difficult problem than PCR. But in spite of having similarities between two systems very few applications supporting both of them exist.

On the other hand, OMR, MICR and Barcode are three closely related systems having different applications. OMR can reduce manual work to great extent when checking the large number of sheets having similar format. MICR is a technique which has been developed to simplify pattern recognition and make the transactions safe, secure and accurate. Lastly, Barcode is the well established standard for secure product information transfer.

The sole aim of all the above mentioned systems is to reduce the workload and eliminate the manual process. Unified Character Recognition System would be significant to the simplification of the workflow in corporate and commercial sector. Also its use is important on individual level. Further to deal with the problem that all the forms of character recognition are not there in the single application 'Unified Character Recognition System' is proposed.

1.2. CHARACTER RECOGNITION

Character recognition systems can contribute tremendously to the advancement of the automation process and can improve the interaction between man and machine in many

applications, including office automation, cheque verification and a large variety of banking, business and data entry applications. Character Recognition is a research area in the field of Pattern Recognition that is widely used in Image Processing and Artificial Intelligence applications. It is the mechanical or electronic translation of scanned images of handwritten, typewritten or printed text into machine-encoded text. It is widely used to convert books and documents into electronic files, to computerize a record-keeping system in an office, or to publish the text on a website. Optical Character Recognition makes it possible to edit the text, search for a word or phrase, store it more compactly, display or print a copy free of scanning artifacts, and apply techniques such as machine translation, text-to-speech and text mining to it. Character Recognition is a field of research in pattern recognition, image processing, artificial intelligence and computer vision.

Character Recognition systems require calibration to read a specific font; early versions needed to be programmed with images of each character, and worked on one font at a time. "Intelligent" systems with a high degree of recognition accuracy for most fonts are now common. Some systems are capable of reproducing formatted output that closely approximates the original scanned page including images, columns and other non-textual components.

1.3. SCOPE

A Character Recognition System has a variety of commercial and practical applications in reading forms, manuscripts and their archival etc.These systems can contribute tremendously to the advancement of the automation process and can improve the interaction between man and machine in many applications, including office automation, cheque verification and a large variety of banking, business and data entry applications. The text that is either printed or handwritten can directly be transferred to the machine. It can be used as a reading machine for the visually handicapped when interfaced with a voice synthesizer. The recognition can be achieved by many methods such as dynamic programming, neural network, nearest neighbor classifier, expert system and a combination of all these techniques.

In this project efforts have been put towards unifying various pattern recognition systems and also increasing the accuracy of handwritten and printed character recognition.

This work can be further extended to the character recognition of other languages as well as for special character recognition. For doing so, sufficient training and also a sufficient number of samples are required so that network could be trained for them. Although, a lot of efforts have been made to complete a great deal of work but still it has tremendous scope for further improvement/enhancement. In future, efforts can be made to improve the recognition accuracy of the network for special characters by using more training samples. An automatic

system for recognizing all handwritten characters can also be developed. This work can be extended for word recognition in any language by including regular expressions. It can also be extended for language translation by including regular expressions and grammar.

2. REVIEW OF LITERATURE

This chapter accounts for the critical literature which was reviewed in order to plan our system and study its numerous image processing tasks. Many types of character recognition systems are studied in order to approach the problem effectively. Different algorithms are implemented and tested to check if the output is matched with the desired performance and the best suited algorithm is selected accordingly. Following sections will cover the crucial topics reviewed.

2.1. PRINTED CHARACTER RECOGNITION (PCR)

There is a clear need for optical character recognitionin order to provide a fast and accurate method tosearch both existing images as well as large archives ofexisting paper documents. However, existing opticalcharacter recognition programs suffer from a flawedtradeoff between speed and accuracy, making it lessattractive for large quantities of documents.

There are two major limitations in the implementation given by Dipayan Sarkar[1] which makeextensive testing difficult. One is speed. The feature extraction process wasimplemented in interpreted R code, which is fairly slow. For a singlepage (consisting of about 1500 glyphs), recognition can take more thanan hour on a moderately fast computer. It should be possible to improvethe performance considerably by internally using C code to do the featureextraction.The other problem was the identification of glyphs in the training stage. It was done by specifying an input image, which is segmented andthe user is asked to identify each glyph. The problem with this approach isthat some glyphs appear far more frequently than others, which may leadto a bias in the recognition step. In particular, capital letters appear veryrarely.

2.2. HANDWRITTEN CHARACTER RECOGNITION (HCR)

HCR involves the following 2 steps:

- Image compensation: This step is used in the trial to compensate the quality of the original image, enhancing certain details of the image as noise or contrast. It includes:
  a) Identification and background noise removal: A low-resolution scanned image, not clean original or a colored envelope, certainly produces a poor result. For this type of image, a threshold factor will be necessary to remove the background color by filtering.
  b) Contrast enhancement: Used to enhance bright images.
  c) Removal of spurious pixels: Can also be eliminated or reduced using the projection histograms

d) Cut: The next step is todetach the central part where we can find the objects of interest, in this in case, the digits of the postal code (CEP).

- Refine: The second level of the decision tree is still based on the projection histogram, using a new refinement rate. The elements not segmented successfully, in the succession of the refinements will be dealt with other segmentation methods in the sequence of the decision tree.

2.3. OPTICAL MARK RECOGNITION (OMR)

Automatic mark reading seems to be a relatively straightforward task of document processing and several OMR algorithms have been already available for quite a long time.

The form reading algorithms of Maciej Smiatacz are designed in such a way that there is no image preprocessing, such as filtering or skew correction. The algorithms operate on 8-bit greyscale images which means that binarization is not performed.

The main type of data, on which the algorithms directly operate, is the vertical orhorizontal projection (sometimes called a “histogram”). The analysis of such intensitydistribution profiles, extracted from the appropriate areas, is the most common techniqueused. Histogram equalization or operations can consume significant amount of time.

2.4. MAGNETIC INK CHARACTER RECOGNITION (MICR)

MICR is a specialized character recognition technology adopted by the banking industry to facilitate check processing. There are number of techniques proposed for recognition of magnetic ink characters.

MICR technique proposed by ‘Jesse Hansen’ involves several steps including pre-processing, feature extraction, and classification. The pre-processing steps proposed in this technique are binarization, morphological operations and segmentation. Binarization – Usually presented with a greyscale image, binarization is then simply a matter of choosing a threshold value. Morphological Operators are used to remove isolated specks and holes in characters. In segmentationconnectivity of shapes, label is checked and isolated.Segmentation is by far the most important aspect of the pre-processing stage. It allows the
recognizer to extract features from each individual character.

Feature extraction process identifies and extracts the useful features. Mainly moment based features such as Total mass (number of pixels in a binarized character), Centroid - Center of mass, Elliptical parameters - Eccentricity (ratio of major to minor axis) or Orientation (angle of major axis), Skewness, Kurtosis, Higher order moments.

In model estimation, using the labelledsets of features for many characters, where the labels correspond to the particular classes that the characters belong to, we wish to estimate a statistical model for each character class.

For the Classification the statistical algorithms or clustering algorithms based on statistical model estimated in previous step are used. This step can increase the computation efficiency. Also, the accuracy of the system is primarily based on this step.

2.5. BARCODE RECOGNITION

There is quite a number of research works carried out on barcode reading and localization using camera. Finding of one of these works is explained here.

Usman Ullah Sheikh[2], real-time barcode reader using Active vision. This is to decode the UPC-A and EAN-13 barcodes using an active vision system, consisting of a camera and user-written software. The camera will feed the software with continuous frames of images from the environment. These images are converted to grayscale and some preprocessing is performed. Image is filtered (such as sharpening and noise reduction) and converted to binary.

An adaptive thresholding algorithm is used to reduce the effects of uneven illumination. Image is then scanned horizontally, vertically and diagonally for barcodes, thus enabling it to decode rotated barcodes. Error correction and predictive decoding is implemented to improve the speed and accuracy of the system. Overall system performance is benchmarked with existing commercially available software.

3. REQUIREMENT ANALYSIS

3.1 TECHNICAL REQUIREMENTS

The technical requirements are as follows:

3.1.1. SOFTWARE REQUIREMENTS

- Operating Systems Microsoft® Windows XP/Vista/7
- MATLAB version 7.0.1 (R14SP1)or higher
  MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:
    - ✓ Math and computation
    - ✓ Algorithm development
    - ✓ Image Processing
    - ✓ Modeling, simulation, and prototyping
    - ✓ Data analysis, exploration, and visualization
    - ✓ Scientific and engineering graphics
    - ✓ Application development, including Graphical User Interface building

3.1.2. HARDWARE REQUIREMENTS

- Pentium Compatible CPU (P-4 or higher)
- Processor 166 MHz or above
- Disk Space Required 4 GB or more
- Memory Requirement 2GB or more

3.2 FUNCTIONAL REQUIREMENTS

- Collection of characters and their entry into database.
- Preprocessing of scanned image to eliminate the factor of noise.
- Processed images are given to segmented who will find glyphs.
- Segmented sub-images are fed to the neural network for recognition.
- Testing the neural network.
- Determining overall efficiency of the system.
- Determining overall recognition speed of the system.

3.3 DESIGN SPECIFICATION

3.3.1 EVALUATION OF THE DESIGN

The evaluation requirements are:

- ACCURACY: the acceptable property of missed recognition and the ability of humans to be able to read coherently and comprehend the meaning of the generated text.
- CONTRAINTS ON SAMPLES: Sampled images used should have text of typical type. Any font size is acceptable.
- SPEED: The speed of recognition should not be too low. An entire page of text should be scanned within 5 to 10 minutes per page depending on the number of characters in the page.

3.3.2 EVALUATION METHODS

- TESTING SPEED: Text documents of various types are tested.
- TESTING ACCURACY: Accuracy can be tested by comparing original text with recognized text.

4. DESIGN

4.1. PROJECT FLOW

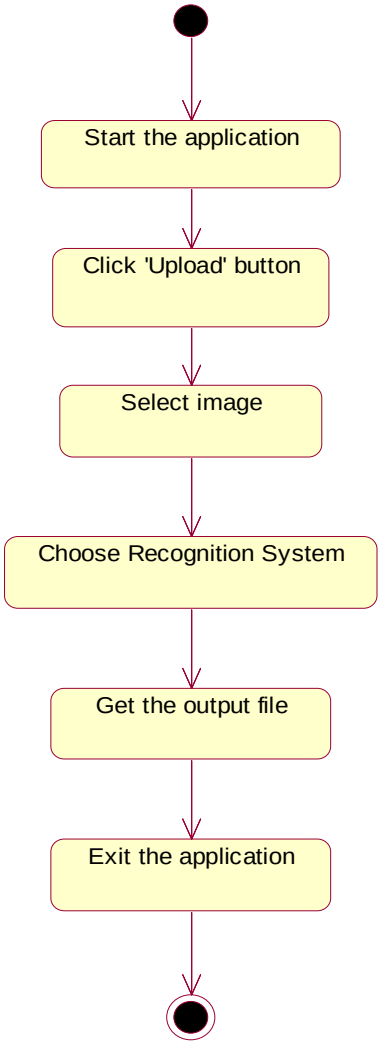


Figure 1: Flow of Project

The input image is uploaded on the system of the size according to the system's requirement. The recognition method for the image is selected from the given options. The output is displayed in a text file.

4.2. UML DIAGRAMS

4.2.1. USE CASE DIAGRAM

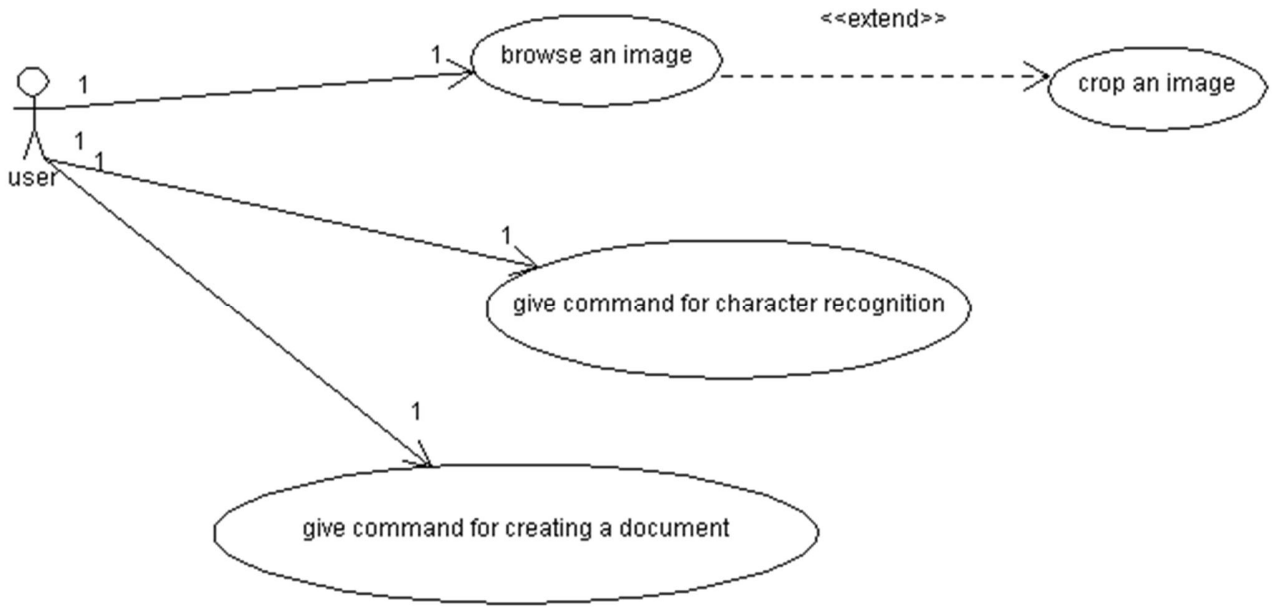


Figure 2: Use Case Diagram

4.2.2. ACTIVITY DIAGRAM

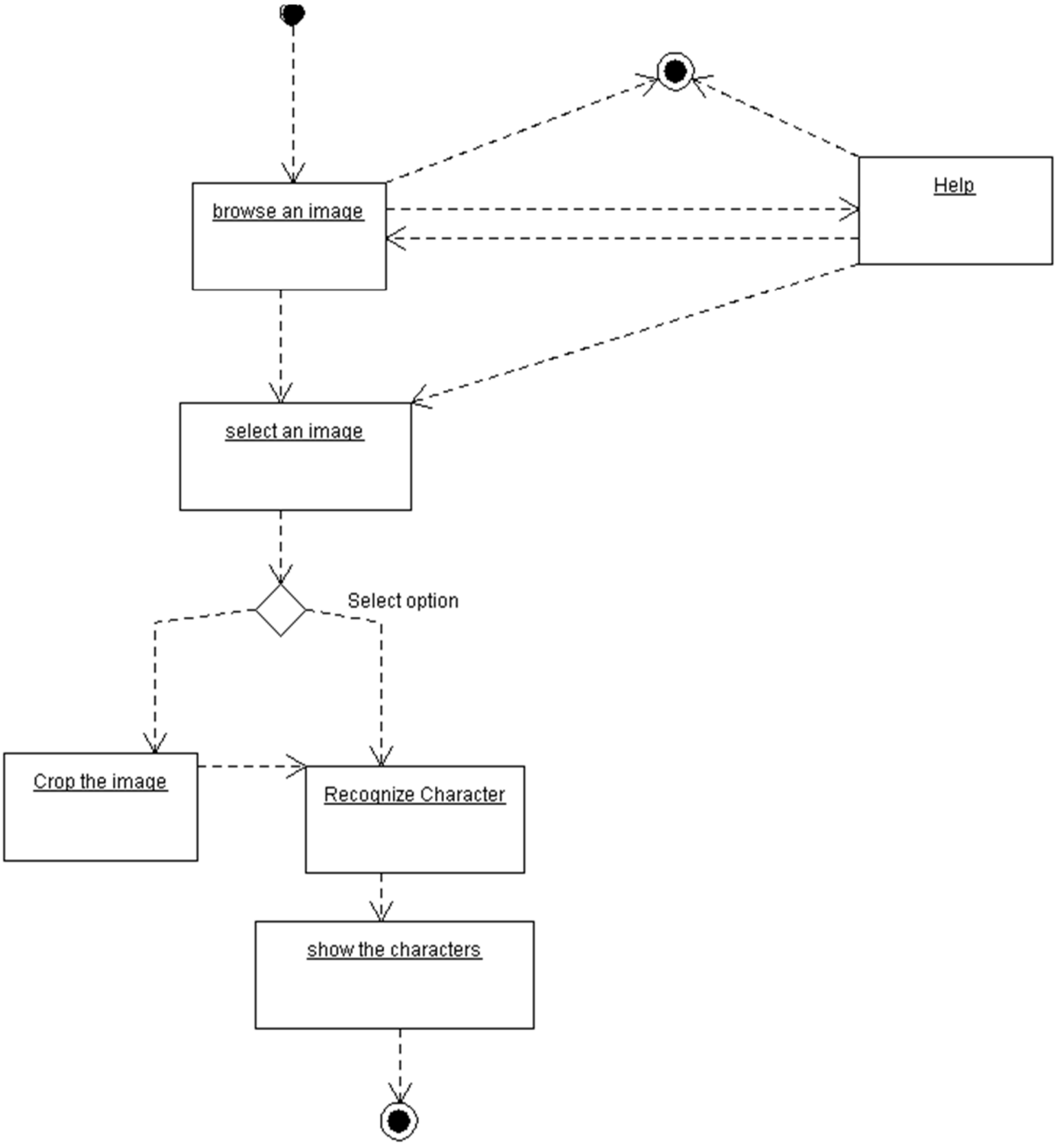


Figure 3: Activity Diagram

4.3. DATA FLOW DIAGRAMS

4.3.1. LEVEL 0

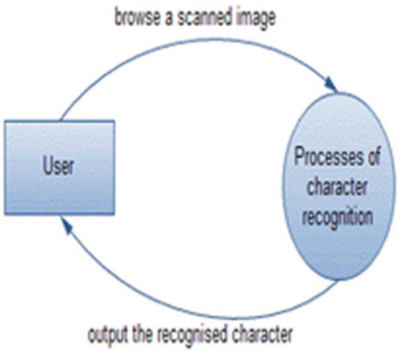


Figure 4: Level 0 DFD

4.3.2. LEVEL 1

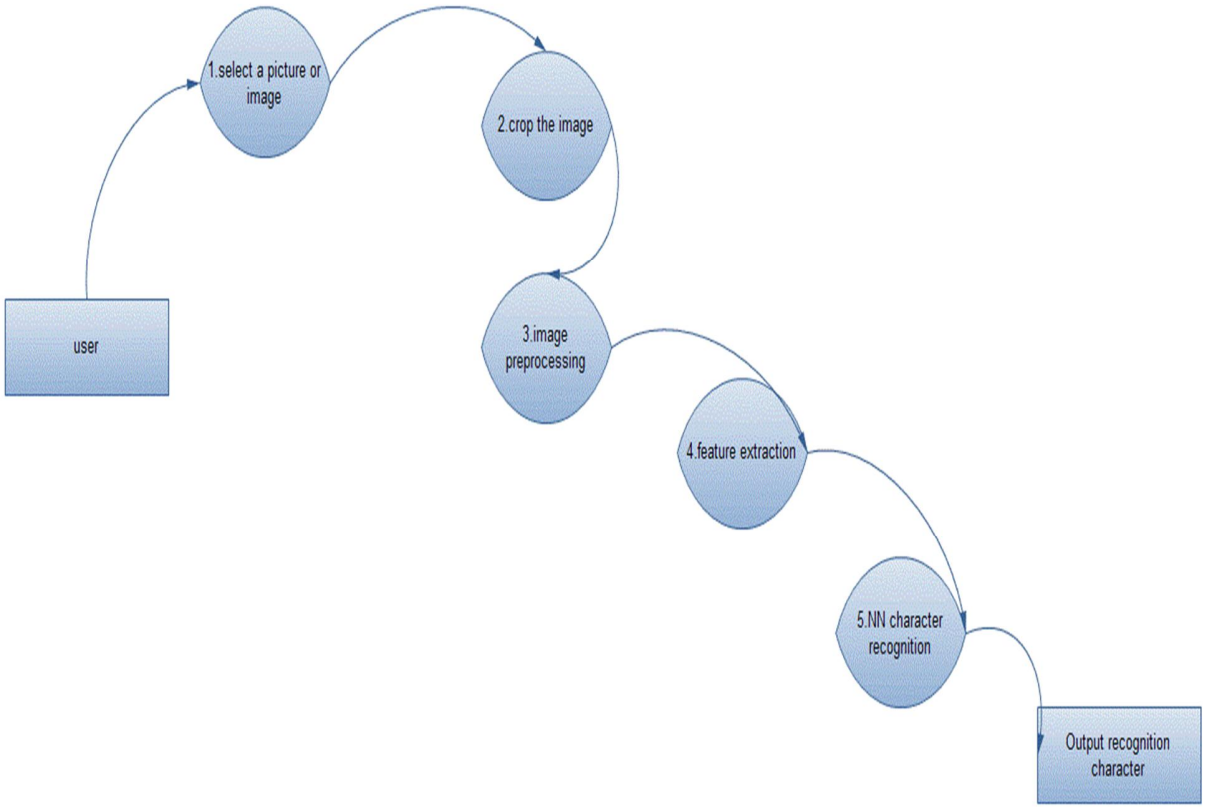


Figure 5: Level 1 DFD

4.3.3. LEVEL 2

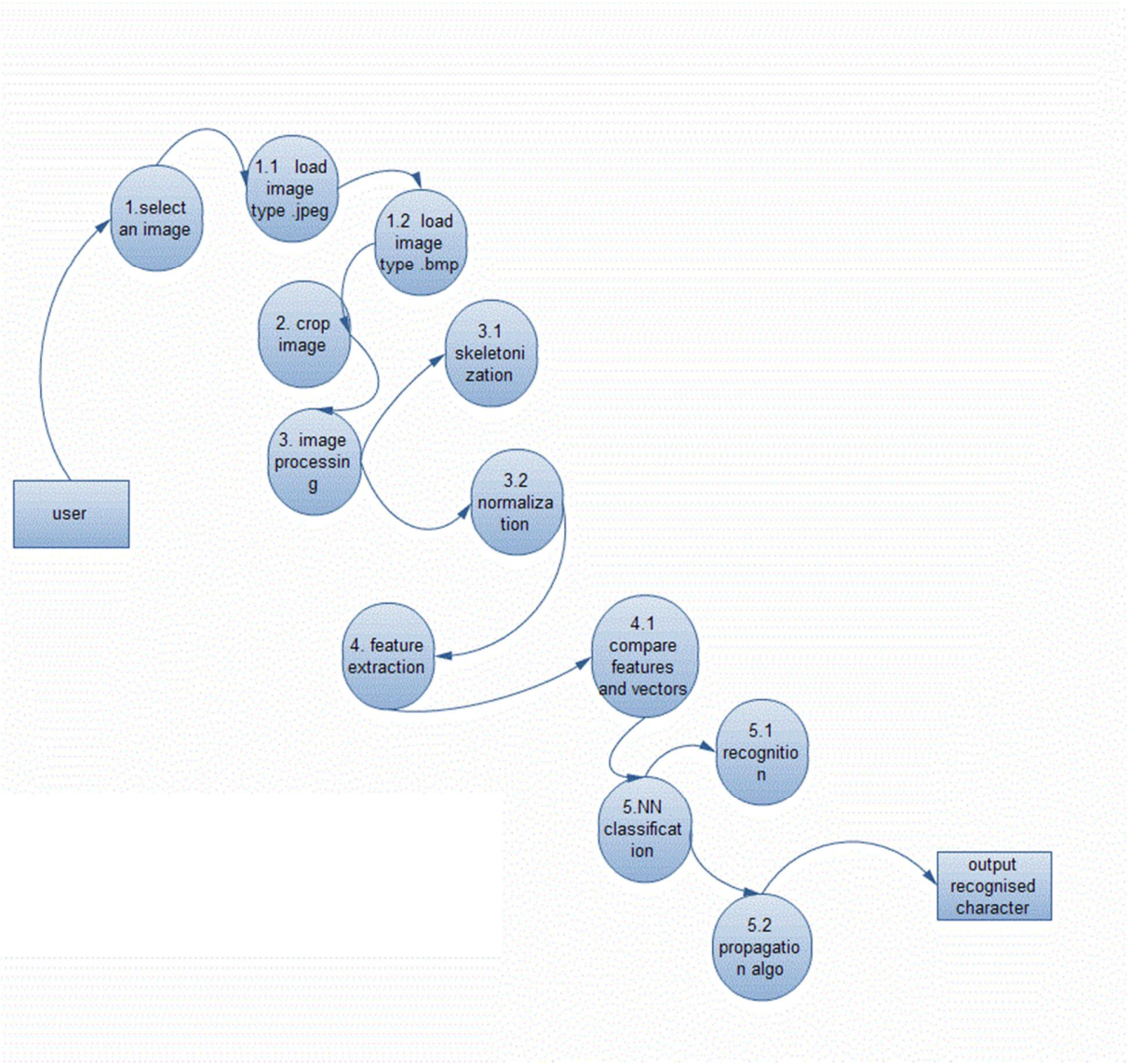


Figure 6: Level 2 DFD

5. IMPLEMENTATION

This section of the report deals with the planning of various image processing tasks and the pre-processing activities involved in the recognition system. We plan to design a system which will be accurate and fast to recognize 5 types of characters. The system should be fairly resistant to noise and should have the ability to recognize characters of different font size as well.

The scanned image is stored in jpeg format. It is digitized and this binary image is then used for feature extraction. The extracted features of the image are input to the neural network. Neural network with back propagation logic id used to train the character set which further is useful in character recognition. The accuracy of the recognition of characters depends on the minimal error in the neural network. All the characters can be recognized if the error is approximately zero.

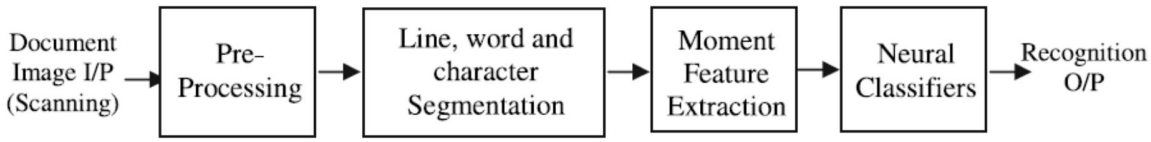


Figure 7: Plan for implementation

5.1. PRE-PROCESSING

5.1.1. BINARIZATION

Modern computers can represent over four billion colors. To represent each color, computers require thirty-two bits. For color images, this means that every pixel will consume at least four bytes of memory. However, optical character recognition is color independent—a black letter is the exact same as a red letter. Binarization is a method to reduce colored images to two colors, black and white. Black and white images only require a single bit per pixel, as opposed to thirty-two for color images. Logically, this greatly reduces the complexity of the image.

5.1.1.1. THRESHOLD ALGORITHM

One algorithm to perform binarization is the threshold algorithm [3]. This algorithm calculates an arbitrary threshold, T, which is a color. Each pixel's color is compared to the chosen

threshold. If the color is above the threshold, then the pixel is converted to a white pixel. If it is below the threshold, the pixel is a black pixel. Although fast and simple, this algorithm has a key flaw. The flaw is the reliance on calculating a single threshold for the entire image. Often the threshold is calculated by averaging the color of every pixel. However, many images may contain very light or dark text which affects the threshold in a negative way. Experimental results showed that low values of the threshold produced letters which appeared to have holes in them, because pixels that should have been black, were chosen to be white. On the other hand, higher values for the threshold produced blurry characters. One method to fix this flaw is called local binarization [2].

5.1.1.2. LOCAL BINARIZATION

Rather than calculating a threshold for the entire image at once, local binarization algorithm analyzes each pixel of the image in a small window; as small as five by five pixels. It analyzes each pixel relative to the pixels nearest it in order to convert it into a black or a white pixel. This compensates for variations in text color, as the threshold can be lower for darker text, and higher for lighter text.

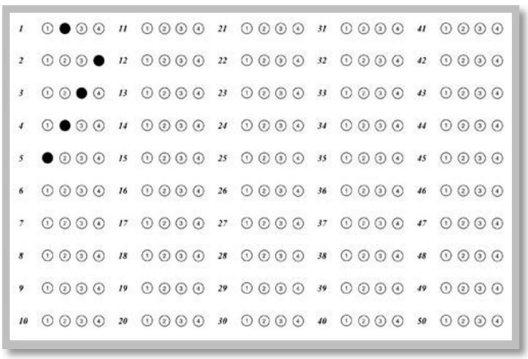


Figure 8: Original OMR Sheet

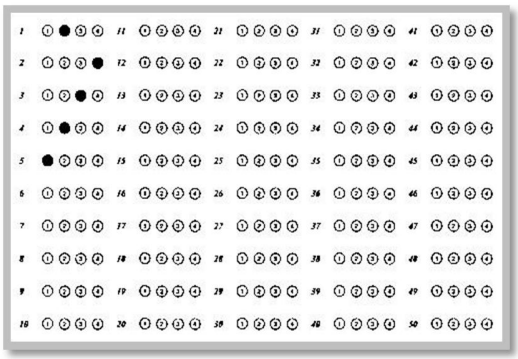


Figure 9: After Binarization

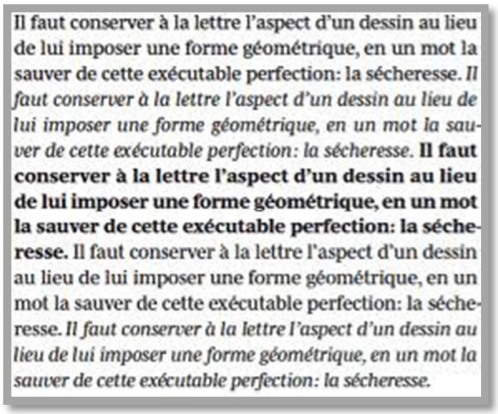

Il faut conserver à la lettre l'aspect d'un dessin au lieu de lui imposer une forme géométrique, en un mot la sauver de cette exécutable perfection: la sécheresse. *Il faut conserver à la lettre l'aspect d'un dessin au lieu de lui imposer une forme géométrique, en un mot la sauver de cette exécutable perfection: la sécheresse.* **Il faut conserver à la lettre l'aspect d'un dessin au lieu de lui imposer une forme géométrique, en un mot la sauver de cette exécutable perfection: la sécheresse.** Il faut conserver à la lettre l'aspect d'un dessin au lieu de lui imposer une forme géométrique, en un mot la sauver de cette exécutable perfection: la sécheresse. *Il faut conserver à la lettre l'aspect d'un dessin au lieu de lui imposer une forme géométrique, en un mot la sauver de cette exécutable perfection: la sécheresse.*

Figure 10: Original Printed Text

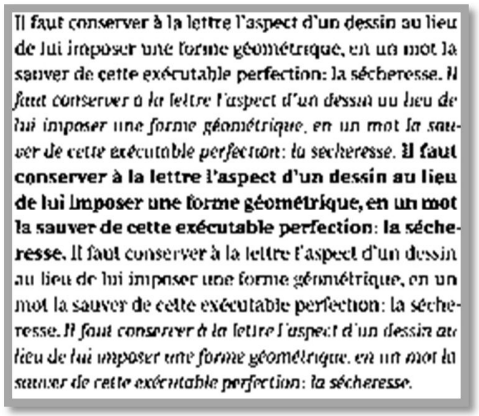

Il faut conserver à la lettre l'aspect d'un dessin au lieu de lui imposer une forme géométrique, en un mot la sauver de cette exécutable perfection: la sécheresse. *Il faut conserver à la lettre l'aspect d'un dessin au lieu de lui imposer une forme géométrique, en un mot la sauver de cette exécutable perfection: la sécheresse.* **Il faut conserver à la lettre l'aspect d'un dessin au lieu de lui imposer une forme géométrique, en un mot la sauver de cette exécutable perfection: la sécheresse.** Il faut conserver à la lettre l'aspect d'un dessin au lieu de lui imposer une forme géométrique, en un mot la sauver de cette exécutable perfection: la sécheresse. *Il faut conserver à la lettre l'aspect d'un dessin au lieu de lui imposer une forme géométrique, en un mot la sauver de cette exécutable perfection: la sécheresse.*

Figure 11: After Binarization

5.1.2. PRE-THINNING

This step aims to reduce the noise due to the binarization process. The pre-thinningalgorithm is as follows:

P3	P2	P1
P4	P	P0
P5	P6	P7

Figure 12: P and the labeling of its 8 neighbours

Input:A digitized image *I*.
Output:A pre-thinned image *I'*.

begin

1. For each pixel *P* in image *I*, let *P0* to *P7* be its 8 neighbours, starting fromthe east neighbour and counted in an anti-clockwise fashion (Figure 12).
2. Let *B(P) = P0 + P2 + P4 + P6* . Let *P'* be the corresponding pixel of *P* in thepre-thinned image I'.
3. If *B(P)* < 2 then set P' to white

Else If *B(P)* > 2 then set P' to black

Else set P' to the value of P;

End

5.1.3. THINNING

Thinning is an algorithm to further reduce the amount of information in the image to process, thereby reducing the complexity of processing the image [4]. Thinning recognizes that a thick bold letter is the exact same as a letter which is one pixel thick. Thinner letters represent the same information more efficiently.

Thinning is a simple algorithm. Moreover, it is fast and has no flaws. Each row of pixels in the image is scanned left to right. In each row, every sequence of connected black pixels is replaced by a single black pixel in the middle of the sequence. Repeated for the entire image, this technique reduces bold lines to thin, single pixel thick lines.

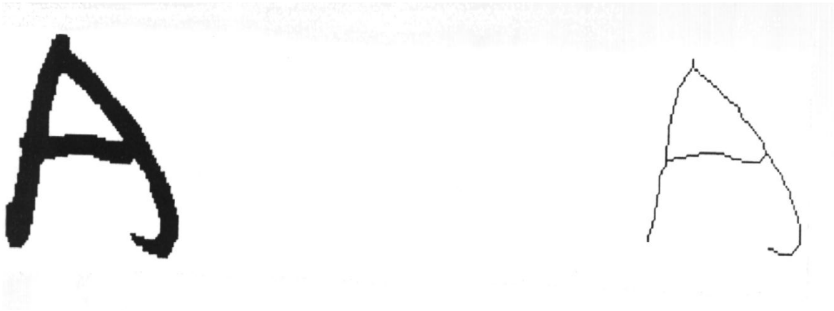


Figure 13: Thinning of character 'A'

Algorithm:

1. Divide the image into two distinct subfields in a checkerboard pattern.
2. In the first subiteration, delete pixel *p* from the first subfield if and only if the conditions G_1, G_2, and G_3 are all satisfied.
3. In the second subiteration, delete pixel *p* from the second subfield if and only if the conditions G_1, G_2, and G_3'are all satisfied.

- Condition G1:

  $X_H(p) = 1$

  Where

  $$X_H(p) = \sum_{i=1}^{4} b_i \qquad b_i = \begin{cases} 1, \text{ if } x_{2i-1} = 0 \text{ and } (x_{2i} = 1 \text{ or } x_{2i+1} = 1) \\ 0, \text{otherwise} \end{cases}$$

  x_1, x_2, ..., x_8 are the values of the eight neighbours of *p*, starting with the east neighbour and numbered in counter-clockwise order.

- Condition G2:

  $2 \le \min\{n_1(p), n_2(p)\} \le 3$

  where

  $$n_1(p) = \sum_{k=1}^{4} x_{2k-1} \vee x_{2k}$$

  $$n_2(p) = \sum_{k=1}^{4} x_{2k} \vee x_{2k+1}$$

- Condition G3:

  $(x_2 \vee x_3 \vee \bar{x}_8) \wedge x_1 = 0$

- Condition G3':

  $(x_6 \vee x_7 \vee \bar{x}_4) \wedge x_5 = 0$

The two subiterations together makeup one iteration of the thinning algorithm. When the user specifies an infinite number of iterations (n=Inf), the iterations are repeated until the image stops changing.

5.1.4. POST-THINNING

After thinning, a post-thinning is used to correct the deformation of the image caused by discontinuity in the thinned image such as gaps, cuts and holes. In order to fill the undesired gaps in the image, we need to understand the fact that the gaps create pseudo end points. From the end points, paths are explored in all directions looking for the nearest black pixel except the direction of arrival for a distance in pixels decided by a threshold limit. Eight directions are possible: East, West, North, South, North East, North West, South East and South West. If such paths can be found within the threshold limit, the pixel in the shortest path is blackened, thereby filling the gaps. This operation is illustrated by Figure 6.

Figure 14: Example of an original character and its thinned image
after applying the post-thinning algorithm.

Input: A character image I and its thinned skeleton I'.

Output: A thinned skeleton I'' with spurious branches and merged splittedfork points.

Method:

1. for each fork point Pi do
    begin
        Calculate the radius Ri of Ci, where Ci is the largest circlecentered at Pi and within the original unthinned image I;
        Create a set S containing only the point Pi ;
    end;

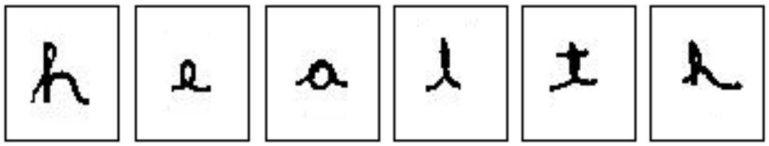
```
2. for every pair of fork points Pi and Pj do
        begin
            if (distance between Pi and Pj) <= Ri +Rj then
                Merge sets Si and Sj, i.e. sets containing Pi and Pj;
            end;
        end;

3. for each of the set S created in (2) do
        begin
            for each point Pi in S do
                begin
                    Reset pixels within the circle Ci ;
                end;
            calculate the average X and average Y of all Pi in S;
            the point with coordinate (X, Y) is the new fork point;
            rejoin the line from the perimeters of circles to the new fork point;
        end;
```

5.2. SEGMENTATION

Segmentation is a method to isolate text in an image. Specifically, it attempts to separate graphics, such as the picture of a tree, from other text contained in an image. It also attempts to separate text from other text, relying on the notion that processing one word or even one letter at a time is faster than processing the entire document at once. Ultimately, segmentation enhances optical character recognition efficiency.

In the algorithm [6], the scanned gray scale image is read into an image matrix which is converted into a monochromatic image matrix which pixel values of 0 for black points and 255 for white points. Then row wise searching is started from the point (0,0) to find out the first black point. This is the assumed top point of the first word of the handwritten text that has been inputted. This point is referred to the “Upper Point”. After the upper point is found, all the black pixels that are connected to this pixel are given a value of 999.

a. b.



c.

Figure 15(a-c): Different stages of handwriting segmentation.

Once this step is complete, then all the characters linked to that word have a value of 999 in the matrix under consideration. After finding all the connected points, a row wise search starts from the bottom to the top to find the first 999 value. This value corresponds to the "Lowest Point". After this point is obtained, the area between the top and bottom point is searched on the left to check if any word has been missed on the left. In case another word is present on the left, then the new top point is obtained and the bottom point is found again else the procedure continues. After this is carried out, the left and right points of the word are found out by column wise searches. After the four points, the Upper, Lower, Right and Left point are found out, the letter can be extracted and stored in a different matrix.

5.3. FEATURE EXTRACTION

Binary tree construction and then retrieving structural information from it is very good approach for feature extraction. But it is little complicated, though efficient, you can not guarantee its accuracy. So, rather than finding primitives from structural information and then using those primitives as the features, we directly use group of pixels as our feature.

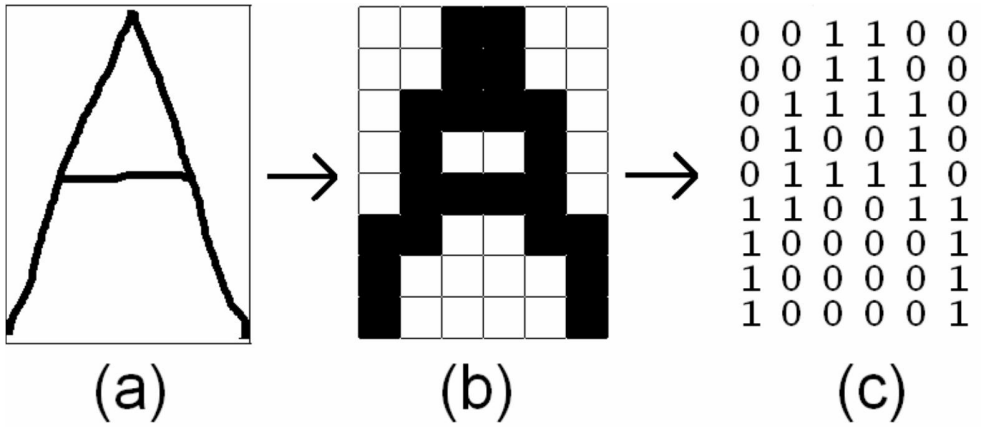


Figure 16: Feature Extraction as a group of pixels

The process of feature extraction is important here for the neural network used in the system. In this process, the input image is sampled into a binary window which forms the input to the recognition system. In the above figure, the alphabet *A* has been digitized into *6 X 8 = 48* digital

cells, each having a single colour, either black or white. It becomes important for us to encode this information in a form meaningful to a computer. For this, we assign a value *+1* to each black pixel and *0* to each white pixel and create the binary image matrix *I* which is shown in the Fig. (16.c). So much of conversion is enough for neural networking which is described next. Digitization of an image into a binary matrix of specified dimensions makes the input image invariant of its actual dimensions. Hence an image of whatever size gets transformed into a binary matrix of fixed pre-determined dimensions. This establishes uniformity in the dimensions of the input and stored patterns as they move through the recognition system.

5.4. CLASSIFICATION

A neural network is used for handwritten and printed character recognition. The neural network used in our context serves as a classification tool which learns from given examples. It has been demonstrated in several studies that neural networks are valuable classifiers for non-linear data [7]. A neural network consists of a network architecture and a learning rule for solving a given pattern recognition problem. There are several available architectures and learning methods which may be selected depending on a given problem. In this project we have selected a multi-layer perceptron architecture using the backpropagation learning scheme. These are popular neural network methods used widely for several types of problems which are solved with supervised learning. In supervised learning as with the handwritten character data, target outputs are known in advance so that the network can learn from past examples as opposed to unsupervised learning where the network attempts to cluster input data into distinct regions by itself.

5.4.1. LEARNING MECHANISM

In the proposed system, a highly simplified architecture of artificial neural networks is used. Also we are going to compare the results with Backpropagation algorithm. For purpose of easy understanding, the learning mechanism of the neural network is described first and its architecture is described next. In the used method, various characters are taughtto the network in a supervised manner. A character is presented to the system and is assigned a particular label. Several variant patterns of the same character are taught to the network under the same label. Hence the network learns various possible variations of a single pattern and becomes adaptive in nature. During the training process, the input to the neural network is the input matrix *M* defined as follows:

If $I(i, j) = 1$ *Then* $M(i, j) = 1$
Else:
If $I(i, j) = 0$ *Then* $M(i, j) = -1$

The input matrix *M* is now fed as input to the neural network. It is typical for any neural network to learn in a supervised or unsupervised manner by adjusting its weights. In the current method of learning, each candidate character taught to the network possesses a corresponding weight matrix. For the *kth* character to be taught to the network, the weight matrix is denoted by *Wk*. As learning of the character progresses, it is this weight matrix that is updated. At the commencement of teaching (supervised training), this matrix is initialized to zero. Whenever a character is to be taught to the network, an input pattern representing that character is submitted to the network. The network is then instructedto identify this pattern as, say, the *kth* character in a knowledge baseof characters. That means that the pattern is assigned a label *k*. In accordance with this, the weight matrix *Wk* is updated in the following manner:

for all i=1 to x

for all j=1 to y

$W_k(i , j) = W_k(i , j) + M(i , j)$

Here *x* and *y* are the dimensions of the matrix *Wk* (and *M*).

The following figure shows the digitization of three input patterns representing *S* that are presented to the system for it to learn.

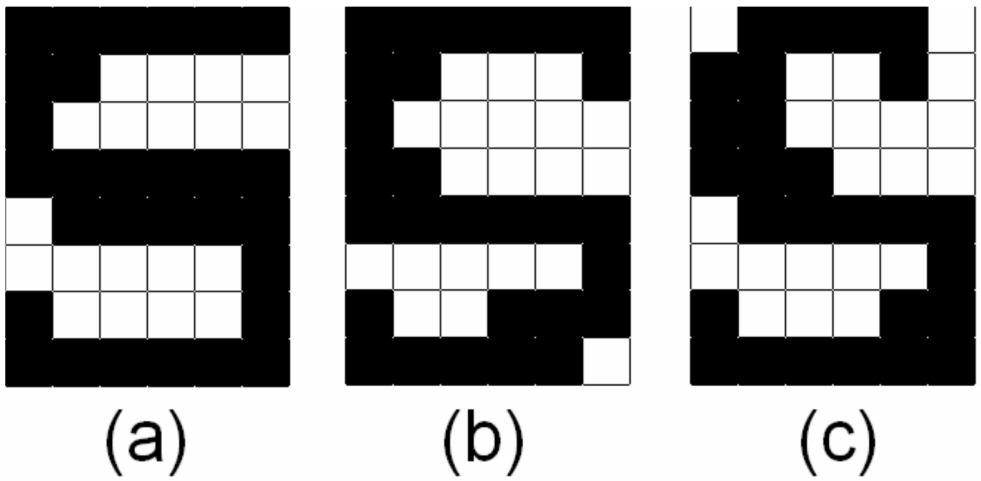


Figure 17: Input Patterns of character 'S'

Note that the patterns slightly differ from each other, just as handwriting differs from person to person (or time to time) and like printed characters differ from machine to machine.

Figure 18 gives the weight matrix, say, $W_S$corresponding to the alphabet *S.* The matrix has been updated thrice to learn the alphabet *S.* It should be noted that this matrix is specific to the alphabet *S* alone. Other characters shall each have a corresponding weight matrix.

0	0	1	1	0	0
0	0	1	1	0	0
0	1	1	1	1	0
0	1	0	0	1	0
0	1	1	1	1	0
1	1	0	0	1	1
1	0	0	0	0	1
1	0	0	0	0	1
1	0	0	0	0	1

Figure 18: Weight Matrix for 'A'

A close observation of the matrix would bring the following points to notice:
1. The matrix-elements with higher (positive) values are the ones which stand for the most commonly occurring image-pixels.
2. The elements with lesser or negative values stand for pixels which appear less frequently in the images.

Neural networks learn through such updating of their weights. Each time, the weights are adjusted in such a manner as to give an output closer to the desired output than before. The weights may represent the importanceor priorityof a parameter, which in the instant case is the occurrence of a particular pixel in a character pattern. It can be seen that the weights of the most frequent pixels are higher and usually positive and those of the uncommon ones are lower and often negative. The matrix therefore assigns importance to pixels on the basis of their frequency of occurrence in the pattern. In other words, highly probable pixels are assigned higher priority while the less-frequent ones are penalized. However, all labelled patterns are treated without bias, so as to include impartial adaptationin the system.

5.4.2. NETWORK ARCHITECTURE

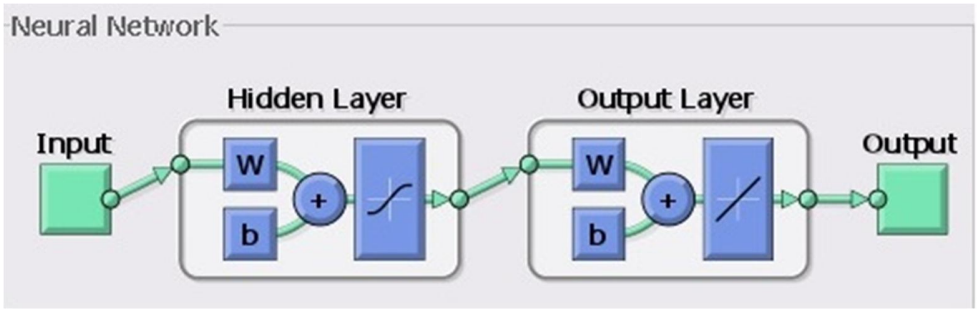


Figure 19: Neural Network Architecture

For greater accuracy in overall character recognition, use of back-propagation algorithm is proposed. The results of simplified network system and back-propagation algorithm will be compared and combined. This model is the most popular in the supervised learning architecture because of the weight error correct rules. It is considered a generalization of the delta rule for nonlinear activation functions and multilayer networks.

In a back-propagation neural network, the learning algorithm has two phases. First, a training input pattern is presented to the network input layer. The network propagates the input pattern from layer to layer until the output pattern is generated by theoutput layer. If this pattern is different from the desired output, an error is calculated and then propagated backward through the network from the output layer to the input layer. Theweights are modified as the error is propagated. The back-propagation training algorithm is an iterative gradient designed to minimize the mean square error between the actual output of multi-layer feed forward perceptron and the desired output.

5.5. USER INTERFACE

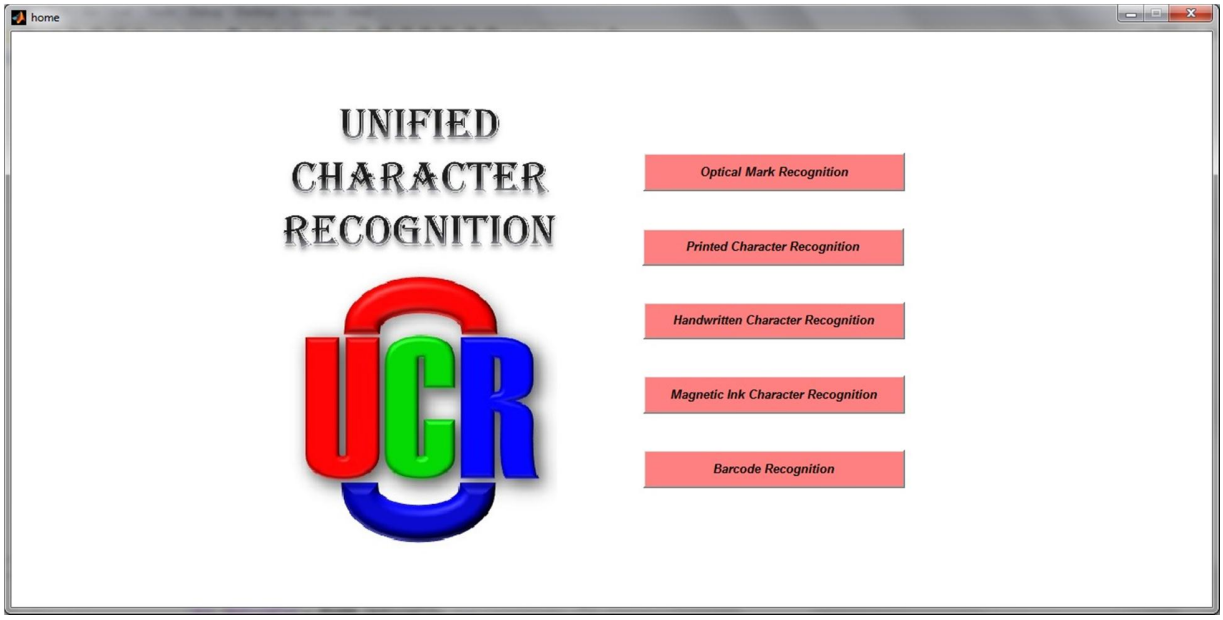


Figure 20: Home Page

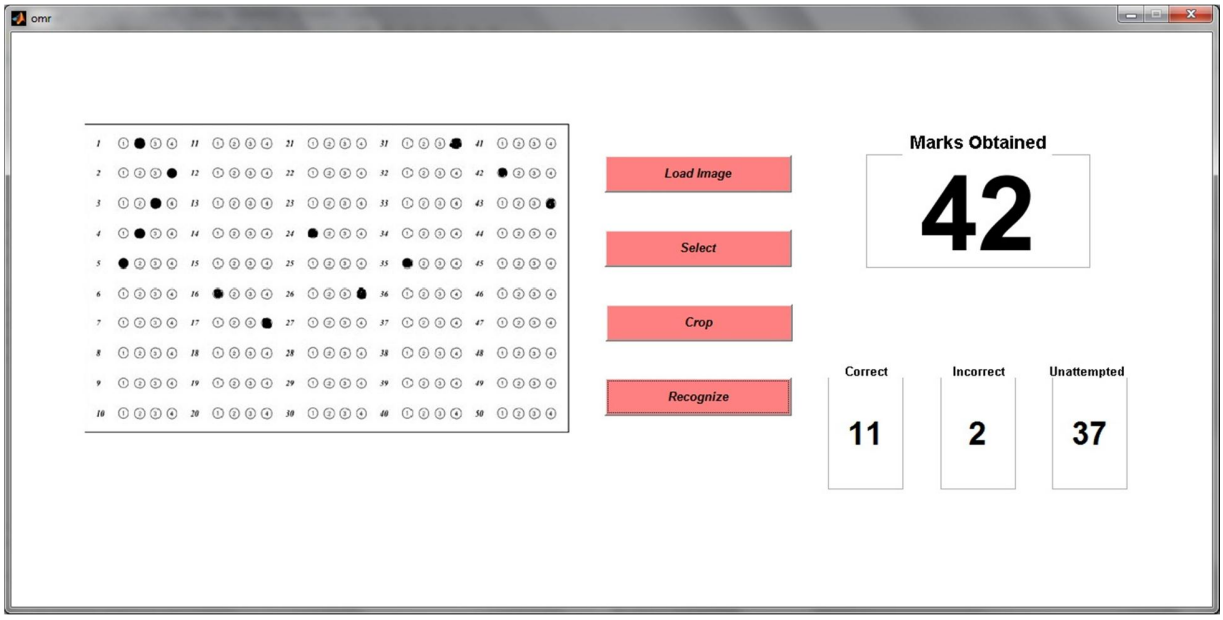


Figure 21: OMR

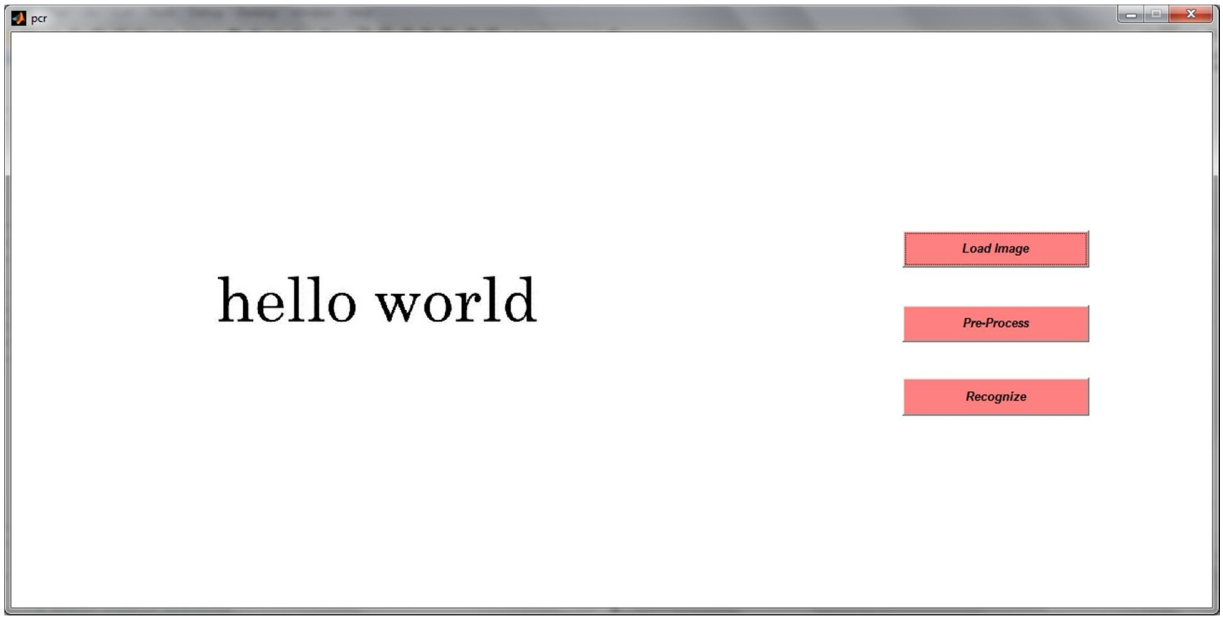


Figure 22: PCR1

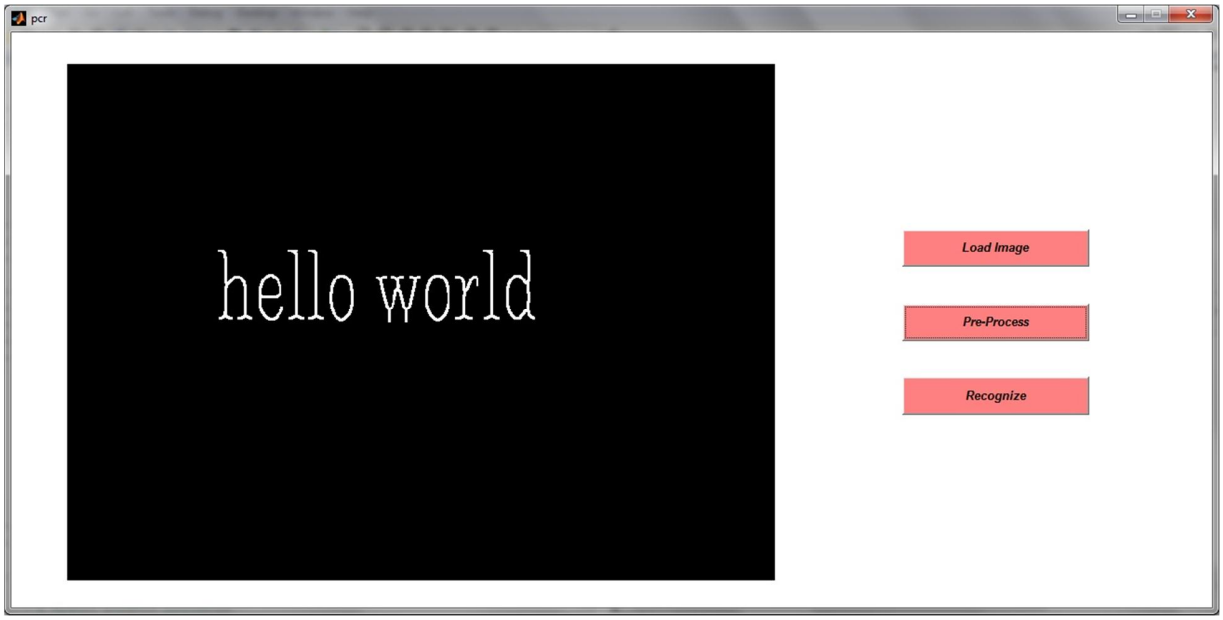


Figure 23: PCR2

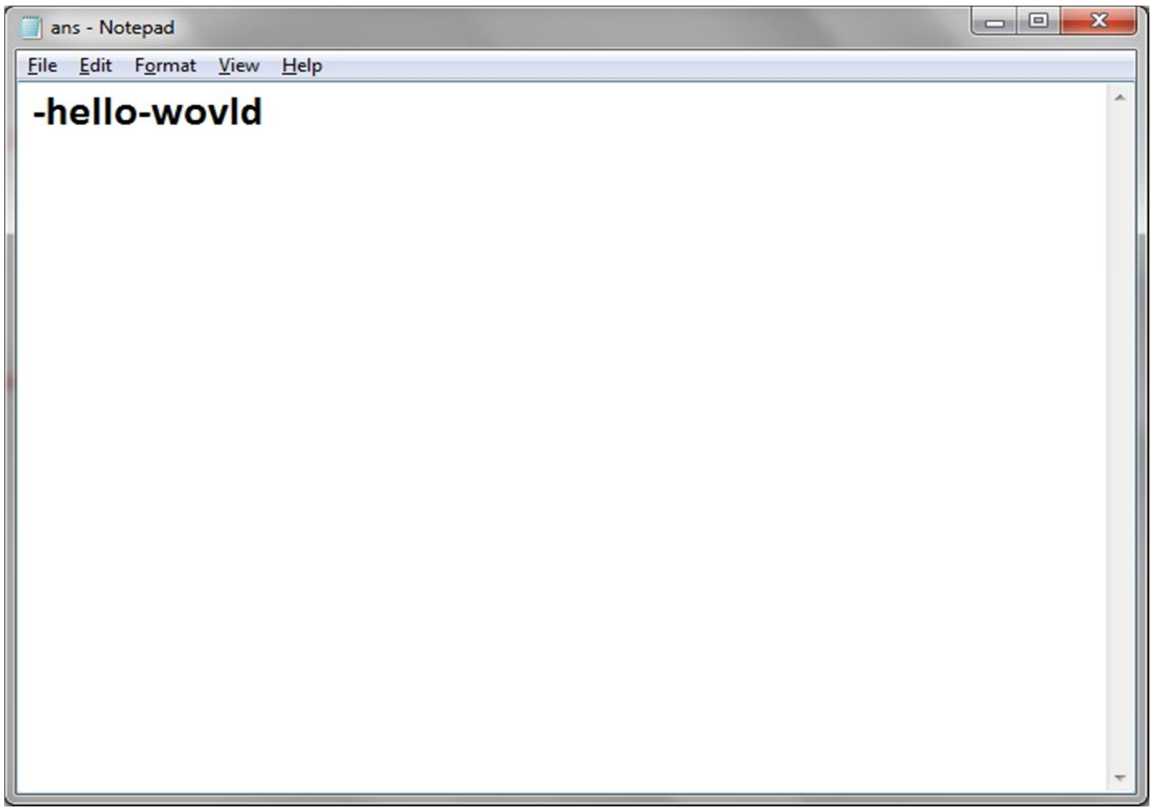


Figure 24: PCR3

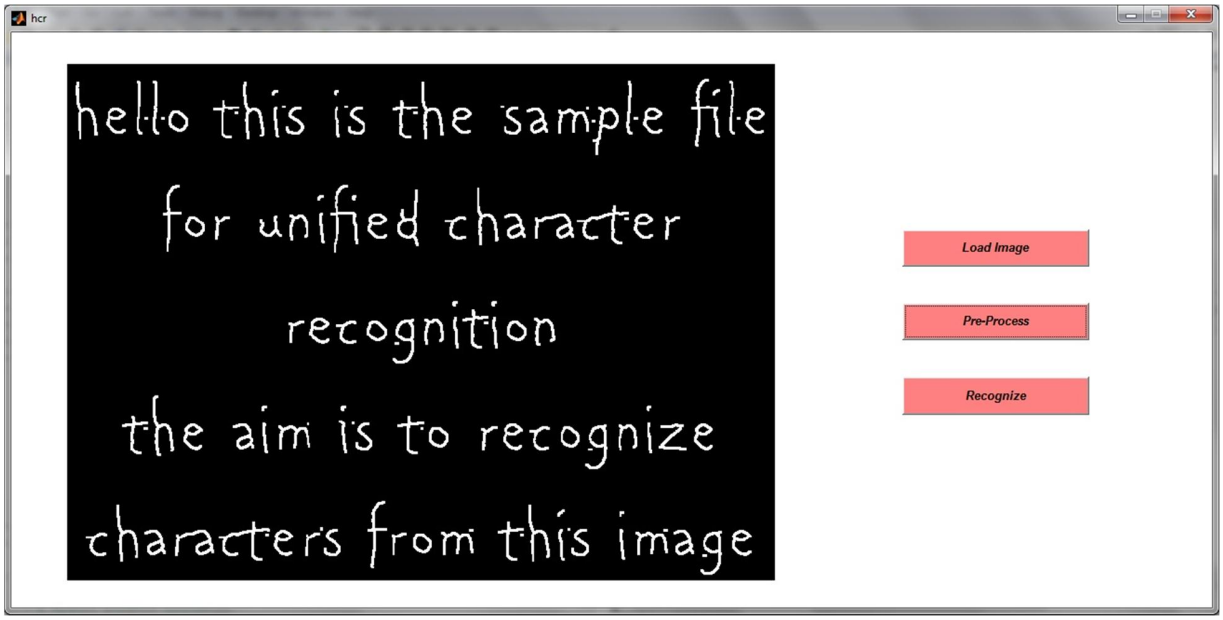


Figure 25: HCR

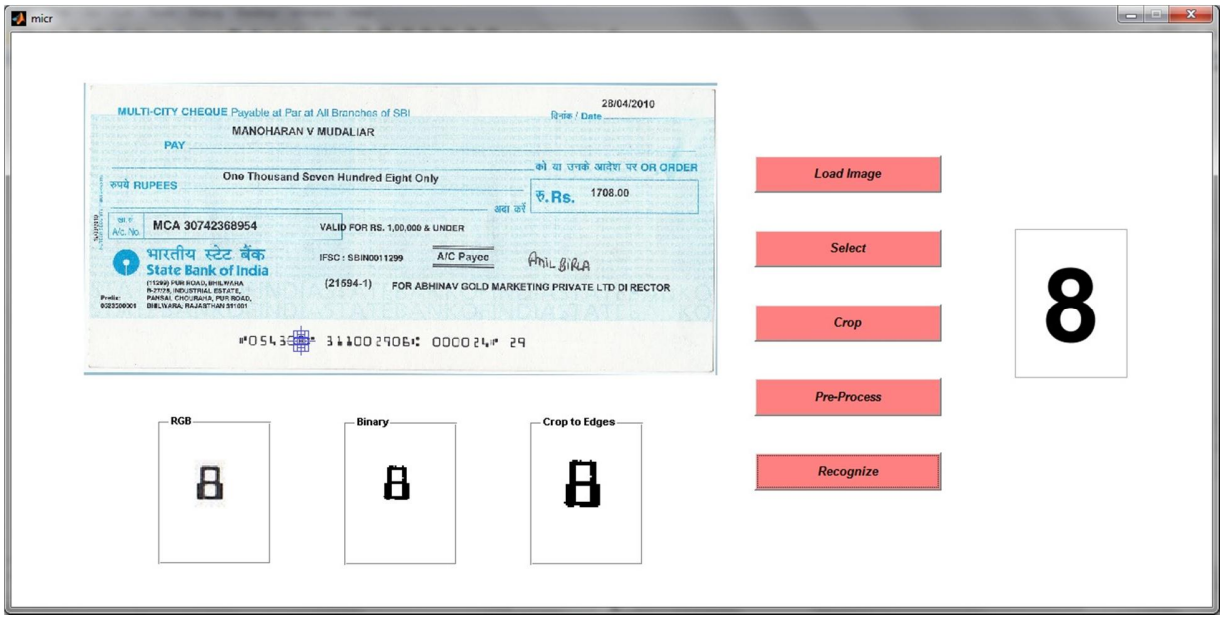


Figure 26: MICR

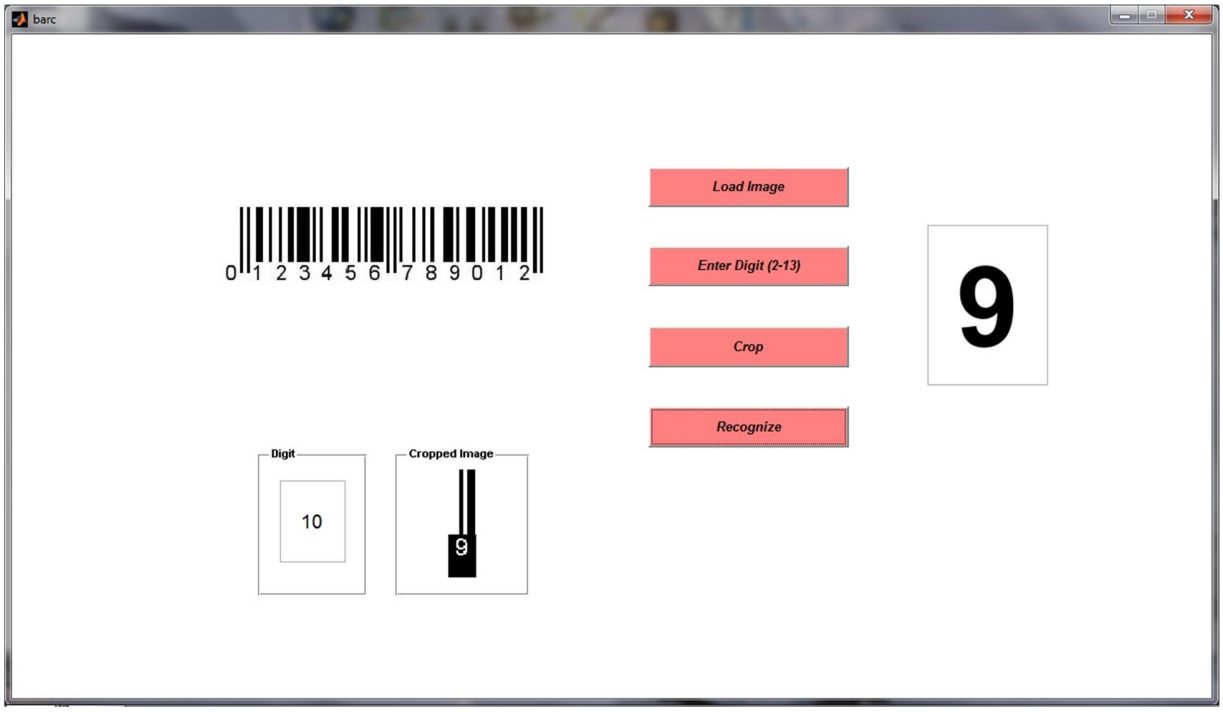


Figure 27: Barcode

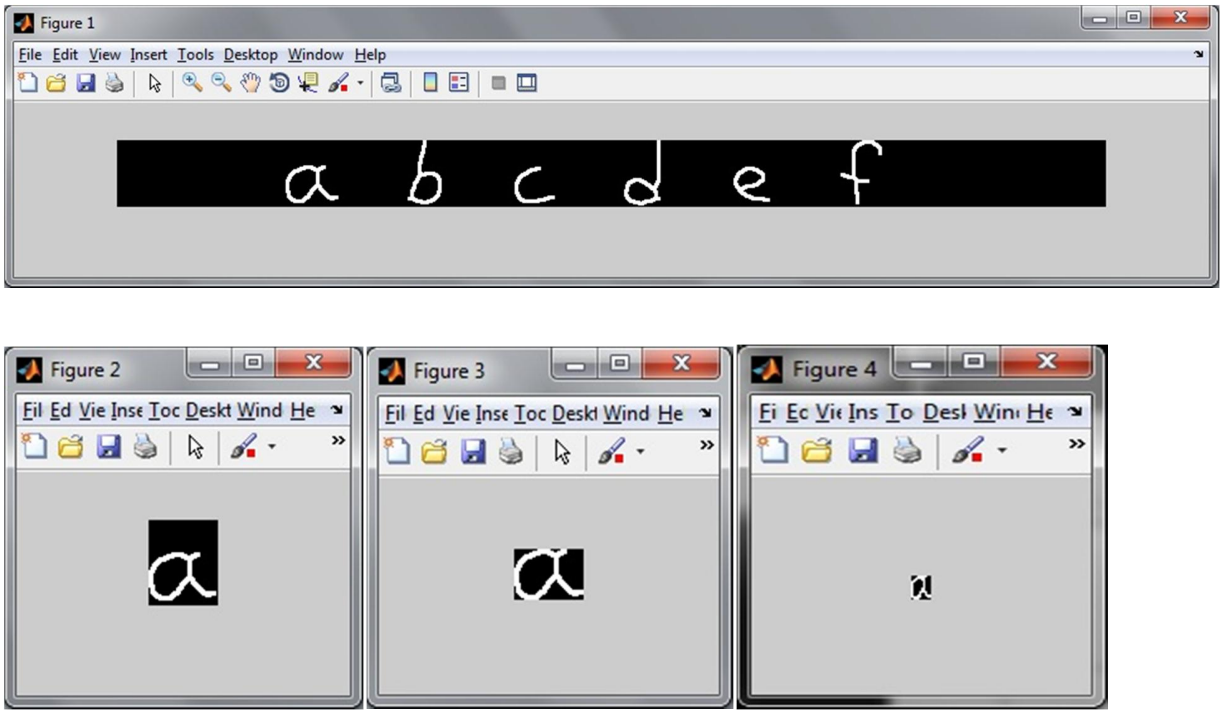


Figure 28: Segmentation

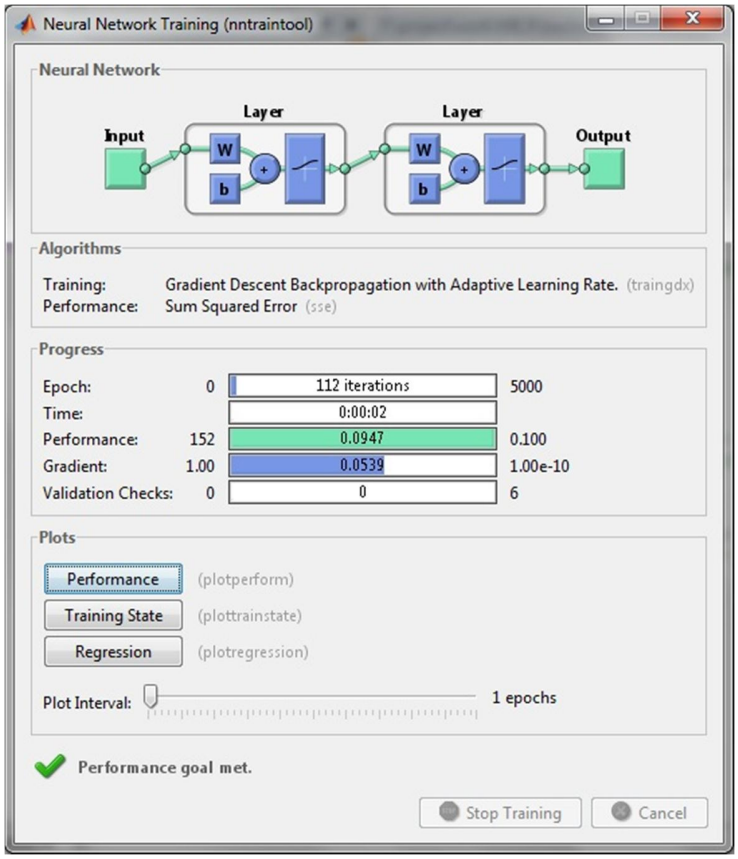


Figure 29: Training

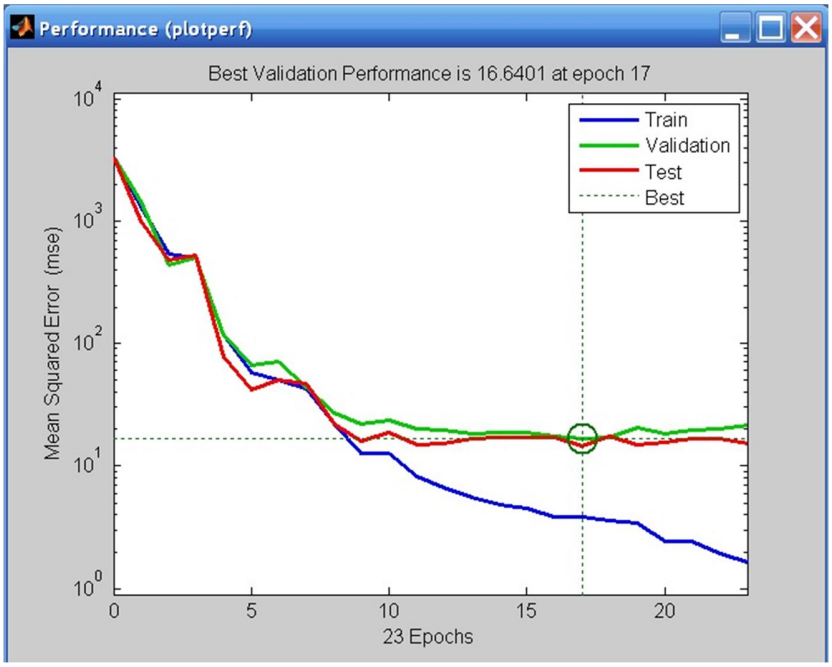


Figure 30: Performance

5.6. SOURCE CODE

Creating a Neural Network:

```
function net = edu_createnn(P,T)
alphabet = P;
targets = T;
[R,Q] = size(alphabet);
[S2,Q] = size(targets);
S1 = 30;
net = newff(minmax(alphabet),[S1 S2],{'logsig' 'logsig'},'traingdx');
net.LW{2,1} = net.LW{2,1}*0.01;
net.b{2} = net.b{2}*0.01;
net.performFcn = 'sse';
net.trainParam.goal = 0.1;
net.trainParam.show = 20;
net.trainParam.epochs = 5000;
net.trainParam.mc = 0.95;
P = alphabet;
T = targets;
[net,tr] = train(net,P,T);
```

6. TESTING AND MAINTAINENCE

6.1. TESTING

The reliability of the neural network pattern recognition system is measured by testing the network with hundreds of input vectors with varying quantities of noise. The script may test the network at various noise levels, and then graph the percentage of network errors versus noise. Noise with a mean of 0 and a standard deviation from 0 to 0.5 may be added to the input vectors. At each noise level, 100 presentations of different noisy versions of each letter are made and the network's output is calculated. The output is then passed through the competitive transfer function so that only one of the 26 outputs (representing the letters of the alphabet), has a value of 1.

Printed Character Recognition

	Net 1 16 Iterations	Net 2 25 Iterations
Train 1	22/26	21/26
Train 2	23/26	24/26
Train 3	24/26	24/26
Train 4	22/26	22/26
Train 5	20/26	20/26
Train 6	24/26	24/26
Train 7	22/26	23/26
Train 8	24/26	26/26
Train 9	22/26	22/26
Train 10	20/26	20/26
Train 11	23/26	24/26
Train 12	24/26	23/26
Sample 1	8/10	9/10
Sample 2	24/30	26/30

Magnetic Ink Character Recognition

	Net 1 35*10*10	Net 2 35*30*10
Train 1	5/10	10/10
Train 2	3/10	10/10
Train 3	2/10	10/10
Train 4	4/10	10/10
Train 5	6/10	10/10
Train 6	2/10	10/10
Sample 1	3/8	8/8
Sample 2	2/9	9/9

The number of erroneous classification is then added and the percentages are obtained. Training the network on noisy input vectors greatly reduces its errors when it has to classify noisy vectors.If a higher accuracy is needed, the network can be trained for a longer time or retrained with more neurons in its hidden layer. Also, the resolution of the input vectors can be

increased to a 10-by-14 grid. Finally, the network could be trained on input vectors with greater amounts of noise if greater reliability were needed for higher levels of noise.

Handwritten Character Recognition

	Net 1 140*30*26	Net 2 140*45*26	Net 3 140*50*26	Net 4 140*55*26	Net 5 140*60*26
Train 1	12/26	22/26	22/26	16/26	3/26
Train 2	14/26	20/26	18/26	14/26	10/26
Train 3	17/26	18/26	24/26	19/26	7/26
Train 4	12/26	15/26	20/26	17/26	10/26
Train 5	6/26	14/26	16/26	13/26	11/26
Train 6	15/26	16/26	17/26	17/26	15/26
Train 7	17/26	19/26	21/26	16/26	10/26
Sample 1	8/30	12/30	14/30	11/30	6/30
Sample 2	12/40	15/40	19/40	14/40	10/40

6.2. MAINTENANCE

Maintenance can be classified into 4 types:

- Corrective maintenance
  It means repair and performance failure or making changes because of indiscriminant previously uncorrected problems.

- Adaptive maintenance
  Over the time original environment for which the system was developed is likely to change. Adaptive maintenance results in the modification of the system to accommodate changes to its external environment.

- Perfective Maintenance
  It means enhancing the performance or modifying the program to respond to users additional or changing needs.

- Preventive Maintenance
  Computer software deteriorates due to change, because of this preventive maintenance must be conducted to enable the system to serve the needs of the end, adapted nad enhanced.

7. CONCLUSION

7.1. EXISTING SYSTEM

Many character recognition systems are available which recognize any one type of characters, that is, a system may recognize only printed characters or handwritten characters and like-wise.

Our system is such that it can accurately recognize handwritten images, printed text, OMR sheets, EAN-13 barcodes and digits written in magnetic ink. It uses three different types of neural networks for PCR, HCR and MICR with various number of hidden layers. The training algorithm used for the Artificial Neural Networks is Error Back Propagation Training (EBPT).

The omr sheet has a format of fifty questions with four options each and the system gives an output in the form of a text file. The accuracy is 100 % for the above mentioned format.

In the barcode section, our primary focus is on EAN-13 since they are most commonly used 1D barcodes. All EAN-13 barcodes are recognized accurately by the UCR system.

The MICR system has an input matrix of size 7*5 ie 35 inputs and the digits are accurately recognized.

In PCR, the printed text file is 90% accurately recognized for the input matrix of size 14*10.

The HCR section identifies an image of non-cursive handwriting that uses an input matrix of size 25*20. The accuracy is around 70-75%.

7.2. FUTURE SCOPE

Future scope of the system is to accurately recognize not only all the different five characters but also special characters like punctuation marks, block letters, &, @, #, % etc. Morever, other types of 1-D barcodes like EAN-8, UPC etc could also be included.

8. APPENDIX

8.1. BIBLIOGRAPHY

8.1.1. RESEARCH PAPERS

[1] Optical Character Recognition using NeuralNetworks, (ECE 539 Project Report), Deepayan Sarkar, December 18, 2003

[2] Usman Ullah Sheikh, Real–time barcode reader using active vision, UTM, 2004. (final project-barcode-literature).

[3] Fu Chang, “Retrieving information from document images: problems and solutions,” International Journal on Document Analysis and Recognition, Vol. 4, No. 1, August 2001, pp. 46-55, doi: 10.1007/PL00013573.

[4] Rangachar Kasturi, Lawrence O’Gorman and Venu Govindaraju, “Document image analysis: A primer,” Sadhana, Vol. 27, No. 1, February 2002, pp. 3-22, doi: 10.1007/BF02703309.

[5] Lam, L., Seong-Whan Lee, and Ching Y. Suen, "Thinning Methodologies-A Comprehensive Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 14, No. 9, September 1992, page 879, bottom of first column through top of second column.

[6] Tripathy N. and Pal U. 2006, ”Handwriting segmentation of constrained Oriya text”, Sadhna, Vol. 31, Part 6, pp. 755-769.

[7] B. Kosko, Neural Networks and Fuzzy Systems - A Dynamical Systems Approach to Machine Intelligence. Prentice Hall, 1992.

[8] Shashank Araokar,“Visual Character Recognitionusing Artificial NeuralNetworks”, University of Mumbai,India.

[9] Fakhraddin Mamedov and Jamal Fathi Abu Hasna, “Character Recognition Using Neural Networks”, Near East University, North Cyprus, Turkey via Mersin-10, KKTC

[10] Singh, S. and Amin, A. “Neural Network Recognition of Hand Printed Characters”,*NeuralComputing and Applications*, vol. 8, no. 1, pp. 67-76, *1999*.

8.2. GLOSSARY

1) Backpropagation learning rule:A learning rule in which weights and biases are adjusted by error-derivative(delta) vectors back propagated through the network.

2) Classification: An association of an input vector with a particular target vector.

3) HCR: Handwritten Character Recognition is the ability of a computer to receive and interpret handwritten text.

4) Learning: The process by which weights and biases are adjusted to achieve some desired network behavior.

5) MLP: Multilayer Perceptron is a network of simple neurons called perceptrons. The perceptron computes a single output from multiple real-valued inputs by forming a linear combination according to its input weights and then possibly putting the output through some nonlinear activation function.

6) MICR: Magnetic Ink Character Recognition , is a recognition technology used primarily by the banking industry to facilitate the processing of cheques.

7) Neuron: The basic processing element of the neural network. Includes weights and bias, a summing junction and an output transfer function.

8) NN:Neural Network is a mathematical model or computational model based on biological neural networks. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation.

9) OCR:Optical Character Recognition is the mechanical or electronic translation of images of text either printed or handwritten into machine-editable text.

10) OMR: Optical Mark Recognition is the process of capturing human-marked data from document forms such as surveys and tests.

11) Perceptron: A single layer network with a hard-limit transfer function. This network is often trained with the perceptron learning rule.

12) Training: A procedure whereby a network is adjusted to do a particular job. Training a neural network model essentially means selecting one model from the set of allowed models that minimizes the cost criterion.