Handwritten Character Recognition (HCR) USING NEURAL NETWORK

Thesis · September 2009

DOI: 10.13140/RG.2.2.21287.24488

1 author:

Hitesh Mohapatra
Veer Surendra Sai University of Technology
35 PUBLICATIONS **1** CITATION

SEE PROFILE

HCR USING NEURAL NETWORK

THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE
REQUIREMENTS
OF

MASTER OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

BY

HITESH MOHAPATRA
REGD.NO:07071706021

UNDER THE GUIDENCE OF

PROF.(DR.) PRASANT KUMAR PATRA
PROF.(DR.) SIBA PRASAD PANIGRAHI

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COLLEGE OF ENGINEERING & TECHNOLOGY
BHUBANESWAR-751003

A Constituent College of Biju Patnaik University of Technology,Odisha

October 2009

Department of Computer Science and Engineering
College Of Engineering And Technology
(a Constituent College of Biju Patnaik University of Technology,Odisha)
Techno Capmus,Ghatikia,Kalinganagar,Bhubaneswar-751003,India

Prof(Dr..)Prashant Kumar Patra **Letter No._____/CSE/_____** **Dated.__/__.____**

Professor and Head

CERTIFICATE

This is to certify that project work entitled "**HCR using Neural Network**" carried out by **Hitesh Mohapatra under the guidance's of Prof(Dr.)P.K.Patra and Prof.(Dr.)S.P.Panigrahi, Regd. No: 0707106021,** and a student of 2nd year M. Tech in Computer Science and Engineering from College of Engineering and Technology , Bhubaneswar,Odisha , has been satisfactorily completed by him and worthy of acceptance for the degree of Master of Technology in Computer Science and Engineering ,Biju Patnaik University of Technology,Rourkela,Odisha.

1._____

2._____

Department of Computer Science and Engineering
College Of Engineering And Technology
(a Constituent College of Biju Patnaik University of Technology,Odisha)
Techno Capmus,Ghatikia,Kalinganagar,Bhubaneswar-751003,India

Prof(Dr..)Prashant Kumar Patra **Letter No._____/CSE/_____** **Dated.__/__.____**

Professor and Head

CERTIFICATE OF APPROVAL

The foregoing thesis is hereby accepted as a credible study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorsed or approve any statement made ,opinion expressed or conclusion drawn therein, but approve the thesis for the purpose for which it is submitted.

Final Examination for the evaluation thesis.

1. _____

2. _____

3. _____

(Signature of Examiners)

** Only in the case the thesis is approved.*

ACKNOWLEDGEMENT

My special gratitude to my project guide Prof.P.K.Patra for his inspiration,adroit guidance,constant supervision and constructive criticism in successful completion of the project.

I am very grateful to express my deep sense of gratitude to Dr.P.K .Patra,Professor and Head of Computer science and Engineering Department,College of Engineering and Technology,Bhubaneswar for his constant encouragement through out the project .

I am very grateful to my external guide, Dr.S.P.Panigrahi ,Professor,Department of Electrical Engineering , Krupajal Engineering College, Bhubaneswar for creating an excellent climate , which made this endeavour possible.

I am very thankful to Miss.Sasmita Nayak, lecturer in CIT,Bhubaneswar for proving adequate and prompt facilities during the course of my project.

I express my gratitude to all the professors and lecturers of our department for their cooperation and keen interest throughout this project.

My sincere thanks to all my friends who helped me during this project.

Hitesh Mohapatra
Regd.No: 0707106021

SYNOPSIS

Handwriting recognition is the ability of a machine to receive and interpret handwritten input from multiple sources like paper documents, photographs, touch screen devices etc. Recognition of handwritten and machine characters is an emerging area of research and finds extensive applications in banks, offices and industries.

The main aim of this project is to design expert system for ,

"HCR(English) using Neural Network".

that can effectively recognize a particular character of type format using the Artificial Neural Network approach. Neural computing Is comparatively new field, and design components are therefore less well specified than those of other architectures. Neural computers implement data parallelism. Neural computer are operated in way which is completely different from the operation of normal computers. Neural computer are trained (not Programmed) so that given a certain starting state (data input); they either classify the input data into one of the number of classes or cause the original data to evolve in such a way that a certain desirable property is optimized.

Keywords:

Off-line Handwritten Recognition,Handwritten Character,Pattern Recognition ,Feature Extraction,Neural Network.

Contents

1 Introduction **1**

1.1 Project Definition 1
1.2 Relevant Theory 1
1.2.1 Benefits of Character Recognition : 1
1.2.2 Implementation of HCR : 2
1.2.3 What is Neural Network 2
1.2.4 Why use Neural Network 2
1.3 Literature Survey 3
1.3.1 Offline Handwritten English Numerals Recognition using Correlation Method 3
1.3.2 Recognition of Handwritten Hindi Characters using Backpropagation Neural Network 3
1.3.3 Devnagiri Character Recognition Using Neural Networks 3
1.3.4 Intelligent Systems for Off-Line Handwritten Character Recognition: A Review 3
1.3.5 Fuzzy Based Handwritten Character Recognition System 4
1.3.6 An Overview of Character Recognition Focused on Off-Line Handwriting 4
1.3.7 Handwritten Devanagari Character Recognition using Neural Network 4
1.3.8 Recognition of Handwritten Devnagari Characters through Segmentation and Artificial neural networks 4
1.3.9 A Recognition System For Handwritten Gurumukhi Characters . .5
1.3.10 Image preprocessing for optical character recognition using neural networks 5
1.3.11 Recognition for Handwritten English Letters: A Review 5
1.3.12 Diagonal Based Feature Extraction For Handwritten Alphabets Recognition System Using Neural Network 5
1.4 Scope 6
1.5 Objective 6

2 Requirement Analysis **7**

2.1 Requirement Specification 7
2.1.1 Functional Requirements 7
2.1.2 Normal Requirements 7
2.1.3 Expected Requirements 8
2.1.4 Excited Requirements 8
2.1.5 Non-Functional requirements 8
2.2 Validation of Requirements 9

2.3 System Requirements 9
2.3.1 Hardware Requirements 9
2.3.2 Software Requirements 9

3 System Design **10**
3.1 Process Model 10
3.1.1 Incremental Model 11
3.1.2 Advantages of Incremental Model 12
3.1.3 Why we use Incremental Model? 12
3.1.4 Characteristics of Incremental Model 12
3.2 Breakdown Structure(Modules) 13
3.2.1 Image Preprocessing 13
3.2.2 Segmentation 13
3.2.3 Feature Extraction 13
3.2.4 Classification 14
3.3 Project Estimation 14
3.3.1 Estimation of KLOC 14
3.3.2 Efforts 14
3.3.3 Development Time for Implementation and Testing 14
3.3.4 Development Time for Project 14
3.3.5 Number Of Persons 15

4 System Analysis **16**
4.1 Projects scheduling and Tracking 16
4.1.1 Project Work Breakdown Structure(Analysis) 16
4.1.2 Project Work Breakdown Structure(Implementation) 18
4.2 Tasks 19
4.3 Project Schedule 21
4.3.1 Time Line Chart 23
4.4 Analysis Model 24
4.4.1 Behavioural Model 24
4.4.2 Functional Modelling 27
4.4.3 Architectural Diagram 29
4.5 Mathematical Model 30
4.5.1 Set Theory 31
4.5.2 Venn Diagram 32
4.5.3 State Diagram 33
4.5.4 Time and Space Computation 34

5 Risk Management **35**
5.1 Risk Identification 35
5.1.1 Product Size Related 35
5.1.2 Customer Related 35
5.1.3 Process Risk 35
5.1.4 Technical Risk 35
5.1.5 Development Environment Related 35
5.2 Risk Projection 35
5.2.1 Risk Table 35
5.2.2 Strategies Used To Manage Risks 36

5.2.3 Risk Table Along With RMMM Plan 36
5.3 Feasibility 37
5.3.1 Technical Feasibility 37
5.3.2 Cost Feasibility 38

6 Technical Specification **39**
6.1 Technology Details Used in Project 39
6.1.1 JAVA Development Kit 39
6.1.2 Eclipse 40
6.2 References to Technology 40

7 Software Implementation **41**
7.1 Introduction 41
7.2 Important Modules And Algorithm Used 41
7.2.1 Module 1: Image Processing 41
7.2.2 Module 2: Segmentation 43
7.2.3 Module 3: Feature Extraction 44
7.2.4 Module 4: Training And Recognition 44

8 Software Testing **48**
8.1 Introduction 48
8.1.1 Unit Testing: 48
8.1.2 Integration testing 48
8.1.3 Validation testing 49
8.1.4 GUI Testing 49
8.2 Test cases 50
8.3 Snap shots of Test Cases and Test Results 52

9 Results **57**

10 Deployment And Maintenance Details **59**
10.1 Installation And Un-installation 59
10.1.1 Installation 59
10.1.2 Un-installation 59
10.2 User Help 59

11 Conclusion and Future Scope **60**
11.1 Conclusion 60
11.2 Future Scope 60

References **60**

List of Figures

3.1 Incremental Model 11
3.2 Breakdown Model 13

4.1 Project Work Breakdown structure(Analysis) 17
4.2 Project Work Breakdown structure(Implementation) 19
4.3 Time line Chart(Analysis Phase) 23
4.4 Time line Chart(Implementation Phase) 23
4.5 Use Case Diagram 24
4.6 Sequence Diagram of Preprocessing 25
4.7 Sequence Diagram of Segmentation 25
4.8 Sequence Diagram of Feature Extraction 26
4.9 Sequence Diagram of Classification 26
4.10 DFD(level 0) for HCR System 27
4.11 DFD(level 1) for Image Reader 27
4.12 DFD(level 1) for Recognition Unit 27
4.13 Data Flow Diagram for ANN) 28
4.14 Control Flow Diagram for HCR System 28
4.15 Deployment Diagram 29
4.16 Venn Diagram 32
4.17 State Diagram 33

6.1 JDK 40
6.2 Eclipse 40

8.1 Integration testing 49
8.2 Information 52
8.3 Setting Project 53
8.4 Violation 53
8.5 Auto-fixable Violations 54
8.6 Metrics 54
8.7 Auto-Fixes 55
8.8 Project Summary 55
8.9 Developers 56
8.10 Console 56

9.1 Modules Result 57
9.2 Recognized Text 57
9.3 Input Image 58
9.4 Recognized Text 58

11.1 Project Planner 65
11.2 Plagiarism Report 66

List of Tables

4.1 Project Schedule 21
4.2 Project Table 22

5.1 Risk Table 36
5.2 Risk Table Along With RMMM Plan 36

8.1 Test case 1 50
8.2 Test case 2 50
8.3 Test case 3 50
8.4 Test case 4 51
8.5 Test case 5 51
8.6 Test case 6 51

11.1 Testing table 64

CHAPTER 1

Introduction

Handwriting recognition is the ability of a machine to receive and interpret handwritten input from multiple sources like paper documents, photographs, touch screen devices etc. Recognition of handwritten and machine characters is an emerging area of research and finds extensive applications in banks, offices and industries.The main aim of this project is to design expert system for , **"HCR using Neural Network"**that can effectivelly recognize a particular character of type format using the Artificial Neural Network approach.

Neural computing Is comparatively new field, and design components are therefore less well specified than those of other architectures. Neural computers implement data parallelism. Neural computer are operated in way which is completely different from the operation of normal computers. Neural computer are trained (not Programmed) so that given a certain starting state (data input); they either classify the input data into one of the number of classes or cause the original data to evolve in such a way that a certain desirable property is optimized.

1.1 Project Definition

This application is useful for recognizing all character(English) given as in input image. Once input image of character is given to proposed system, then it will recognize input character which is given in image.Recognition and classification of characters are done by Neural Network.The main aim of this project is to effectively recognize a particular character of type format using the Artificial Neural Network approach.

1.2 Relevant Theory

1.2.1 Benefits of Character Recognition :

1.The idea of Neural Network in HCR will brings us the reading of various combined style of writing a character.

2.In forensic application HCR will be an effective method for evidence collection.

3.It will also help to reduce noise from the original character.

4.Our method develop accuracy in recognizing character in divert font and size.

5.More set of sample invites more accuracy rate because of heavy training and testing session.

1.2.2 Implementation of HCR :

HCR works in stages as preprocessing, segmentation, feature extraction and recognition using neural network. Preprocessing includes series of operations to be carried out on document image to make it ready for segmentation. During segmentation the document image is segmented into individual character or numeric image then feature extraction technique is applied on character image. Finally feature vector is presented to the selected algorithm for recognition.Here this extracted features are provided to NN for recognition of character.

1.2.3 What is Neural Network

An Artificial Neural Network (ANN) is an information-processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information.The key element of this paradigm is the novel structure of the information processing system. It is composed of large no. of highly interconnected processing element (neurons) working in union to solve specific problems. ANN's like peopling, learning by example. An ANN is configured for a specific application. such as pattern recognition or data classification, through a learning process .Learning in a Biological system involves adjustments to the synaptic connections that exist between the neuron.

1.2.4 Why use Neural Network

Neural network with their remarkable ability to derive meaning from complicated or imprecise data can be use to extract pattern and detect trend that are too complex to be noticed by either human or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer "what if" questions. Other Advantages Include:

- Adaptive Learning: An ability to learn how to do tasks based on the data given for training or initial experience.
- Self-Organization:An ANN can create its own organization or representation of the information it receives during learning time.
- Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
- Fault Tolerance Via Redundant Information coding: partial destruction of network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

1.3 Literature Survey

1.3.1 Offline Handwritten English Numerals Recognition using Correlation Method

In this paper author has proposed system is to efficiently recognize the offline handwritten digits with a higher accuracy than previous works done. Also previous handwritten number recognition systems are based on only recognizing single digits and they are not capable of recognizing multiple numbers at one time.So the author has focused on efficiently performing segmentation for isolating the digits. [1]

1.3.2 Recognition of Handwritten Hindi Characters using Backpropagation Neural Network

Automatic recognition of handwritten characters is a difficult task because characters are written in various curved and cursive ways, so they could be of different sizes, orientation, thickness, format and dimension. An offline handwritten Hindi character recognition system using neural network is presented in this paper. Neural networks are good at recognizing handwritten characters as these networks are insensitive to the missing data. The paper proposes the approach to recognize Hindi characters in four stages 1) Scanning, 2) Preprocessing, 3) Feature Extraction and, 4) Recognition. Preprocessing includes noise reduction, binarization, normalization and thinning. Feature extraction includes extracting some useful information out of the thinned image in the form of a feature vector. The feature vector comprises of pixels values of normalized character image. A Back propagation neural network is used for classification. Experimental result shows that this approach provides better results as compared to other techniques in terms of recognition accuracy, training time and classification time. The average accuracy of recognition of the system is 93 %. [2]

1.3.3 Devnagiri Character Recognition Using Neural Networks

Neural network approach is proposed to build an automatic offline character recognition system. Devnagari is an Indo-Aryan language spoken by about 71 million people mainly in the Indian state of Maharashtra and neighboring states. One may find so much work for Indian languages like Hindi, kanada, Tamil, Bangala, Malayalam etc but devnagari is a language for which hardly any work is traceable especially for character recognition. In this paper, work has been performed to recognize Devnagari characters using multilayer perceptron with hidden layer. Various patterns of characters are created in the matrix (n*n) with the use of binary form and stored in the file. We have used the back propagation neural network for efficient recognition and rectified neuron values were transmitted by feed forward method in the neural network. [3]

1.3.4 Intelligent Systems for Off-Line Handwritten Character Recognition: A Review

Handwritten character recognition is always a frontier area of research in the field of pattern recognition and image processing and there is a large demand for Optical Character

Recognition on hand written documents. This paper provides a comprehensive review of existing works in handwritten character recognition based on soft computing technique during the past decade. [4]

1.3.5 Fuzzy Based Handwritten Character Recognition System

This paper presents a fuzzy approach to recognize characters. Fuzzy sets and fuzzy logic are used as bases for representation of fuzzy character and for recognition. This paper describes a fuzzy based algorithm which first segments the character and then using fuzzy system gives the possible characters that match the given input and then using defuzzication system finally recognizes the character. [5]

1.3.6 An Overview of Character Recognition Focused on Off-Line Handwriting

Character recognition (CR) has been extensively studied in the last half century and progressed to a level sufficient to produce technology driven applications. Now, the rapidly growing computational power enables the implementation of the present CR methodologies and creates an increasing demand on many emerging application domains, which require more advanced methodologies.[6]

1.3.7 Handwritten Devanagari Character Recognition using Neural Network

In this digital era, most important thing is to deal with digital documents, organizations using handwritten documents for storing their information can use handwritten character recognition to convert this information into digital. Handwritten Devanagari characters are more difficult for recognition due to presence of header line, conjunct characters and similarity in shapes of multiple characters. This paper deals with development of grid based method which is combination of image centroid zone and zone centroid zone of individual character or numerical image. In feature extraction using grid or zone based approach individual character or numerical image is divided into n equal sized grids or zones then average distance of all pixels with respect to image centroid or grid centroid is computed. In combination of image centroid and zone centroid approach it computes average distance of all pixels present in each grid with respect to image centroid as well as zone centroid which gives feature vector of size 2xn features. This feature vector is presented to feed forward neural network for recognition. Complete process of Devanagari character recognition works in stages as document preprocessing, segmentation, feature extraction using grid based approach followed by recognition using feed forward neural network. [7]

1.3.8 Recognition of Handwritten Devnagari Characters through Segmentation and Artificial neural networks

Handwritten character recognition is the ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touchscreens and other devices.Handwritten Marathi Characters are more complex for recognition than corresponding English characters due to many possible variations in order, number, direction and shape of the constituent strokes. The main purpose of this paper is to introduce

a new method for recognition of offline handwritten devnagari characters using segmentation and Artificial neural networks. The whole process of recognition includes two phases-segmentation of characters into line, word and characters and then recognition through feed-forward neural network.[8]

1.3.9 A Recognition System For Handwritten Gurumukhi Characters

This paper represent Handwritten Gurmukhi Character Recognition system using some statistical features like zone density, projection histograms , 8 directional zone density features in combination with some geometric features like area, perimeter, eccentricity, etc. The image document is first pre-processed by using many techniques like binarization, orphological operations (erosion and dilation) applied to remove noise and then segmented into isolated characters. The highest accuracy obtained by using these features and back propagation classifier is 98 %.[9]

1.3.10 Image preprocessing for optical character recognition using neural networks

Primary task of this master's thesis is to create a theoretical and practical basis of preprocessing of printed text for optical character recognition using forward-feed neural networks. Demonstration application was created and its parameters were set according to results of realized experiments.[10]

1.3.11 Recognition for Handwritten English Letters: A Review

Character recognition is one of the most interesting and challenging research areas in the field of Image processing. English character recognition has been extensively studied in the last half century. Nowadays different methodologies are in widespread use for character recognition. Document verification, digital library, reading bank deposit slips, reading postal addresses, extracting information from cheques, data entry, applications for credit cards, health insurance, loans, tax forms etc. are application areas of digital document processing. This paper gives an overview of research work carried out for recognition of hand written English letters. In Hand written text there is no constraint on the writing style. Hand written letters are difficult to recognize due to diverse human handwriting style, variation in angle, size and shape of letters. Various approaches of hand written character recognition are discussed here along with their performance.[11]

1.3.12 Diagonal Based Feature Extraction For Handwritten Alphabets Recognition System Using Neural Network

An off-line handwritten alphabetical character recognition system using multi layer feed forward neural network is described in the paper. A new method, called, diagonal based feature extraction is introduced for extracting the features of the handwritten alphabets. Fifty data sets, each containing 26 alphabets written by various people, are used for training the neural network and 570 different handwritten alphabetical characters are used for testing. The proposed recognition system performs quite well yielding higher levels of recognition

accuracy compared to the systems employing the conventional horizontal and vertical methods of feature extraction. This system will be suitable for converting handwritten documents into structural text form and recognizing handwritten names.[12]

1.4 Scope

1. System will be designed in way to ensure that offline Handwritten Recognition of English characters.

2. Our old and epic HCR literature can be restore in digital form.

3. Use of Neural Network for classification.

4. Large number of training data set will improve the efficiency of the suggested approach.

1.5 Objective

1. Use Neural signs in literature domain.

2. Reduced man-power to convert old literature into digitized form manually.

3. proposed system served as guide and working in character recognition areas.

4. Making rich the digitized library with English language.

CHAPTER 2

Requirement Analysis

Requirement analysis results in the specification of software's operational characteristics indicates software's interface with other system elements and establish constraints that software must meets. Requirement analysis allows the software engineer(sometime called Analyst or Modeler in this role) to elaborate on basis requirements during earlier requirement engineering task and build models that depict user scenarios,functional activities , problem classes and their relationship s,system and class behavior and the flow of data as it is transformed.

The requirements analysis task is a process of discovery , refinement , modeling and specification. The scope , initially established by us and refined during project planning , is refined in details.Model of the required data , information and control flow and operations behavior are created.

2.1 Requirement Specification

2.1.1 Functional Requirements

The functional requirements for a system describe what system do.

1. The developed system should recognize handwritten English character present in the image.
2. System shall show the error message to the user when given input is not in the required format.
3. System must provide the quality of service to user.
4. System must provide accuracy for character recognition.

2.1.2 Normal Requirements

These are the requirements clearly stated by the customer hence requirement must be present for customer satisfaction.

N1 : Application should have graphical user interface.

N2 : Input of characters with various font size and styles should recognize.

N3 : Database should identify computer based English character by comparison .

N4 : Application should be able for matching the stored patterns on input handwritten character.

N5 : Minimum 10*50 (characters * patterns) should be available for each character.

2.1.3 Expected Requirements

These requirements are implicit type of requirements.These requirements are not clearly stated by customer but even though customer expects them.

Exp1 : Instead of only one character application should take set of characters or text.

Exp2 Application should be user friendly and also easy to install.

Exp3 : By using Neural Network bringing more accuracy in character recognition process.

Exp4 : Application also recognize English numerals.

Exp5 : Minimum 26*50 (character * patterns) should be available for each character.

2.1.4 Excited Requirements

These requirements are neither stated by customer nor expected. But to make the customer satisfied , developer may include some unexpected requirements.

Exc1: Application interpret all the English alphabets through NN training process.

Exc2: Using this application Continuous handwritten characters need to be recognize.

Exc3: Alphanumerical characters with special symbol should be recognized with proposed system.

Exc4: Development of HCR system for noisy images.

2.1.5 Nonfunctional requirements

As the name suggest these are the requirements that are not directly interacted with specific functions delivered by the system.

Performance: Handwritten characters in the input image will be recognised with an accuracy of ablut 90

Functionality: This software will deliver on the functional requirements.

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.

2.2 Validation of Requirements

.

The project "HCR using Neural Network" will be recognized as successful implementation if it provide all the required images on the basis of suitable input with minimum time. The requirement specification define should be validated such a that the successful implementation of product can be recognized. Hence validation specifies classes of tests to be performed to validate function, performance and the constraints.

With respect to the system under consideration the following issues are to be validated to ensure consistency of system.

V1: The Neural Network are known to be capable of providing good recognition rate at the present as compare to other methods.

V2: Handwritten Character Recognition system give much better result in terms of performance and accuracy in comparison with existing usual approach due to the application of artificial way character recognition and Neural Network in detection of characters.

V3: Handwritten Character Recognition technology provides image definition, image pre-processing and image segmentation and recognition capabilities and still maintains high level of accuracy in the field of image processing.

2.3 System Requirements

.

2.3.1 Hardware Requirements

- Intel i3 Processer
- 128 MB RAM
- 10 GB Hard Disk.

2.3.2 Software Requirements

- Windows 7/8/8.1
- Language: Java(J2SE) JDK 1.7.
- Eclipse

CHAPTER 3

System Design

3.1 Process Model

Process Model are processes of the same nature that are classified together into a model. Thus, a process model is a description of a process at the type level. Since the process model is at the type level, a process is an instantiation of it. The same process model is used repeatedly for the development of many applications and thus, has many instantiations. One possible use of a process model is to prescribe how things must/should/could is done in contrast to the process itself which is really what happens. A process model is roughly anticipation of what the process will look like. What the process shall be determined during actual system development.

The goal of a process model is to be:

- Descriptive
  1. Track what actually happens during a process.
  2. Take the point of view of an external observer who looks at the way a process has been performed and determines the improvements that must be made to make it perform more effectively or efficiently.
- Prescriptive
  1. Define the desired processes and how they should/could/might be performed.
  2. Establish rules, guidelines, and behavior patterns which, if followed, would lead to the desired process performance. They can range from strict enforcement to flexible guidance.
- Explanatory
  1. Provide explanations about the rationale of processes.
  2. Explore and evaluate the several possible courses of action based on rational arguments.
  3. Establish an explicit link between processes and the requirements that the model needs to fulfil.
  4. Pre-defines points at which data can be extracted for reporting purposes.

3.1.1 Incremental Model

Incremental model is used as the process model in our system. Figure 3.1 shows the process model of the system. To save actual problems in an industry setting, Software Engineering must incorporate a development strategy that encompasses the process, method and the tool layers; this strategy is often referred as process model. A process model for Software Engineering is chosen base on the nature of the Project and its application. For our project, we have selected Incremental Model.

1. Using these models, a limited set of customer requirements are implemented quickly and are delivered to customer.

2. Modified and expanded requirements are implemented step by step.

3. It combines elements of linear Sequential Model with the iterative Philosophy of prototyping.

4. Each linear sequence produces a deliverable Increment of the Software.

5. Each linear Sequence is divide into 4 parts:-

- Analysis
- Design
- Code
- Testing

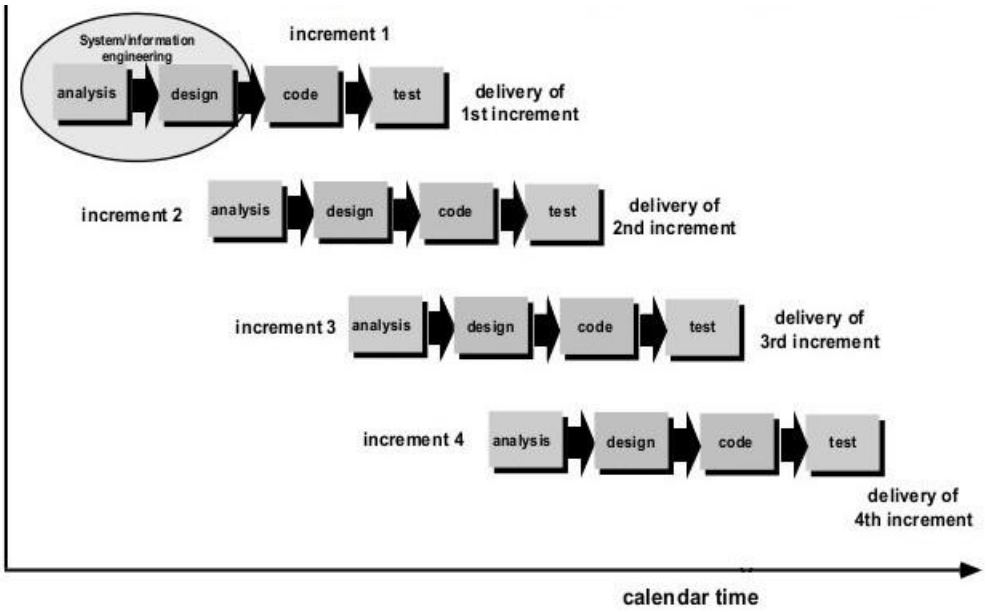

Figure 3.1: Incremental Model

1. **Analysis**
   It includes understanding of information domain, required functions, behavior, performance and interface. Requirements for the system and software are documented and reviewed with customer.

2. **Design**
   It is multiple processes that include four attributes of program data structure, software architecture, interface representation and procedural detail.

3. **Coding**
   Translation of design to machine code is done by this step.

4. **Testing**
   It focuses on Logical internals of Software and ensures that all statement is correct to uncover all hidden errors. For an incremental model, the first Increment is developed as a core model, which is used by the customer. Then as things are added after the first delivery, product gets and better.

3.1.2 Advantages of Incremental Model

1. Generates working Software quickly and early during the software life cycle.

2. More Flexible-less costly to change Scope and Requirements.

3. Easier to test and debug during a smaller iteration.

4. Customer can respond to each built.

3.1.3 Why we use Incremental Model?

The main aim of using the model is the reason that we have to add more features in the existing modules to increase project reliability and usability. Using this model we can adapt to the changing requirements of the customer which helps in developing the project in relatively small amount of time. The next increment implements customer suggestions plus some additional requirements in the previous increment. The process is repeated until the project is completed.

3.1.4 Characteristics of Incremental Model

1. Using these models a limited set of customers' requirements are implemented quickly and are delivered to customer then modified and expanded requirements are implemented step by step.

2. Each increments produces the product which is submitted to customer and suggests some change and increment implements that changes with some extra requirements to previous.

3. Incremental model does not facilitate the development of project in one go. This is useful for developing modules and then testing them which helps us to modularize the entire project for better handling.

So finally, it is easier to develop project in increments. We can develop a working prototype 1st with just basic functions and then build upon this prototype in later increments. This will help to reduce the complexity of system by dividing entire system in different levels of priority. We have discarded the other process models based on following points:

1. When staff is less then we can go for step by step evolution of increments.
2. In case of increment process model, scope for intermediate requirements change and testing.

3.2 Breakdown Structure (Modules)

As shown in Fig 3.2 implemented of proposed work is divided in the following modules:

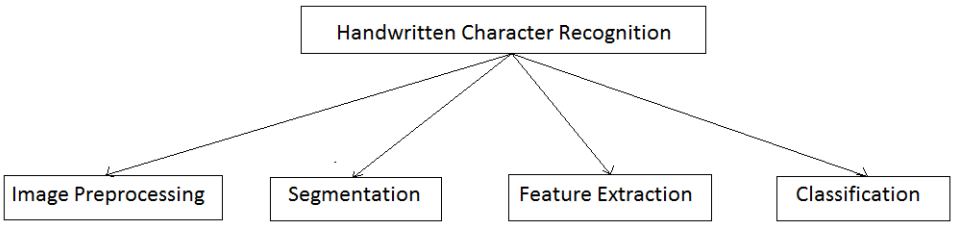

Figure 3.2: Breakdown Model

1.Image Preprocessing

2.Segmentation

3.Feature Extraction

4.Classification

3.2.1 Image Preprocessing

The image is preprocessed using different image processing algorithms like Inverting image , Gray Scale Conversion and image thinning.

3.2.2 Segmentation

After preprocessing of the image segmentation is done. This is done with the help of following steps:

1.Remove the borders

2.Divide the text into rows

3.Divide the rows (lines) into words

4.Divide the word into letters

3.2.3 Feature Extraction

Once the character is segmented we generate the binary glyphs and calculate the summation of each rows and columns values as an features.

3.2.4 Classification

In this phase, we are going to train and test the Neural Network.

3.3 Project Estimation

3.3.1 Estimation of KLOC

The number of lines required for implementation of various modules can be estimated as follows:

1.Graphcal User Interface: 1.3 KLOC

2.Image Processing: 0.3 KLOC

3.Segmentation Module: 0.4 KLOC

4.Feature Extraction: 0.4 KLOC

5.Training: 0.6 KLOC

6.Testing: 0.4 KLOC

7.Texture code: 0.95 KLOC

Thus the total number of lines required is approximately 4.35 KLOC

3.3.2 Efforts

```
E=3.2 * (KLOC)^1.05 (Boehm Simple Model)
E=3.2 * (4.35)^1.05
E=15.87 person  months
```

3.3.3 Development Time for Implementation and Testing

D=E / N
D=15.87 / 5
D=3.17 months

3.3.4 Development Time for Project

Requirement analysis requires 3 months.
Implementation and Testing requires 3.17 months.
D= 3 + 3.17 months
D= 6.17 months

3.3.5 Number of Persons

5 Persons are required to complete the project with given time span successfully.

D1: Hitesh Mohapatra

CHAPTER 4

System Analysis

4.1 Projects scheduling and Tracking

We have selected an appropriate process model we have identified the software engineering tasks that we have to perform, we estimated the amount of work and the number of people, and we know the deadline. Now we have to create a network of software engineering tasks that will enable us to get the job done on time. Once the network is created, we have to assign responsibility for each task, make sure it gets done, and adapt the network as risks become reality. In a nutshell, it is called software project scheduling and tracking.

It is important because in order to build a complete system, many software engineering tasks occur in parallel, and the result of work performed during tasks may have performed effect on work to be conducted in another task. These interdependencies are very difficult to understand without a detailed schedule.

4.1.1 Project Work Breakdown Structure(Analysis)

T1: Communication
Software development process starts with the communication between customer and developer. According to need of project, we gathered the requirements related to project.

T2: System Design
It includes the process model used for the development of the system. The Breakdown Structure (Modules) is also defined in this task. Different Modules used in the system are viewed in Breakdown Structure.

T3: Project Planning
It includes complete estimation and scheduling complete time line chart for project development and tracking. it is also tasks required to define resources, time line, and other project related information.

T4: Modeling (Analysis & Design)
It includes detailed requirements analysis and project design. In requirements analysis, according to the customer requirements the analysis of the system is carried out and what will be the starting of system, in which direction it travels and what will be the destination is given by analysis phase. In design according to analysis, system design takes place.

T5: Risk Management

It includes identifying the risks during project development and according to that managing the risks which are affecting the project development.

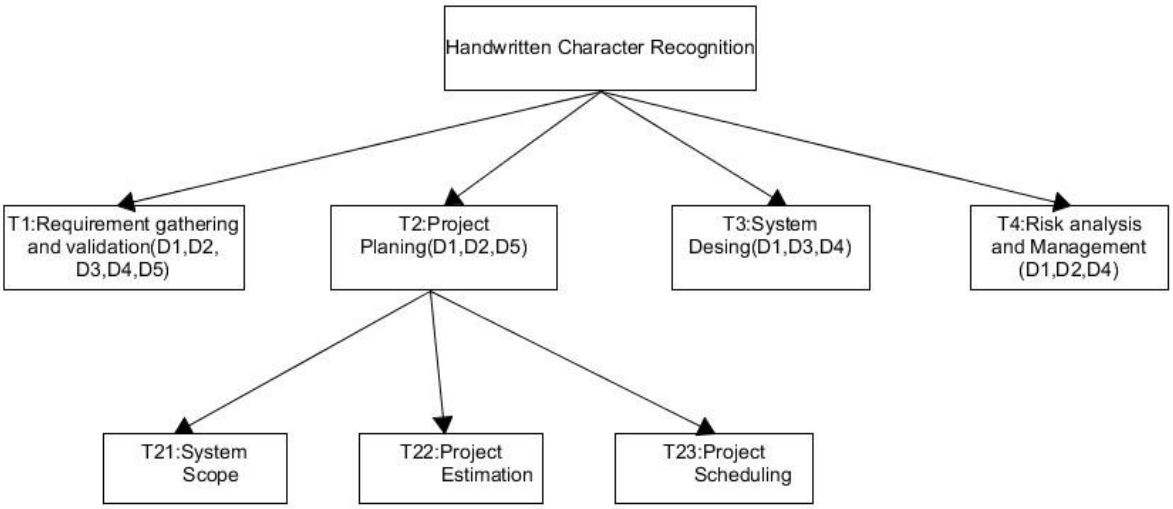

Figure 4.1: Project Work Breakdown structure(Analysis)

4.1.2 Project Work Breakdown Structure(Implementation)

Module 1 :Preprocessing

Gray Scale :

An image is an array, or a matrix, of square pixels(picture elements) arranged in columns and rows. In an (8 bit) gray scale image each picture element has an assigned intensity that ranges from 0 to 255. A gray scale image is what people normally call a black and white image,but the name emphasizes that such an image will also include many shades of gray. A normal gray scale image has 8 bit color depth- 256 gray scales.A " true color images 24 bit color depth 8 * 8 * 8 bits 256 * 256 * 256 colors = 16 million colors. Some gray scale images have more gray scales, fro instance 16 bit =65536 gray scales.There are two general groups of images :vector graphics(or line art) and bitmaps(pixel based or images).

Thinning :

Thinning algorithm is a Morphological operation that is used to remove selected foreground pixels from binary images.It preserves the topology (extent and connectivity) of the original region while throwing away most of the original foreground pixels.

Module 2 :Segmentation

In this part the characters will be identified as letters and the image will be converted to text.After the image is cleaned up and becomes a binary image which contains only the text, the binary image is then saved and the memory is cleaned up.This step is very important to increase the speed of the system.After the following steps should be done.

- Divide the text into rows
- Divide the rows into words
- Divide the word into letters

Module 3:: Feature Extraction

In pattern recognition in image processing, Feature extraction is a special form of dimensionality reduction. When the input data to an algorithm is too large to be processed and it is suspected to be notoriously redundant (much data,but not much information) then the input data will be transformed into reduced representation set of features(also named feature vector). Transforming the inputbdata into the set of features is called features extraction.

Module 4:Recognition Using Neural Network

The back end for recognition is neural network. The number of input to each network is associated with the size of the feature vector for each text.The no of output considered for this data sets are 27(Characters a-z and 1 reject neuron for non character pattern.

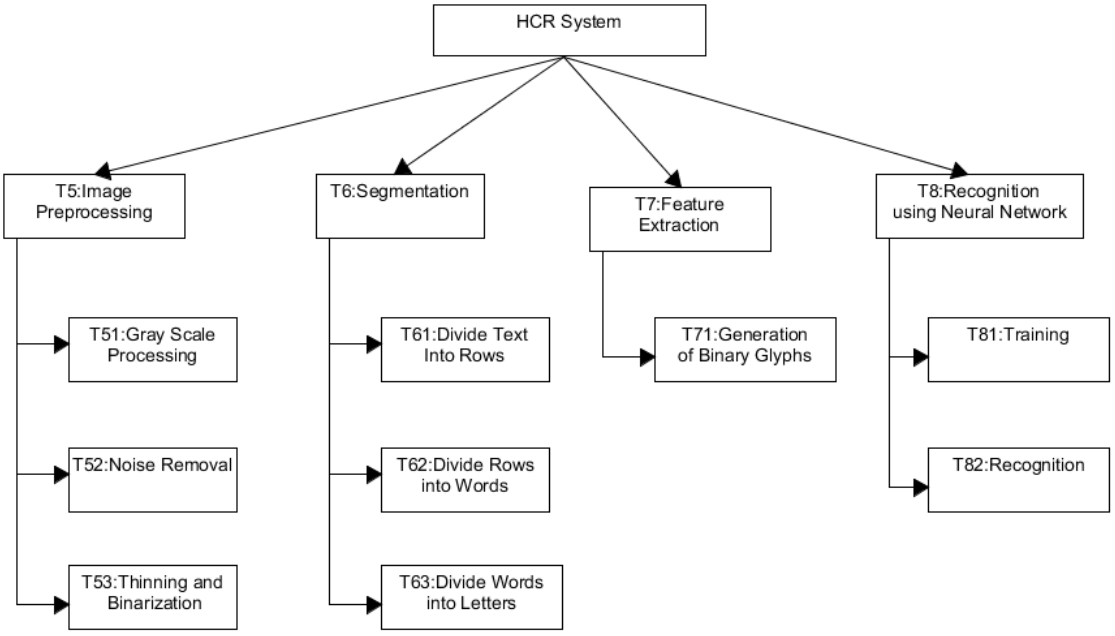

Figure 4.2: Project Work Breakdown structure(Implementation)

4.2 Tasks

In project management, a task is an activity that needs to be accomplished within a defined period of time or by a deadline. A task can be broken down into assignments which should also have a defined start and end date or a deadline for completion. One or more assignments on a specific task normally render the completed. Tasks can be linked together to create dependencies.

T1: Searching of Project Definition
T2: Collection of Required Literature
T3: Requirement Gathering and validation
T4: Determine process model
T5: Determine KLOC breakdown structure
T6: Detail information gathering of each module
T7: Project Scheduling, timeline chart, project Table
T8: UML Modeling
T9: Mathematical Model
T10: Risk Analysis and Risk Management
T11: GUI Module
T12: Implementation of Image Preprocessing Algorithm
T13: Implementation of Segmentation Module
T14: Implementation of Feature Extraction Module
T15: Implementation of Classification Module
T16: Implementation of Neural Network algorithm
T17: Unit Integration
T18: Testing
T19: Make Required Modifications with Documentation
T20: code Review

4.3 Project Schedule

In project management, a schedule consists of a list of projects terminal elements with intended start and finish dates. Terminal elements are the lowest elements in a schedule, which are not further subdivided. Those items are often estimated in terms of resource requirements, budget and duration, linked by dependencies and schedule events. Table describes the schedule for the project development. It also highlights all the tasks to be carried out along with their duration, dependencies, and developers assigned to accomplish those tasks.

Table 4.1: Project Schedule

Tasks	Days	Dependencies	Developers Assigned
T1	16	-	D1
T2	04	T1	D1
T3	19	T2	D1
T4	07	T3	D1
T5	12	T4	D1
T6	10	T2,T3,T5	D1
T7	18	T4,T6	D1
T8	07	T4,T6	D1
T9	05	T6	D1
T10	08	T7,T8	D1
T11	12	T5,T6	D1
T12	15	T5,T6,T7	D1
T13	17	T5,T6,T7	D1
T14	15	T5,T6,T7	D1
T15	11	T5,T6,T7	D1
T16	13	T5,T6,T7	D1
T17	06	T5,T6,T7	D1
T18	06	T5,T6,T7	D1
T19	10	T12-T18	D1
T20	13	T11-T19	D1

Table 4.2: Project Table

Task	Starting Time	Ending Time	Developers	Remark
T1	20/6/15	5/7/15	D1	
T2	6/7/15	9/7/15	D1	
T3	10/7/15	28/7/15	D1	
T4	28/7/15	3/8/15	D1	
T5	4/8/15	15/8/15	D1	
T6	16/8/15	25/8/15	D1	
T7	26/8/15	2/9/15	D1	
T8	3/9/15	9/9/15	D1	
T9	10/9/15	14/9/15	D1	
T10	15/9/15	22/9/15	D1	
T11	14/12/15	25/12/15	D1	
T12	26/12/15	9/1/15	D1	
T13	10/1/15	16/1/15	D1	
T14	17/1/15	31/1/15	D1	
T15	1/2/15	11/2/15	D1	
T16	12/2/15	24/2/15	D1	
T17	25/2/15	1/3/15	D1	
T18	2/3/15	7/3/15	D1	
T19	8/3/15	17/3/15	D1	
T20	18/3/15	31/3/15	D1	

4.3.1 Time Line Chart

4.3.1.1 Time line Chart(Analysis Phase) 4.3.1.2 Time line Chart(Implementation Phase)

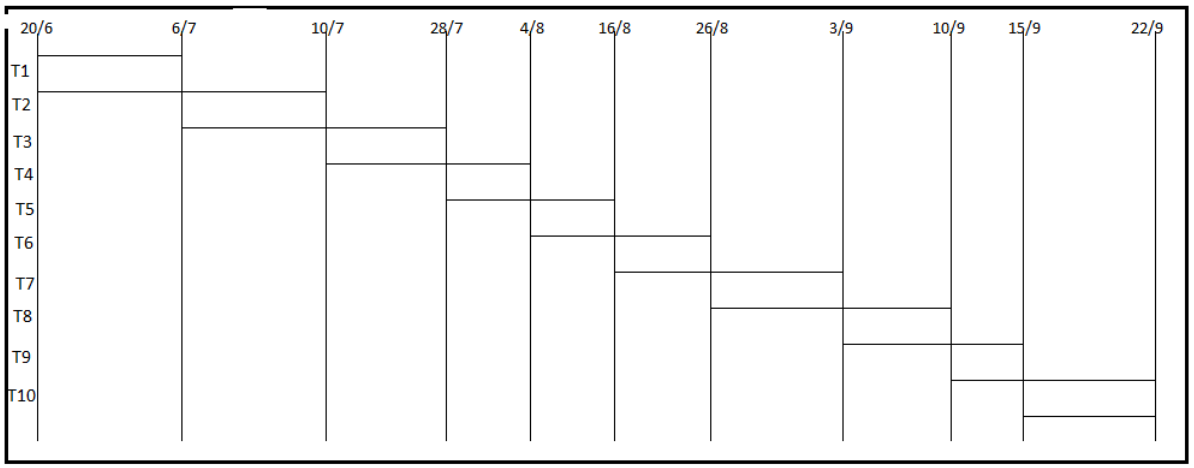

Figure 4.3: Time line Chart(Analysis Phase)

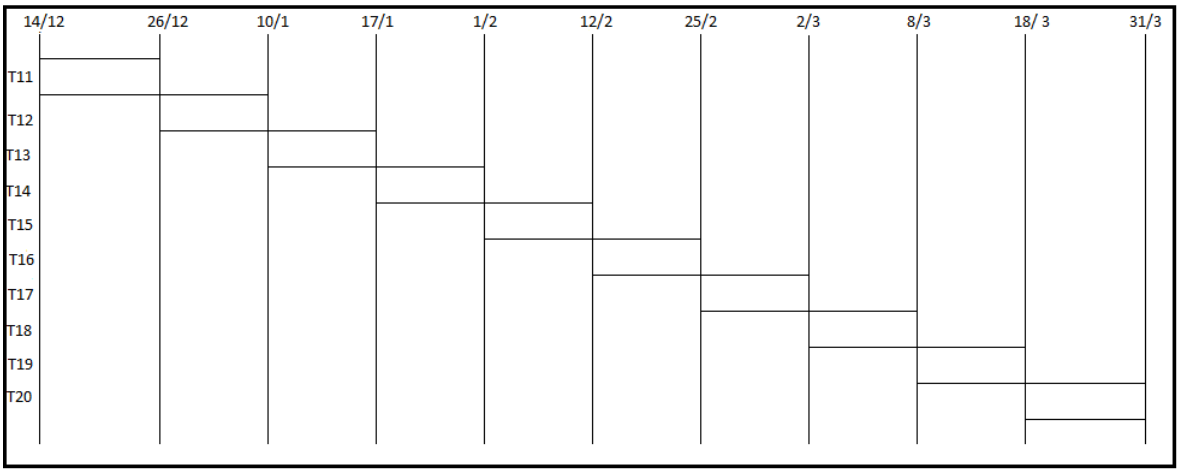

Figure 4.4: Time line Chart(Implementation Phase)

4.4 Analysis Model

4.4.1 Behavioral Model

4.4.1.1 Use case Diagram

The use case view models functionality of the system as perceived by outside uses. A use case is a coherent unit of functionality expressed as a transaction among actors and the system.

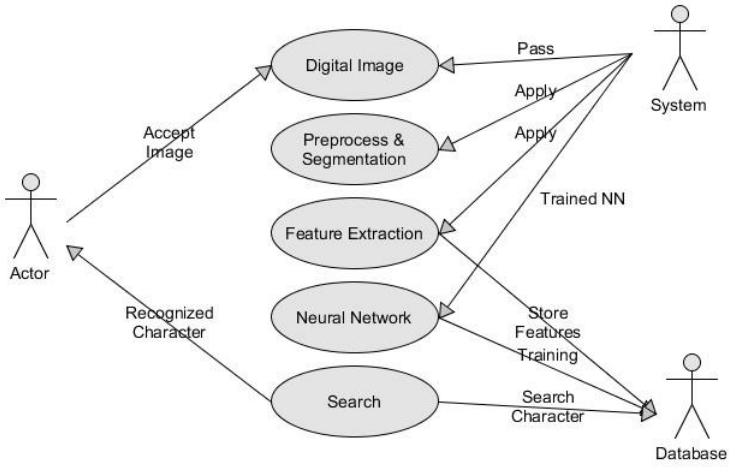

Figure 4.5: Use Case Diagram

4.4.1.2 Sequence Diagram

A sequence diagram is a graphical view of a scenario that shows object interaction in a time-based sequence what happens first, what happens next. Sequence diagram establish the role of objects and helps provide essential information to determine class responsibilities and interfaces. This type of diagram is best used during early analysis phase in design because they are simple and easy to comprehend. Sequence diagram are normally associated with use cases.

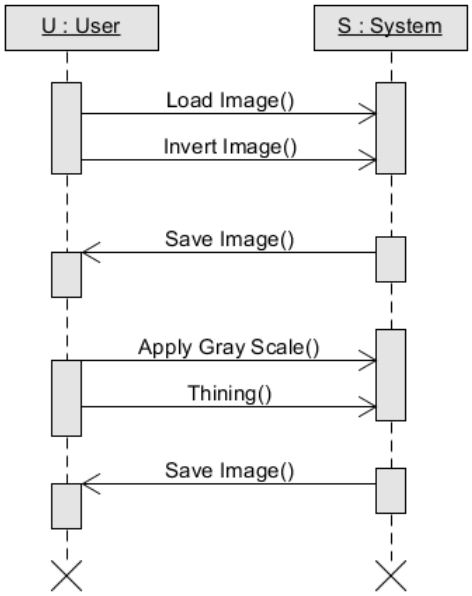

Figure 4.6: Sequence Diagram of Preprocessing

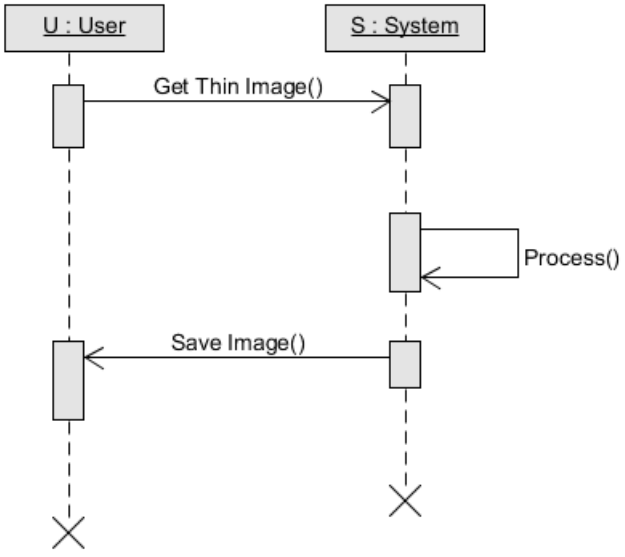

Figure 4.7: Sequence Diagram of Segmentation

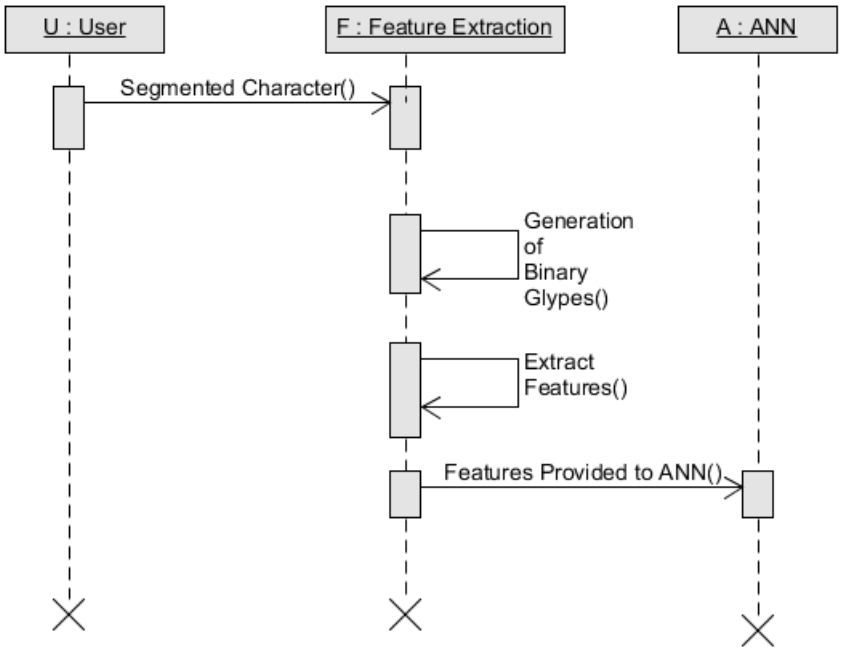

Figure 4.8: Sequence Diagram of Feature Extraction

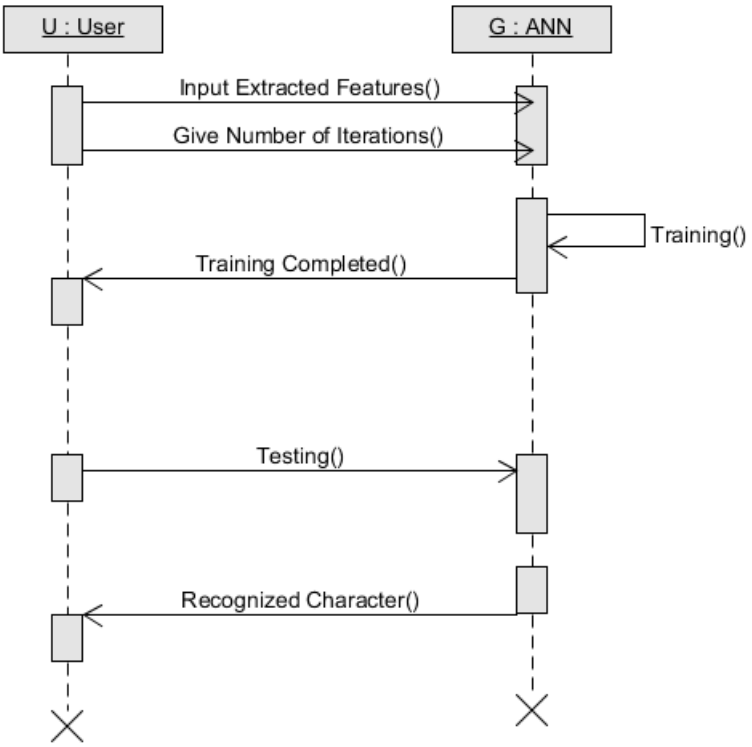

Figure 4.9: Sequence Diagram of Classification

4.4.2 Functional Modelling

4.4.2.1 Data Flow Diagram

Data flow diagram (DFD) is also called as Bubble Chart is a graphical technique, which is used to represent information flow, and transformers those are applied when data moves from input to output. DFD represents system requirements clearly and identify transformers those becomes programs in design. DFD may further partitioned into different levels to show detailed information flow e.g. level 0, level 1 etc.

Figure 4.10: DFD(level 0) for HCR System

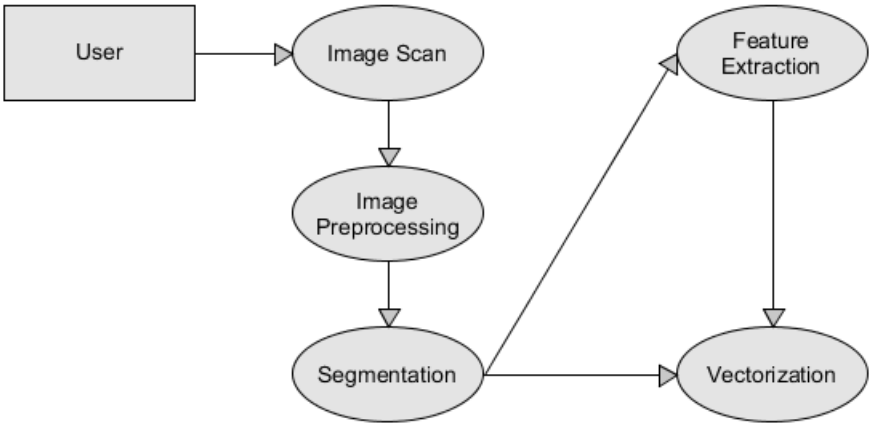

Figure 4.11: DFD(level 1) for Image Reader

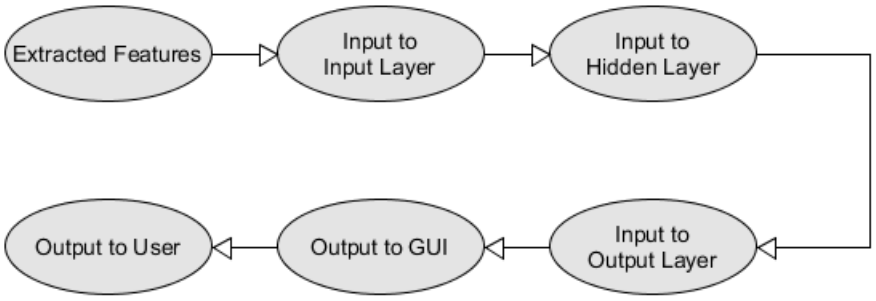

Figure 4.12: DFD(level 1) for Recognition Unit

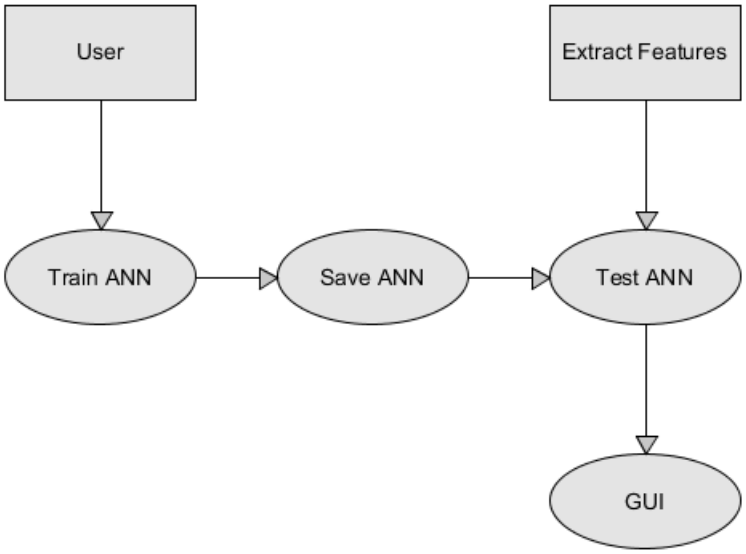

Figure 4.13: Data Flow Diagram for ANN)

4.4.2.2 Control Flow Diagram

The large class of applications having following characteristics requires control flow modelling:

- The applications that are driven by events rather than data.
- The applications that produce control flow information rather than reports or displays.
- The application that process information in specific time.

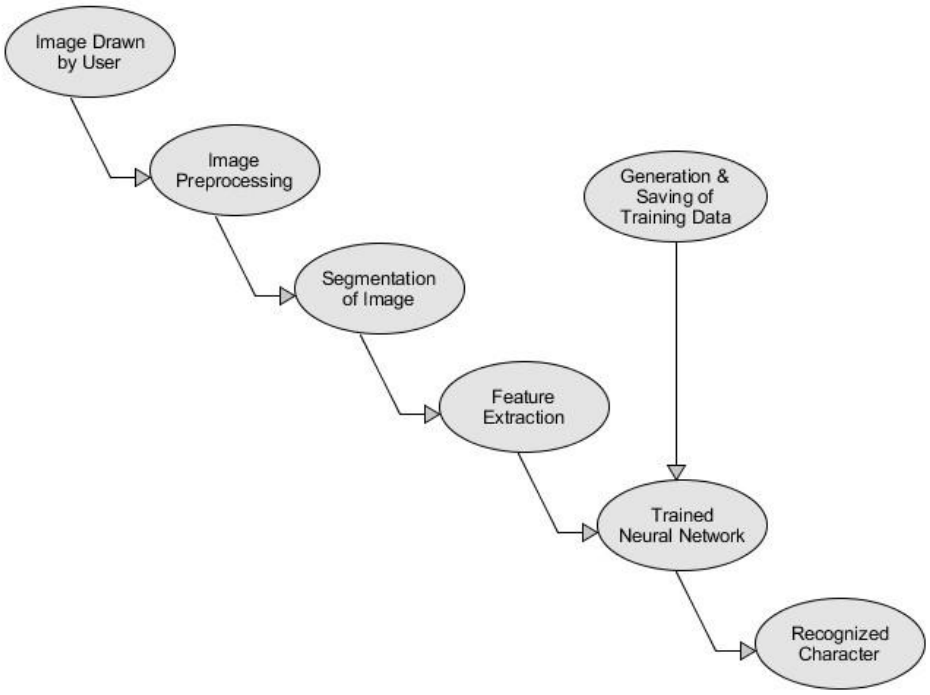

Figure 4.14: Control Flow Diagram for HCR System

4.4.3 Architectural Diagram

4.4.3.1 Deployment Diagram

A deployment diagram shows the allocation of processes to processors in the physical design of a system. A deployment diagram may represent all or part of the process architecture of a system.

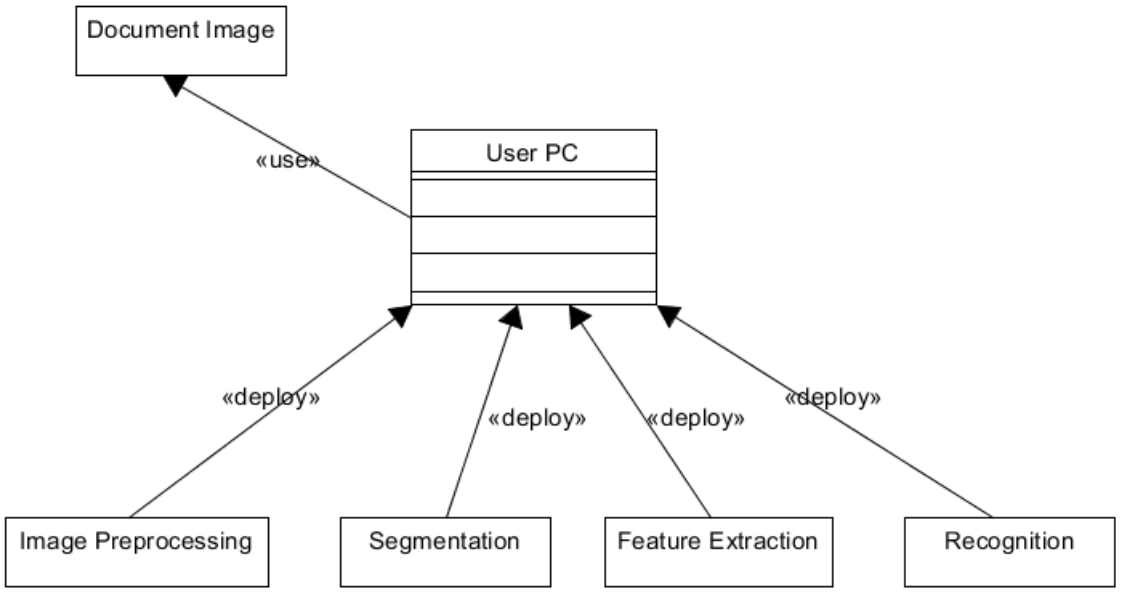

Figure 4.15: Deployment Diagram

4.5 Mathematical Model

when solving problems we have to decide the difficulty of our problem. There are two types of classes provided for that.These are as follows.

- class P
- class NP
- **Class P**

  P is the class of decisions that are polynomially bounded. P is defined only for decision problems. It may seem rather extravagant to use the existence of a polynomial time bound as the criterion for defining the class of more or less reasonable problems polynomials can be quite large. There are, however a number of good reasons for this choice.

  First, while it is not true that every problem in P has an acceptably efficient algorithm, we can certainly say that if a problem is not in P, it will be extremely expensive and probably impossible to solve in practice.

  A second reason for using a polynomial bound to define p is that polynomials have nice closure properties. An algorithm for a complex problem may be obtained by combining several algorithms for simpler problems. Some of the simpler algorithms may work on the output or intermediate result of others.

  A third reason for a using polynomial bound is that it makes P independent of the particular formal model of computation used. A number of formal models are used to prove rigorous theorems about the complexity of algorithms and problems.

- **Class NP**

  NP is the class of decision problems for which a given proposed solution for a given input can be checked quickly (in polynomial time) to see if it really is a solution. More formally, inputs for a system and proposed solution must be described by strings of symbols from some finite set.

  There may be decision problems where there is no natural interpretation for solutions and proposed solutions. A decision problem is abstractly just some function from a set of input string to the set yes, no. A formal definition of NP considers all decision problems.

>NP-Hard Problems

NP-hard (Non-Deterministic polynomial time hard), in computational complexity theory, is a class of problems that are, informally at least as hard as the hardest problem in NP.

A problem H is NP-hard if and only if there is an NP-complete L that is polynomial time Turing reducible to H.

NP-hard problems may be of any type: Decision problems, search problems or optimization problems.

>NP-Complete Problems

NP-Complete is the term used to describe decision problems that are the hardest ones in NP in the sense that, if there were a polynomially bounded algorithm for an NP-complete

problem, then there would be a polynomially bounded algorithm for each problem in N.

Conclusion-: As we have seen all the classes of problem. Our Topic is ”HCR Using Neural Network.” is of P class problem.

4.5.1 Set Theory

Set theory is the branch of mathematical logic that studies sets, which are collections of objects. Although any type of object can be collected into a set, set theory is applied most often to objects that are relevant to mathematics. A set is a collection of objects which are called the members or elements of that set. If we have a set we say that some objects belong (or do not belong) to this set, are (or are not) in the set. We say also that sets consist of their elements.

$S = \{I, P, R, O\}$

Where,

I = Set of inputs.

P = Set of Process.

R = Set of rules.

O = Set of Outputs.

$I = \{I1\}$

Where,

I1 = Input image.

$P = \{P1, P2, P3, P4\}$

Where,

P1 = Accept image.

P2 = Preprocess on image.

P3 = Feature Extraction.

P4 = Classify and Display Recognize character.

$O = \{O1\}$

Where,

O1 = Recognize character.

4.5.2 Venn Diagram

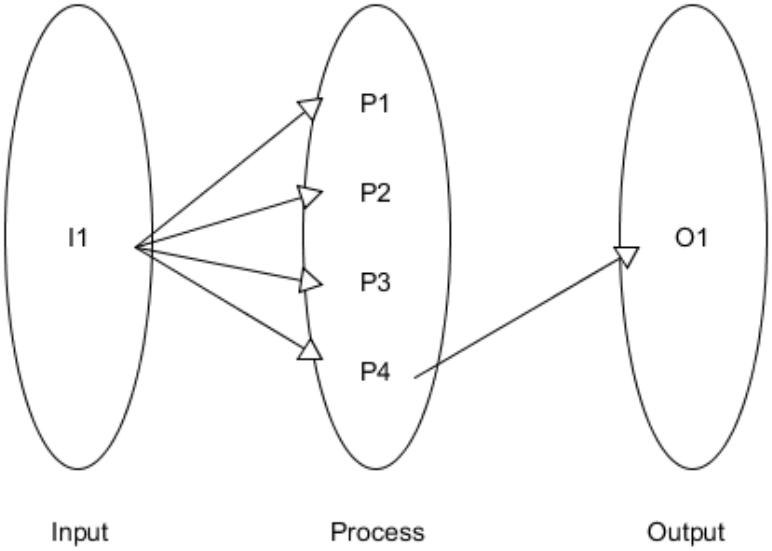

Figure 4.16: Venn Diagram

4.5.3 State Diagram

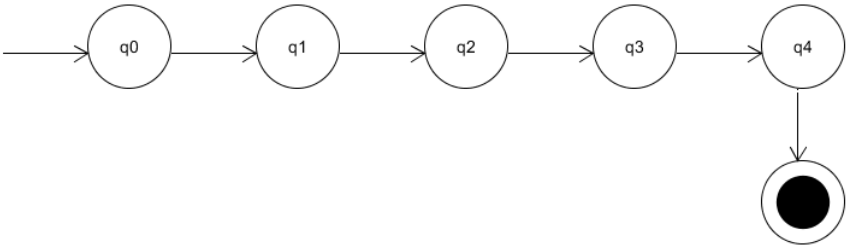

Figure 4.17: State Diagram

q0=Image Acquisition

q1=Preprocessing

q2=Segmentation

q3=Feature Extraction

q4=Classification and Recognition

q5=Close

4.5.4 Time and Space Computation

The space complexity can be defined as amount of memory required by an algorithm to run. To compute space complexity we used two factors:

- Constant Characteristic
- Instant Characteristic

Space requirement S(p) can be given as: S(p)= C + Sp
where,
C is constant i.e. fixed part that denotes space of inputs, outputs, instructions used. Sp is space dependent on instance characteristic i.e. variable part and it include space for recursion stack.
The time complexity of an algorithm is the amount of processor time required by an algorithm to run to completion. It is difficult to compute time complexity in terms of physically clocked time. For instance in multi tier system, executing time depends on many factors such as:

- System Load
- Number of other programs running
- Instructions used and speed of underlying hardware

IR2 tree algorithm time complexity:
The processing time T for the searching nearest hotels/lodges is proportional to the number of hospital register to the particular location T = O(NC), where, N is the number of hotels/lodges and C is the number of users, C can be represented by S as C = O(S). Thus, the total processing time T could be estimated as:
T = O(NC) = O(S2)

CHAPTER 5

Risk Management

5.1 Risk Identification

5.1.1 Product Size Related

R1 Extra line of codes or redundant algorithm may cause wastage of memory.

5.1.2 Customer Related

R2 As the customer is not the technical person, so it creates a problem while understanding the extra requirements of customer.

R3 If customer provides irrelevant information then it may generate some unknown risk.

5.1.3 Process Risk

R4 During segmentation, processing on blur or noisy image may occur.

5.1.4 Technical Risk

R5 Without extracting feature of character complexity of ANN will be increased.

5.1.5 Development Environment Related

R6 If customer asks for change or gives some unexpected modification in later stages of development then it is difficult to alter the entire system design in accordance with that change.

R7 Lack on training on tools and inexperience may cause difficulty in completing project modules.

5.2 Risk Projection

5.2.1 Risk Table

The table 1 shown lists of all possible risks that may occure at any stage during development of the project.

Risk	Category	Probability	Impact
R1	Product Size	More	High
R2	Customer	More	High
R3	Customer	Less	Low
R4	Execution	Less	High
R5	Technical	Less	Low
R6	Development	Less	Low
R7	Development	Less	High

Table 5.1: Risk Table

5.2.2 Strategies Used To Manage Risks

S1 We can Avoid Risk R1 by using minimizing our redundant code.

S2 Regular meeting with customer reduce the risk to some extent.

S3 To reduce above stated risk R3, Properly design the system with a flexibility to adopt changes in later stages and also maintain all necessary documentation for the same.

S4 Before processing for segmentation use suitable noise removal algorithm.

S5 Extract the Feature of character to reduce complexity of ANN.

S6 As the number of persons are sufficient, and ANN part of project is complicated and enlarged so, we have selected incremental process model so that the project can be completed and handed over to customer in same way.

S7 We will try to improve the quality of the software as customer demand changes time to time.

S8 We can avoid R7 by giving well training on tools.

5.2.3 Risk Table Along With RMMM Plan

Table 2 clearly shows the impact of the risks and RMMM plan to deal with any such risks.

Risk	Category	Probability	Impact	RMMM plan
R1	Product Size	More	High	S1
R2	Customer	More	High	S2
R3	Customer	Less	Low	S3
R4	Execution	Less	High	S4
R5	Technical	Less	Low	S5
R6	Development	Less	Low	S6,S7
R7	Development	Less	High	S8

Table 5.2: Risk Table Along With RMMM Plan

5.3 Feasibility

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.Dimensions of Software Feasibility are as follows:

- Technology:
  Is project technically feasible?
  Is it within state of art?
  Can defect be reduce to a level matching application's need?
- Finance:
  Is it financially feasible?
  Can development be completed at a cost the software organization and its client or market can afford?
- Time:
  Will project's time to market beat competition?
- Resources:
  Does the organization have resources needed to success?

Two key considerations involved in the feasibility analysis are:
1. Technical Feasibility.
2. Cost Feasibility.

5.3.1 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement,as only minimal or null changes are required for implementing this system.

Technical feasibility assessment can be done through following ways:

1.NP-Complete

2.NP-Hard

3.Satisfiability

1.NP-Complete:
P Class:Class of all deterministic polynomial language problems.
NP Class:Class of all non-deterministic polynomial language problems.
NP Complete Problems are always solves within given time and space.

2.NP-Hard:

These are problems for which there are no efficient solutions are found. Generally complexity of these problems are more than P,NP,NP-Complete. These may include higher multiplicative constants, exponents terms or high order polynomial.

3.SAT (Satisfiability):

Boolean formula is satisfiable if there exists at least one way of assigning value to its variable so as to make it true and we denote it by using SAT. The problem of deciding whether given formula is satisfiable or not.

5.3.2 Cost Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

CHAPTER 6

Technical Specification

6.1 Technology Details Used in Project

6.1.1 JAVA Development Kit

STEP 1: Download JDK.

1. Goto Java SE download site that is www.oracle.com visit technetwork than java click on javase then downloads to visit index.html page.
2. Click the "Download" button under "JDK" of "Java SE 7".
3. Choose your operating platform, e.g., Windows x86 (for 32-bit Windows OS - "jdk-7u2-windows-i586.exe" 84MB); or Windows x64 (for 64-bit Windows OS).

STEP 2: Install JDK/JRE

1. Run the downloaded installer, which installs both the JDK (Java Development Kit) and JRE (Java Runtime). By default, the JDK and JRE will be installed into C drive:Program Files,java and into jdk1.7.0" and C drive:Program Files,java and into jre7", respectively.Refer JDK installed directory as JAVA HOME,by accepting terms and conditions and by clicking to next we can use jdk for our java programs.

Setting PATH and CLASS PATH variables in order to compile or execute java (Environmental variable):

1. Click My computer ,properties and then advanced settings.
2. Now click Environment Variables, here variables are divided into two sections.
3. User variables : whenever it is modified the corresponding language only effected.
4. System variables : whenever it is modified not only a single language, corresponding all languages are only effected.
5. Set variable name = classpath and variable value as directory where java is installed.One more environment variable ,we have to set for Java home directory similar to path and

class path.For verification ,whether java is installed successfully or not ,just goto command prompt,type javac if you see java compiling commands then you have installed it successfully.

Figure 6.1: JDK

6.1.2 Eclipse

STEP 3:install eclipse.
1.Download latest version of eclipse .
2.install eclipse on our system.
3.Eclipse stores your projects in a folder called a workspace.Choose a workspace folder to use for this session.

Figure 6.2: Eclipse

6.2 References to Technology

www.eclipse.org/downloads.
www.install/windows/jdk_installation-windows.html

CHAPTER 7

Software Implementation

7.1 Introduction

GUI Specification For the creation of GUI we have used the java swing toolkit.
Swing
Swing library is an official Java GUI toolkit released by Sun Microsystems. The main characteristics of the Swing toolkit

1.platform independent

2.customizable

3.extensible

4.configurable

5.lightweight

7.2 Important Modules And Algorithm Used

Following is a description of the various implementation steps , which were applied in order to achieve the final target our project.

7.2.1 Module 1: Image Processing

Preprocessing includes steps that are required to shape the input image into a form suitable for segmentation. Color image is converted into gray scale.Image transform into binary image that means in the form of black in white image.

Gray Scale Conversion

As each color pixel is described by a triplet (R,G,B) of intensities for red, green and blue color. we can map that to a single number giving a gray scale value. There are many approaches to convert color image into gray scale.Here average method is used for color to gray scale conversion.

Algorithm : Gray Scale Conversion
Input : Scanned Handwritten Document Image

Output : Gray Scaled Document Image
Step 1: Start
Step 2: Select Input Document Image.
Step 3: Repeat for x=0 to Width of Image.
Step 4: Repeat for y=0 to Height of Image.
Step 5: Extract RGB value for each of pixel as RGB(i,j)

```
int col = inPixels[x][y];
int r = col & 0xff;
int g = (col >> 8) & 0xff;
int b = (col >>16) & 0xff;
int gs = (r + g + b)/3;
```

Step 6: Set Pixel with computed gray level as :

```
inPixels[x][y] = (gs | (gs << 8) | (gs << 16));
gimage.setRGB(x, y, inPixels[x][y]);
```

Step 7: Display Gray Scale image.
Step 8: Stop

Binarization

Image Binarization converts an image pf upto 256 gray level to a black and white image. The simplest way to use image binarization is to choose a threshold value and clasify all pixels with values above the threshold as black and all other pixels are white.

Algorithm : Image Binarization(Threshoulding)
Input : Gray Scaled Image
Output : Black and White Image
Step 1: Start
Step 2: Select Gray Scaled Document Image.
Step 3: Repeat for x=0 to Width of Image.
Step 4: Repeat for y=0 to Height of Image.
Step 5: Set the Threshold.
Step 6: Extract RGB value for each of pixel as RGB(i,j)

```
int col = inPixels[x][y];
int r = col & 0xff;
int g = (col >> 8) & 0xff;
int b = (col >>16) & 0xff;
int gs = (r + g + b) / 3;
```

Step 7: If pixel(gs) is Above Threshold Then

```
{
  r=g=b=0;
}
else
{
  r=g=b=255;
}
```

Step 8: Set Pixel with computed Threshold level as :

```
inPixels[x][y] = (b | (g << 8) | (r << 16));
```

bimage.setRGB(x, y, inPixels[x][y]);
Step 9: Display Binarized Image.
Step 10: Stop

7.2.2 Module 2: Segmentation

Once image preprocessing is done it is necessary to segment document into lines, lines into words and words into characters.When characters has been extracted from document we can extract features from it for recognition.Segmentation of image is performed to separate the characters from the image.Characters separation from the input image involves three steps as:

- Line Segmentation
- Word Segmentation
- Character Segmentation

Line Segmentation

To perform line segmentation, we need to scan each horizontal pixel row starting from the top of document. The lines are separated where we finds a row with no black pixels.This row acts as a separation between two lines.

Algorithm : Line Segmentation
Input : Binarized Document Image
Output : Segmented lines from Document Image
Step 1: Start
Step 2: Select Document Image.
Step 3: Repeat for x = 0 to Height of Image.
Step 4: Repeat for y = 0 to Width of Image.
Step 5: Scans Each pixels from Horizontal pixel row.
Step 6: Extract RGB value for each pixels inPixels[x][y]
Step 7: If pixel with no Black pixel is found then
{
Segment line from document image
}
Step 8: Stop

Word Segmentation

To perform word segmentation, we need to scan each vertical pixel column starting from the left of line. The words are separated where we finds a column with no black pixels for more than predefined columns. This column acts as a separation between two words.

Algorithm : Word Segmentation
Input : Segmented lines from Image and avgpxl = average pixel width for word separation
Output : Segmented words from line

Step 1: Start
Step 2: Select Document Image.
Step 3: Repeat for x = 0 to Height of Segmented line image.
Step 4: Repeat for y = 0 to Width of Segmented line image..
Step 5: Scans Each pixels from Vertical pixel column.
Step 6: Extract RGB value for each pixels inPixels[x][y]
Step 7: If pixel with no Black pixel is found for more than avgpxl then

{
Segment Word from lines
}

Step 8: Stop

Character Segmentation

To perform character segmentation, we need to scan each vertical pixel column starting from the left of word. The characters are separated where we finds a column with no black pixels columns. This column acts as a separation between two character.

Algorithm : Character Segmentation
Input : Segmented words from lines
Output : Segmented characters from words
Step 1: Start
Step 2: Select Document Image.
Step 3: Repeat for x = 0 to Height of Segmented word.
Step 4: Repeat for y = 0 to Width of Segmented word.
Step 5: Scans Each pixels from Vertical pixel column.
Step 6: Extract RGB value for each pixels inPixels[x][y]
Step 7: If pixel with no Black pixel is found then

{
Segment characters from words
}

Step 8: Stop

7.2.3 Module 3: Feature Extraction

As individual characters has been separated, character image can be re sized to 15 x 20 pixels. If the features are extracted accurately then the accuracy of recognition is more.Here we have use the 15 x 20 means 300 pixels as it is for feature vector. This extracted feature are stored in .dat file

7.2.4 Module 4: Training And Recognition

The Features extracted from previous modules are given as an input for Neural Network. The Kohonen algorithm is an automatic classification method which is the origin of Self-Organizing Maps.This SOM is used for training and recognition.

Training

```
 * This method is called to train the network. It can run
for a very long time and will report progress back to the
owner object.
@exception java.lang.RuntimeException

public void learn ()
throws RuntimeException
{
int i, key, tset,iter,n_retry,nwts;
int won[],winners ;
double work[],correc[][],rate,best_err,dptr[];
double bigerr[] = new double[1] ;
double bigcorr[] = new double[1];
KohonenNetwork bestnet;
Preserve best here
totalError = 1.0 ;
for ( tset=0 ; tset¡train.getTrainingSetCount(); tset++ ) {
dptr = train.getInputSet(tset);
if ( vectorLength( dptr ) ¡ 1.E-30 ) {
throw(new RuntimeException("Multiplicative normalization has null training case ")) ;
}
}
bestnet = new KohonenNetwork(inputNeuronCount,outputNeuronCount,owner) ;
won = new int[outputNeuronCount];
correc = new double[outputNeuronCount][inputNeuronCount+1];
if ( learnMethod==0 )
work = new double[inputNeuronCount+1];
else
work = null ;
rate = learnRate;
initialize () ;
best_err = 1.e30 ;
main loop:
n_retry = 0 ;
for ( iter=0 ; ; iter++ )
evaluateErrors ( rate , learnMethod , won ,bigerr , correc , work ) ;
totalError = bigerr[0] ;
if ( totalError ¡ best_err ) {
best_err = totalError ;
copyWeights ( bestnet , this ) ;
}
winners = 0 ;
for ( i=0;i¡won.length;i++ )
if ( won[i]!=0 )
winners++;
if ( bigerr[0] ¡ quitError )
break ;
```

```
if ( (winners ¡ outputNeuronCount) && (winners ¡ train.getTrainingSetCount()) ) {
forceWin ( won ) ;
continue ;
}
adjustWeights ( rate , learnMethod , won , bigcorr, correc ) ;
owner.updateStats(n_retry,totalError,best_err);
if ( halt ) {
owner.updateStats(n_retry,totalError,best_err);
break;
}
Thread.yield();
if ( bigcorr[0] ¡ 1E-5 ) {
if ( ++n_retry ¿ retries )
break ;
initialize () ;
iter = -1 ;
rate = learnRate ;
continue ;
}
if ( rate ¿ 0.01 )
rate = reduction ;
}
copyWeights( this , bestnet ) ;
for ( i=0 ; i¡outputNeuronCount ; i++ )
normalizeWeight ( outputWeights[i] ) ;
halt = true;
n_retry++;
owner.updateStats(n_retry,totalError,best_err);
}

    Called to initialize the Kononen network.

    public void initialize()
{
int i ;
double optr[];
clearWeights() ;
randomizeWeights( outputWeights ) ;
for ( i=0 ; i¡outputNeuronCount ; i++ ) {
optr = outputWeights[i];
normalizeWeight( optr );
}
}
```

Recognition

```
/**
* Present an input pattern and get the
winning neuron.
@param input input pattern
@param normfac the result
@param synth synthetic last input
@return The winning neuron number.
*/
public int winner(double input[] ,double normfac[] ,double synth[])
{
int i, win=0;
double biggest, optr[];
normalizeInput( input , normfac , synth ) ;
biggest = -1.E30;
for ( i=0 ; i < optr = outputWeights[i];
output[i] = dotProduct (input , optr ) * normfac[0]+ synth[0] * optr[inputNeuronCount] ;
output[i] = 0.5 * (output[i] + 1.0) ;
if ( output[i] > biggest ) {
biggest = output[i] ;
win = i ;
}
if ( output[i] > 1.0 )
output[i] = 1.0 ;
if ( output[i] < 0.0 )
output[i] = 0.0 ;
}
return win ;
}
```

CHAPTER 8

Software Testing

8.1 Introduction

Software Testing is the process of testing the functionality and correctness of software.Software testing is an empirical technical investigation conducted to provide stakeholders with information about the quality of the product to the context in which it is intended to operate.This includes,but is not limited to,the process of executing a program or application with the intent of finding errors.

8.1.1 Unit Testing:

In this each module is tested individually.Criteria selected for identifying unit test module is to identify module that has core functionality implementation.Module could be an individual or procedure.
The following is a list of functions for unit testing that will tested:

- Select the scanned input image of handwritten document.
- Apply Preprocessing.
- Apply Segmentation.
- Apply Feature Extraction.
- Extract Digital character.

8.1.2 Integration testing

Integration testing integrates individual modules and tested as a group.Integration testing takes as its input modules that have been unit tested,groups them in larger aggregates,applies tests defined in an integration test plan to those aggregates,and delivers as its output the integrated system for testing.

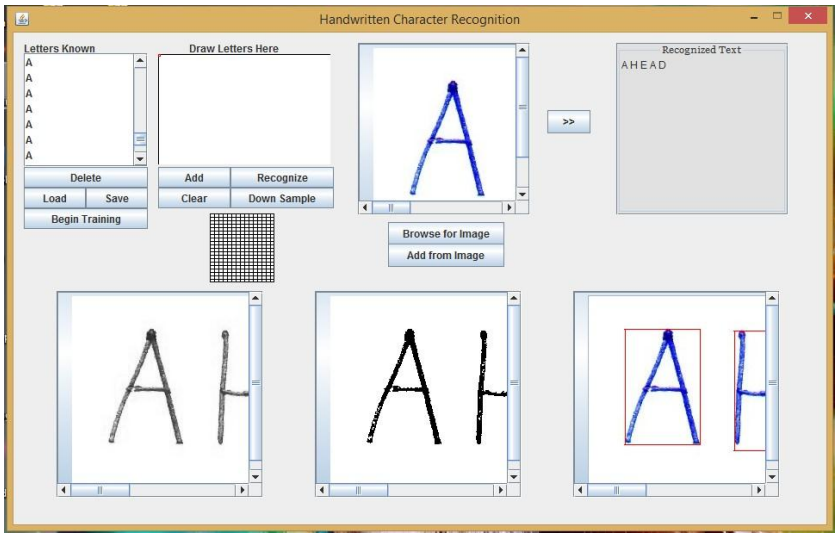

Figure 8.1: Integration testing

8.1.3 Validation testing

The process of evaluating software during or at the end of the development process in to determine whether ot satisfies specified requirements.

8.1.4 GUI Testing

GUI testing is the process of testing a product's graphical user interface to ensure it meet its specification,ensuring the navigation between icons/buttons with source code.

8.2 Test cases

Table 8.1: Test case 1

Use case ID	1
Test Case Name	Check image format
Test case Description	Valid image format MUST be provided to continue
Steps	1.Open application 2.Give image with proper format input
Expected Results	Input image accepted and display
Actual Results	As Expected.

Table 8.2: Test case 2

Usecase ID	2
Test Case Name	Application should allow selecting any type of image.
Test case Description	Application should allow selecting any type of image, which we want.
Steps	2.Click on Browse for image button
Expected Results	After choose any image, image is selected
Actual Results	As Expected.

Table 8.3: Test case 3

Usecase ID	3
Test Case Name	Add character from image
Test case Description	Application should allow to select character for training of character of an image, which we want to add.
Steps	1.Browse image 2.Click on Add from image button and select character and click OK button.
Expected Results	Character should add successfully.
Actual Results	As Expected.

Table 8.4: Test case 4

Use case ID	4
Test Case Name	Start Training
Test case Description	Start training character.
Steps	1.Start training of character which are in .dat file by user
Expected Results	Training should be done and accurate.
Actual Results	As Expected.

Table 8.5: Test case 5

Use case ID	5
Test Case Name	Select image for recognition.
Test case Description	Image should be accepted.
Steps	1.accept any type image.
Expected Results	Application should be able to do accept image for recognition.
Actual Results	As Expected.

Table 8.6: Test case 6

Use case ID	6
Test Case Name	Character recognition from the selected image.
Test case Description	Character from the image should be displayed .
Steps	1.Accept any scanned image. 2.Click on >>(Recognition)button
Expected Results	Application should be able to display Recognized text from image.
Actual Results	As Expected.

8.3 Snap shots of Test Cases and Test Results

AppPerfect Java Code Test is a Java code analysis software designed to perform the following two key tasks: Automate Java code review and Enforce Good Java Coding Practices. AppPerfect Code Test analysis your Java and Java Server Pages (JSP) source code and applies over 750 Java coding rules to apply the collective knowledge of leading experts in the Java programming field to your code.

Locating and fixing Java coding problems during source code development time is arguably the cheapest way to resolve problems. As your project goes past the development phase, in to testing and deployment, the cost of fixing problems grows exponentially. By conducting source code analysis and successfully identifying and correcting all such issues, software developers can eliminate the risk and potential costs early in the software development cycle.

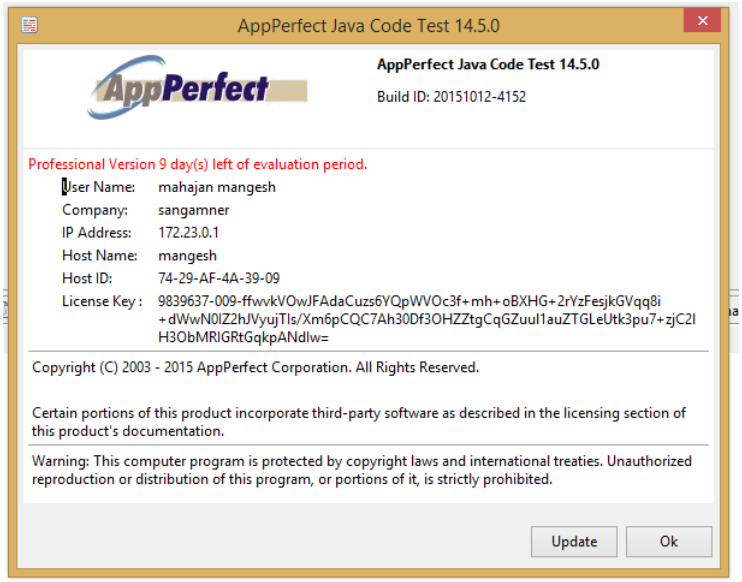

Figure 8.2: Information

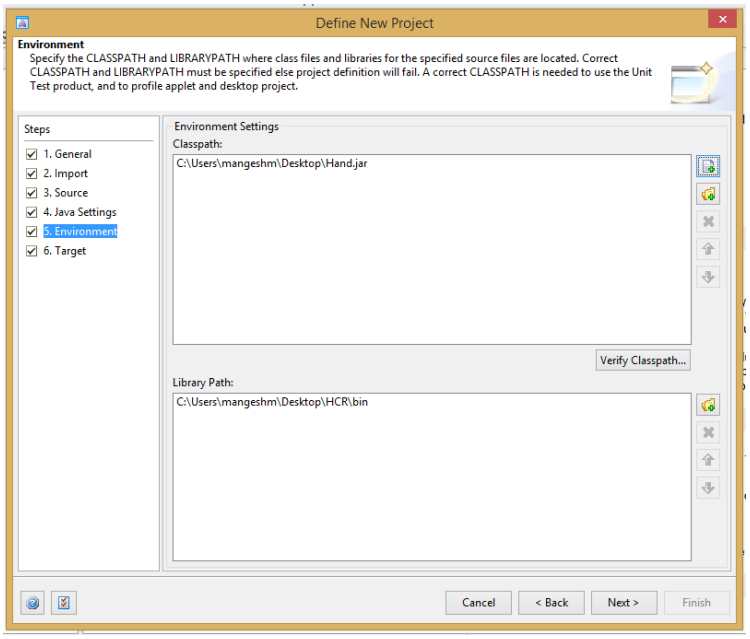

Figure 8.3: Setting Project

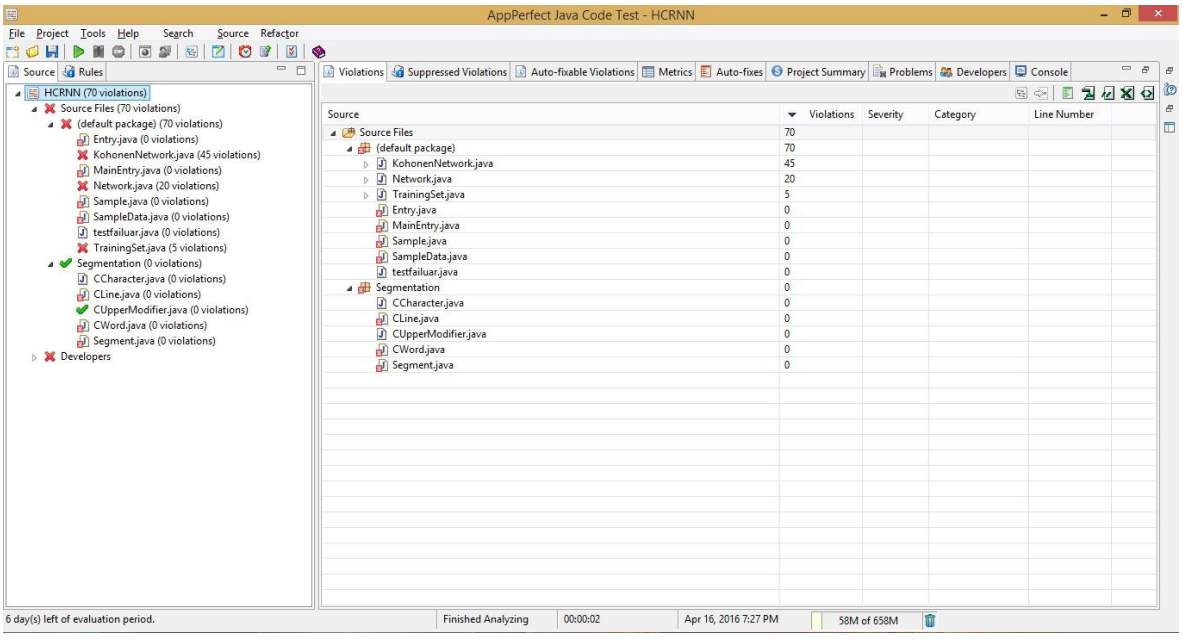

Figure 8.4: Violation

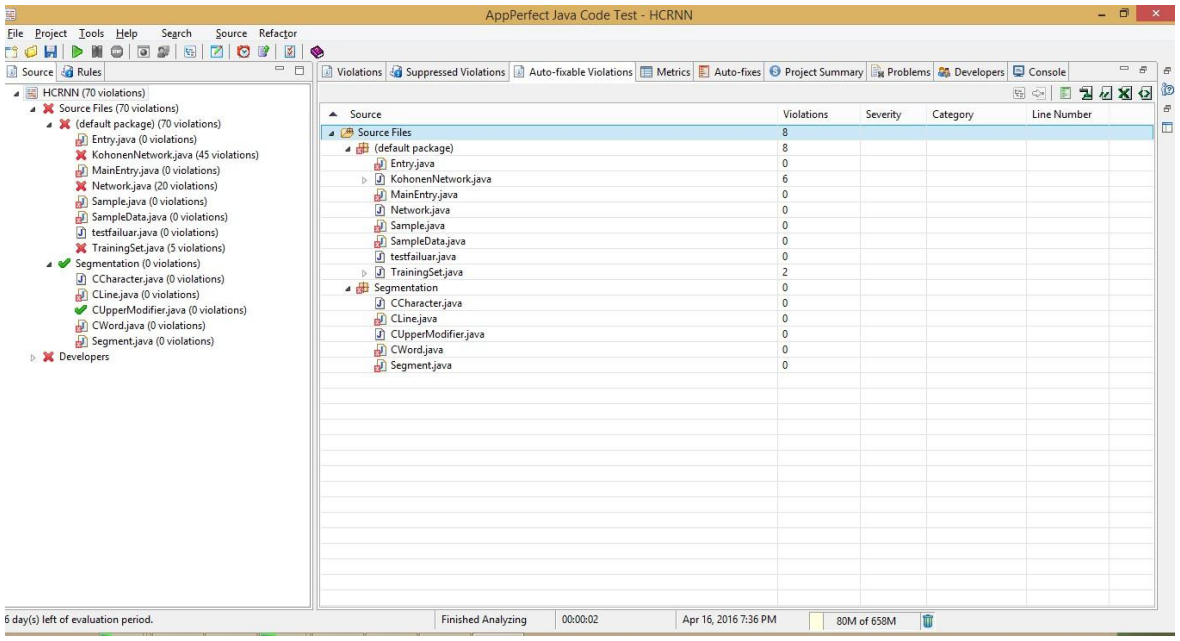

Figure 8.5: Auto-fixable Violations

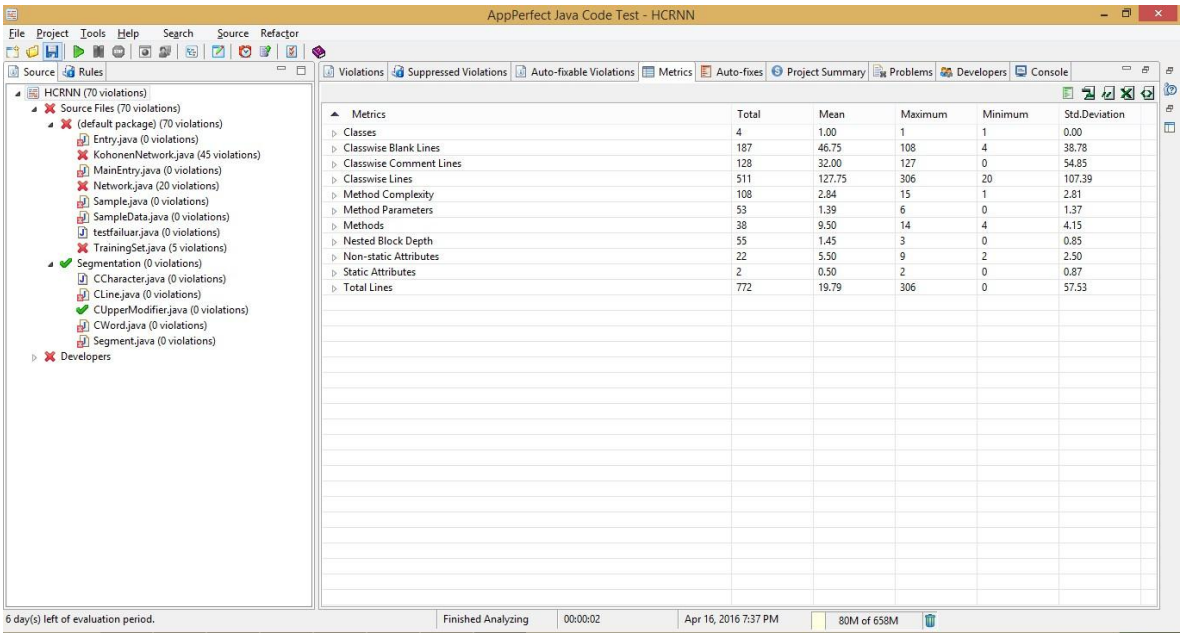

Figure 8.6: Metrics

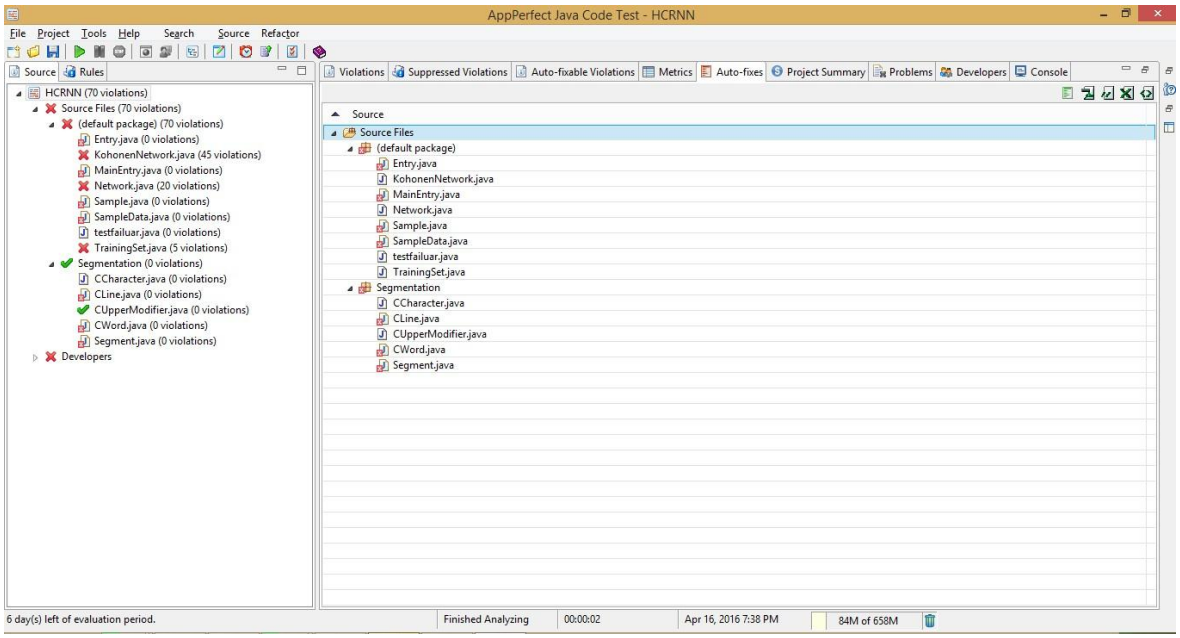

Figure 8.7: Auto-Fixes

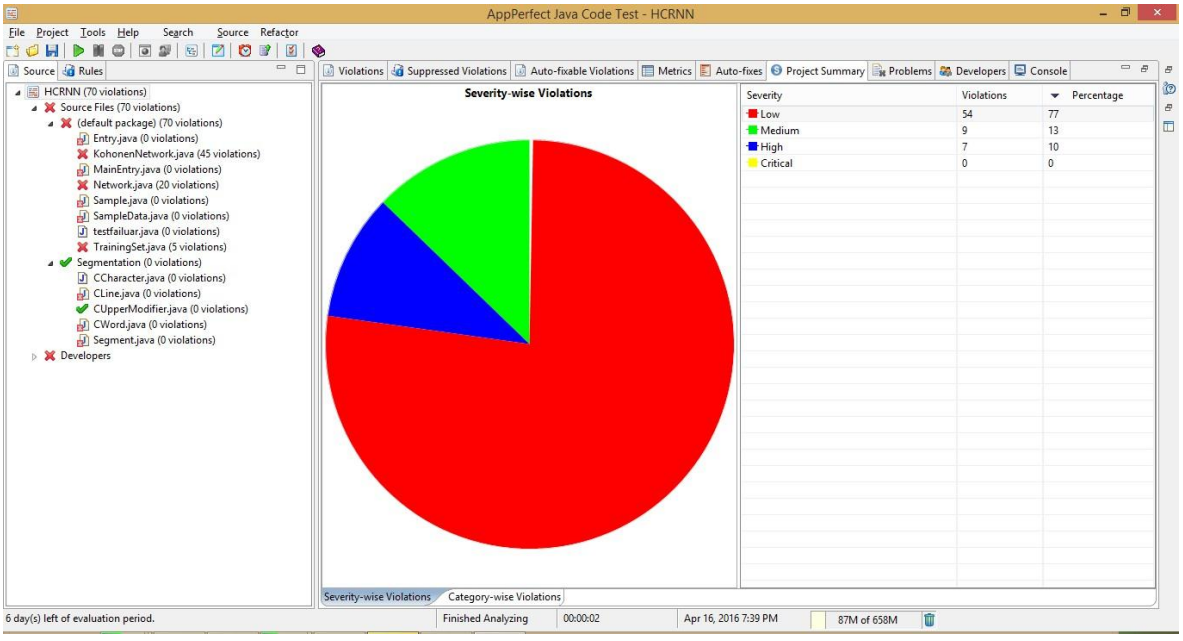

Figure 8.8: Project Summary

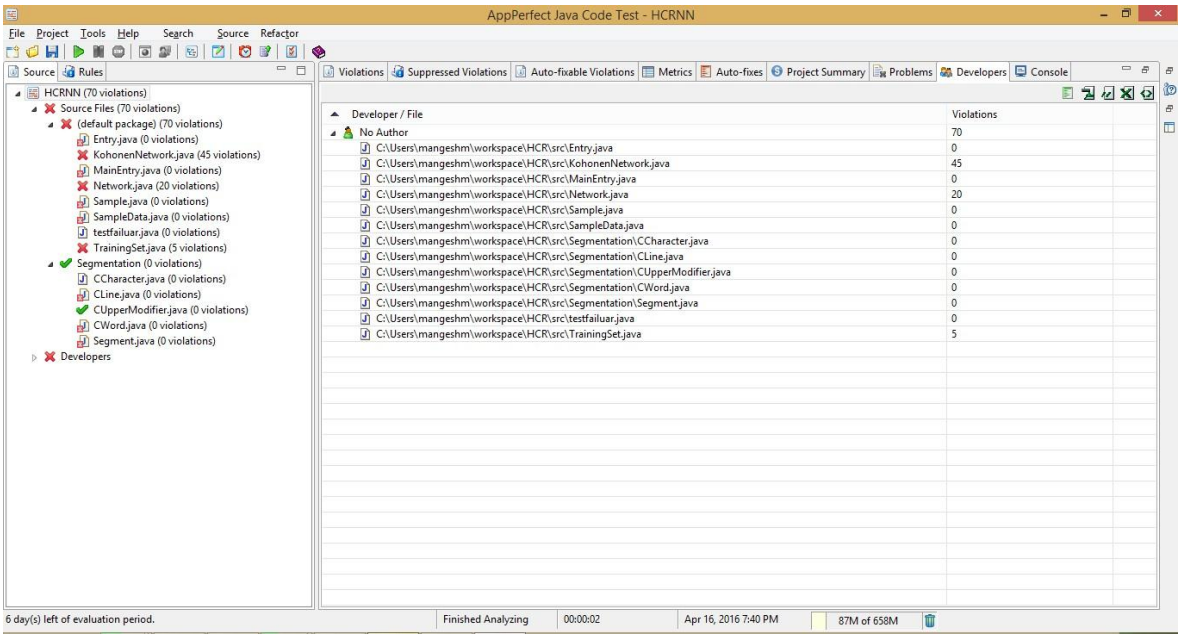

Figure 8.9: Developers

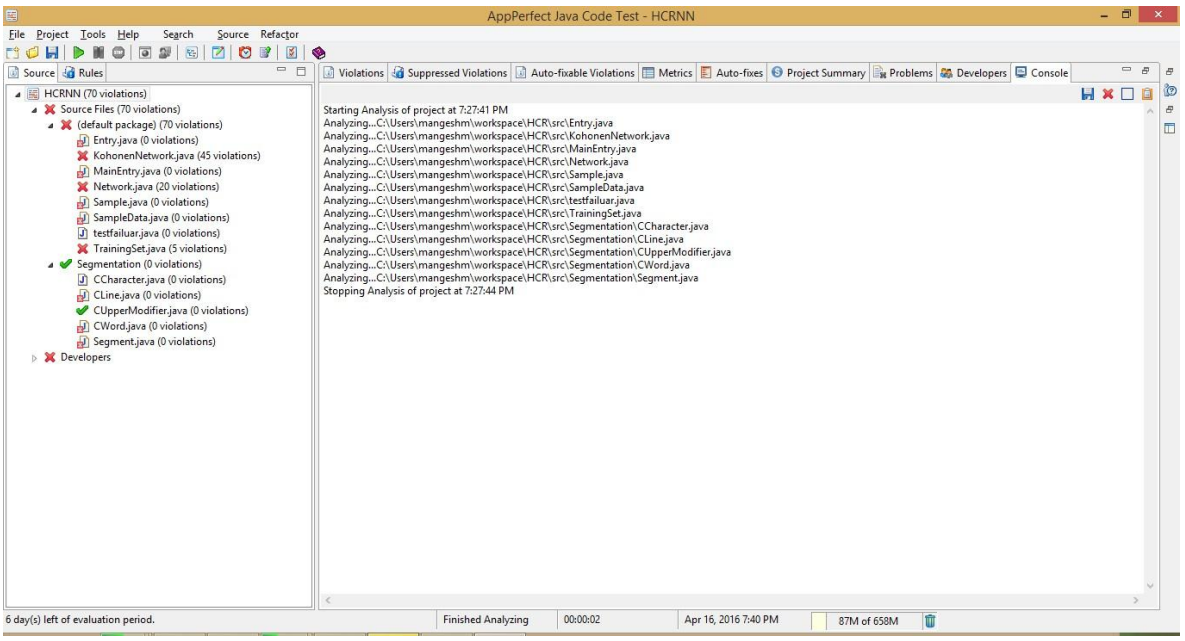

Figure 8.10: Console

CHAPTER 9

Results

"HCR Using Neural Network "is aimed at recognizing the handwritten characters. The "handwritten Character Recognition System "is implemented using a neural network.in this system original image is converted into gray scale image then After gray scaling image is converted in black and white and segmented form. After preprocessing and segmentation operation system show final output.

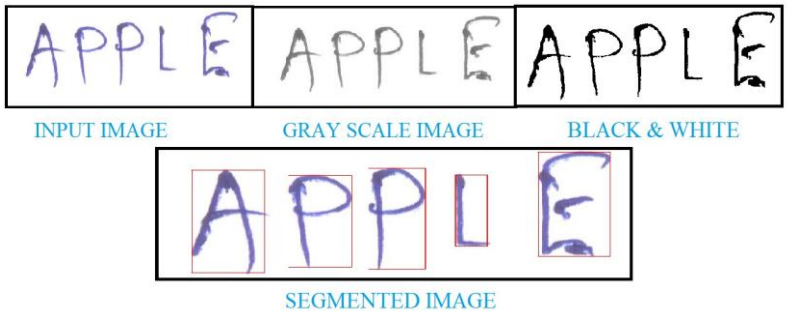

Figure 9.1: Modules Result

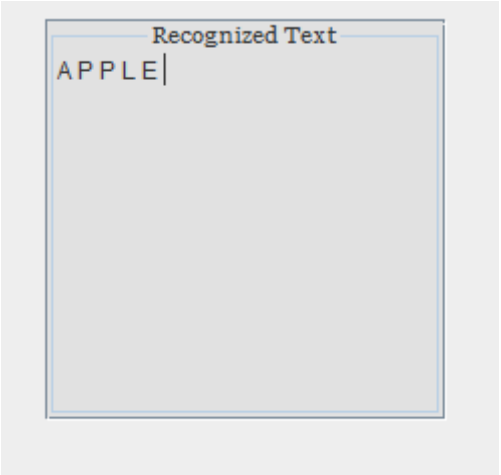

Figure 9.2: Recognized Text

Figure 9.3: Input Image

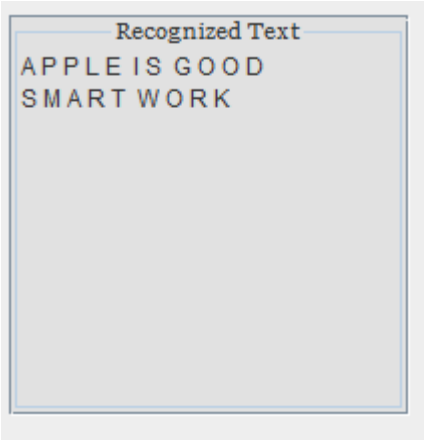

Figure 9.4: Recognized Text

CHAPTER 10

Deployment And Maintenance Details

10.1 Installation And Un-installation

10.1.1 Installation

For installation of application just double click on the executable jar file,the application will smoothly work on the windows 7 or onward versions.

10.1.2 Un-installation

For un-installation of the application user can simply delete the jar file,the application will be delete permanently.

10.2 User Help

HCR is easy process. Just follow these steps to guide you through the HCR process.

1.Run a Project.

2.Load and Train Datasets.

3.Save datasets.

4.Select an Image through browse for image button.

5.After selecting image you can click on Recognize button.

6.When you click on Recognize button they show us grayscale, black and white and segmented image as well as final output also shown in recognize text box.

HCR English 2015-2016. This is free software, and you are welcome to redistribute it under certain conditions.

CHAPTER 11

Conclusion and Future Scope

11.1 Conclusion

Many regional languages throughout world have different writing styles which can be recognized with HCR systems using proper algorithm and strategies. We have learning for recognition of English characters. It has been found that recognition of handwritten character becomes difficult due to presence of odd characters or similarity in shapes for multiple characters. Scanned image is pre-processed to get a cleaned image and the characters are isolated into individual characters.

Preprocessing work is done in which normalization, filtration is performed using processing steps which produce noise free and clean output. Managing our evolution algorithm with proper training, evaluation other step wise process will lead to successful output of system with better efficiency. Use of some statistical features and geometric features through neural network will provided better recognition result of English characters. This work will be helpful to the researchers for the work towards other script.

11.2 Future Scope

This work further extended to the character recognition for other languages. It can be used to convert the fax and news papers into text format.In order to recognize words,sentences or paragraphs we can use multiple ANN for classification.It can be used in post office for reading postal address.

References

[1]Isha Vats, Shamandeep Singh, *"Offline Handwritten English Numerals Recognition using Correlation Method"*,International Journal of Engineering Research and Technology (IJERT): ISSN: 2278-0181 Vol. 3 Issue 6, June 2014. Access Date: 09/07/2015.

[2]Gunjan Singh,Sushma Lehri, *" Recognition of Handwritten Hindi Characters using Back propagation Neural Network"*,International Journal of Computer Science and Information Technologies ISSN 0975-9646, Vol. 3 (4) , 2012,4892-4895. Access Date:09/07/2015.

[3]S S Sayyad, Abhay Jadhav, Manoj Jadhav, Smita Miraje, Pradip Bele, Avinash Pandhare, *'Devnagiri Character Recognition Using Neural Networks"* ,International Journal of Engineering and Innovative Technology (IJEIT)Volume 3, Issue 1, July 2013. Access Date: 09/07/2015.

[4]Shabana Mehfuz,Gauri katiyar, *'Intelligent Systems for Off-Line Handwritten Character Recognition: A Review"* ,International Journal of Emerging Technology and Advanced Engineering Volume 2, Issue 4, April 2012. Access Date: 09/07/2015.

[5]Prof. Swapna Borde, Ms. Ekta Shah, Ms. Priti Rawat, Ms. Vinaya Patil, *"Fuzzy Based Handwritten Character Recognition System"* ,International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622,VNCET 30 Mar'12. Access Date: 09/07/2015.

[6]Rahul KALA, Harsh VAZIRANI, Anupam SHUKLA and Ritu TIWARI, *"An Overview of Character Recognition Focused on Off-Line Handwriting"*,IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS PART APPLICATIONS AND REVIEWS, VOL. 31, NO. 2, MAY 2001. Access Date:09/07/2015.

[7]Ms. Seema A. Dongare , Prof. Dhananjay B. Kshirsagar, Ms. Snehal V. Waghchaure *,"Handwritten Devanagari Character Recognition using Neural Network "* ,IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727 Volume 16, Issue 2, Ver. X (Mar-Apr. 2014), PP 74-79 . Access Date:09/07/2015.

[8]Mitrakshi B. Patil, Vaibhav Narawade, *"Recognition of Handwritten Devnagari Characters through Segmentation and Artificial neural networks"* ,International Journal of Engineering Research and Technology (IJERT) Vol. 1 Issue 6, August - 2012. ISSN: 2278-0181. Access Date:09/07/2015.

[9]Mandeep Kaur, Sanjeev Kumar, *"A RECOGNITION SYSTEM FOR HANDWRITTEN GURMUKHI CHARACTERS"*International Journal of Engineering Research and Technology (IJERT) Vol. 1 Issue 6, August - 2012 ISSN: 2278-0181. Access Date:09/07/2015.

[10]Miroslav NOHAJ, Rudolf JAKA, *"Image preprocessing for optical character recognition using neural networks"*Journal of Patter Recognition Research, 2011. Access Date:09/07/2015.

[11]Nisha Sharma et al, *"Recognition for handwritten English letters: A Review"*International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 7, January 2013. Access Date:09/07/2015.

[12]J.Pradeep et al,, *"Diagonal based feature extraction for handwritten alphabets recognition System using neural network"*International Journal of Computer Science and Information Technology (IJCSIT), Vol 3, No 1, Feb 2011. Access Date:09/07/2015.

Appendix A: Project Quality and Testing Report

Table 11.1: Testing table

Test case ID	Module Name	Description	Expected Output	Actual Output	Remark
TC_00_01	Image Acquisition	Uploading image	Image file should be selected and uploaded successfully	Image file selected and uploaded successfully	Pass
TC_00_02	Image Acquisition	Wrong input file uploaded	error message should be displayed	error message displayed	Pass
TC_01_01	Image Preprocessing	To preprocess image which is uploaded	Gray scale conversion should be done	Gray scale conversion done	Pass
TC_01_02	Image Preprocessing	To preprocess image which is uploaded	Binarization(Conversion from RGB to B/W image)	Binarization done	Pass
TC_02_01	Segmentation	Line Segmentation	Line Segmentation should be done	Line Segmentation should done	Pass
TC_02_02	Segmentation	Word Segmentation	word Segmentation should be done	Word Segmentation should done	Pass
TC_02_03	Segmentation	Character Segmentation	Character Segmentation should be done	Character Segmentation should done	Pass
TC_03_01	Feature Extraction	Feature extraction	Feature should be extract	Feature extracted successfully	Pass
TC_04_01	Training	Training of characters	Character should be train successfully	Character trained successfully	Pass
TC_05_01	Recognition	Recognition of character	Character should be recognized successfully	Character recognized successfully	Pass

Appendix B: Project Planner and Progress Report

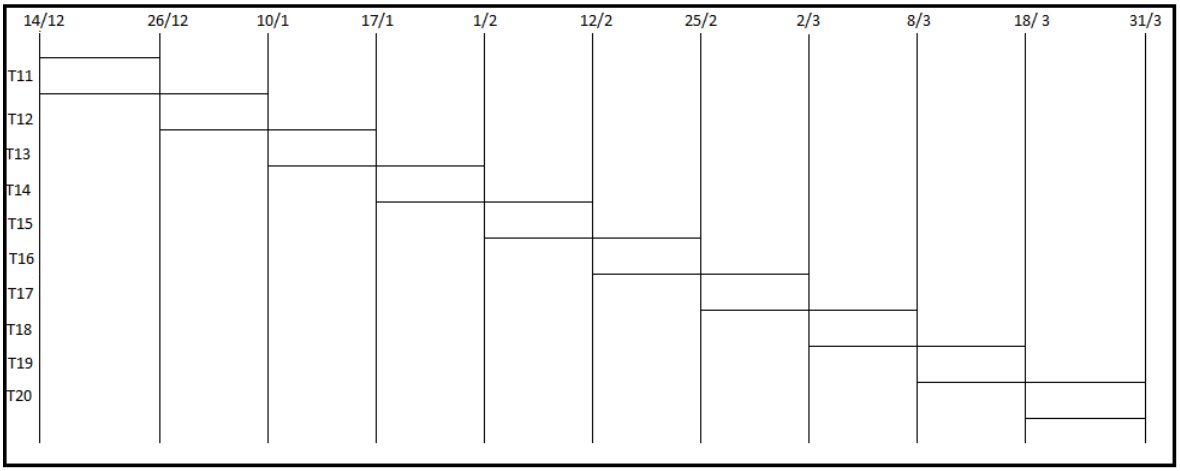

Figure 11.1: Project Plan