A

Project

On

“OPTICAL CHARACTER RECOGNITION USING NEURAL NETWORK”

Submitted By

Mr. Rupesh Vaman Gawas.

Mr. Ketan Krishna Kadam .

Under the guidance of

Prof. Poonam Kadam.

DEPARTMENT OF COMPUTER ENGINEERING

Metropolitan Institute of Technology & Management

University Of Mumbai

2020-2021

CERTIFICATE

Metropolitan Institute of Technology & Management

Project Report Entitled

"OPTICAL CHARACTER RECOGNITION USING NEURAL NETWORK"

Submitted By

Mr.Rupesh Vaman Gawas.

Mr. Ketan Krishna Kadam.

As the partial fulfillment of degree of B. E. COMPUTER ENGINEERING is approved.

(Prof. Poonam Kadam)

Internal Guide

(Prof. Poonam Kadam)

Project Co-coordinator

(Prof. Manoj Khadilkar)

Head of Dept.

(Prof. S.C Nawle)

Principal

Internal Examiner

External Examine

ACKNOWLEDGMENT

We would like to express our sincere gratitude our guide, **Prof. Poonam Kadam** for the help, guidance and encouragement, she provided during the dissertation. This work would have not been possible without her valuable time, patience and motivation. We thank her for making our stint thoroughly pleasant and enriching. It was great learning and an honor being her student.

We are deeply indebted to **Prof. Manoj khadilkar** and the entire team in the **computer department**. They supported us with scientific guidance, advice and encouragement, they were always helpful and enthusiastic and this inspired us in my work.

We take the privilege to express our sincere thanks **Prof. S. C. Nawle**, our Principal for providing the encouragement and much support throughout my work.

Mr.Rupesh Vaman Gawas.

Mr. Ketan Krishna Kadam.

ABSTRACT

Propose a neural network based size and colour invariant character recognition system using feed-forward neural network. Our feed-forward network has two layers. One is input layer and another is output layer. The whole recognition process is divided into four basic steps such as Pre-processing, normalization, network establishment and boundary detection, the input character matrix is normalized into matrix for size invariant recognition and fed into the proposed network which consists of input and output neurons.

INDEX

LIST OF FIGURES

Chapter1

Introduction

In the running world, there is growing demand for the software systems to recognize characters in computer system when information is scanned through paper documents as we know that we have number of newspapers and books which are in printed format related to different subjects. These days there is a huge demand in "storing the information available in these paper documents in to a computer storage disk and then later reusing this information by searching process". One simple way to store information in these paper documents in to computer system is to first scan the documents and then store them as IMAGES. But to reuse this information it is very difficult to read the individual contents and searching the contents form these documents line-by-line and word-by-word. The reason for this difficulty is the font characteristics of the characters in paper documents are different to font of the characters in computer system. As a result, computer is unable to recognize the characters while reading them. This concept of storing the contents of paper documents in computer storage place and then reading and searching the content is called DOCUMENT PROCESSING. Sometimes in this document processing we need to process the information that is related to languages other than the English in the world. For this document processing we need a software system called CHARACTER RECOGNITION SYSTEM. This process is also called DOCUMENT IMAGE ANALYSIS (DIA).

Thus our need is to develop character recognition software system to perform Document Image Analysis which transforms documents in paper format to electronic format. For this process there are various techniques in the world. Among all those techniques we have chosen Optical Character Recognition as main fundamental technique to recognize characters. The conversion of paper documents in to electronic format is an on-going task in many of the organizations particularly in Research and Development (R&D) area, in large business enterprises, in government institutions, so on. From our problem statement we can introduce the necessity of Optical Character Recognition in mobile electronic devices such as cell phones, digital cameras to acquire images and recognize them as a part of face recognition and validation.

To effectively use Optical Character Recognition for character recognition in-order to perform Document Image Analysis (DIA), we are using the information in Grid format. This system is thus effective and useful in Virtual Digital Library's design and construction.

1.2 Project Scope

The scope of our product Optical Character Recognition on a neural network is to provide an efficient and enhanced software tool for the users to perform Document Image Analysis, document processing by reading and recognizing the characters in research, academic, governmental and business organizations that are having large pool of documented, scanned images. Irrespective of the size of documents and the type of characters in documents, the product is recognizing them, searching them and processing them faster according to the needs of the environment.

1.3 Advantages:

1. Increasing the DATABASE used for training the ANN, so as to enable it to recognize stylized fonts also.
2. Using better algorithms for training the ANN, so as to decrease the Time complexity while handling larger databases.
3. OCR recognizes machine-printed or hand-printed characters.
4. Quick retrieval for editing and reprocessing.

Chapter 2

Literature Survey

In the running world there is a growing demand for the users to convert the printed documents in to electronic documents for maintaining the security of their data. Hence the basic OCR system was invented to convert the data available on papers in to computer process able documents, So that the documents can be editable and reusable. The existing system/the previous system of OCR on a grid infrastructure is just OCR without grid functionality. That is the existing system deals with the homogeneous character recognition or character recognition of single languages.

Vijay Laxmi Sahu, Babita Kubde (January 2013) This paper explains that classification methods based on learning from examples have been widely applied to character recognition from the 1990s and have brought forth significant improvements of recognition accuracies. This class of methods includes statistical methods, artificial neural networks, support vector machines, multiple classifier combination, etc. In this paper, the characteristics of the classification methods that have been successfully applied to character recognition, and show the remaining problems that can be potentially solved by learning methods have been discussed.

We now describe the architecture used to learn the feature representations and train the classifiers used for our detection and character recognition systems. The basic setup is closely

Related to a convolutional neural network, but due to its training method can be used to rapidly construct extremely large sets of features with minimal tuning.
Our system proceeds in several stages:

1) Apply an unsupervised feature learning algorithm to a set of image patches harvested from the training data to learn a bank of image features.

2) Evaluate the features convolutional over the training images. Reduce the number of features using spatial pooling.

3) Train a linear classifier for either text detection or character recognition.

Chapter 3

Problem Statement

A character can be written in a number of ways differing in shape and properties, such as Tilt, stroke, Cursively and Overall shape. A plethora of Fonts are available for use in any commonly used Word Processing Application Software. Yet, while perceiving any text written in a variety of ways, humans can easily recognize and read each character. This is because the human perception processes the information by the features that define a character's shape in an overall fashion. Thus, while modeling the human perception model in machines, ANN can be applied for classification of characters.

Furthermore, OCR is aimed at developing the ability to 'read', also known as Computer Vision. Thus, a recognized character should be carefully classified, if the same 'symbol' may signify more than one character at different places.

Chapter 4

Existing System:

4.1 Template -matching and correlation techniques.

- The matrix containing the image of the input character is directly matched with a set of prototype characters representing each possible class.
- The technique is simple and easy to implement in hardware and has been used in many commercial OCR machines.
- However, this technique is sensitive to noise and style variations and has no way of handling rotated characters.

4.2 Drawbacks of Existing System:

- More time complexity.
- Unable to recognize stylized fonts.
- Blur image cannot be recognized.
- Brocken characters cannot be read easily.

Chapter 5

Requirement Analysis

5.1 Functional Requirements

The relationship between the input and output to the system is determined by

the functional requirement of the SRS.

5.1.1 Technical Issues

Many a software project has failed due to an incomplete or inaccurate analysis process, especially technical issues. Technical issues are a key step while developing a software application.

5.1.2 Risk Analysis

Project Risk Analysis is for Cost estimates of known accuracy and risk on capital investment projects. Their main challenge is to determine how to model and visualize the complex relationships between risks, define and monitor the risks impacts, analyse the probability of risk occurrence, mitigate the negative impact of risks, and monitor the course of the project with risks and uncertainties.

5.1.3 Interface Requirements

The system performance is adequate. However, Virtual travel agency is working with the user internet connection, 60 percentage of the performance is up to the client side.

5.1.4 Hardware Requirements

a) Processor type: Pentium III-compatible processor or faster.

b) Processor speed: Minimum: 1.0 GHz, Recommended: 2.0 GHz or faster

c) RAM: 512 MB or more

d) HARD DISK: 20GB or more

e) Monitor: VGA or higher resolution 800x600 or higher resolution

5.1.5 Software Requirements

a) Operating system: Windows 8 and above

b) Front End: Asp .Net 2.0

.c) Coding Language: Visual C Sharp .Net

d) Back end: SQL Server 2013

5.1.6 Performance Requirements

The project has the following performance requirements

a) The prime requirement is that no error condition causes a project to exit abruptly.

b) Any error occurred in any process should return an understandable error message

c) The response should be fairly fast, the action participants should not be con fused at any point of time about action that is happening.

d) The system performance is adequate.

5.2 Non-Functional Requirements

5.2.1 Reliability

The project is guaranteed to provide reliable results for the entire user. The system shall operate 95 perc. of the time. The number of defect should not exceed 10 per function. In addition , before the submission of the final release the calendar must be tested in case of the defects over 10 per function.

5.2.2 Usability

a) Since GUI interface is used, it can be used by a user.

b) Since the system is placed on for online users any type user can use the system.

c) The system detects the fraud and reports to the user.

5.2.3 Scalability

The need for scalability has been a driver for much of the technology innovations of the past few years. The industry has developed new software languages, new design strategies, and new communication and data transfer protocols, in part to allow web sites to grow as needed.

5.2.4 Maintainability

Maintainability is our ability to make changes to the product over time. We need strong maintainability in order to retain our early customers. We will address this by anticipating several types of change, and by carefully documenting our design and implementation

Chapter 6

Project Design

6.1 Proposed Method:

Artificial Neural Network:

Artificial neural networks (ANNs) or connectionist systems are a computational model used in machine learning, computer science and other research disciplines, which is based on a large collection of connected simple units called artificial neurons, loosely analogous to axons in a biological brain. Connections between neurons carry an activation signal of varying strength.

If the combined incoming signals are strong enough, the neuron becomes activated and the signal travels to other neurons connected to it. Such systems can be trained from examples, rather than explicitly programmed, and excel in areas where the solution or feature detection is difficult to express in a traditional computer program. Like other machine learning methods, neural networks have been used to solve a wide variety of tasks, like computer vision and speech recognition, that are difficult to solve using ordinary rule-based programming.

Typically, neurons are connected in layers, and signals travel from the first (input), to the last (output) layer. Modern neural network projects typically have a few thousand to a few million neural units and millions of connections; their computing power is similar to a worm brain, several orders of magnitude simpler than a human brain. The signals and state of artificial neurons are real numbers, typically between 0 and 1. There may be a threshold function or limiting function on each connection and on the unit itself, such that the signal must surpass the limit before propagating. Back propagation is the use of forward stimulation to modify connection weights, and is sometimes done to train the network using known correct outputs However, the success is unpredictable: after training, some systems are good at solving problems while others are not. Training typically requires several thousand cycles of interaction.

The goal of the neural network is to solve problems in the same way that a human would, although several neural network categories are more abstract. New brain research often stimulates new patterns in neural networks. One new approach is use of connections which span further to connect processing layers rather than adjacent neurons. Other research being explored with the different types of signal over time that axons propagate, such as deep learning, interpolates greater complexity than a set of boolean variables being simply on or off. Newer types of network are more free flowing in terms of stimulation and inhibition, with connections interacting in more chaotic and complex ways. Dynamic neural networks are the most advanced, in that they dynamically can, based on rules, form new connections and even new neural units while disabling others.

6.2 System flow diagram

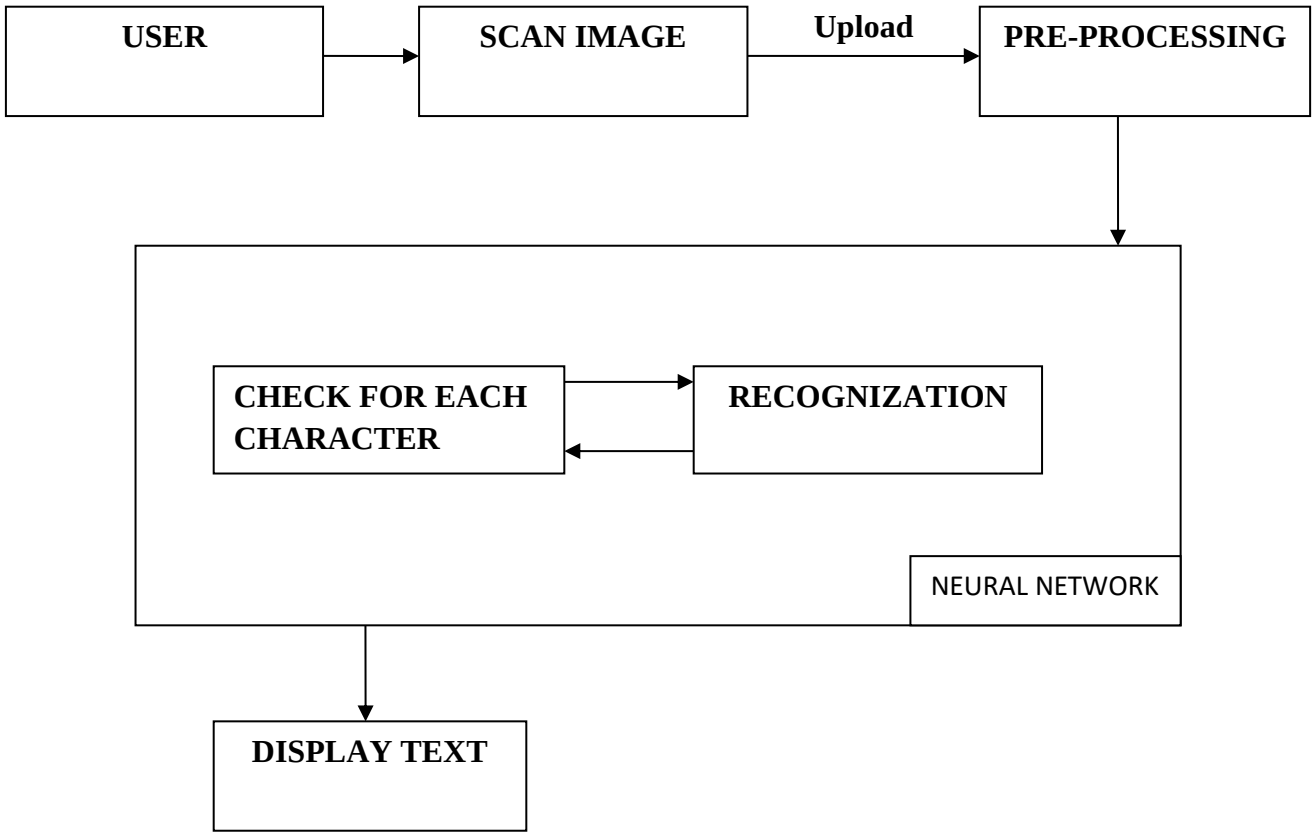


Fig 6.2 System flow diagram of OCR system

User:

In this block, user will open the OCR for recognition of image.

Scan Image:

User will scan the image of document and upload into the OCR.

Pre-Processing:

In this block Pre-processing process is start. It consists of digitization, noise removal and boundary detection of the digitized character matrix.

Neural Network:

After Pre-processing we get character matrix, which is further giver to Neural Network.

Neural Network recognized for each and every character based on its trained database.

Display Text:

In this block, Recognized character in display on screen that can be edited or saved.

6.3 Algorithms:

6.3.1 Algorithm for Training of Neural network

1) Take a random training sample for any character (it is better to go ahead in sequence) and then generate normalized matrix.

2) Generate the corresponding 12×8 initial weight_matrix by taking 3 for a 1 and -3 for 0 of the input matrix.

3) Calculate the weighted sum Oi(net activation) as follows:

$$O_i = \sum_{j=1}^{n} w_{ji} x_j$$

Where, i=0, 1, 2……..n.

4) Calculate the sum of positive weight, Pwi of the weight_matrix.

5) Calculate Yi=f (Oi) =Oi/Pwi.
Where,
Oi →Net activation for each character i (e.g. 0, 1, 2,……9,A, B……Z)
Pwi → the sum of positive weight of the weight matrix for the each character.

6) Pick the maximum Yi.

7) Check if the corresponding neuron fires. If neuron fires then save the weight matrix into a file.

8) Check with other training samples of the same character and update the weight described as above.

9) Fixed the weight matrix and save into a file after the final training samples for that character.

10) Repeat steps 1 to 8 for any other character.

11) If the training is complete then the network is established.

6.3.2 Algorithm for Recognition:

1) Take a random sample for any character and then generate 12×8 matrix.

2) Calculate the weighted sum Oi(net activation) as follows:

$$O_i = \sum_{j=1}^{n} w_{ji} x_j$$

Where, i=0, 1, 2……..n.

3) Calculate the sum of positive weight, Pwi of the weight matrix.

4) Calculate Yi=f (Oi) =Oi/Pwi.

Where,

Oi →Net activation for each character i (e.g. 0, 1, 2,……9,A, B……Z)

Pwi→ the sum of positive weight of the weight matrix for the each character.

5) Pick the maximum Yi.

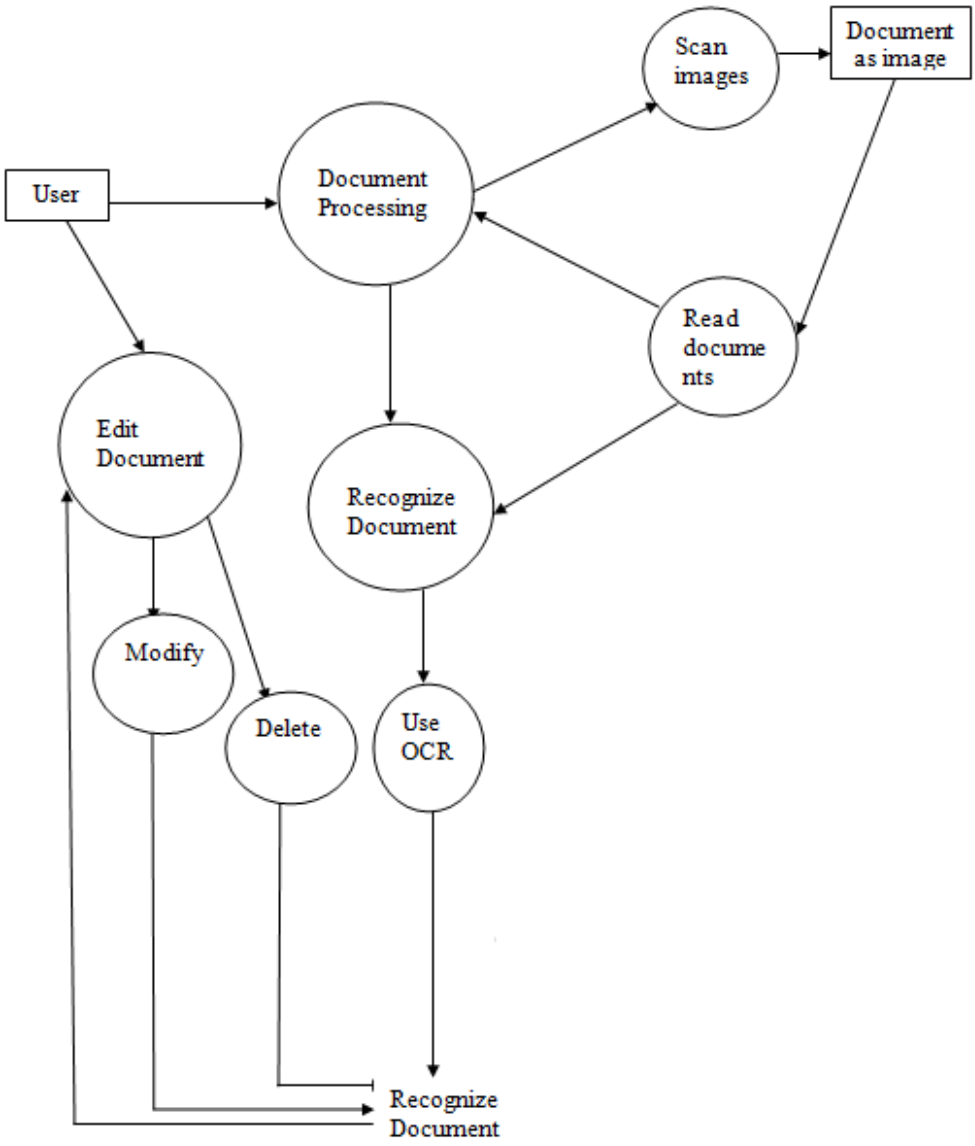
6) Check if the corresponding neuron fires.

7) If neuron fires, recognize corresponding character.

8) Repeat steps 1 to 7 for any other character.

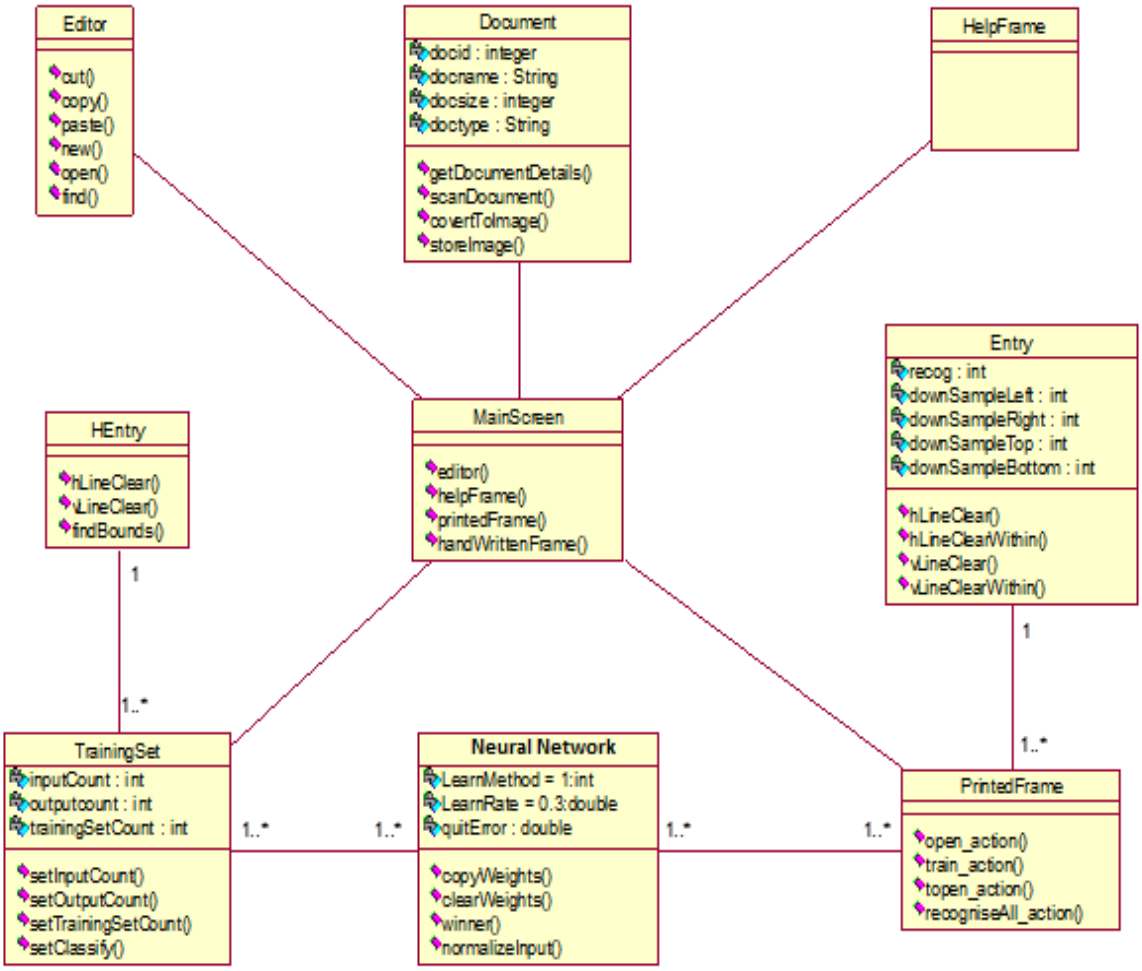
6.4 Design Details

6.4.1 DFD Diagram:



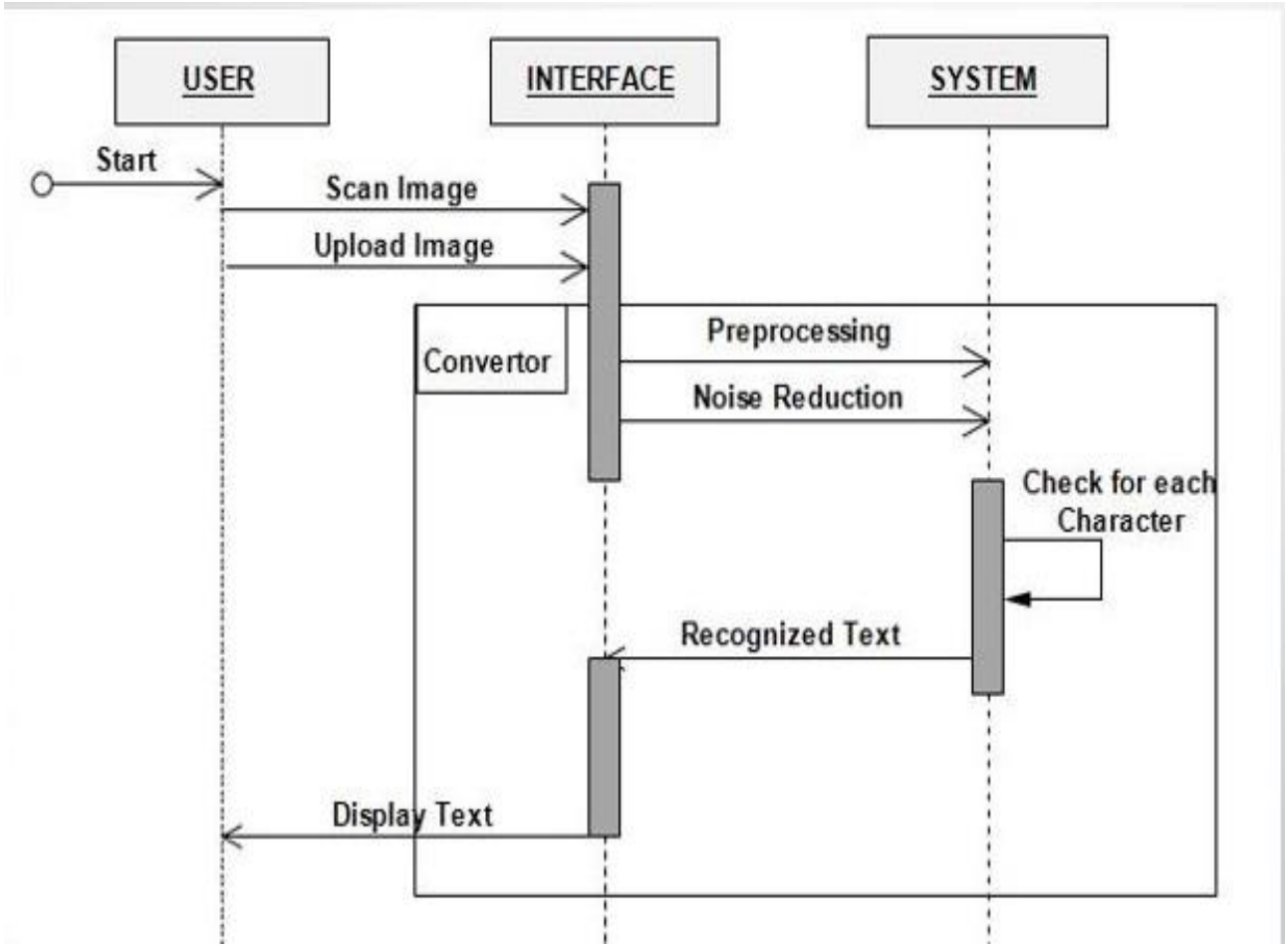
6.4.1 DFD diagram use for the working in OCR

6.4.2. Class Diagram:



6.4.2 Class Diagram

6.4.3 Sequence diagram:



6.4.3 Sequence Diagram

Chapter 7

Implementation Details

7.1 Methodology

The whole recognition process consists of four basic steps: pre-processing, normalized character matrix creation, network establishment and recognition. Pre-processing consists of digitization, noise removal and boundary detection of the digitized character matrix.

7.1.1 Input Character Image:

Our system is able to recognize any coloured printed character image with white background and font size is between 18 and 96.

7.1.2. Digitization and Matrix Creation from Character Image

In order to able to recognize characters by computer the character image is first digitized into a matrix i.e. transformed into a binary form for the ease of handling by the computer.

Colour image is first converted to the grey scale image as follows:

Y= (int) (0.33 * R + 0.56 *G +0.11 *B)

Where, R is the red component of a colour pixel, G is the green component of a colour pixel, B is the blue component of a colour pixel and Y is the value of a pixel in a gray scale level. The image is then converted to binary. Just replace all the gray levels of the image into binary value 0(for gray level value 129 to 255) treated as absence of writing or 1 (for gray level value 0 to 128) treated as presence of writing.

7.1.3. Boundary Detection

After creating the digitized binary matrix from the input character image, the detection of boundary is very much important to recognize character correctly. The boundary detection procedure is therefore given by

i. For top boundary detection, scan the character matrix starts at the top-left corner and remove all rows from top having only 0's. To detect top boundary there must be at least two consecutive 1's in two consecutive rows. Then the first row of the two consecutive rows from top will be selected as top boundary.

ii. For bottom boundary detection, scan the character matrix starts at the bottom-left corner and remove all rows from bottom having only 0's. To detect bottom boundary there must be at least two consecutive 1's in two consecutive rows. Then the first row of the two consecutive rows from bottom will be selected as bottom boundary.

iii. For left boundary detection, scan the character matrix starts at the top-left corner and remove all columns from left having only 0's. To detect left boundary there must be at least two consecutive 1's in two consecutive columns. Then the first column of the two consecutive columns from left will be selected as left boundary.

iv. For right boundary detection, scan the character matrix starts at the top-right corner and remove all columns from right having only 0's. To detect right boundary there must be at least two consecutive 1's in two consecutive columns. Then the first column of the two consecutive columns from right will be selected as right boundary.

7.1.4. Normalization

Normalization is the process of equating the size of all extracted character bitmaps (binary array).For size invariant character recognition, we have converted the boundary detected input character matrix into normalized matrix. The normalization procedure is as follows:

I. Take top row and left column assuming they contain salient features.

II. Then take alternate row and column until desired character matrix is found. For example, the size of input character (after boundary detection) is 15×11.Then during normalization,
   a. take first row
   b. delete row 2, 4, 6.
   c. Take all the remaining row d. Converted the matrix to 12×11

III. Finally the matrix is converted to desired form.

7.2 Result of Implementation

Here are the screenshots of the Optical characters Reorganization.

7.2.1 GUI Design:

7.2.1.1 Main Form:

Open the menu bar item on screen page

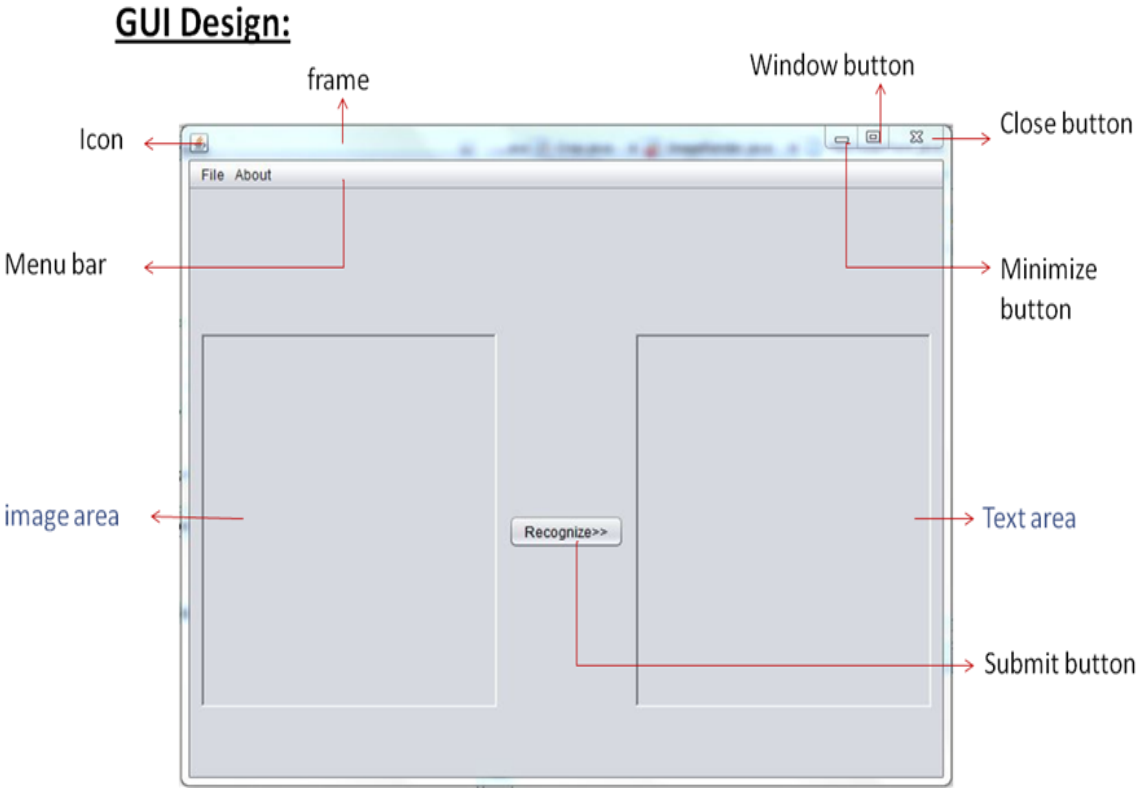


Fig 7.2.1.1 Frame menus of Implimented System

7.2.1.2 File Chooser Form

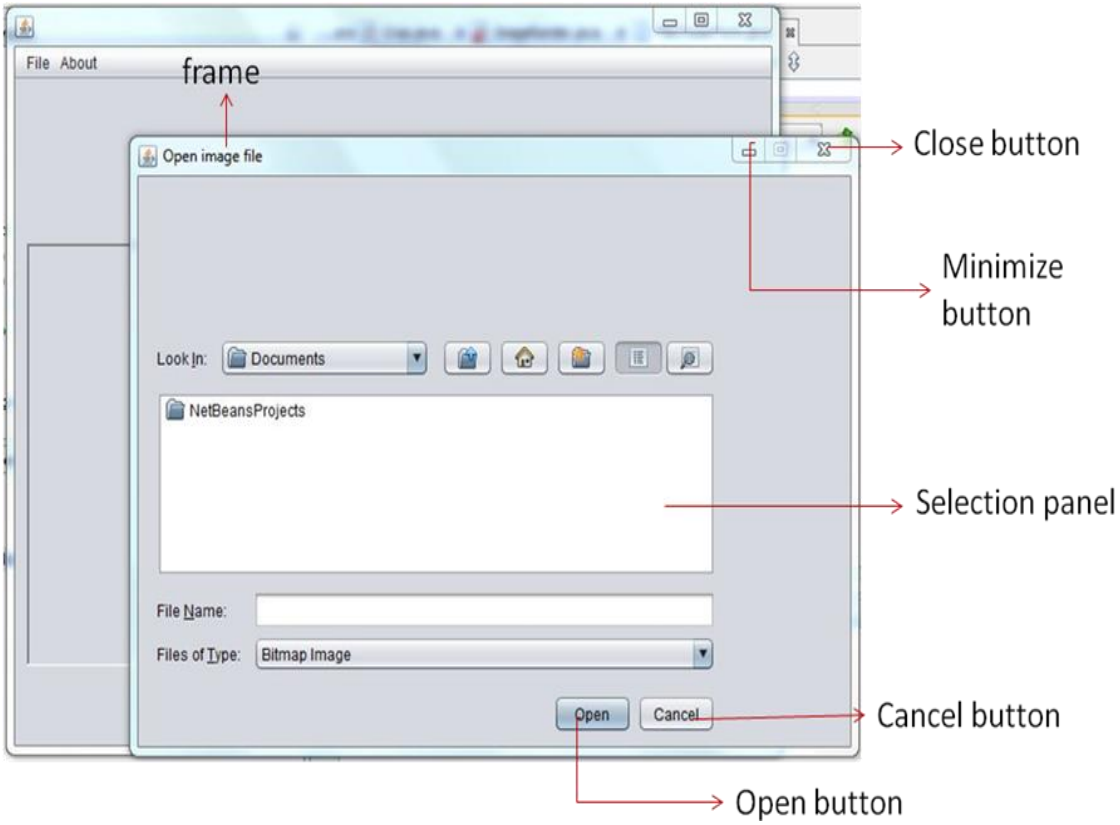


Fig 7.2.1.2 File Chooser form of Implimented System

7.2.1.3 Final Result Page

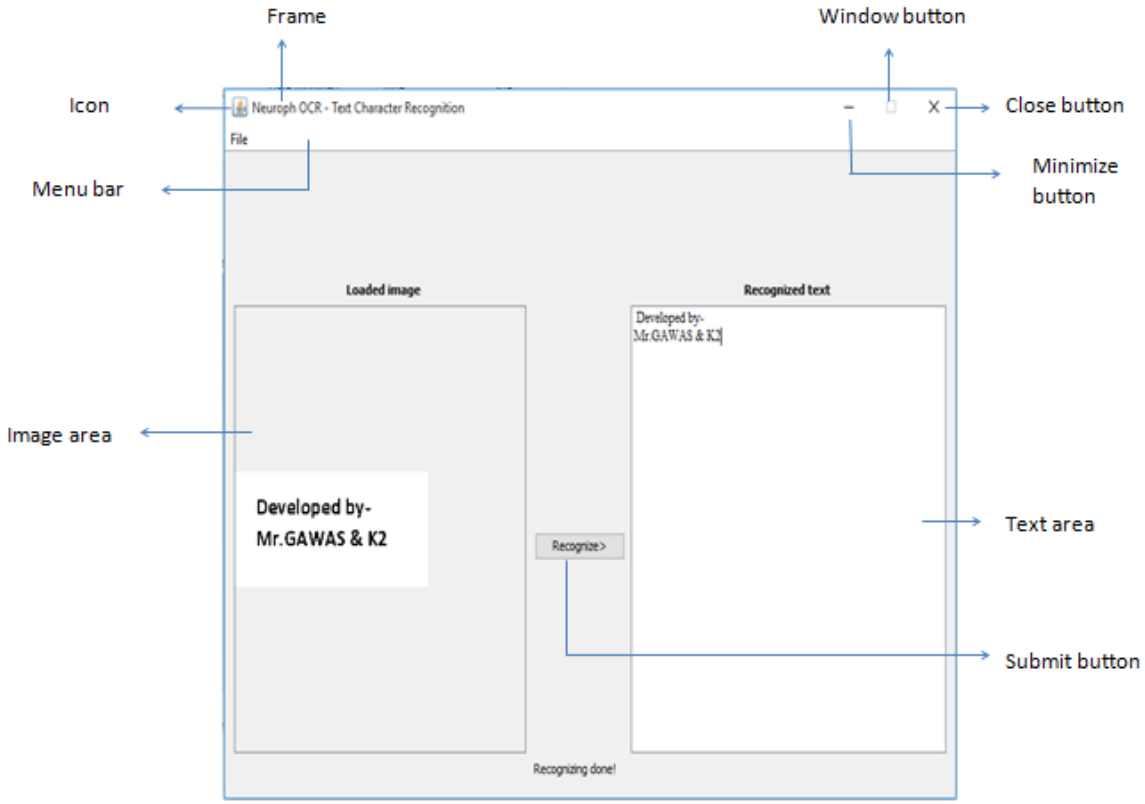


Fig 7.2.1.3 Final Result of OCR of Implemented System

Chapter 8

Technology Used

8.1 Neural Network:

An Artificial Neural Network (ANN), usually called neural network (NN), is a mathematical model or computational model that is inspired by the structure and/or functional aspects of biological neural networks. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase.

Modern neural networks are nonlinear statistical data modelling tools. They are usually used to model complex relationships between inputs and outputs or to find patterns in data. An artificial neural network (ANN), usually called "neural network" (NN), is a mathematical model or computational model that tries to simulate the structure and/or functional aspects of biological neural networks. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase.

8.1.1 ANN Training and Classification

Before the character recognition can take place, the ANN is 'trained', so that it can develop the capability of mapping various inputs to the required outputs and effectively classify various characters. For training the ANN, we use the 'Vectors' generated by the 'Database Templates' using the above mentioned Feature Extraction techniques. The above mentioned 7 different types of features have been used to generate 11 parameters (some of same type but different values), which are fed to the ANN. Thus, a matrix of 11x36 values is fed to the ANN to receive 36 different values at the output, one for each character in the database.

It may be noted that the ANN uses Backpropagation algorithm for Learning. The 'Target' values are specified by the system programmer to accommodate for small recognition errors, which may be changed from application to application.

8.2 Java:

Java is a recently developed, concurrent, class-based, object-oriented programming and runtime environment, consisting of:

- A programming language
- An API specification
- A virtual machine specification

Java has the following characteristics:

- **Object oriented** - Java provides the basic object technology of C++ with some enhancements and some deletions.
- **Architecture neutral** - Java source code is compiled into architecture-independent object code. The object code is interpreted by a Java Virtual Machine (JVM) on the target architecture.
- **Portable** - Java implements additional portability standards. For example, ints are always 32-bit, 2's-complemented integers. User interfaces are built through an abstract window system that is readily implemented in Solaris and other operating environments.
- **Distributed** - Java contains extensive TCP/IP networking facilities. Library routines support protocols such as HyperText Transfer Protocol (HTTP) and file transfer protocol (FTP).
- **Robust** - Both the Java compiler and the Java interpreter provide extensive error checking. Java manages all dynamic memory, checks array bounds, and other exceptions.
- **Secure** - Features of C and C++ that often result in illegal memory accesses are not in the Java language. The interpreter also applies several tests to the compiled code to check for illegal code. After these tests, the compiled code causes no operand stack over- or underflows, performs no illegal data conversions, performs only legal object field accesses, and all opcode parameter types are verified as legal.
- **High performance** - Compilation of programs to an architecture independent machine-like language, results in a small efficient interpreter of Java programs. The Java environment also compiles the Java bytecode into native machine code at runtime.
- **Multithreaded** - Multithreading is built into the Java language. It can improve interactive performance by allowing operations, such as loading an image, to be performed while continuing to process user actions.
- **Dynamic** - Java does not link invoked modules until runtime.

8.3 NetBeans:

NetBeans is coded in Java and runs on most operating systems with a Java Virtual Machine (JVM), including Solaris, Mac OS, and Linux.

NetBeans manages the following platform features and components:

- User settings
- Windows (placement, appearance, etc.)
- NetBeans Visual Library
- Storage
- Integrated development tools
- Framework wizard

NetBeans uses components, also known as modules, to enable software development. NetBeans dynamically installs modules and allows users to download updated features and digitally authenticated upgrades.

NetBeans IDE modules include NetBeans Profiler, a Graphical User Interface (GUI) design tool, and NetBeans JavaScript Editor.

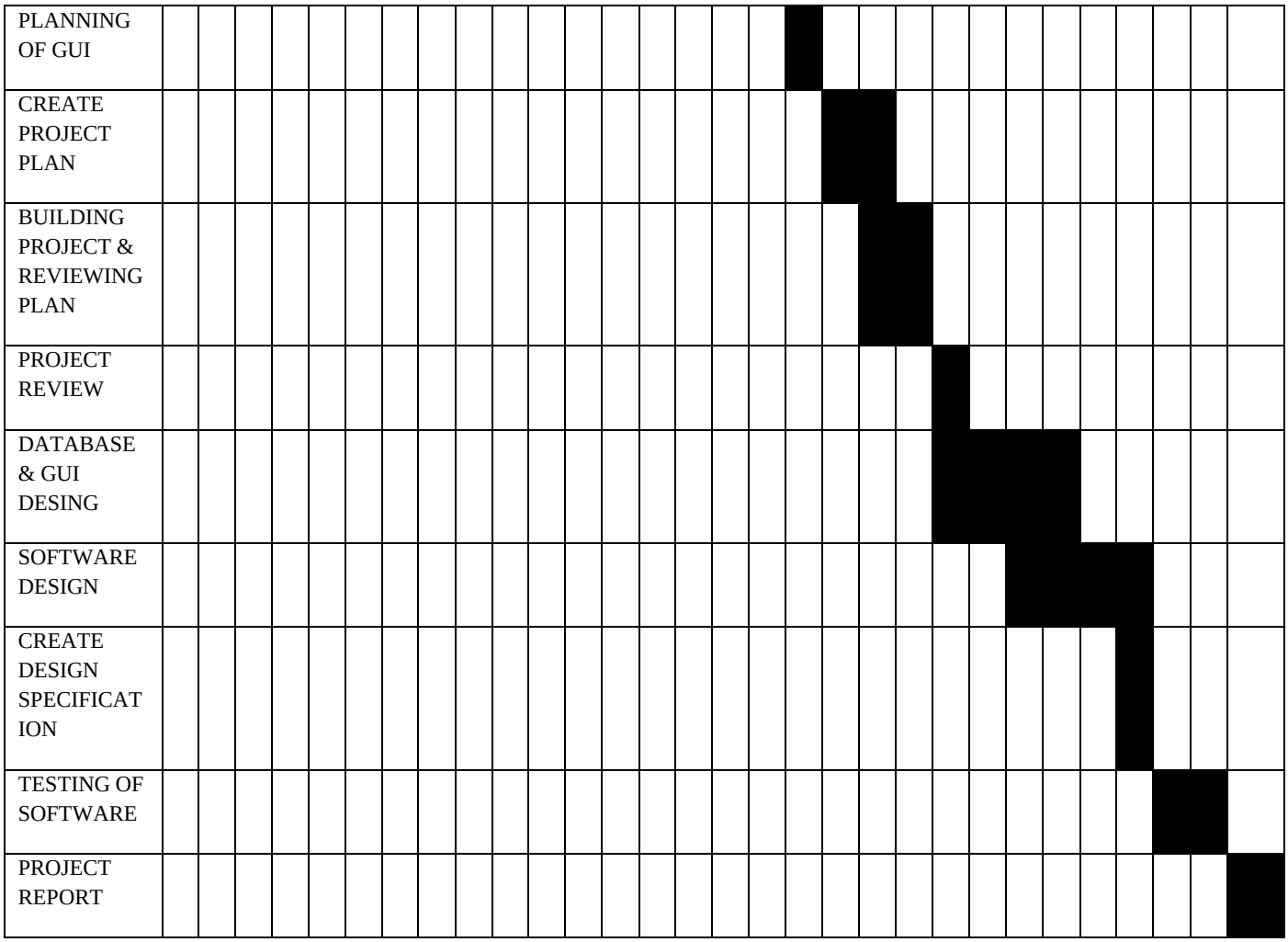
NetBeans framework reusability simplifies Java Swing desktop application development, which provides platform extension capabilities to third-party developers.

Chapter 9

Project timeline and Task Distribution

9.1 Timeline chart :

MONTHS	AUG				SEPT				OCT				NOV/DEC				JAN				FEB				MAR				AR	
WORK	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2
REQUIREMENT GATHERING	■	■	■																											
UNDURSTANDING THE DEPTH OF SUBJECT				■	■																									
LEARNING ALGORITHM					■	■																								
ANALYSIS OF PREPOSED SOFTWARE						■																								
SOFTWARE REQUIRMENT ANALYSIS							■	■																						
DESINGING CORE MODULE								■	■	■																				
DATA PROCESS MODELLING											■	■																		
PERSONAL SPECIFICATION																	■													



9.1 **Project timeline**

9.2 Task Distribution:

SR.NO.	WEEKS	NAME OF STUDENTS	DESCRIPTION
1	Jan	Rupesh Gawas	Functional Specification (Netbean Platform)
		Ketan Kadam	Planning of GUI
2	Jan	Rupesh Gawas	Create Project Plan
		Ketan Kadam	Building Project & Reviewing Plan (Learning of an Algorithms)
		Rupesh Gawas	
	Jan	Ketan Kadam	Implementation of GUI
		Rupesh Gawas	
3	Feb	Rupesh Gawas	Project Review
		Ketan Kadam	
		Ketan Kadam	Database & Interface Design (Design Training Database)
		Rupesh Gawas	
		Ketan Kadam	Software Design
		Rupesh Gawas	
	Feb	Ketan Kadam	Create Design Specification
		Rupesh Gawas	Design Complete
4	Feb	Rupesh Gawas	Design Coding
		Ketan Kadam	
	March	Ketan Kadam	Project Review
		Rupesh Gawas	
	March	Ketan Kadam	System Programming (JAVA Language)
		Rupesh Gawas	
5	March/April	Rupesh Gawas	Testing
		Ketan Kadam	
6	April	Ketan Kadam	Project Review
		Rupesh Gawas	
7	April	Ketan Kadam	Maintenance
		Rupesh Gawas	

9.3 Test Cases

Test id	Test Objective	Steps	Expected Result	Actual Result	Status
1	Image Browsing	Click on the Browse button	Image should be display on image area	Image displayed successfully	Pass
2	Recognizing	Click on Recognize button	Recognize character	Characters are display	Pass
3	Save Document	Click on Save Menu item	Document should be save	Document save successfully	Pass

Chapter 10

Future Work:

In proposed methodology of OCR system we have implemented for only English language.

It will also implement of Devnagari characters or any other characters.
This will provide more usability for user.

Chapter 11

Conclusion

The system has its advantages such as Less Time Complexity, Very Small Database and High Adaptability to untrained inputs, with only a small number of features to calculate as compared to the method followed in Yet, the system has a large scope for further developments.

We have proposed an artificial neural network-based simple colour and size invariant character recognition system to recognize English alphanumeric characters.

Our proposed system gives excellent result for numeric digits and letters when they are trained and tested separately but produce satisfactory result when they are processed together. In addition our system is computationally inexpensive and easier to implement.

Chapter 12

References:

- J. Kamruzzaman and S. M. Aziz ,"A Neural Network Based Character Recognition System Using Double Backpropagation," Malaysian Journal of Computer Science, Vol. 11 No. 1, pp. 58-64, June 1998.

- Guyanet. al., "Design of a Neural Network Character Recognizer for a Touch Terminal", Pattern Recognition, Vol. 24, No. 2, pp. 105-119, 1991.

- B. Widrow, R. Winter and R. A. Bayter., "Layered Neural Nets for Pattern Recognition", IEEE Trans. Acoustics, Speech and Signal Process. Vol. 36, No. 7, pp. 1109-1118,1998.

- F.Li, ands. Gao ," Character Recognition System Based on Back-Propagation Neural Network," International Conference on Machine Vision and Human-Machine Interface (MVHI),pp.393- 396,2010.