Chapter 1

1 Introduction

The purpose of this project is to take handwritten English characters as input, process the character, train the neural network algorithm, to recognize the pattern and modify the character to a beautified version of the input.

This project is aimed at developing software which will be helpful in recognizing characters of English language. This project is restricted to English characters only. It can be further developed to recognize the characters of different languages. It engulfs the concept of neural network.

One of the primary means by which computers are endowed with humanlike abilities is through the use of a neural network. Neural networks are particularly useful for solving problems that cannot be expressed as a series of steps, such as recognizing patterns, classifying them into groups, series prediction and data mining.

Pattern recognition is perhaps the most common use of neural networks. The neural network is presented with a target vector and also a vector which contains the pattern information, this could be an image and hand written data. The neural network then attempts to determine if the input data matches a pattern that the neural network has memorized.

A neural network trained for classification is designed to take input samples and classify them into groups. These groups may be fuzzy, without clearly defined boundaries. This project concerns detecting free handwritten characters.

1.1 Objectives

- To provide an easy user interface to input the object image.
- User should be able to upload the image.
- System should be able to pre-process the given input to suppress the background.
- System should detect text regions present in the image.
- System should retrieve text present in the image and display them to the user.

1.2 Methods

The proposed method comprises of 4 phases:

1. Pre-processing.
2. Segmentation.
3. Feature Extraction.
4. Classification and Recognition.

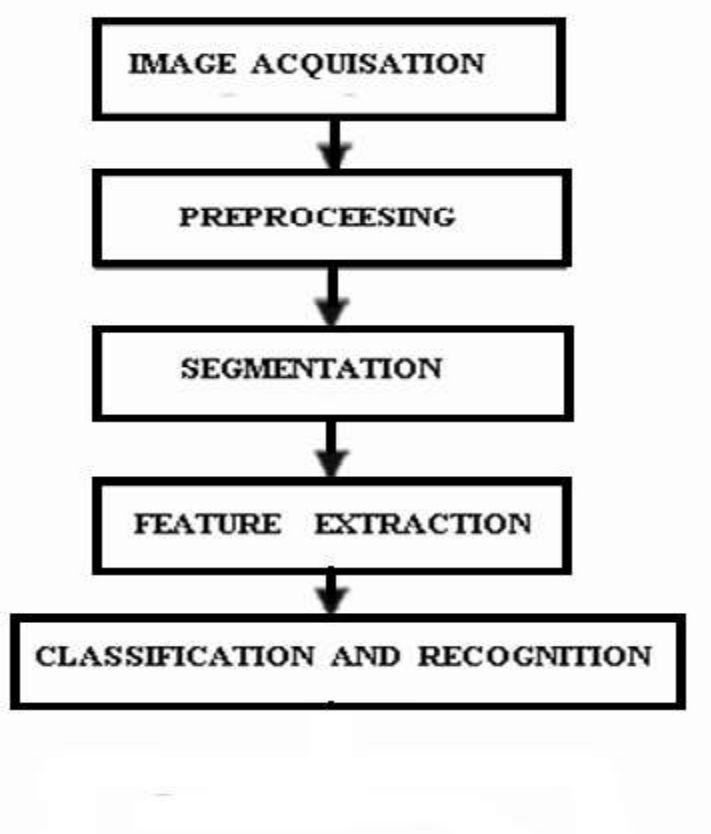


Figure 1.2.1: Process Flow

Chapter 2

2 Literature Survey

A few state of the art approaches that use hand written character recognition for text identification have been summarized here:

- **Handwritten Character Recognition using Neural Network**

  Chirag I Patel,Ripal Patel, Palak Patel.

  Objective of this paper is to recognize the characters in a given scanned documents and study the effects of changing the Models of ANN. Today Neural Networks are mostly used for Pattern Recognition task. The paper describes the behaviors of different Models of Neural Network used in OCR. OCR is widespread use of Neural Network. We have considered parameters like number of Hidden Layer, size of Hidden Layer and epochs. We have used Multilayer Feed Forward network with Back propagation. In Preprocessing we have applied some basic algorithms for segmentation of characters, normalizing of characters and De-skewing. We have used different Models of Neural Network and applied the test set on each to find the accuracy of the respective Neural Network.

- **Handwritten Character Recognition Using Gradient Features**

  AshutoshAggarwal, Rajneesh Rani, RenuDhir.

  Feature extraction is an integral part of any recognition system. The aim of feature extraction is to describe the pattern by means of minimum number of features that are effective in discriminating pattern classes. The gradient measures the magnitude and direction of thegreatest change in intensity in a small neighbourhood of eachpixel. (In what follows, "gradient" refers to both thegradient magnitude and direct ion). Gradients are computedby means of the Sobel operator.In this paper an effort is made towards recognition of English Characters and obtained recognition accuracy of 94%.Due to its logical simplicity, ease of use and high recognition rate, Gradient Features should be used for recognition purposes.

- **Character Recognition Using Matlab's Neural Network Toolbox**

  Kauleshwar Prasad, Devvrat C. Nigam, AshmikaLakhotiya and DheerenUmre.

  Recognition of Handwritten text has been one of the active and challenging areas of research in the field of image processing and pattern recognition. It has numerous applications which include, reading aid for blind, bank cheques and conversion of any hand written document into structural text form. In this paper we focus on recognition of English alphabet in a given scanned text document with the help of Neural Networks. Using Mat lab Neural Network toolbox, we tried to recognize handwritten characters by projecting them on different sized grids. The first step is image acquisition which acquires the scanned image followed by noise filtering, smoothing and normalization of scanned image, rendering image suitable for segmentation where image is decomposed into sub images. Feature Extraction improves recognition rate and misclassification. We use character extraction and edge detection algorithm for training the neural network to classify and recognize the handwritten characters.

- **Neural based handwritten character recognition**

  Hanmandlu M, Murali Mohan K.R,Kumar H.

  This paper explores the existing ring based method (W.I.Reber, 1987), the new sector based method and the combination of these, termed the Fusion method for the recognition of handwritten English capital letters. The variability associated with the characters is accounted for by way of considering a fixed number of concentric rings in the case of the ring based approach and a fixed number of sectors in the case of the sector approach. Structural features such as end points, junction points and the number of branches are used for the preclassification of characters, the local features such as normalized vector lengths and angles derived from either ring or sector approaches are used in the training using the reference characters and subsequent recognition of the test characters. The recognition rates obtained are encouraging.

- **A feature extraction technique based on character geometry for character recognition.**

  Dinesh Dileep.

  This paper describes a geometry based technique forfeature extraction applicable to segmentation-based word recognitionsystems. The proposed system extracts the geometric features of thecharacter contour. These features are based on the basic line typesthat form the character skeleton. The system gives a feature vectoras its output. The feature vectors so generated from a training setwere then used to train a pattern recognition engine based on NeuralNetworks so that the system can be benchmarked.

- **A Review of Gradient-Based and Edge-Based Feature Extraction Methodsfor Object Detection.**

  Sheng Wang.

  In computer vision research, object detection based on image processing is the task of identifying a designated object on a static image or a sequence of video frames. Projects based on such research works have been widely adapted to various industrial and social applications. The field to which those applications apply includes but not limited to, security surveillance, intelligent transportation nsystem, automated manufacturing, and quality control and supply chain management. In this paper, we are going to review a few most popular computer vision methods based on image processing and pattern recognition. Those methods have been extensively studied in various research papers and their significance to computer vision research has been proven by subsequent research works. In general, we categorize those methods into to gradient-based and edge based feature extraction methods, depending on the low level features they use. In this paper, the definitions for gradient and edge are extended. Because an image can also be considered as a grid of image patches, it is therefore reasonable to incorporate the concept of granules to gradient for a review.

Chapter 3

3 Problem Definition

The purpose of this project is to take handwritten English characters as input, process the character, train the neural network algorithm, to recognize the pattern and modify the character to a beautified version of the input.

This project is aimed at developing software which will be helpful in recognizing characters of English language. This project is restricted to English characters and numerals only. It is also helpful in recognizing special characters. It can be further developed to recognize the characters of different languages. It engulfs the concept of neural network.

One of the primary means by which computers are endowed with humanlike abilities is through the use of a neural network. Neural networks are particularly useful for solving problems that cannot be expressed as a series of steps, such as recognizing patterns, classifying them into groups, series prediction and data mining.

Pattern recognition is perhaps the most common use of neural networks. The neural network is presented with a target vector and also a vector which contains the pattern information, this could be an image and hand written data. The neural network then attempts to determine if the input data matches a pattern that the neural network has memorized.

A neural network trained for classification is designed to take input samples and classify them into groups. These groups may be fuzzy, without clearly defined boundaries. This project concerns detecting free handwritten characters.

Chapter 4

4 System Requirement Specifications

4.1 Hardware and Software Requirements

	Windows
MATLAB V.13 (R2013a)	**Windows7**
Processor	Dual core, core2duo, Intel I3
RAM	2GB RAM
DISK Space	Disk space varies depending on size of partition and installation of online help files. The MathWorks Installer will inform you of the hard disk space requirement for your particular partition
Graphics adapter	8-bit graphics adapter and display (for 256 simultaneous colors
CD-ROM drive	for installation from CD.

Table 4.1.1: Minimum Requirements

		Windows		
	Processor	**RAM**	**DISK Space**	**Graphics adapter**
MATLAB V.13 (R2013a)	Intel I3	2GB	1 GB for MATLAB only, 5 GB for a typical installation	A 32-bit or 64-bit OpenGL capable graphics adapter is strongly recommended

Table 4.1.2 Recommended Requirements

4.2 High Level Specifications

- MATLAB V.13(R2013a)
- Intel Dual core or core2duo, Intel I3XP based personal computer
- 2GB RAM recommended
- 8-bit graphics adapter and display (for 256 simultaneous colors). A 32-bit or 64bit OpenGL capable graphics adapter is strongly recommended.

4.3 Low Level Specifications

- Microsoft Windows supported graphics accelerator card, printer, and sound card.
- Microsoft Word 8.0 (Office 97), Office 2000.
- TCP/IP is required on all platforms when using a license server.
- Some license types require a license server running FLEXlm 8.0d, which is provided by the Math Works installer.

4.4 Functional Requirements

- The system should process the input given by the user only if it is an image file (JPG, PNG etc.)
- System shall show the error message to the user when the input given is not in the required format.
- System should detect characters present in the image.
- System should retrieve characters present in the image and display them to the user.

4.5 Non Functional Requirements

- Performance: Handwritten characters in the input image will be recognized with an accuracy of about 90% and more.
- Functionality: This software will deliver on the functional requirements mentioned in this document.
- Availability: This system will retrieve the handwritten text regions only if the image contains written text in it.
- Flexibility: It provides the users to load the image easily.
- Learn ability: The software is very easy to use and reduces the learning work.
- Reliability: This software will work reliably for low resolution images and not for graphical images.

Chapter 5

5 System Modeling and Design

Purpose

The purpose of this design document is to explore the logical view of architecture design, sequence diagram, data flow diagram, user interface design of the software for performing the operations such as pre-processing, extracting features and displaying the text present in the images.

Scope

The scope of this design document is to achieve the features of the system such as pre-process the images, feature extraction, segmentation and display the text present in the image.

5.1 Block Diagram and Algorithm

The proposed methodology uses some techniques to remove the background noise, andfeatures extraction to detect and classify the handwritten text.

The proposed method comprises of 4 phases:

1. Pre-processing.
2. Segmentation.
3. Feature Extraction.
4. Classification and Recognition.

The block schematic diagram of the proposed model is given in Fig.5.1.1

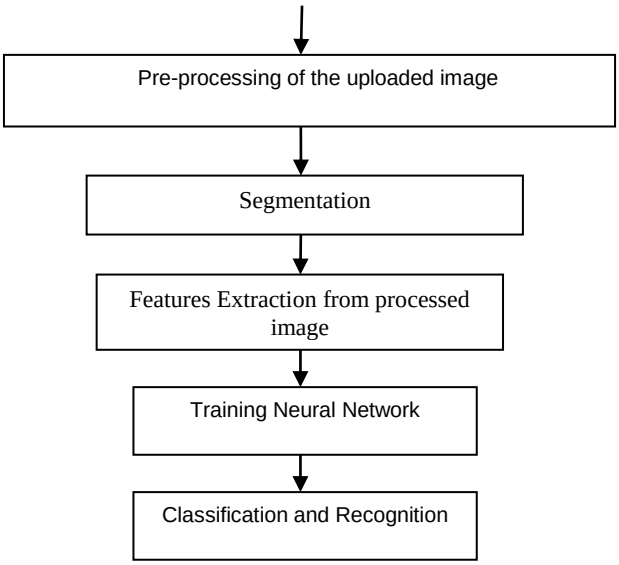


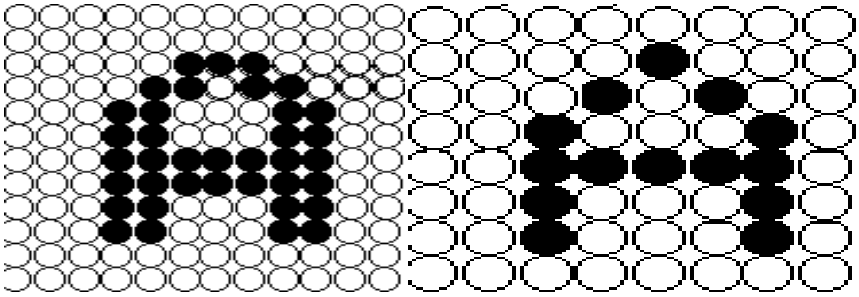
Figure 5.1.1: Block diagram of proposed method

5.1.1 Pre-processing

The pre-processing is a series of operations performed on scanned input image. It essentially enhances the image rendering it suitable for segmentation. The role of pre-processing is to segment the interesting pattern from the background. Generally, noise filtering, smoothing and normalization should be done in this step. The pre-processing also defines a compact representation of the pattern. Binarization process converts a gray scale image into a binary image. Dilation of edges in the binarized image is done using sobel technique.

5.1.2 Segmentation

In the segmentation stage, an image of sequence of characters is decomposed into sub-images of individual character. The pre-processed input image is segmented into isolated characters by assigning a number to each character using a labelling process. This labelling provides information about number of characters in the image. Each individual character is uniformly resized into pixels. Normalization: Afterextracting the character we need to normalize the size of the characters. There are large variations in the sizes of each Character hence we need a method to normalize the size.



Original Image **Normalized Image**

Figure 5.1.2.1: Normalization of Image

Character Extraction Algorithm

1. Create a Traverse List: - List of pixels which have been already traversed. This list is initially empty.
2. Scan row Pixel-by-Pixel.
3. Whenever we get a black pixel check whether the pixel is already in the traverse list, if it is simply ignore and move on else apply Edge-detection Algorithm.
4. Add the List of Pixels returned by Edge-detection Algorithm to Traverse List.
5. Continue the steps 2 - 5 for all rows.

Edge Detection Algorithm

The Edge Detection Algorithm has a list called traverse list. It is the list of pixel already traversed by the algorithm.

```
EdgeDetection(x,y,TraverseList);
     1) Add the current pixel to TraverseList. The
current position of pixel is (x,y).
     2) NewTraverseList= TraverseList + current
position(x,y).
          If pixel at (x-1,y-1) then
               Check if it is not in TraverseList.
               Edgedetection(x-1,y-1,NewTraverseList);
          end if
          If pixel at (x-1,y) then
               Check if it is not in TraverseList.
```

```
            Edgedetection(x-1,y+1,NewTraverseList);
        end if
        If pixel at (x,y+1) then
            Check if it is not in TraverseList.
            Edgedetection(x,y+1,NewTraverseList);
        Endif
    3)return
```

5.1.3 Feature Extraction

There are two techniques employed based on the efficiencies obtained, while training the neural network. They are as follows

- Feature Extraction based on Character Geometry.
- Feature Extraction Using Gradient Features.

5.1.3.1 Feature Extraction Based on Character Geometry.

It extracts different line types that form a particular character. It also concentrates on the positional features of the same. The feature extraction technique explained was tested using a Neural Network which was trained with the feature vectors obtained from the system proposed.

Universe of Discourse

Universe of discourse is defined as the shortest matrix that fits the entire character skeleton. The Universe of discourse is selected because the features extracted from the character image include the positions of different line segments in the character image. So every character image should be independent of its Image size.

Original Image **Universe of Discourse**

Figure 5.1.3.1.1: Universe of Discourse

Zoning

After the universe of discourse is selected, the image is divided into windows of equal size, and the feature is done on individual windows. For the system implemented, two types of zoning were used. The image was zoned into 9 equal sized windows. Feature extraction was applied to individual zones,rather than the whole image. This gives more information about fine details of character skeleton. Also positions of different line segments in a character skeleton become a feature if zoning is used. This is because, a particular line segment of a character occurs in a particular zone in almost cases. For instance, the horizontal line segment in character 'A' almost occurs in the central zone of the entire character zone.

To extract different line segments in a particular zone, the entire skeleton in that zone should be traversed. For this purpose, certain pixels in the character skeleton were defined as starters, intersections and minor starters.

Starters

Starters are those pixels with one neighbour in the character skeleton. Before character traversal starts, all the starters in the particular zone is found and is populated in a list.

Figure 5.1.3.1.2: Starters are rounded

Intersections

The definition for intersections is somewhat more complicated. The necessary but insufficient criterion for a pixel to be an intersection is that it should have more than one neighbour. A new property called true neighbours is defined for each pixel. Based on the number of true neighbours for a particular pixel, it is classified as an intersection or not. For this, neighbouring pixels are classified into two categories, Direct pixels and diagonal pixels. Direct pixels are all those pixels in the neighbourhood of the pixel under consideration in the horizontal and vertical directions. Diagonal pixels are the remaining pixels in the neighbourhood which are in a diagonal direction to the pixel under consideration. Now for finding number of

true neighbours for the pixel under consideration, it has to be classified further based onthe number of neighbours it have in the character skeleton. Pixels under consideration are classified as those with 3 neighbours: If any one of the direct pixels is adjacent to anyone of the diagonal pixels, then the pixel under consideration cannot be an intersection, else if none of the neighbouring pixels are adjacent to each other than its an intersection. 4 neighbours: If each and every direct pixel has an adjacent diagonal pixel or vice-versa, then the pixel under consideration cannot be considered as an intersection. 5 or neighbours: If the pixel under consideration has five or more neighbours, then it is always considered as an intersection once all the intersections are identified in the image, then they are populated in a list.

Figure 5.1.3.1.3: Intersections

Minor Starters

Minor starters are found along the course of traversal along the character skeleton. They are created when pixel under consideration have more than two neighbours. There are two conditions that can occur Intersections: When the current pixel is an intersection. The current line segment will end there and all the unvisited neighbours are populated in the minor starters list. Non-intersections: Situations can occur where the pixel under consideration has more than two neighbours but still it's not an intersection. In such cases, the current direction of traversal is found by using the position of the previous pixel. If any of the unvisited pixels in the neighbourhood is in this direction, then it is considered as the next pixel and all other pixels are populated in the minor starters list. If none of the pixels is not in the currentdirection of traversal, then the current segment is ended there and the entireneighbourhood are populated in the minor starters list.

When the algorithm proposed is applied to character 'A', in most cases, the minor starters found are given in the image.

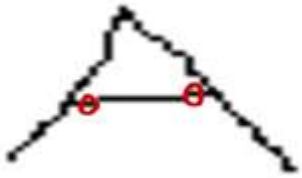


Figure 5.1.3.1.4: Minor Starters

After the line type of each segment is determined, feature vector is formed based on this information. Every zone has a feature vector corresponding to it. Under the algorithm proposed, every zone has a feature vector with a length of 8.

The contents of each zone feature vector are:

1) Number of horizontal lines.
2) Number of vertical lines.
3) Number of Right diagonal lines.
4) Number of Left diagonal lines.
5) Normalized Length of all horizontal lines.
6) Normalized Length of all vertical lines.
7) Normalized Length of all right diagonal lines.
8) Normalized Length of all left diagonal lines.
9) Normalized Area of the Skeleton.

The number of any particular line type is normalized using the following method,Value = 1 - ((number of lines/10) x 2).

Normalized length of any particular line type is found using the following method,Length = (Total Pixels in that line type)/ (Total zone pixels).

The feature vector explained here is extracted individually for each zone. So if there are N zones, there will be 9N elements in feature vector for each zone. For the system proposed, the original image was first zoned into 9 zones by dividing the image matrix. The features were then extracted for each zone. Again the original image was divided into 3 zones by dividing in the horizontal direction. Then features were extracted for each such zone.

After zonal feature extraction, certain features were extracted for the entire image based on the regional properties namely Euler Number: It is defined as the difference of Number of Objects and Number of holes in the image. For instance, a perfectly drawn ’A’ would have Euler number as zero, since number of objects is 1 and number of holes is 2, whereas ‘B’ would have Euler number as -1, since it have two holes

Regional Area: It is defined as the ratio of the number of the pixels in the skeleton to the total number of pixels in the image. Eccentricity: It is defined as the eccentricity of the smallest ellipse that fits the skeleton of the image.

5.1.3.2 Gradient Feature Extraction.

The gradient measures the magnitude and direction of the greatest change in intensity in a small neighbourhood of each pixel. (In what follows, "gradient" refers to both the gradient magnitude and direction). Gradients are computed by means of the Sobel operator. The Sobel templates used to compute the horizontal (X) & vertical (Y) components of the gradient are shown in Fig.

1	2	1
0	0	0
-1	-2	-1

-1	0	1
-2	0	2
-1	0	1

Horizontal Component Vertical Component

Figure 5.1.3.2.1: Sobel masks for Gradient

Given an input image of size D1×D2, each pixel neighbourhood is convolved with these templates to determine these X and Y components, Sx and Sy, respectively.
Eq. (1) and (2) represents their mathematical representation:
(1)

$$S(i, j) = I(i-1, j+1) + 2 * I(i, j+1) + I(i+1, j+1)$$
$$-I(i-1,j-1)-2*I(i,j-1)-I(i+1,j-1). \quad (1)$$

$$S(i, j) = I(i-1, j-1) + 2* I(i-1, j) + I(i-1, j+1)\ y\ -I(i+1,j-1) - 2* I(i+1, j) - I(i+1, j+1)$$

(2)

Here, (i, j) range over the image rows (D1) and columns (D2), respectively. The gradient strength and direction can be computed from the gradient vector [Sx, Sy].
After obtaining gradient vector of each pixel, the gradient image is decomposed into four orientation planes or eight direction planes (chain code directions) as shown in

Fig.3.

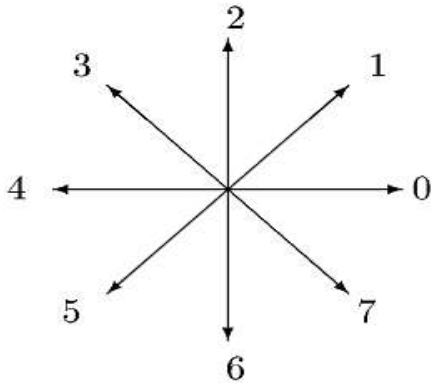


Figure 5.1.3.2.2: directions of chain codes

Generation of Gradient Feature Vector

A gradient feature vector is composed of the strength of gradient accumulated separately in different directions as described below: (1) the direction of gradient detected as above is decomposed along 8 chain code directions. (2) The character image is divided into 81(9 horizontal × 9 vertical) blocks. The strength of the gradient is accumulated separately in each of 8 directions, in each block, to produce 81 local spectra of direction. (3) The spatial resolution is reduced from 9×9 to 5×5 by down sampling every two horizontal and every two vertical blocks with 5×5 Gaussian Filter to produce a feature vector of size 200 (5 horizontal, 5 vertical, 8 directional resolution). (5) The variable transformation (y = x0.4) is applied to make the distribution of the features Gaussian-like. The 5 × 5 Gaussian Filter used is the high cut filter to reduce the aliasing due to the down sampling.

5.1.4 Classification

Artificial Neural Network

Animals recognize various objects and make sense out of large amount of visual information, apparently requiring very little effort. Simulating the task performed by animals to recognize to the extent allowed by physical limitations will be enormously profitable for the system. This necessitates study and simulation of Artificial Neural Network. In Neural Network, each node perform some simple computation and each connection conveys a signal from one node to another labelled by a number called the “connection strength” or weight indicating the extent to which signal is amplified or diminished by the connection.

Different choices for weight results in different functions are being evaluated by the network. If in a given network whose weight are initial random and given that we know the task to be accomplished by the network , a learning algorithm must be used to determine the values of the weight that will achieve the desired task. Learning Algorithm qualifies the computing system to be called Artificial Neural Network. The node function was predetermined to apply specific function on inputs imposing a fundamental limitation on the capabilities of the network. Typical pattern recognition systems are designed using two pass. The first pass is a feature extractor that finds features within the data which are specific to the task being solved (e.g. finding bars of pixels within an image for character recognition). The second pass is the classifier, which is more general purpose and can be trained using a neural network and sample data sets. Clearly, the feature extractor typically requires the most design effort, since it usually must be hand-crafted based on what the application is trying to achieve.

Back propagation was created by generalizing the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer functions. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as defined by you. Networks with biases, a sigmoid layer, and a linear output layer are capable of approximating any function with a finite number of discontinuities.

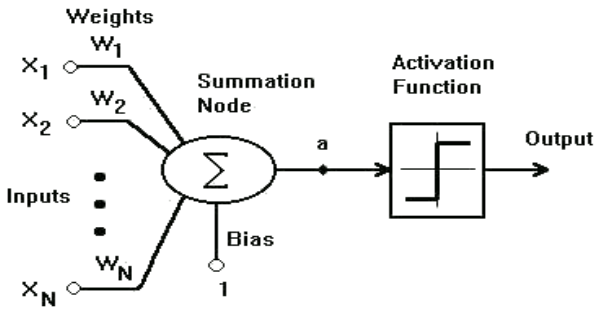


Figure 5.1.4.1: Typical Neural Network

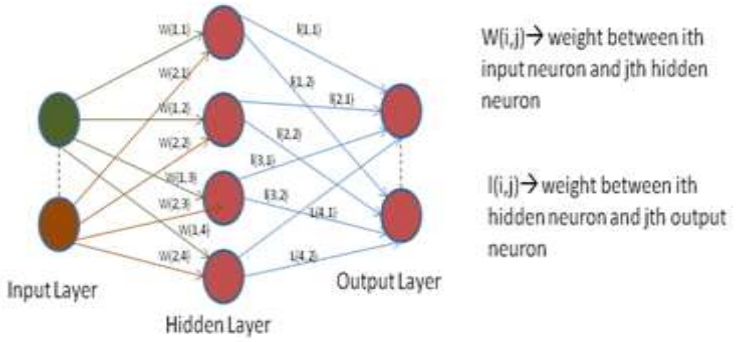
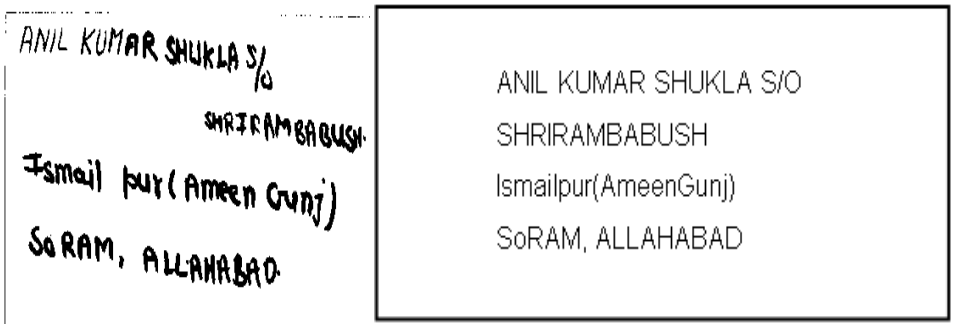


Figure 5.1.4.2: Neural Network

Once the network is trained, the match pattern is obtained to generate the associated character.



Sample Input **Sample Output**

Figure 5.1.4.3: Sample Input & Output

Output will be the beautified version of the uploaded image and will be saved in a .doc or in text file.

5.2 Use of tools for design

Image Processing Toolbox

Image Processing Toolbox™ provides a comprehensive set of reference-standard algorithms, functions, and apps for image processing, analysis, visualization, and algorithm development. You can perform image enhancement, image deblurring, feature detection, noise reduction, image segmentation, geometric transformations, and image registration. Many toolbox functions are multithreaded to take advantage of multi core and multiprocessor computers.

Image Processing Toolbox supports a diverse set of image types, including high dynamic range, giga pixel resolution, embedded ICC profile and tomography. Visualization functions let you explore an image, examine a region of pixels, adjust the contrast, create contours or histograms, and manipulate regions of interest (ROIs). With toolbox algorithms you can restore degraded images, detect and measure features, analyze shapes and textures, and adjust color balance. Image Processing Toolbox is included in MATLAB and Simulink Student Version.

Neural Network Toolbox

Neural Network Toolbox™ provides functions and apps for modeling complex nonlinear systems that are not easily modeled with a closed-form equation. Neural Network Toolbox supports supervised learning with feedforward, radial basis, and dynamic networks. It also supports unsupervised learning with self-organizing maps and competitive layers. With the toolbox you can design, train, visualize, and simulate neural networks. You can use Neural Network Toolbox for applications such as data fitting, pattern recognition, clustering, time-series prediction, and dynamic system modeling and control.

Rational Rose

Rational Rose is an object-oriented Unified Modeling Language (UML) software design tool intended for visual modeling and component construction of enterprise-level software applications. In much the same way a theatrical director blocks out a play, a software designer uses Rational Rose to visually create (model) the framework for an application by blocking out classes with actors (stick figures), use case elements (ovals), objects (rectangles) and messages/relationships (arrows) in a sequence diagram using drag-and-drop symbols. Rational Rose documents the diagram as it is being constructed and then generates code in the designer's choice of C++, Visual Basic, Java, Oracle8, Corba or Data Definition Language.

5.3 Flow Chart

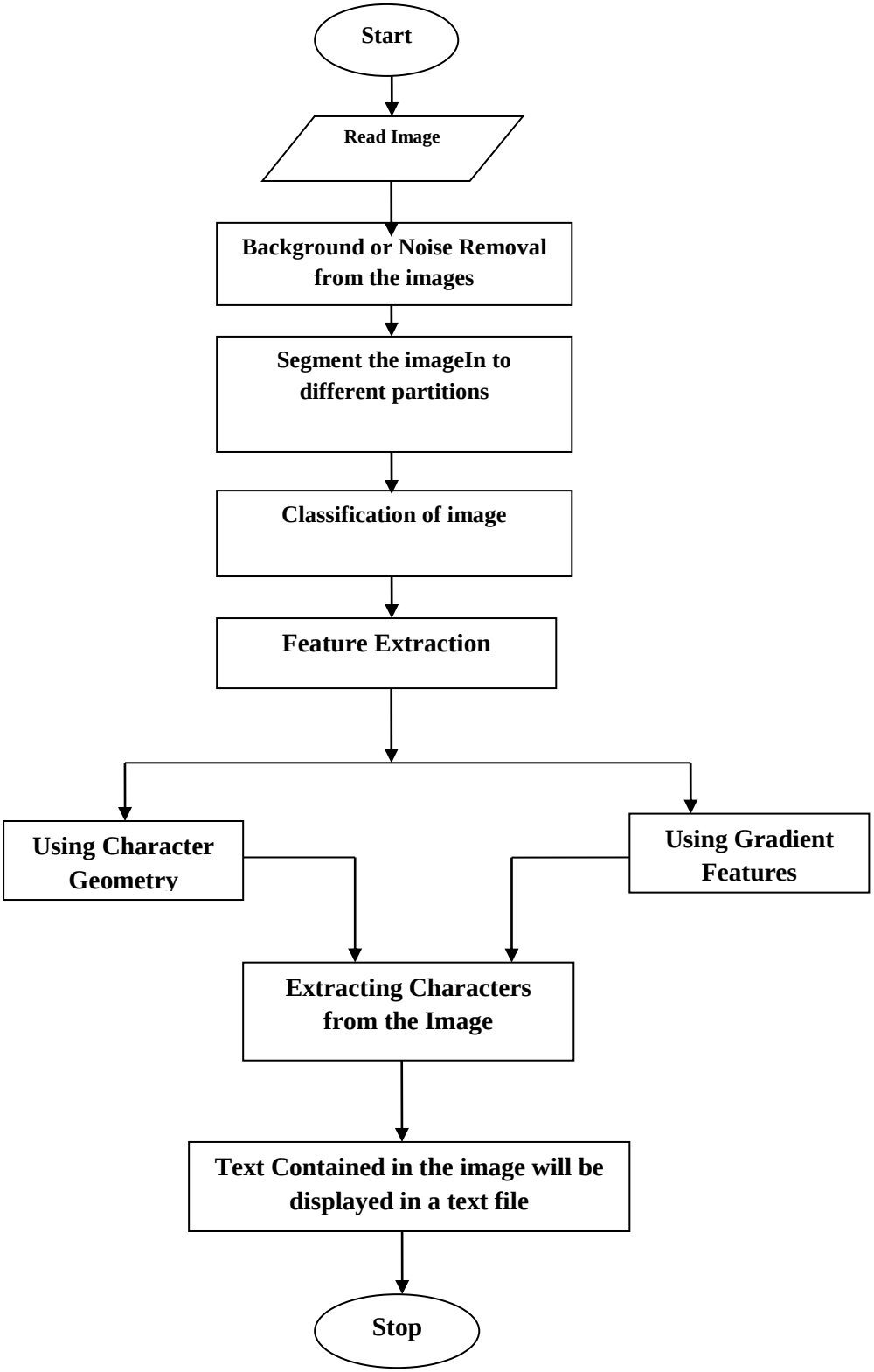


Fig: 5.3.1Flow chart

5.4 UML diagrams

5.4.1 Use Case Diagram

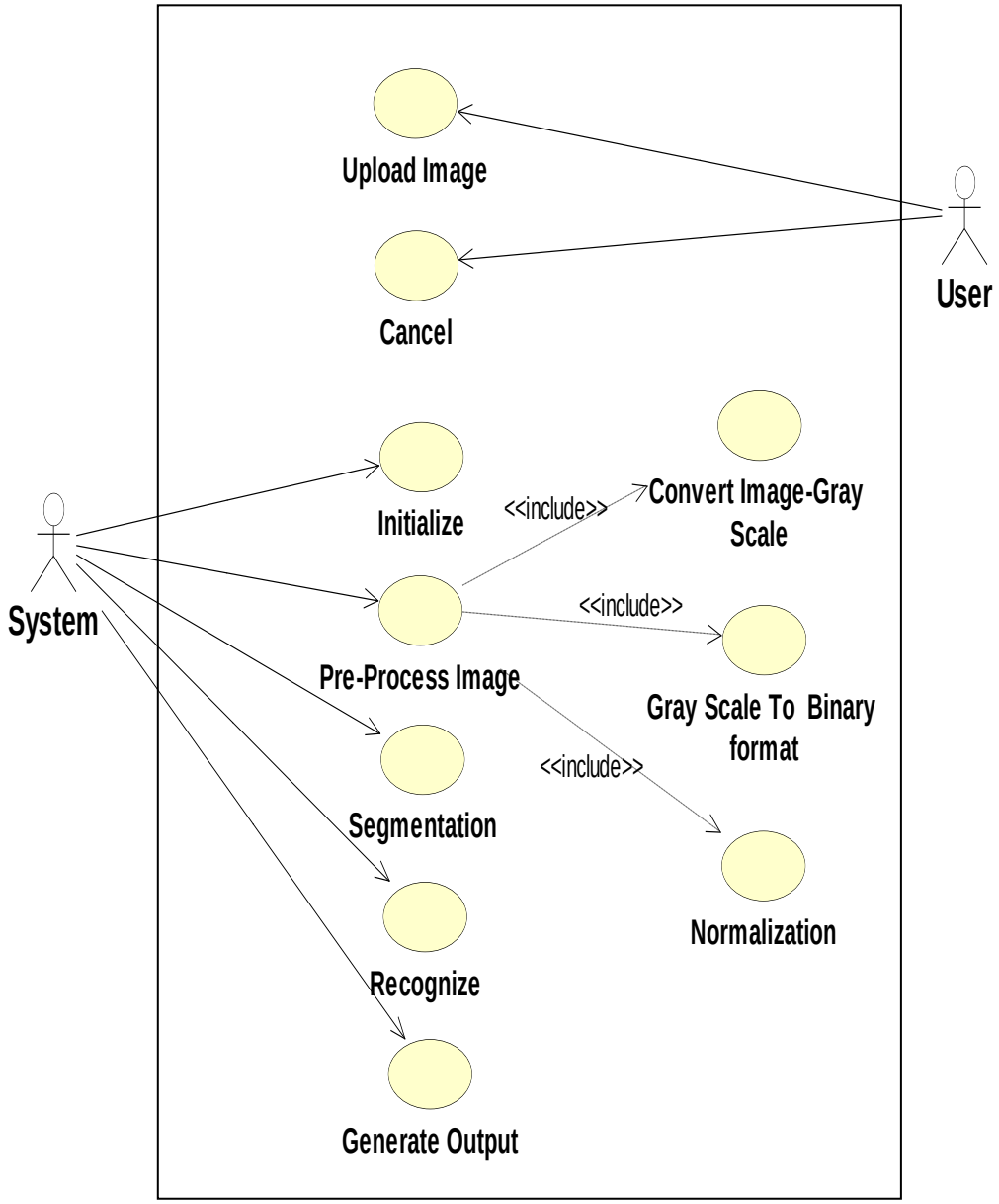


Figure 5.4.1: Use Case Diagram

User Module

User Case	Description
Actor	**User**
Precondition	Input image should be available.
Main Scenario	User uploads image.
Extension Scenario	If the image is not compatible. Not possible to upload file.
Post Condition	Image successfully uploaded.

Figure 5.4.1.1: User module

Pre-processing Module

User Case	Description
Actor	**System**
Precondition	Uploaded input image
Main Scenario	Pre-processing is carried out by converting the image from RGB format to binary format.
Post Condition	Extract characters Before segmentation

Figure 5.4.1.2: Pre-processing module

Segmentation Module

User Case	Description
Actor	**System**
Precondition	Pre-processed image should be available.
Main Scenario	The pre-processed input image is segmented into isolated characters by assigning a number to each character using a labeling process. This labeling provides information about number of characters in the image. Each individual character is uniformly resized into pixels.
Extension Scenario	If the image is not compatible. Not possible to upload file.
Post Condition	Image successfully uploaded.

Figure 5.4.1.3 Segmentation module

5.4.2 Sequential Diagram

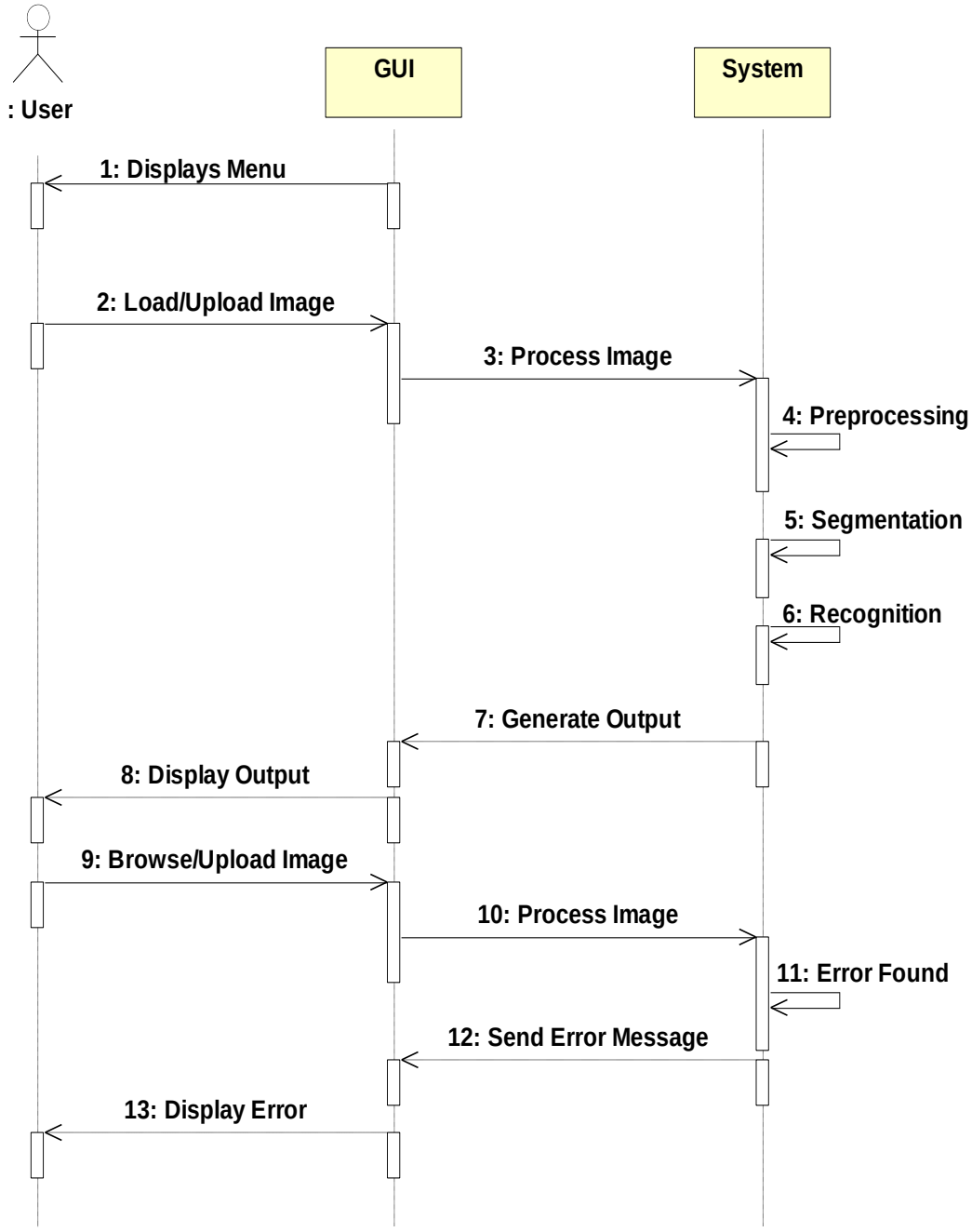


Figure 5.4.2: Sequence Diagram

5.4.3 Activity Diagram

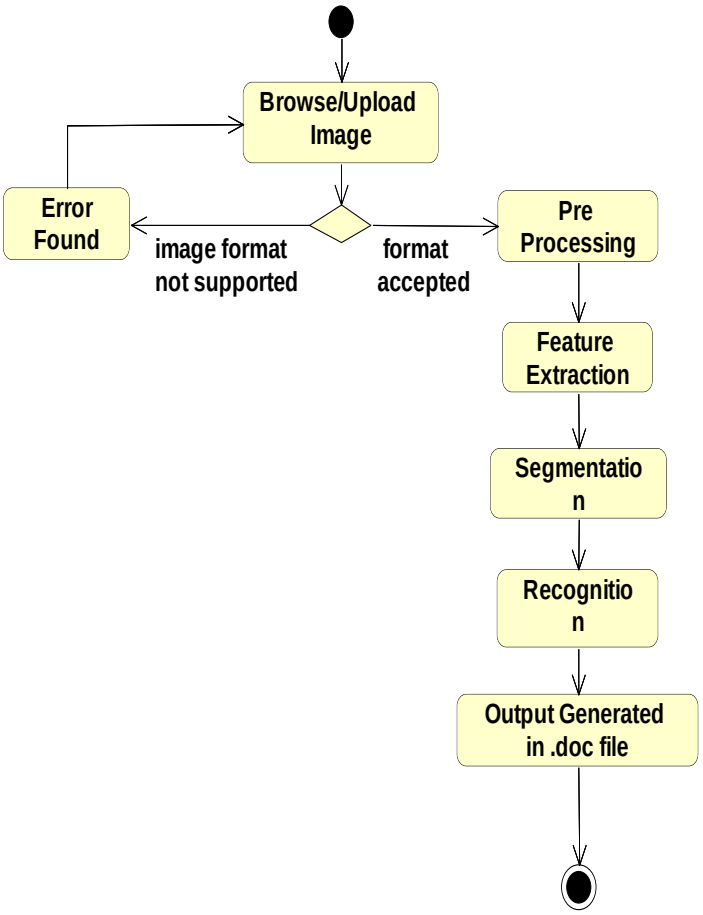


Figure 5.4.3: Activity Diagram

5.4.4 Architecture of the system

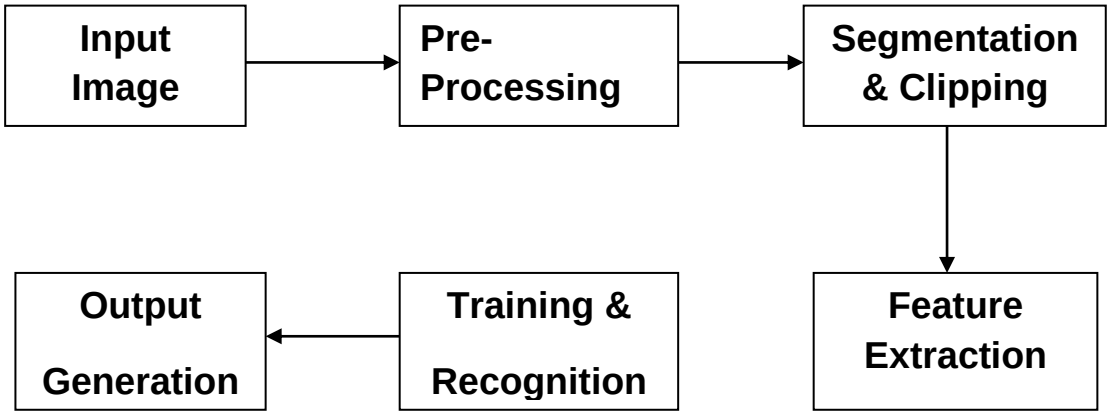


Figure 5.4.4: Architecture of the proposed system

5.5 Design of the User Interface

GUIDE, the MATLAB graphical user interface development environment, provides a set of tools for creating graphical user interfaces (GUIs). These tools simplify the process of laying out and programming GUIs.

The GUIDE Layout Editor enables you to populate a GUI by clicking and dragging GUI components — such as buttons, text fields, sliders, axes, and so on — into the layout area. It also enables you to create menus and context menus for the GUI.
The three main principle elements required in the creation of Graphical User Interface are:

- **Components**: Each item on the MATLAB GUI is a graphical component. The types of components include graphical controls (pushbuttons, edit boxes, lists, sliders, etc, static elements (frames and text strings),menus and axes. Axes, which are used to display the graphical data are created using function axes.
  Graphical control and static elements are created by the function uicontrol, and menus are created by function uimenu and uicontextmenu. Axes which are used to display graphical data are created by the function axes.

- **Figures:** The components of the GUI must be arranged within the figure, which is a window on the computer screen. In the post figure have been created automatically whenever we have plotted data. However empty figure can be created with the function figure and can be used to hold any combination of components.

- **Call back:** Finally, there must be some way to perform an action if a user click a mouse on a button or type information on a keyboard .A mouse click or a key press is an event, and the MATALAB program must respond to each event if the program is to perform its function .The code executed in response to an event is called a callback. There must be a callback to implement the function of each graphical user component on the GUI.

5.6 Risks Identified

Matlab is an interpreted language. The main disadvantage of interpreted languages is execution speed. When a language is compiled, all of the code is analyzed and processed efficiently, before the programmer distributes the application. With an interpreted language, the computer running the program has to analyze and interpret the code (through the interpreter) before it can be executed (each and every time), resulting in slower processing performance.

The values of 39 and 35 hidden neurons for gradient features and character geometry respectively are chosen based on experiments conducted on several different images and are used by classifiers to produce correct classification results. The variations in the hidden neuron values might tend to produce wrong result. Hence these values should be carefully chosen.

Chapter 6

6 Implementation

6.1 Software and Hardware Used

		Windows 8	
	Processor	**RAM**	**DISK Space**
MATLAB V.13 (R2013a)	**Dual core, core2duo, Intel I3**	**2048 MB**	**1 GB for MATLAB only,5 GB for a typical installation**

Table 6.1.1: Software and Hardware Used

6.2 Software Development Platform/Environment/framework

- 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.

- The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects, which together represent the state-of-the-art in software for matrix computation.

6.2.1 The MATLAB language

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

6.2.2 The MATLAB working environment

This is the set of tools and facilities that you work with as the MATLAB user or programmer. It includes facilities for managing the variables in your workspace and importing and exporting data. It also includes tools for developing, managing, debugging, and profiling M-files, MATLAB's applications.

6.2.3 MATLAB Image Processing Toolbox

We have used MATLAB Image Processing Toolbox for the development of this software. Image processing involves changing the nature of an image in order to improve pictorial information of the image for human interpretation for autonomous human perception. The Image Processing Toolbox is a collection of functions that extend the capability of the MATLAB numeric computing environment. The toolbox supports a wide range of operations on the image.

Key Features

- Image enhancement, including filtering, filters design, deblurring and contrast enhancement.
- Image analysis including features detection, morphology, segmentation, and measurement.
- Spatial transformations and image registration.
- Support for multidimensional image processing.
- Support for ICC version 4 color management system.
- Modular interactive tools including ROI selection, histograms and distance measurements.
- Interactive image and video display.
- DICOM import and export.

6.2.4 MATLAB Neural Network Toolbox

Key Features

- Supervised networks, including multilayer, radial basis, learning vector quantization (LVQ), time-delay, nonlinear autoregressive (NARX), and layer-recurrent
- Unsupervised networks, including self-organizing maps and competitive layers
- Apps for data-fitting, pattern recognition, and clustering
- Parallel computing and GPU support for accelerating training (using Parallel Computing Toolbox)
- Preprocessing and postprocessing for improving the efficiency of network training and assessing network performance
- Modular network representation for managing and visualizing networks of arbitrary size
- Simulink® blocks for building and evaluating neural networks and for control systems applications

6.2.5 Working of the Front End

When you save your GUI layout, GUIDE automatically generates an M-file that you can use to control how the GUI works. This M-file provides code to initialize the GUI and contains a framework for the GUI callbacks the routines that execute in response to user-generated events such as a mouse click. Using the M-file editor, you can add code to the callbacks to perform the functions you want.

Home Page

1. The system displays the Home page with some options.
2. The user will browse for the input image.
3. User clicks on LOAD IMAGE Button to upload the image.

Processing Module

1. This GUI will receive the query image.
2. Displays the noiseless image of the uploaded image.

3. The characters are extracted from the image and displayed.
4. Output will be displayed in a .txt /.doc file.

6.3 Screen Shots of the Front End

Home Page

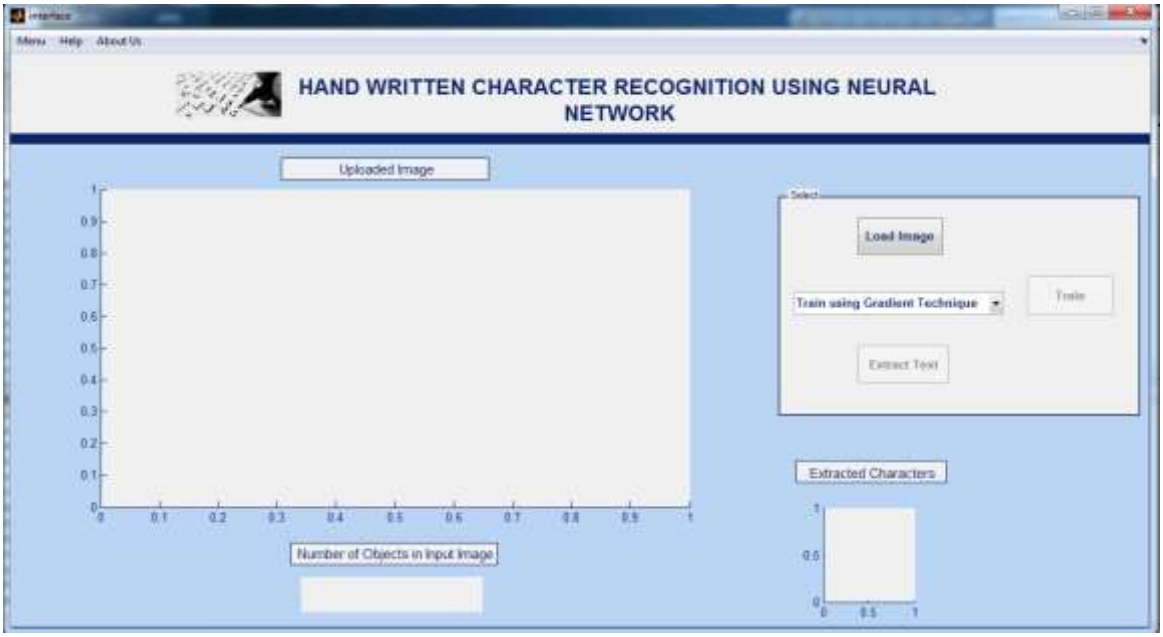


Figure 6.3.1: Home Page

Uploaded Image

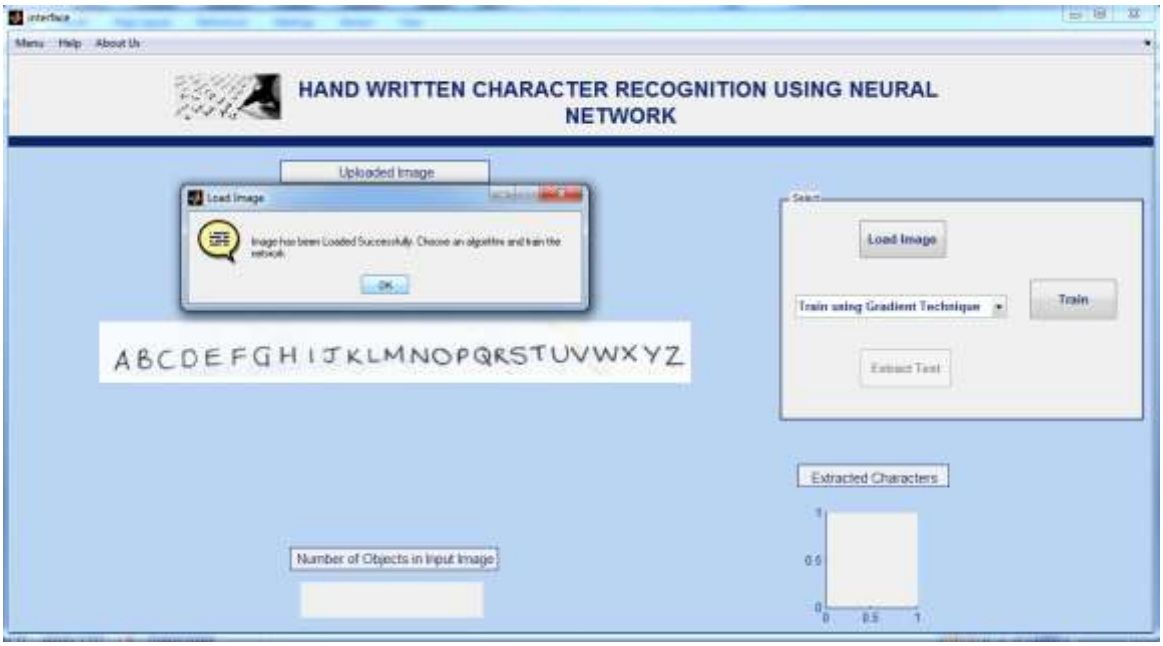


Figure 6.3.2: Uploading Image

Neural Network Training

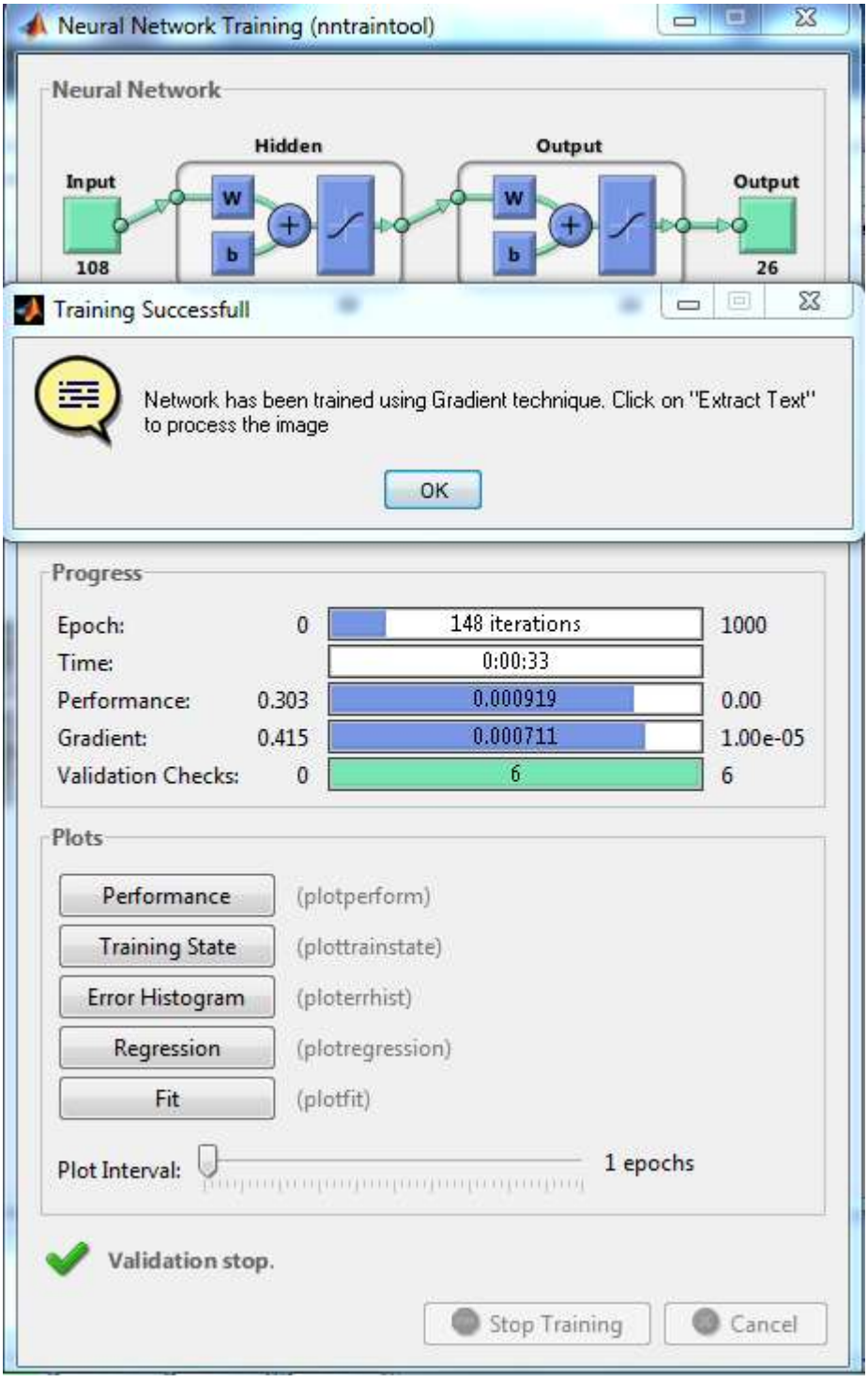


Figure 6.3.3: Neural Network Training

Character Extraction

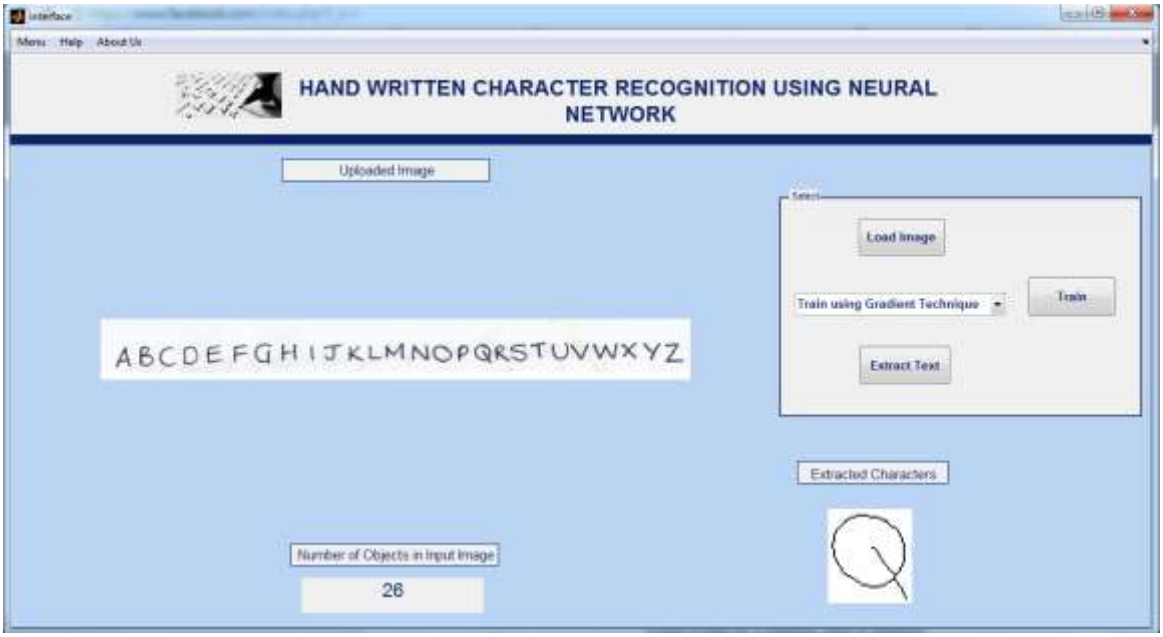


Figure 6.3.4: Extracting Characters

Output File

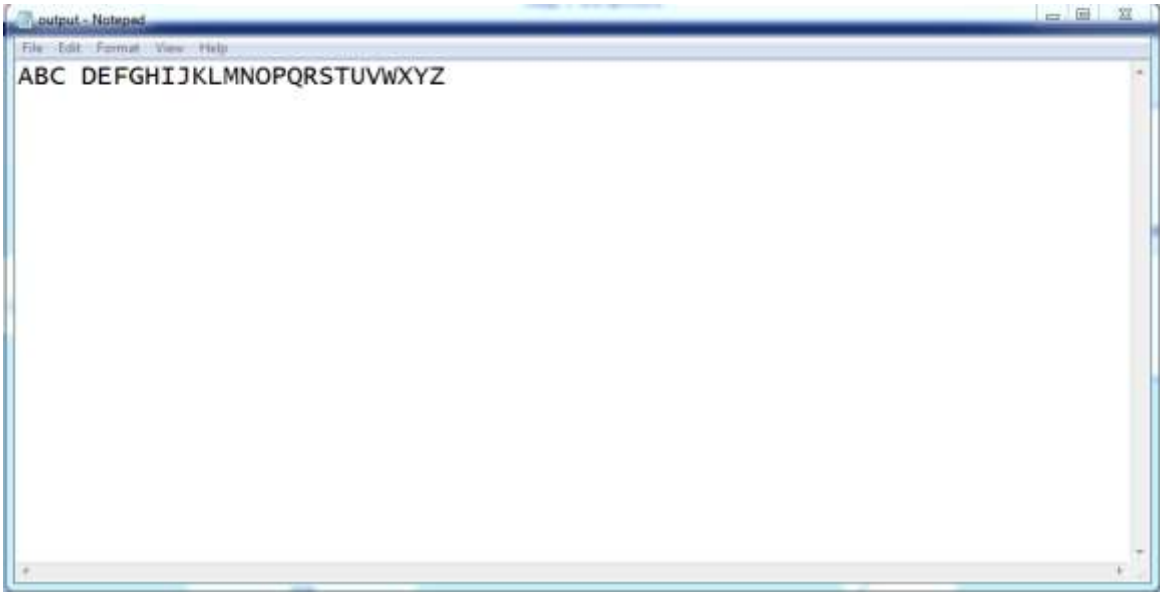


Figure 6.3.5: Output File

6.4 Snapshot and description of Experimental Set up

Requirements

- An internet connection

- A DVD reader Or ISO file

- Matlab media – DVD

- Matlab serial number

1. Load the DVD into the PC you want to install Matlab onto. The DVD should

Automatically start the installation program whereby you will see the first splash screen.

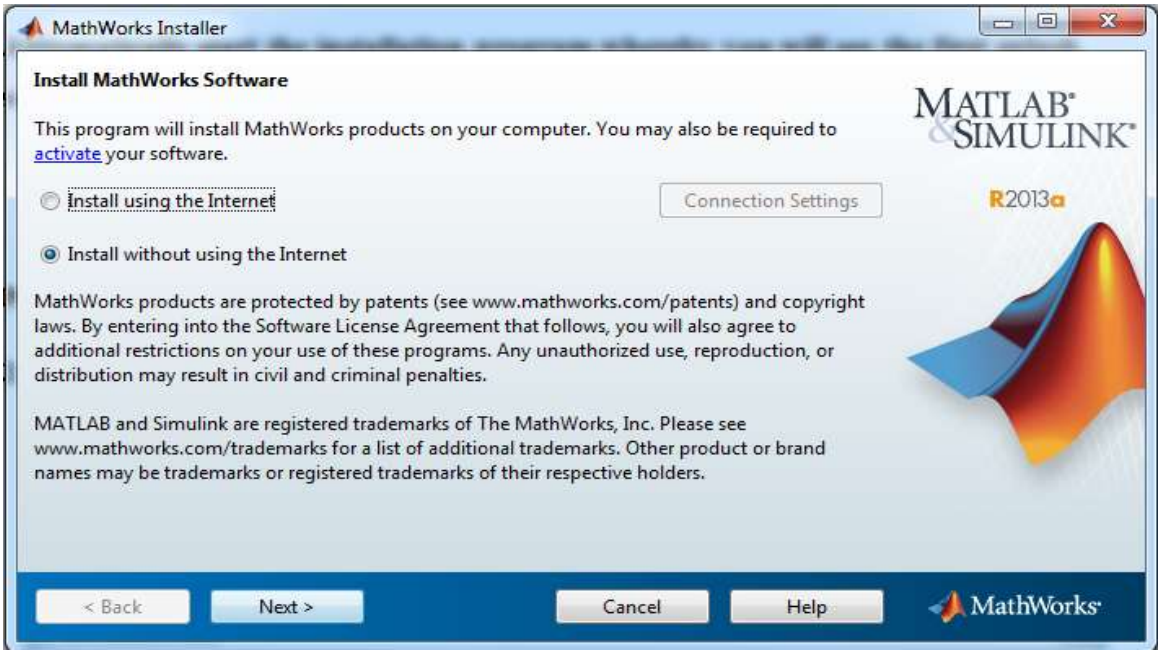


Figure 6.4.1: setup 1

2. You need to agree to the Math works license.

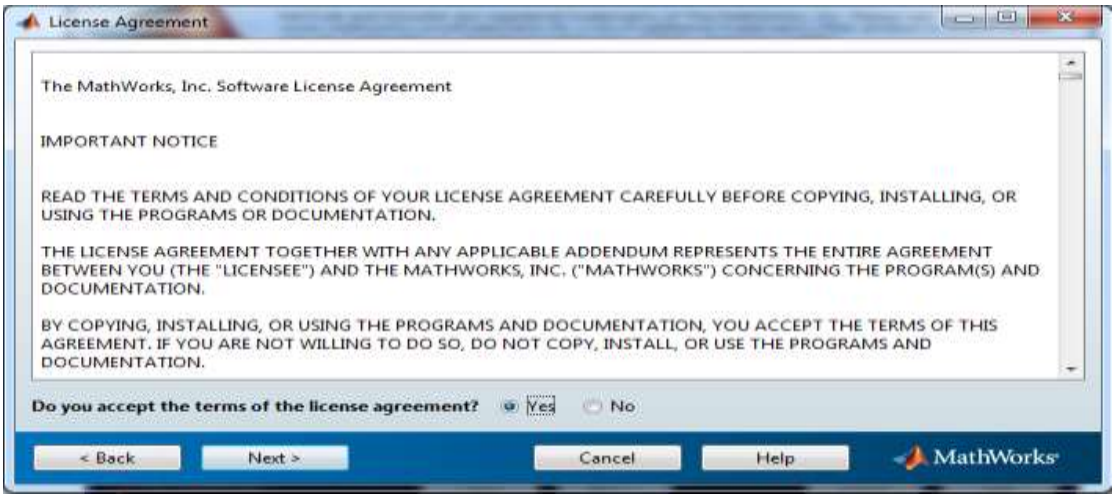


Figure 6.4.2: setup 2

3. Choose the 'Typical' installation

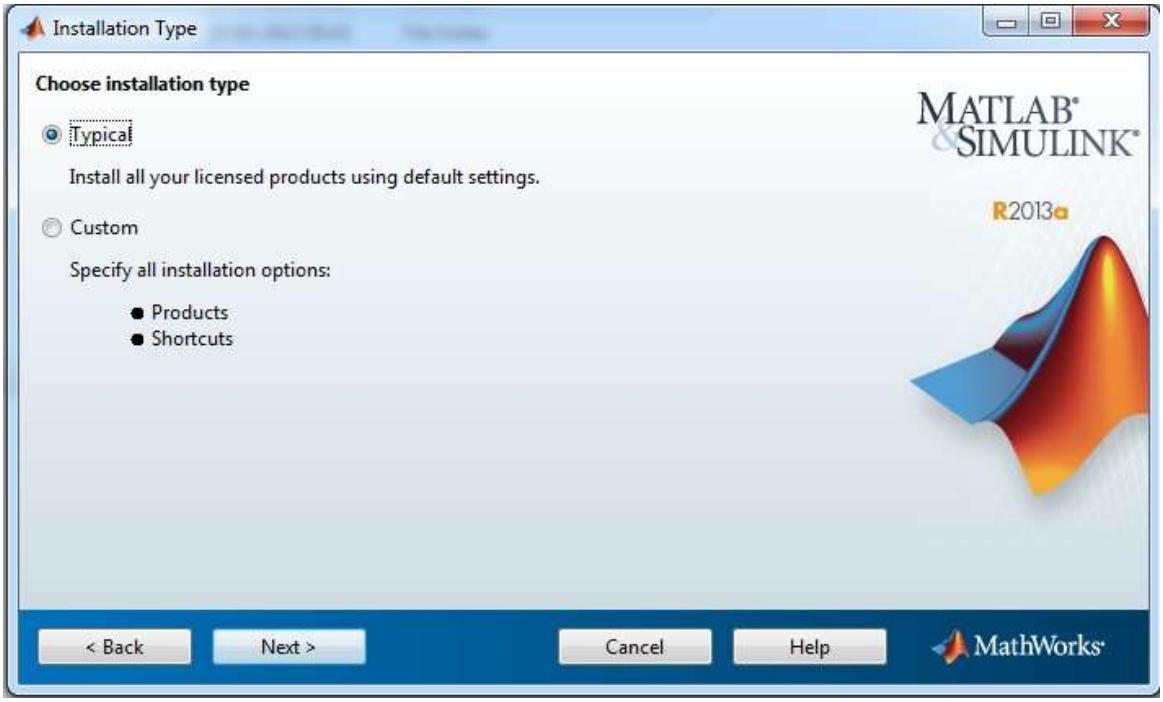


Figure 6.4.3: setup 3

4. Choose the location of the installation.

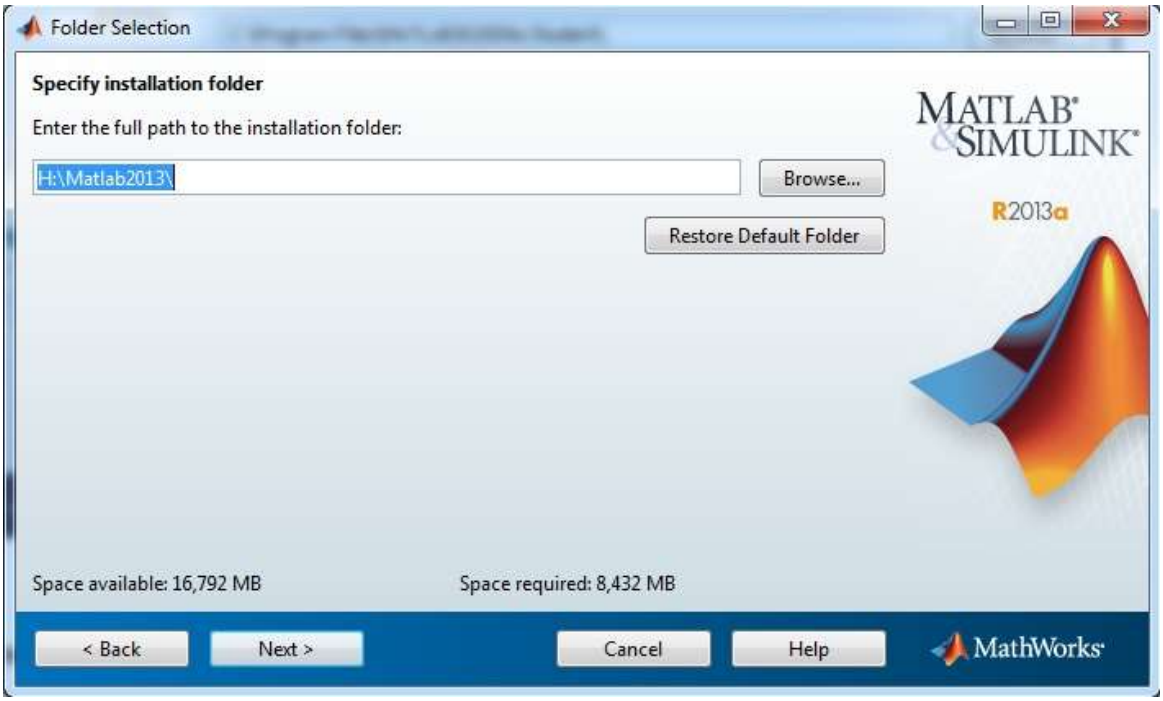


Figure 6.4.4: setup 4

5. Confirm the installation settings by pressing Install

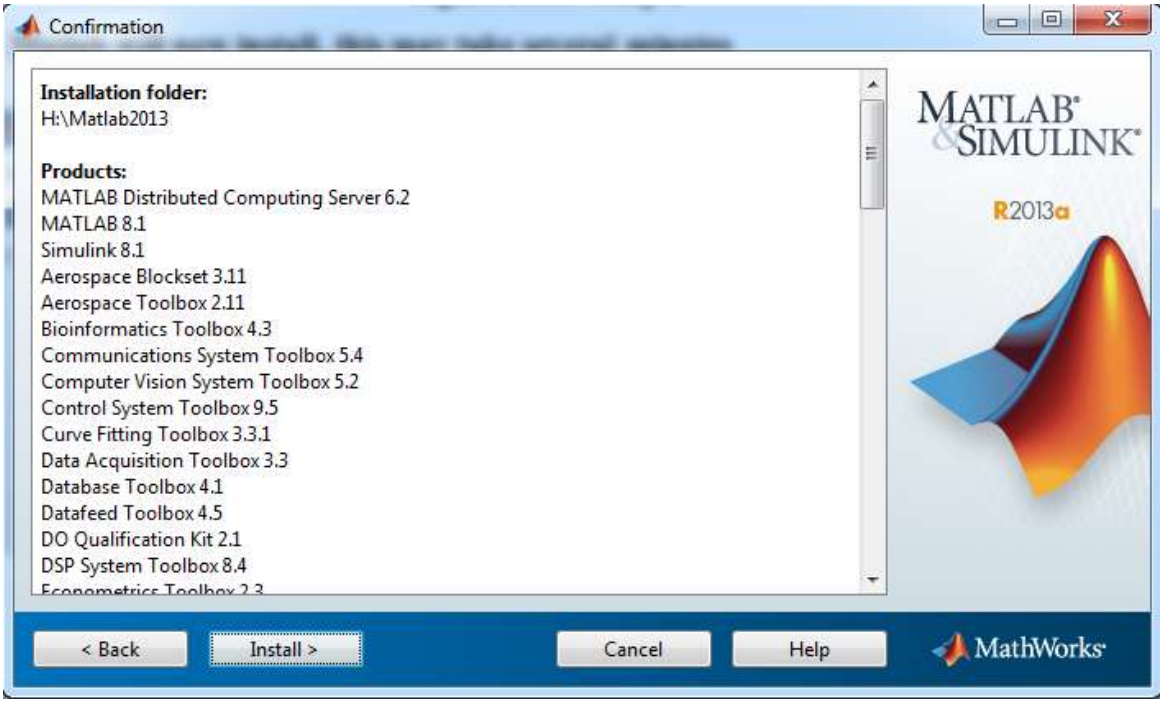


Figure 6.4.5: setup 5

7. Matlab will now install, this may take several minutes

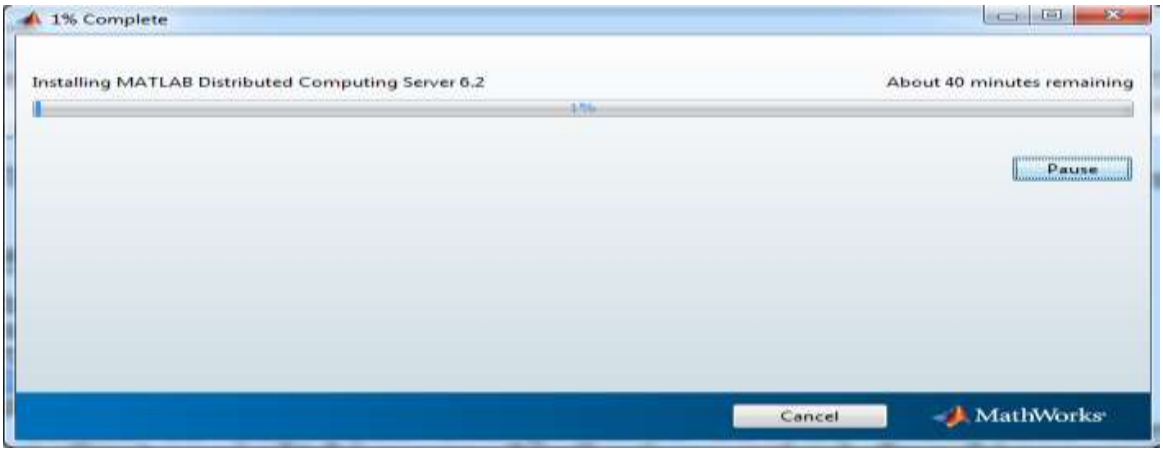


Figure 6.4.6: setup 6

8. After the installation has completed, you then need to license your install.

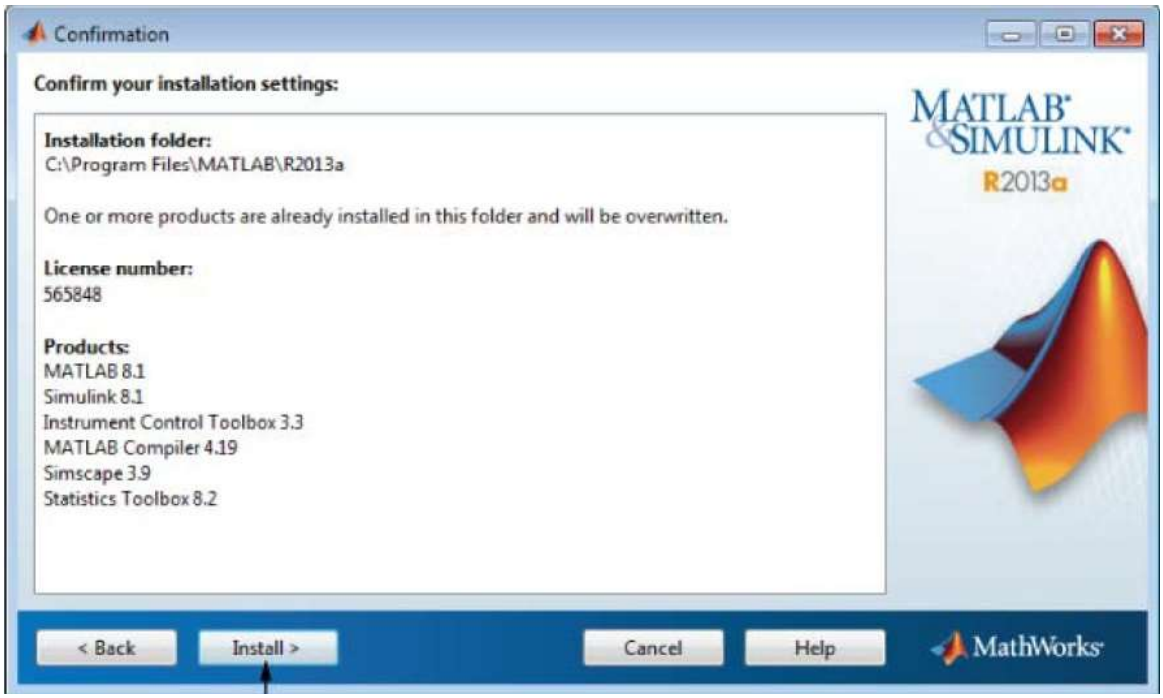


Figure 6.4.7: setup 7

You need to have the serial number ready. This number can be located on the DVD case.

9. Matlab will initially make an internet connection the Mathworks prior to you entering the serial number.

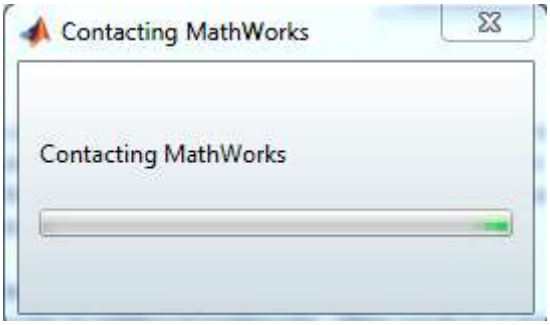


Figure 6.4.8: setup 8

12. Enter the serial number and your @cam email address e.g. ab123@cam.ac.ukand Continue with the rest of the registration process then below figure will appear.

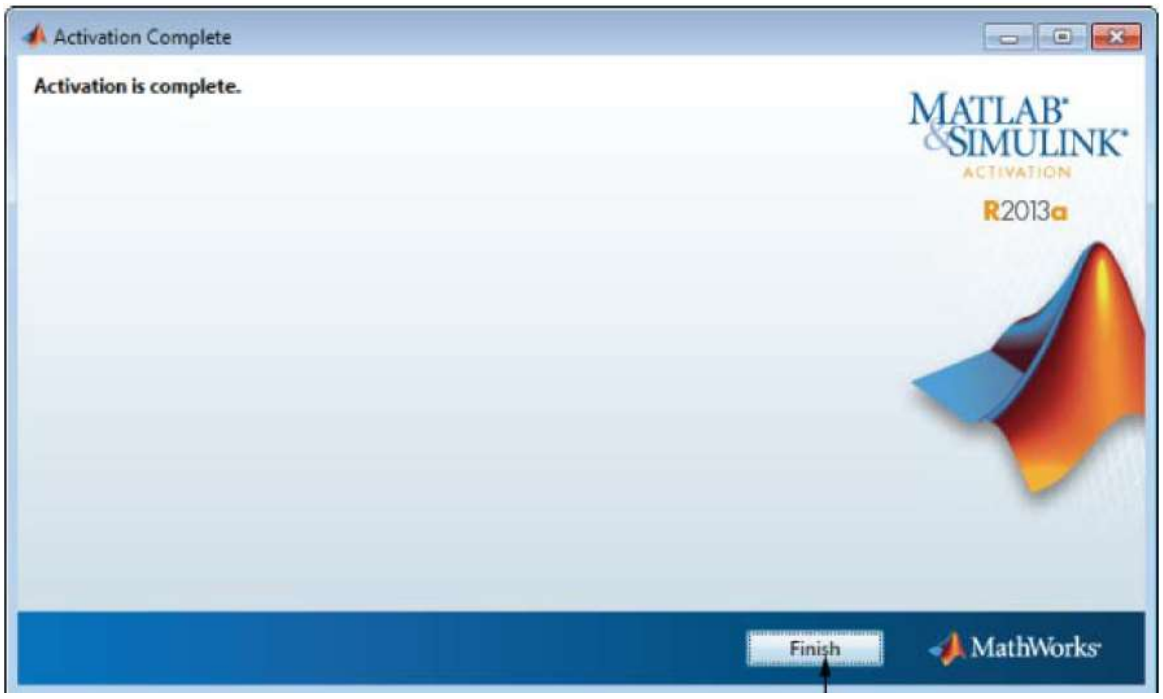


Figure 6.4.9: setup 9

Chapter 7

7 Testing

7.1 Verification

The set of Test Cases are used to test the functionality of each module if that module works properly then that Test Cases marked as Pass or else Fail.

Test Id.	Test Case	Input Description	Expected Output	Test Status
1	Uploading Image	When user clicks on open button open field box will be opened to select the image file	Image file should be selected and uploaded	Pass
2	To preprocess images	Image will be taken for preprocessing	Conversion from RGB to B/W image (Binarization)	Pass
3	Feature extraction	A Gray Scale Image	Character features should be extracted	Pass
4	Output file	Normalized character to the neural network	file containing only the text	Pass

Table 7.1: Verification

7.2 Validation

The below table is used to determine whether or not a system satisfies the acceptance criteria and to determine whether or not to accept the system.

SI no	Functions	Required Output	Actual Output
1	Upload the image with valid format	Image should be uploaded if supported	Valid image is uploaded successfully
2	Invalid image format	Error message should be displayed	Error message is displayed if the image format is not supported
3	Pre-processing of the uploaded image	Image should be pre-processed in order to convert to Gray scale	Image is preprocessed
4	Extraction of features	Character features such as edges and curves are calculated	Image features are extracted
5	Displaying result	Text of the file displayed	Text contained in the file is displayed

Table 7.2: Validation

7.3 Failure modes and action on failure

SlNo.	Event	Action
1.	Wrong input file uploaded	Error message should be displayed to the user and home screen must be displayed
2	System shut down	Tasks should be canceled and process should be restarted again

Table 7.3: Failure modes of system

7.4 Test Results

Character Geometry

Epochs	Hiddenlayers	Config	Classification%
43	10	85-10-26	27.8
104	20	85-20-26	77.4
148	30	85-30-26	93.8
172	35	85-35-26	94.3
117	39	85-39-26	93.3
53	45	85-45-26	12.5
60	50	85-50-26	83.7
76	55	85-55-26	78.3
71	60	85-60-26	18
110	65	85-65-26	49.8
112	70	85-70-26	49.2

Table 7.4.1: Using Character Geometry

Gradient Features

Epochs	Hidden layers	Config	Classification %
47	10	108-10-26	10
189	20	108-20-26	87.4
144	30	108-30-26	82
137	35	108-35-26	82.2
148	39	108-39-26	94.5
109	45	108-45-26	76.8
113	50	108-50-26	86.6
98	55	108-55-26	55.2
94	60	108-60-26	68.3
102	65	108-65-26	89.1
130	70	108-70-26	91.1

Table 7.4.2 Using Gradient Features

7.4 Evaluation

- The Handwritten Character Recognition system was tested on several different scanned images containing handwritten textwritten with different styles and the results were highly encouraging.
- The proposed method performs preprocessing on the image for removing the noise and further uses feature extraction using gradient technique OR using character geometry which gives relatively good classification compared to OCR.
- The method is advantageous as it uses nine features to train the neural network using character geometry and twelve features using gradient technique. The advantage lies in less computation involved in feature extraction, training and classification phases of the method.
- The proposed methodology has produced good results for images containing handwritten text written in different styles, different size and alignment with varying background. It classifies most of the handwritten characters correctly if the image contains less noise in the characters and also in the background. Characters written with legible handwriting are classified more accurately.

Chapter 8

8 Conclusion

8.1 Summary of work done

- The effectiveness of the method that uses feature extraction using character geometry and gradient technique from scanned images containing handwritten characters is presented.
- The feature extraction methodshave performed well in classification when fed to the neural network and preprocessing of image using edge detection and normalization are the ideal choice for degraded noisy images.
- The method of training neural network with extracted features from sample imagesof each character has detection accuracy to a greater extent.
- The proposed methodology has produced good results for images containing handwritten text written in different styles, different size and alignment with varying background
- The system is developed in MATLAB and evaluated for a set of sample images containing handwritten texton Intel dual core computer.
- The method is advantageous as it uses nine features to train the neural network using character geometry and twelve features using gradient technique.

8.2 Proposal for enhancement or re-design

- As the feature extraction methods such as gradient technique and character geometryused in the method does not classify characters of different language, the method can be extended for language independent classification from the images of other languages with little modifications. The performance of the method has been tested for classifying English text written in upper case, but needs further exploration.
- Refinement of the segmented characters can be done in order to achieve higher accuracy rate.
- The performance of the neural network can be increased by adding some more features other than the existing ones.

- The classification rate can be increased by training the neural network with more number of test images.

References

[1] Chirag I Patel, Ripal Patel, Palak Patel, "Handwritten Character Recognition Using Neural Networks", International Journal of Scientific & Engineering Research Volume 2, Issue 5, May-2011.

[2] Kauleshwar Prasad, Devvrat C Nigam, AshmikaLakhotiya, DheerenUmre, "Character Recognition Using Matlab's Neural Toolbox", International Journal of u- and e- Service, Science and Technology Vol. 6, No. 1, February, 2013.

[3] AshutoshAggarwal, Rajneesh Rani, RenuDhir, "Handwritten Character Recognition Using Gradient Features", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 5, May 2012.

[4] Vinita Dutt, Sunil Dutt, "Handwritten Character Recognition Using Artificial Neural Network", Advances in Computing: 2011; 1(1): 18-23.

[5] Rahul Kala, Harsh Vazirani, AnupamShukla, RituTiwari, "Offline Handwriting Recognition", International Journal of Computer Science issues, volume 7, March-2010.

[6] Dinesh Dileep, "A Feature Extraction Technique Based on Character Geometry for Character Recognition".

[7] Alexander J. Faaborg, "Using Neural Networks to Create an Adaptive Character Recognition System", Cornell University, Ithaca NY, (May 14, 2002)

[8] Swapnil A. Vaidya, Balaji R. Bombade "A Novel Approach of Handwritten Character Recognition using Positional Feature Extraction",IJCSMC, Vol. 2, Issue. 6, June 2013.

[9] Sheng Wang "A Review of Gradient-Based and Edge-Based Feature Extraction Methods for Object Detection",Computer and Information Technology (CIT), 2011 IEEE 11th International Conference.

Glossary

ANN	Artificial neural networks (ANNs) are computational models inspired by an animal's central nervous systems(in particular the brain) which is capable of machine learning as well as pattern recognition. Artificial neural networks are generally presented as systems of interconnected "neurons" which can compute values from inputs.
Binary image	This image format also stores an image as a matrix but can only color a pixel black or white (and nothing in between). It assigns a 0 for black and a 1 for white.
Contrast	The difference in color and light between parts of an image
Image processing	Transforming digital information representing images
Segmentation	Image segmentation is the process of dividing an image into multiple parts. This is typically used to identify objects or other relevant information in digital images
Noise	Image noise is random (not present in the object imaged) variation of brightness or color information in images, and is usually an aspect of electronic noise. Image noise is an undesirable by-product of image capture that adds spurious and extraneous information.
Gray scale image	A gray scale or grey scale digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information. Images of this sort, also known as black-and-white, are composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest