HANDWRITTEN CHARACTER RECOGNITION SYSTEM

A

Major Project Report

Submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Engineering

In

Computer Science & Engineering

Submitted to

RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA,

BHOPAL (M.P.)

Guided By

Prof. Deepak Singh Chouhan

Submitted By

Aashi Pandey (0875CS151002)

Ashish Aswani (0875CS151034)

Avani Kala (0875CS151037)

Avani Sharma (0875CS151038)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ACROPOLIS TECHNICAL CAMPUS,

INDORE (M.P.) 452020

2018-2019

Declaration

I hereby declare that the work, which is being presented in the project entitled **Handwritten Character Recognition System** partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering**, submitted in the Department of Computer Science & Engineering at **Acropolis Technical Campus, Indore** is an authentic record of my own work carried under the supervision of **Prof. Deepak Singh Chouhan**. I have not submitted the matter embodied in this report for the award of any other degree.

Aashi Pandey (0875CS151002)

Ashish Aswani (0875CS151034)

Avani Kala (0875CS151037)

Avani Sharma (0875CS151038)

Prof. Deepak Singh Chouhan

Supervisor

Project Approval Form

I hereby recommend that the project **Handwritten Character Recognition System** prepared under my supervision by **Aashi Pandey (0875CS151002), Ashish Aswani (0875CS151034), Avani Kala (0875CS151037) & Avani Sharma (0875CS151038)** be accepted in partial fulfillment of the requirement for the degree of Bachelor of Engineering in Computer Science & Engineering.

Prof. Deepak Singh Chouhan

Supervisor

Recommendation concurred in 2018-2019

Prof. Deepshikha Yadav

Project In-Charge

Prof. Uday Moghe

Project Coordinator

Acropolis Technical Campus

Department of Computer Science & Engineering

Certificate

The project work entitled **Handwritten Character Recognition System** submitted by **Aashi Pandey (0875CS151034), Ashish Aswani (0875CS151034), Avani Kala (0875CS151037) & Avani Sharma (0875CS151038)** is approved as partial fulfillment for the award of the degree of Bachelor of Engineering in Computer Science & Engineering by Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.).

Internal Examiner

Name: …….

Date: …. /… /…….

External Examiner

Name: ……..

Date: …. /…/……

Acknowledgment

With boundless love and appreciation, we would like to extend our heartfelt gratitude and appreciation to the people who helped us to bring this work in reality. We would like to have some space of acknowledgment for them.

Foremost, we would like to express our sincere gratitude to our supervisor, **Prof. Deepak Singh Chouhan** whose expertise, consistent guidance, ample time spent and consistent advice that helped us to bring this study into success.

To the project in-charges **Prof. Sumit S. Jain, Prof. Brajesh Chaturvedi, Prof. Deepshikha Yadav** for their constructive comments, suggestions, and critiquing even in hardship.

To the honorable **Prof. Prashant Lakkadwala**, Head, Department of Computer Science & Engineering for his favorable responses regarding the study and providing the necessary facility.

To the honorable **Dr. Sanjay T. Purkar**, Principal, ATC, Indore for his unending support, advises and effort to make possible.

Finally, we would like to pay our thanks to faculty members and staff of the Department of Computer Science & Engineering for their timely help and support.

We also like to pay thanks to our **parents** for their eternal love, support and prayers. Withoutthem, it is not possible.

Aashi Pandey (0875CS151002)

Ashish Aswani (0875CS151034)

Avani Kala (0875CS151037)

Avani Sharma (0875CS151038)

Abstract

This project presents a simple learning rule for recognition of mouse dragged character on our computer screen using the artificial neural network. We will use Kohonen self-organization map for pattern classification which employs unsupervised learning algorithm. A new method called diagonal based feature extraction is introduced for extracting the features of the handwritten alphabets. The results are expected to be quite encouraging in terms of the percentage of characters being successfully recognized. One advantage of the proposed scheme is that the system is quite tolerant to changing conditions and inputs. The system will consistently learn. Moreover, the recognition ratio is expected to be excellent in the proposed system. The proposed recognition system will perform 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.

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

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

Table of Content

Pg. No.

List of Figures

Pg. No.

List of Tables

Pg. No.

Abbreviations

Symbol	Word
• OCR	Optical Character Recognition
• ICR	Intelligent Character Recognition
• SOM	Self-Organizing Map

CHAPTER 1

INTRODUCTION

Chapter 1

Introduction

Character recognition is the process to classify the input character according to the predefined character class. With increasing the interest of computer applications, modern society needs the input text into computer readable form. The purpose of this project is to take handwritten English characters as input, process the character, train the neural network algorithm, to recognize the pattern and modify the character to a beautified version of the input. Some research for handwritten characters is already done by researchers with artificial neural networks. In this project, we use the Kohonen neural network. A network, by its self-organizing properties, is able to infer relationships and learn more as more inputs are presented to it. This project is restricted to English characters only. It can be further developed to recognize the characters of different languages. It engulfsthe concept of neural network. One of the primary means by which computers are endowed with human-like abilities is through the use of a neural network. Neural networks are particularly useful for solving problems that cannot be expressed as a series of steps, such as recognizing patterns, classifying them into groups, series prediction and data mining. Pattern recognition is perhaps the most common use of neural networks. The neural network is presented with a target vector and also a vector which contains the pattern information; this could be an image and handwritten data. The neural network then attempts to determine if the input data matches a pattern that the neural network has memorized. A neural network trained for classification is designed to take input samples and classify them into groups. These groups may be fuzzy, without clearly defined boundaries. This project concerns detecting free handwritten characters.

1.1 Rationale

Handwriting recognition has been one of the most fascinating and challenging research areas in the field of image processing and pattern recognition in recent years. It contributes immensely to the advancement of an automation process and can improve the interface between man and machine in numerous applications. Several research works have been focusing on new techniques and methods that would reduce the processing time while

providing higher recognition accuracy. In general, handwriting recognition is classified into two types as off-line and on-line handwriting recognition methods. In the off-line recognition, the writing is usually captured optically by a scanner and the completed writing is available as an image. But, in the onlinesystem, the two-dimensional coordinates of successive points are represented as a function oftime and the order of strokes made by the writer are also available. The online methods have been shown to be superior to their off-line counterparts in recognizing handwritten characters due to the temporal information available with the former. However, in the off-line systems, the neural networks have been successfully used to yield comparably high recognition accuracy levels. Several applications including mail sorting, bank processing, document reading and postal address recognition require off-line handwriting recognition systems. As a result, the off-line handwriting recognition continues to be an active area for research towards exploring the newer techniques that would improve recognition accuracy.

1.2 Goal

The goal of a character recognition system is to transform a handwritten text document on paper into a digital format that can be manipulated by word processor software. The system is required to identify a given input character form by mapping it to a single character in given character set. Each handwritten character is split into a number of segments (depending on the complexity of the alphabet involved) and each segment is handled by a set of purpose-built a neural network. The final output is unified via a lookup table. Neural network architecture is designed for different values of the network parameters like the number of layers, a number of neurons in each layer, the initialvalues of weights, the training coefficient and the tolerance of the correctness. The optimal selection of these network parameters certainly depends on the complexity of the alphabet.

1.3 Objectives

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

1.4 Overview

We are using Kohonen neural network algorithm for development because other machine learning algorithms need more supervised learning. Also, the accuracy of Kohonen neural algorithm is good with fewer complexities.

1.5 The contribution of the project

With the help of this software, we will be able to convert free handwritten text to the beautiful text in the computer. It will be done by recognizing their pattern.

1.5.1 Market Potential

- This project will help in education advancements by checking patterns.
- It will also be useful in the banking sector by recognizing the signatures which is also a very tedious task.

1.5.2 Innovativeness

As we know there are few solutions available in the market like Google Smart whiteboard but the limitation of this system is that it cannot recognize patterns. Our system will even recognize patterns and handwritten characters which will create a great impact on the market.

1.5.3 Usefulness

With the help of this project, we will take handwritten English characters as input, process the character, train the neural network algorithm, to recognize the pattern and modify the character to a beautified version of the input. With the help of this software, we will be able to convert free handwritten text to the beautiful text in the computer. It will be done by recognizing their pattern. Several applications including mail sorting, bank processing, document reading and postal address recognition require off-line handwriting recognition systems. As a result, the off-line handwriting recognition continues to be an active area for research towards exploring the newer techniques that would improve recognition accuracy.

1.6 Report Organization

The remaining section of the report is structured as follows:

Chapter 2 provides a detailed project plan. Different steps taken for information collection are also discussed in this chapter.

Chapter 3 provides details analysis on requirement engineering.

Chapter 4 & 5 provides details of conceptual design and methodology.

CHAPTER 2

PROJECT PLAN

Chapter 2

Project Plan

2.1 Risk Management

Risk management means risk containment and mitigation. First, you've got to identify and plan. Then be ready to act when a risk arises, drawing upon the experience and knowledge of the entire team to minimize the impact to the project.

- Identify risks and their triggers.
- Classify and prioritize all risks.
- Craft a plan that links each risk to mitigation.
- Monitor for risk triggers during the project.
- Implement the mitigating action if any risk materializes.
- Communicate risk status throughout the project.

2.1.1 Project Risk

Some of the major risks that may happen during the life cycle of our project:

- An extra line of codes or redundant algorithm may cause wastage of memory.
- If the customer provides irrelevant information then it may generate some unknown risk.
- If a 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.
- Lack of training on tools and inexperience may cause difficulty in completing project modules.

2.2 Project Schedule

A project schedule is a listing of a project's milestones, activities, and deliverables, usually with intended start and finish dates. Those items are often estimated by other information included in the project schedule of resource allocation, budget, task duration, and linkages of dependencies and scheduled events. A schedule is commonly used in the project

planning. The project schedule is the tool that communicates what work needs to be performed, which resources of the organization will perform the work and the timeframes in which that work needs to be performed.

2.2.1 Milestones & Deliverables

Table 2.2.1 Milestones & Deliverable

Deliverable	Content	Due Date
Deliverable 0	Preliminary Project Plan	06/08/18
Deliverable 1	Interim Progress • Project Plan • Improved Understanding Document • PowerPoint	13/08/18
Deliverable 2	Final Synopsis • Project Plan • Improved Understanding Document	16/08/18
Deliverable 3	Interim Progress • Project Plan • Improved Understanding Document	11/10/18
Deliverable 4	System Design Report • Project Plan • Improved Understanding Document • Traceability Matrix	30/10/18
Deliverable 5	Implementation of Image Preprocessing Algorithm & Segmentation Module	01/02/19
Deliverable 6	Implementation of Feature Extraction Module & Classification Module	15/02/19
Deliverable 7	Implementation of Neural Network Algorithm	10/03/19
Deliverable 8	Unit Integration	20/03/19
Deliverable 9	Testing	25/03/19
Deliverable 10	Deployment	01/04/19

2.2.2 Schedule Chart

Table 2.2.2 Schedule Chart

Deliverable	Content	Start Date	End Date
Deliverable 0	Preliminary Project Plan	25/7/18	06/8/18
Deliverable 1	Project Plan	06/8/17	13/8/18
	Improved Understanding Document		
	PowerPoint		
Deliverable 2	Project Plan	13/08/18	20/8/18
	Improved Understanding Document		
	Final Synopsis		
Deliverable 3	Project Plan	20/8/18	11/10/18
	Improved Understanding Document		
	Traceability Matrix		
Deliverable 4	System Design Report	11/10/18	30/10/18
Deliverable 5	Implementation of Image Preprocessing Algorithm & Segmentation Module	30/10/18	01/02/19
Deliverable 6	Implementation of Feature Extraction Module & Classification Module	01/02/19	15/02/19
Deliverable 7	Implementation of Neural Network Algorithm	15/02/19	10/03/19
Deliverable 8	Unit Integration	10/03/19	20/03/19
Deliverable 9	Testing	20/03/19	25/03/19
Deliverable 10	Deployment	25/03/19	01/04/19

2.3 Role & Responsibility of Team Members

Table 2.3 Roles & responsibility of team members

Team Members	Project Role
Aashi Pandey	Requirement Analysis & Front-end
Ashish Aswani	Designing & Back-end development
Avani Kala	Requirement Gathering & database
Avani Sharma	Risk Management & Front-end

2.4 Process Model Adopted

We have used a combination of an iterative and incremental process model for this application. Iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented. In each iteration, design modifications are made and new functional capabilities are added. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental).

The Unified Process groups increments/iterations into phases: inception, elaboration, construction, and transition:-

- **Inception:** Inception identifies project scope, requirements (functional and non-functional) and risks at a high level but in enough detail that work can be estimated.
- **Elaboration:** Elaboration delivers a working architecture that mitigates the top risks and fulfills the non-functional requirements.
- **Construction:** Construction incrementally fills-in the architecture with production-ready code produced from analysis, design, implementation, and testing of the functional requirements.
- **Transition:** Transition delivers the system into the production operating environment.

Each of the phases may be divided into one or more iterations, which are usually time-boxed rather than feature-boxed. Architects and analysts work one iteration ahead of developers and testers to keep their work-product backlog full.

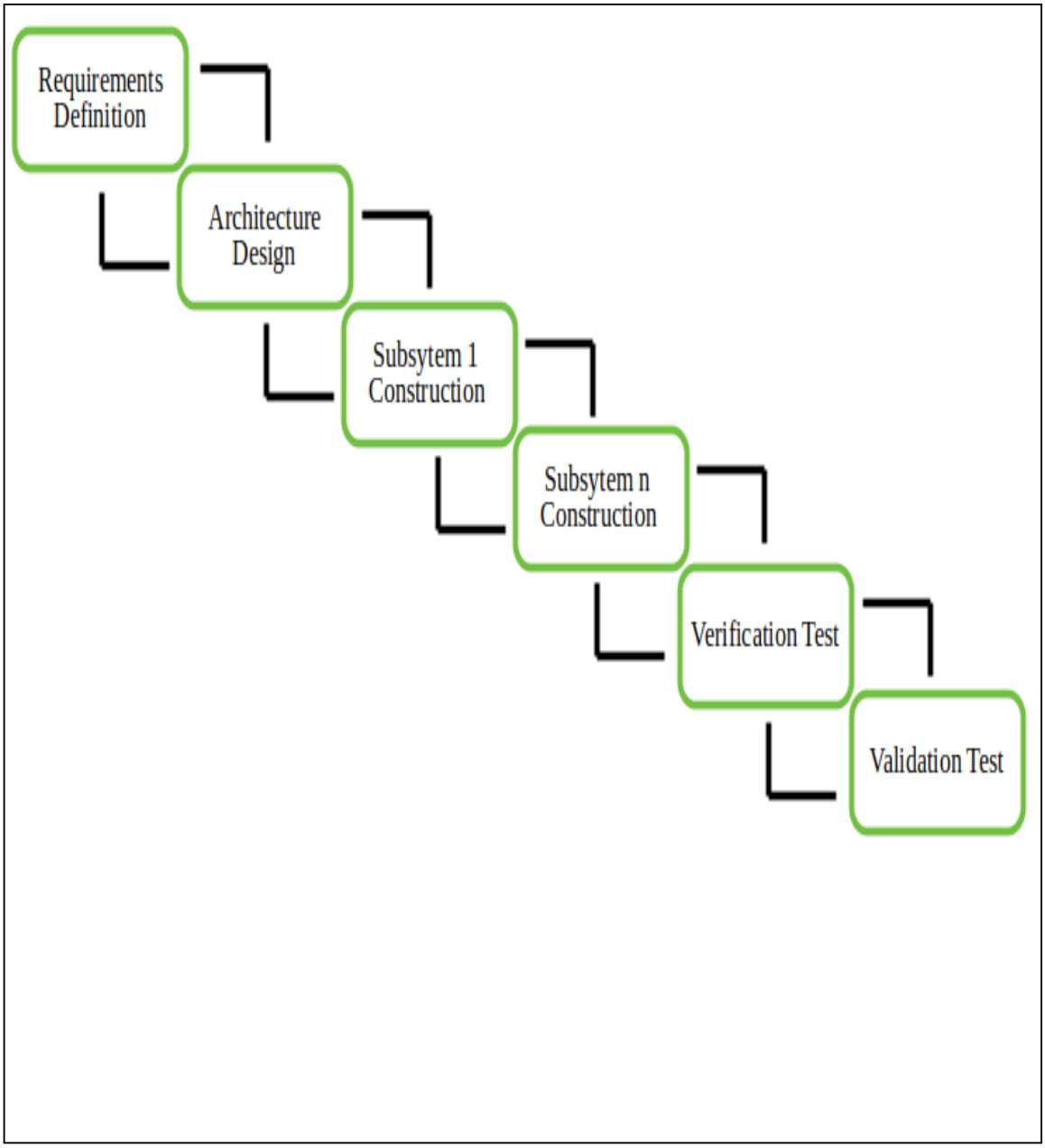

Figure 2.4 Process model

CHAPTER 3

REQUIREMENT ENGINEERING

Chapter 3

Requirement Engineering

3.1 User Roles and Responsibilities

The user has complete ownership of the application and responsible for any misuse of the application.

Responsibilities:

- Start and stop resources.
- Use the system operations controls.

3.2 Requirements

3.2.1 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 accuracy for character recognition.

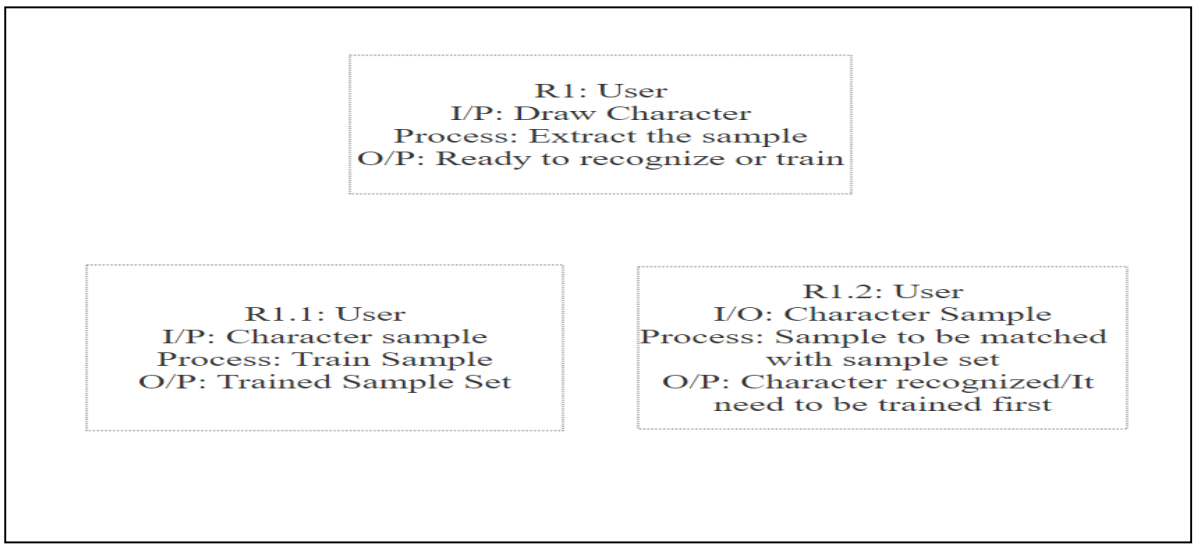

Figure 3.2.1 Functional requirements

3.2.2 Non-functional Requirements

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

3.3 Use Case Diagram

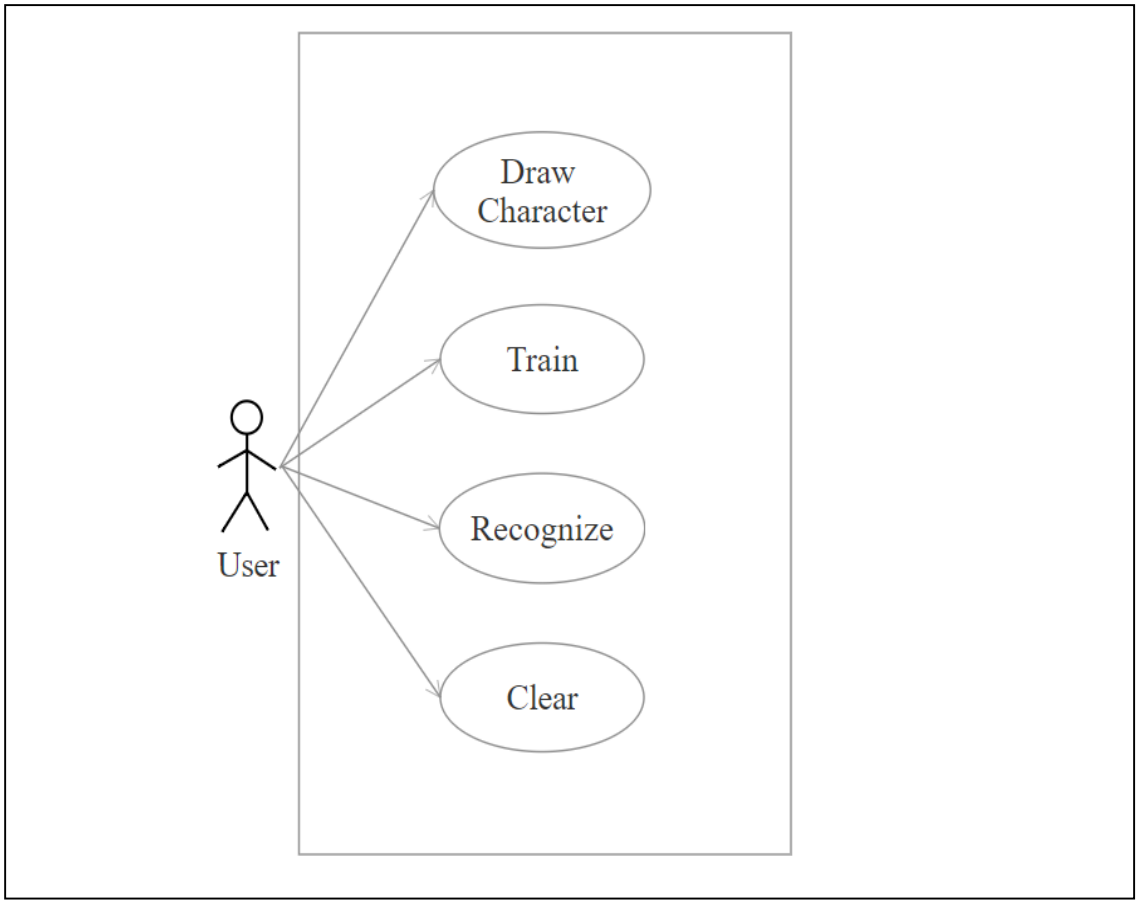

Figure 3.3 Use case diagram

3.3.1 Use Case Description

- **Draw Character**

  Use Case: Canvas

  Actor: User

  Description: User will be able to draw character through input console.

  Precondition: N.A.

  Post condition: Ready to recognize and train.

  Basic Flow:

  Subject Area: Institution

  Actor: Student

  Trigger: As soon as the touch is performed on the canvas it will start recording the pattern.

  Alternative Flow: N.A.

- **Training Character**

  Use Case: Train

  Actor: User

  Description: With the help of this use case the characters will be trained.

  Precondition: Character must be drawn on the canvas.

  Post condition: Ready to recognize.

  Basic Flow:

  Subject Area: Institution

  Actor: Student

  Trigger: As soon as the canvas records the pattern then, it will start observing the pattern.

  Alternative Flow: Maybe the pattern is already recorded then it will not train that pattern.

- **Recognizing Character**

  Use Case: Recognize

  Actor: User

  Description: With the help of this use case the characters will be recognized.

Precondition: Character must be recorded in the database.

Post condition: N.A.

Basic Flow:

Subject Area: Institution

Actor: Student

Trigger: As soon as the touch is performed on the canvas it will start recording the pattern.

Alternative Flow: N.A.

- **Clear Database**

  Use Case: Clear

  Actor: User

  Description: With the help of this use case the characters stored in the database will be deleted

  Precondition: Characters must be available in the database

  Post condition: N.A.

- **Basic Flow:**

  Subject Area: Institution

  Actor: Student

  Trigger: The stored characters will be deleted.

- **Alternative Flow:**

  Description: No character available in database.

  Termination Outcome: N.A.

3.4 Replacement of legacy system

There are softwares available in the market such as ORC and ICR but they don't recognize the human handwriting and possess less accuracy. Also, the cost of the already existing system was too high.

Thus a more accurate system with better performance could be achieved by using a Kohonen Neural Network algorithm. With the simplicity of the computation and quickness, Kohonen neural network algorithm provides better final distortion.

Also, no additional adaptation parameter for tuning is required and thus gives deterministic reproducible results.

In fact, it is nowadays extensively used for data mining, data visualization, and exploratory data analysis. Some users drift towards the Kohonen algorithm as it is a deterministic algorithm which can go faster in some cases.

3.5 Requirement Review

A requirements review is a manual process that involves people from both client and contractor organizations. They check the requirements document for anomalies and omissions. The review process may be managed in the same way as program inspections. Alternatively, it may be organized as a broader activity with different people checking different parts of the document.

Requirements reviews can be informal or formal. Informal reviews simply involve contractors discussing requirements with as many system stakeholders as possible. It is surprising how often communication between system developers and stakeholders ends after elicitation and there is no confirmation that the documented requirements are what the stakeholders really said they wanted. Many problems can be detected simply by talking about the system to stakeholders before making a commitment to a formal review.

In a formal requirements review, the development team should ‘walk’ the Conflicts, contradictions, errors and omissions in the requirements should be pointed out by reviewers and formally recorded in the review report. It is then up to the users, the system procurer and the system developer to negotiate a solution to these identified problems. Reviews can be used throughout software development for quality assurance and data collection. Requirements review is a review by a group of people to find errors and point out other matters of concern in the requirement specification of the system. The review group should include the author of requirement documents, someone who understands the needs of the client, a person of the design team, and the person(s) responsible for maintaining the requirement document. It is also good practice to include some people not directly involved with product development like a software quality engineer.

Author of requirement documents, someone who understands the needs of the client, a person of the design team, and the person(s) responsible for maintaining the requirement document. It is also good practice to include some people not directly involved with product development like a software quality engineer.

The goal of a character recognition system to transform a handwritten text document on paper into a digital format that can be manipulated by word processor software is expected to be accomplished. Each handwritten character is split into a number of segments (depending on the complexity of the alphabet involved) and each segment is handled by a set of purpose-built a neural network. Also the system will detect text regions present in the image and retrieve text present in the image and display them to the user. The final output is unified via a lookup table.

CHAPTER 4

ANALYSIS, CONCEPTUAL DESIGN & TECHNICAL ARCHITECTURE

Chapter 4

Analysis, Conceptual Design & Technical Architecture

4.1 Technical Architecture

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.

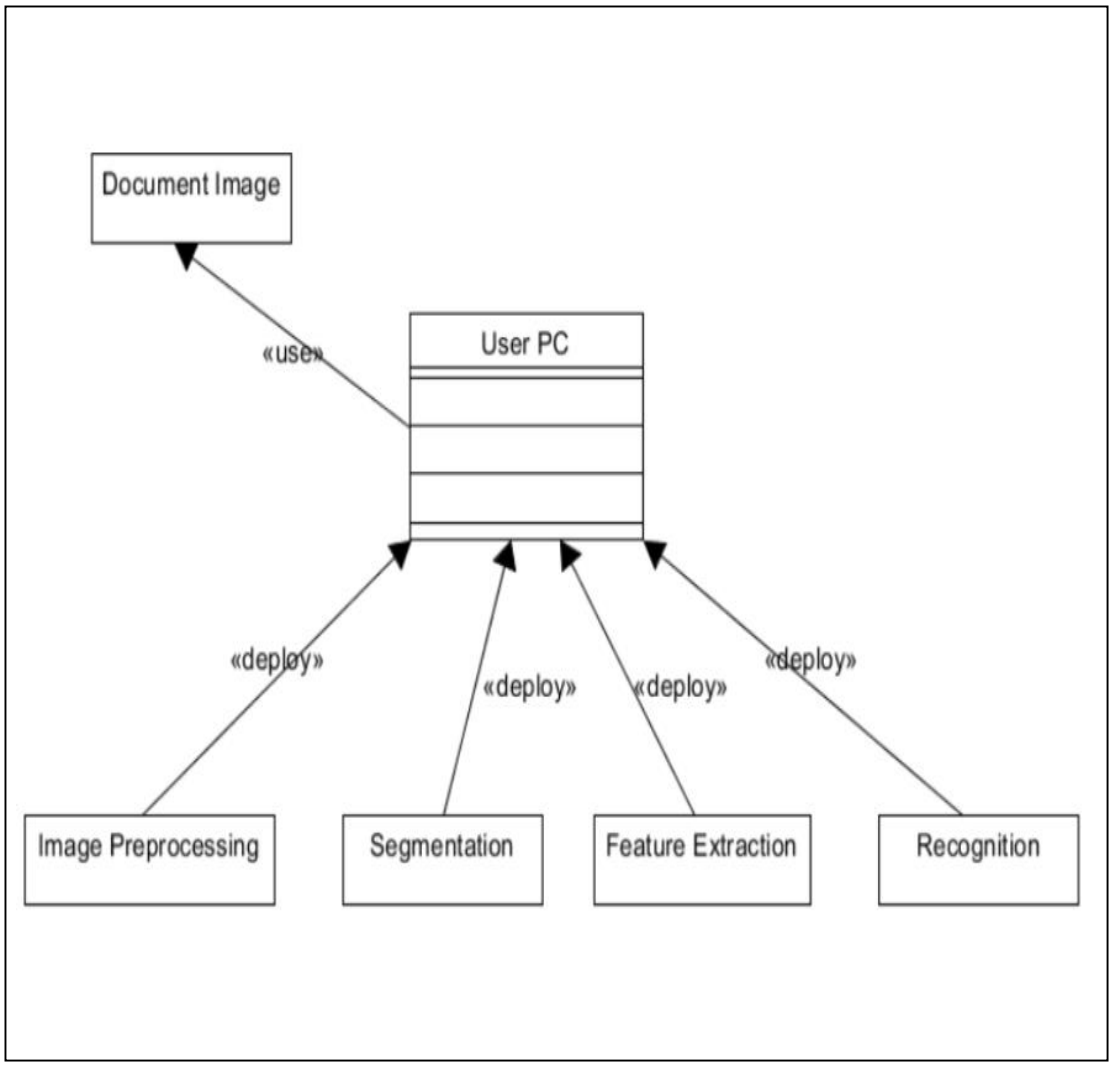

Figure 4.1 Technical Architecture

4.2 Sequence Diagrams

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.

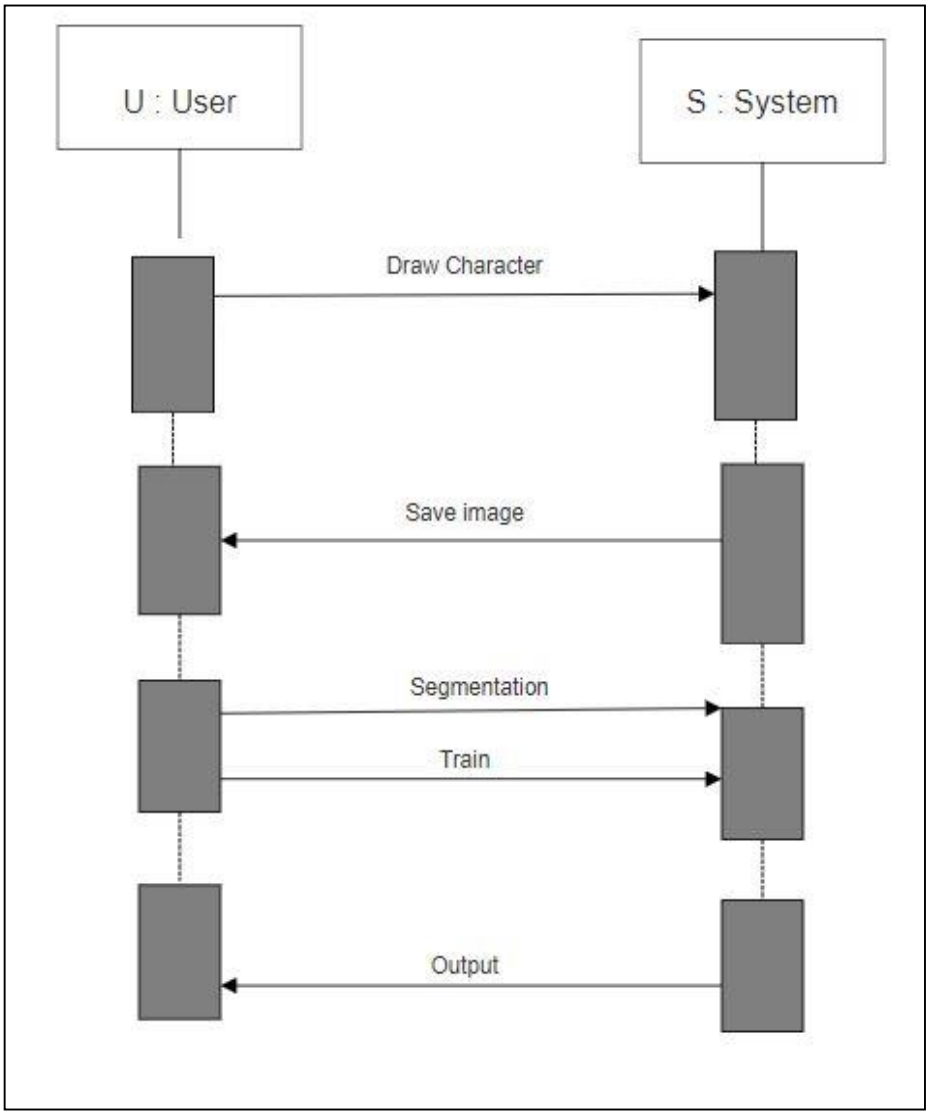

Figure 4.2.1 Sequence diagram (Train sample)

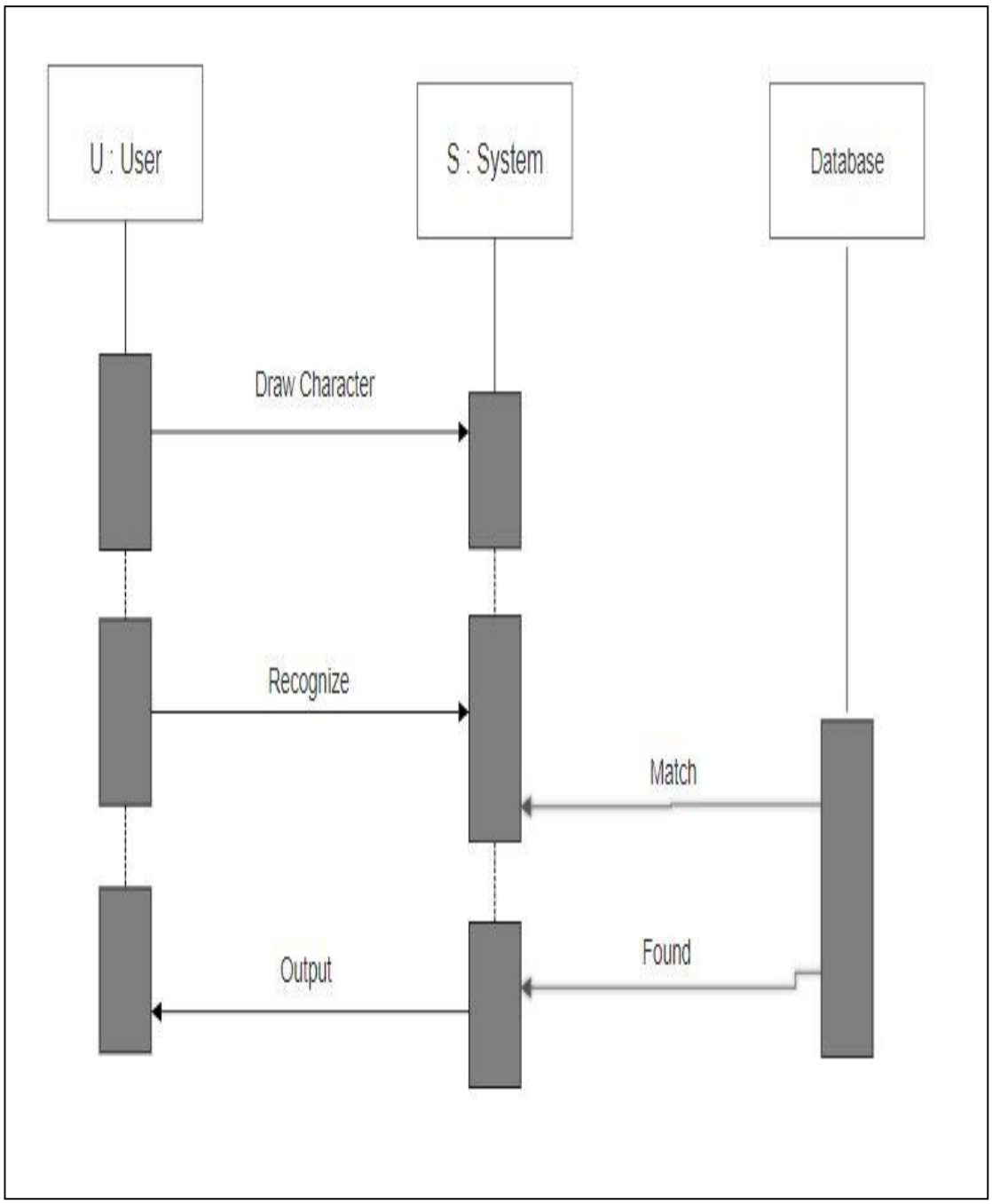

Figure 4.2.2 Sequence diagram (Recognize character)

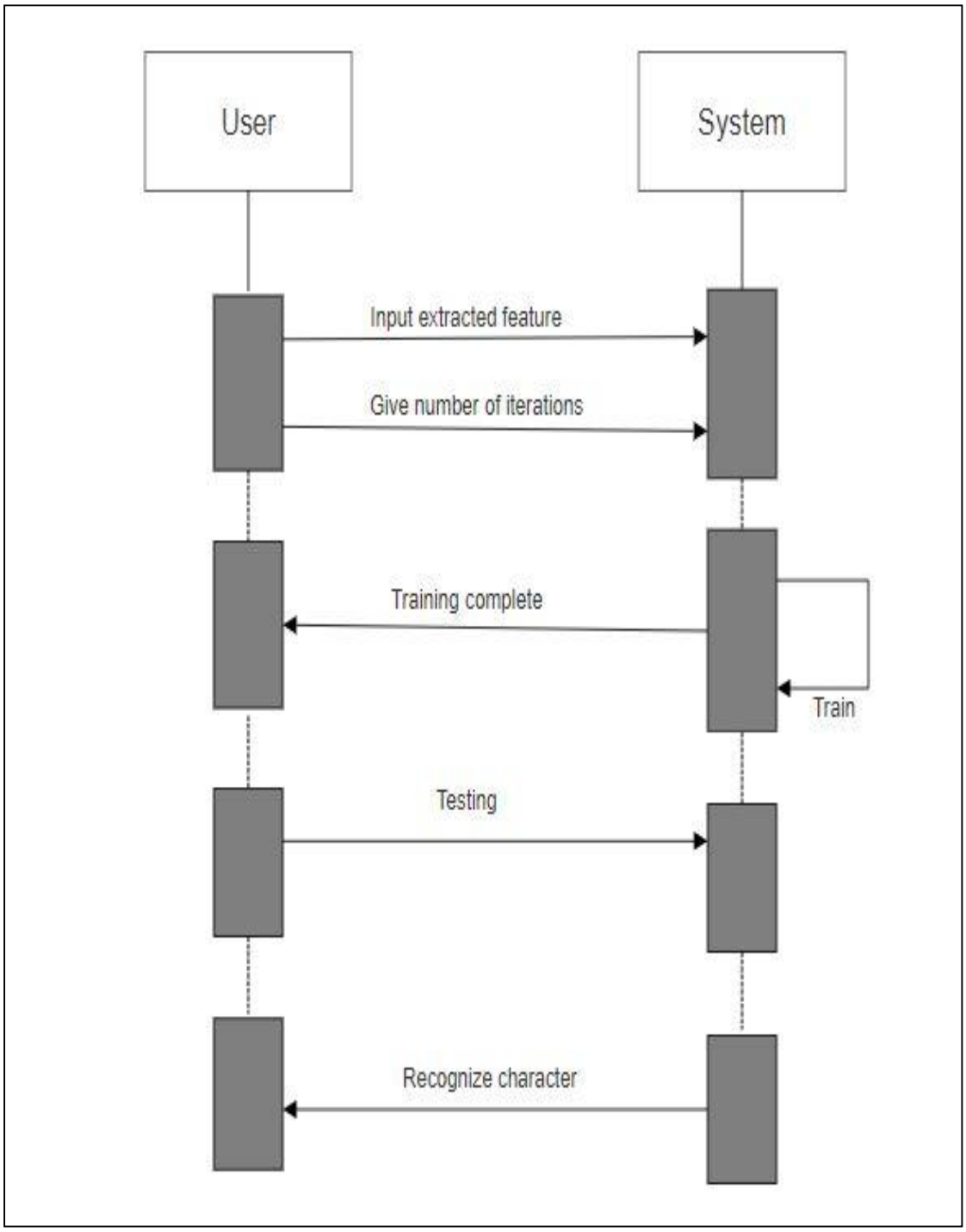

Figure 4.2.3 Sequence diagram (Train and recognize)

4.3 Class Diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

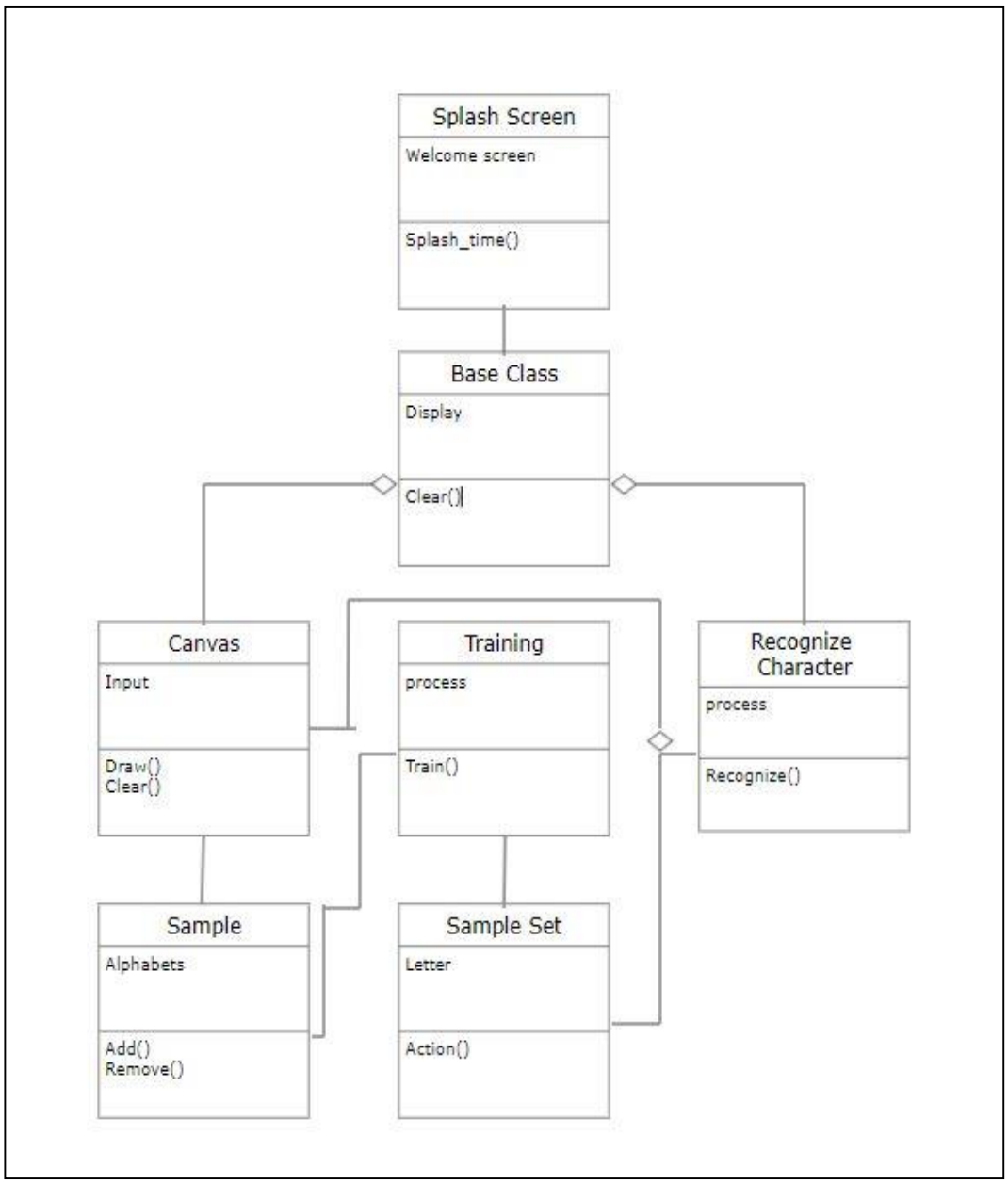

Figure 4.3 Class diagram

4.4 Data Flow Diagrams

A data-flow diagram (DFD) is a way of representing a flow of a data of a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

Level 0 DFD

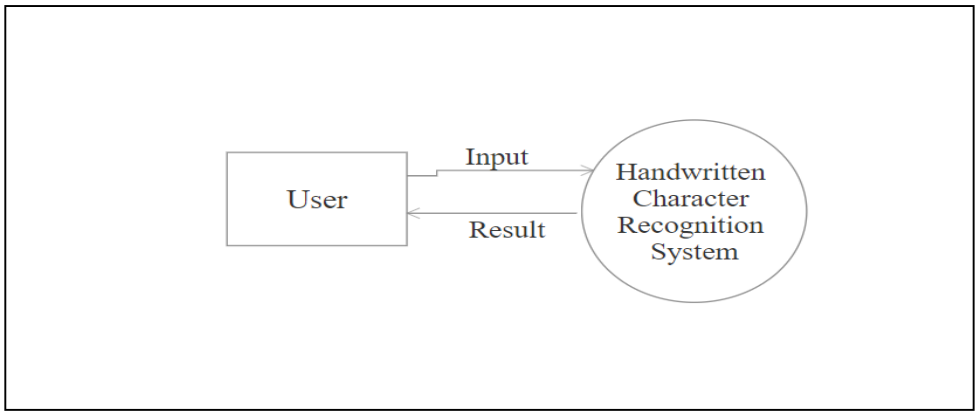

Figure 4.4.1 Level 0 DFD

Level 1 DFD

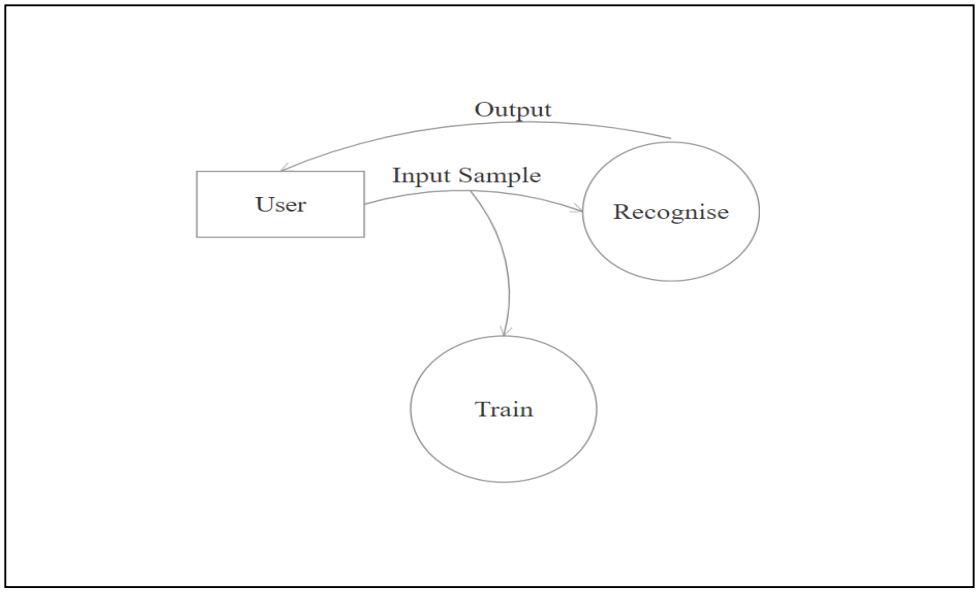

Figure 4.4.2 Level 1 DFD

CHAPTER 5

METHODOLOGY

Chapter 5
Methodology

5.1 Proposed Algorithm

Kohonen Neural Network method is an unsupervised learning process studying the distribution of a set of patterns without any class information. The basic idea of this technique is understood from how human brain stores images/patterns that have been recognized through eyes, and the the human brain is able to reveal the images/ patterns back. Therefore, the application of this model is widely used in object recognition. The Kohonen Self-Organizing Map (SOM) designed by Teuvo Kohonen is a variation of the traditional Artificial Neural Network. It is a third generation neural network, meaning that many of its functional characteristics are thought to mirror those found in biological fact. A SOM consists of a collection of nodes of neurons that are each connected to every other node and each node has associated with it a set of input weights w. The SOM also has associated with it a metric for determining which nodes are in the neighborhood N of a given node. When the network is presented with a vector xi at its input, it computes the neural response SJ of the node j using the formula: Sj=wj * xi.

1. Normalize both JW and xi before computing the dot product, SJ, and refer to the node that produces the largest value of s as node k. Since the dot product of the normalized wk and xi vectors is the cosine of the angle between them, we can conclude that the winning node is the one with the weight vector closest to the input vector in its spatial orientation. We can then say that node k giving the largest s and is closest to recognizing the input vector. We allow the nodes to learn by applying a Δw to their weights using the formula: $\Delta wk=\alpha(xi-wk)$

2. Where α is a constant in the range [0,1] called the learning constant. The learning process is applied to the maximum response neuron and neurons in its defined neighborhood.

This training process can be described by the following algorithm:

1. A cycle: for every input vector xi

a. Apply vector input to the network and evaluate the dot products of the normalized weights on each node and a normalized input vector. Call these dot products.

b. Find the node k with the maximal response sk.

c. Train node k, and all the nodes in some neighborhood of k, according to the learning equation above.

d. Calculate a running average of the angular distance between the values of wk and their associated input vectors.

e. Decrease the learning rate.

2. After every M cycles, called the period, decrease the size of the neighborhood N.

3. Repeat steps 1-2 for some finite period of time or until the average angular distance. One advantage to this scheme is that the system is quite tolerant to changing conditions and inputs. The system consistently learns. Moreover, the recognition ratio is excellent in the proposed system.

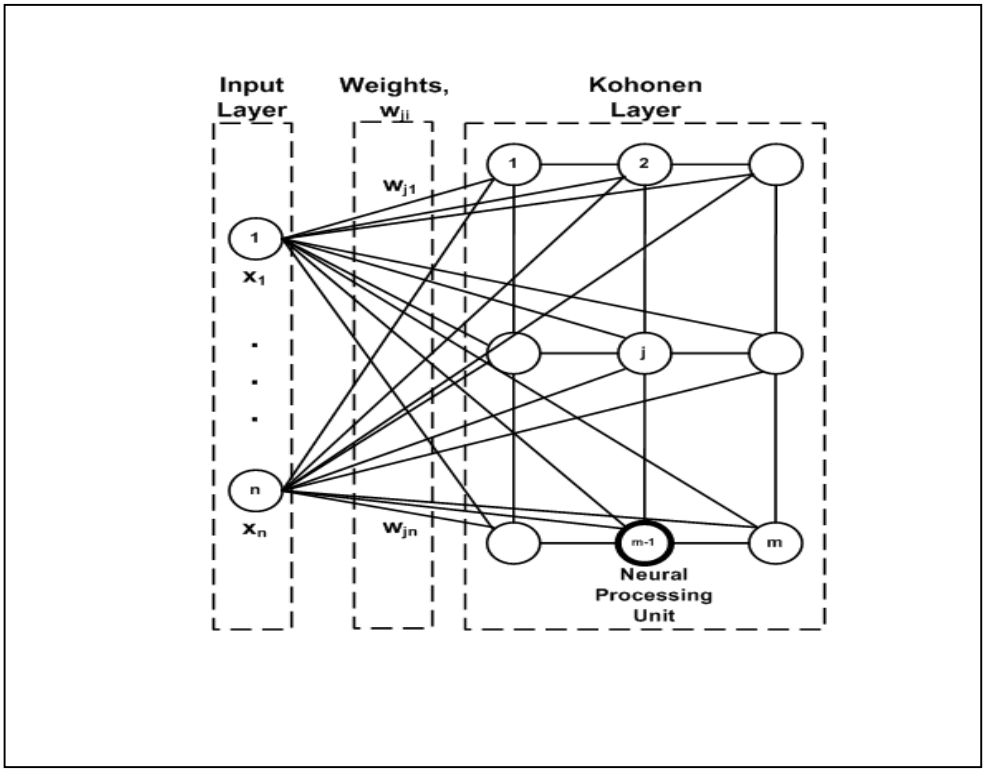

Figure 5.1 Kohonen Structure

The overall method of the implemented system is illustrated in the following figure:

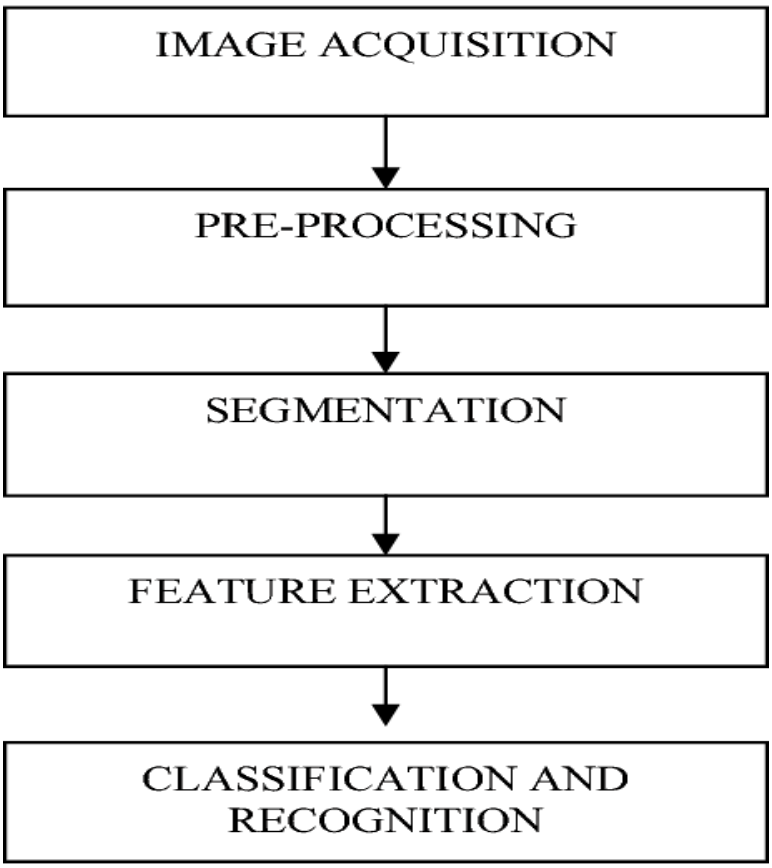

Figure 5.1.1 Stages of Handwritten Character Recognition

The proposed method comprises of following phases:

1. The userneeds to draw the character on the given canvas through input consoles.
2. The mapping canvas will do binarization of the character i.e. the character will be converted into a binary form for processor understanding.
3. After binarization, the information of the character will be saved into the database.
4. With the help of information available, the character will be trained by the Kohonen algorithm and the character will be ready to recognize.
5. The given character will be compared with the database of trained samples.
6. The character will be recognized if the given input is matched with the sample set else it will say that the character needs to be trained first.

5.2 Tools Required

Resources required for development

Hardware configurations:

- CPU Intel Core i3-4005U
- Monitor Resolution - 800X600 minimum
- RAM - 4.00 GB
- Hard Disk– 100 GB

Software configurations:

- Window 7 or above
- Java Compiler 8
- Net Beans IDE 8.1

Resources required for system application

Hardware configurations:

- CPU Intel Core 2 Duo
- MONITOR RESOLUTION - 800X600 minimum
- RAM - 2.00 GB
- HARD DISK - 128GB

Software Requirements:

- Window 7 or above

CHAPTER 6

IMPLEMENTATION & TESTING

Chapter 6

Implementation & Testing

6.1 User Interface Design

For the creation of GUI we have used the java swing toolkit.

Swing library is an official Java GUI toolkit released by Sun Microsystems.

The main characteristics of the Swing toolkit are-

Light Weight − Swing components are independent of native Operating System's API as Swing API controls are rendered mostly using pure JAVA code instead of underlying operating system calls.

Rich Controls − Swing provides a rich set of advanced controls like tree, tabbed pane, slider, color picker, and table controls.

Highly Customizable − Swing controls can be customized in a very easy way as visual appearance is independent of internal representation.

Pluggable look-and-feel − Swing based GUI application look and feel can be changed at run-time, based on available values.

6.2 Implementation Approaches

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

6.2.1 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 image

Output: Gray scaled 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 of up to 256 gray level to a black and white image. The simplest way to use image binarization is to choose a threshold value and classify all pixels with values above the threshold as black and all other pixels are white.

Algorithm: Image binarization

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

6.2.2 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 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

Step1: 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 characters.

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

6.2.3 Module 3: Feature Extraction

As an individual character 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. These extracted features are stored in .dat file.

6.2.4 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 )
```

```
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 ;
}
```

6.3 Testing Approaches

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.

6.3.1 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.

Table 6.3 (a) Test case 1

Test Case Name	Drawing character on canvas
Test Case Description	Character must be drawn on the canvas with the help of mouse
Steps	1. Open application 2. Draw character on canvas
Expected Results	Input character accepted
Actual Results	As expected

Table 6.3 (b) Test case 2

Test Case Name	Add character
Test Case Description	Application should allow to select a character for training
Steps	1. Draw character 2. Click on add button
Expected Results	Character should add successfully
Actual Results	As expected

Table 6.3 (c) Test case 3

Test Case Name	Training
Test Case Description	Start training character
Steps	1. Draw character on canvas 2. Click on train button
Expected Results	Character should be trained
Actual Results	As expected

Table 6.3 (d) Test case 4

Test Case Name	Recognition
Test Case Description	Draw character to recognize
Steps	1. Draw character on canvas 2. Click on recognize button
Expected Results	Character should be recognized
Actual Results	As expected

6.3.2 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 6.3.2 Integration Testing

CHAPTER 7

RESULTS & DISCUSSION

Chapter 7

Results & Discussion

Result:

"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 the character is drawn on the canvas and the character the character needs to be trained so that it can be recognized later. Now the trained character is ready to recognize.

Discussion:

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

Conclusion

Many regional languages throughout the 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.

Future Scope

This work further extended to the character recognition for other languages. It can be used to convert the fax and newspapers 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.

APPENDIX A

PROJECT SYNOPSIS

ACROPOLIS TECHNICAL CAMPUS, INDORE

Department of Computer Science & Engineering

Synopsis

on

Handwritten Character Recognition System

1. Problem Domain

- **Problem Identification:**

Handwriting is the human way of communicating each other by using written media. By the advancement in technology and development of science, there is still difference between computer printing and human handwriting and computer is unable to read human writing much efficiently. Therefore, it is needed that computer should be able to receive input in the form of handwriting data and able to recognize the handwriting input.

- **Problem in existing system:**

There are few software's available in the market which are ORC i.e. optical character recognition but it doesn't recognize the human handwriting and for doing so ICR i.e. intelligent character recognition technology has evolved which is under development and possess less accuracy, apart from this GOOGLE has developed the machine called Smart Whiteboard which is based on ICR but the disadvantage is that it is machine dependent and expensive.

2. Solution Domain

- **Suggested Solution:**

Since the existing solutions are expensive and less efficient we can overcome this problem by handwritten character recognition system which will be developed using Kohonen Neural Networks and this system will be inexpensive and more efficient.

The application can be operated on any personal computer. The application can be built using Java programming language which is an open source language.

- **Purpose of the project/Innovativeness & Usefulness:**

The purpose of this project is to take handwritten English characters as input, process the character, train the neural network algorithm, to recognize the pattern and modify the character to a beautified version of the input. With the help of this software, we will be able to convert free hand written text to the beautiful text in computer. It will be done by recognizing their pattern.

Improvement over the existing system:

The primary goal of this project is to develop a neural network system which will be machine independent and platform independent as well.

1. Handwritten Character Recognition Software will segregate the input samples and classify them into groups. These groups may be fuzzy, without clearly defined boundaries. This project concerns detecting free handwritten characters.
2. It is much faster than existing system and it doesn't even require human resource for performing the task.
3. The previous system doesn't possess pattern recognizing capability and this system can recognize previously trained patterns.
4. Handwritten Character Recognition Software provides an accuracy rate of up to 99%.

3. System Domain

- **Required Resources**

Resources required for development

Hardware Requirements:

CPU Intel Core i3-4005U

Hard Disk - 128GB

RAM - 4.00 GB

Monitor Resolution – 800*600 minimum

Software Requirements:

Window 7 or above

Java Complier 8

Net Beans IDE 8.1

Resources required for system application

Hardware Requirements:

CPU Intel Core 2 Duo

Monitor Resolution – 800*600 minimum

RAM - 2.00 GB

Hard Disk - 128GB

Software Requirements:

Windows 7 or above

- ## Methodology to be adopted/ Planning of work

Methodology: The methodology to be adopted for the development of this project is incremental approach.

Stages to be involved for the development of the system:

Design & Development: The designing of each class will be done accordingly and then all the classes will be developed and if all the classes are developed then we will move to the testing.

Testing: The each class will be tested using test cases and after testing the each class we will move to the implementation.

Implementation: After development and testing the application will be built and ready for the deployment.

The proposed methodology is expected to produce good results for images containing handwritten text written in different styles, different size and alignment with varying background.

4. Application Domain

- ## Scope of Project:

a) This project will help in education advancements by checking patterns.

b) It will also be useful in banking sector by recognizing the signatures which is also very tedious work.

c)

- **Impact of the work on real life / end user:**

  With the help of this software, we will be able to convert free hand written text to the beautiful text in computer. It will be done by recognizing their pattern.

5. Expected outcomes/Benefits

The system is developed to provide improved facilities. The overall manual recognition work and time consumption will be reduced. Some of the benefits of the system are listed below-

1. Minimize manual data entry.
2. Minimize time required for processing.
3. Reduces paperwork.
4. Eliminate duplicate data entry and errors.
5. Increased Security and confidentiality.
6. Much more efficient as compared to manual processing.
7. No need to check the records for recognizing the pattern.

6. References

1. Kohonen Neural Networks https://ubiquity.acm.org last accessed on 15/10/2018.
2. Pattern Recognition Introduction https://www.geeksforgeeks.org last accessed on 15/10/2018.
3. Java Tutorial-Learn Core and Advanced Java: https://www.w3schools.in last accessed on 15/10/2018.
4. Lulu C. Munggaran, Suryarini Widodo "Handwritten Pattern Recognition Using Kohonen Neural Network Based on Pixel Character", Volume-5, Dated 25/11/2014, Page no. 1-6.

Guided By-

Prof. Deepak Singh Chouhan

Group Members-

Aashi Pandey (0875CS151002)

Ashish Aswani (0875CS151034)

Avani Kala (0875CS151037)

Avani Sharma (0875CS151038)

APPENDIX B

GUIDE INTERACTION REPORT

APPENDIX B: GUIDE INTERACTION REPORT

S. No.	Date	Task Performed	Remark
1.	8/10/18	Ideation of topic	
2.	15/10/18	Synopsis rechecking	
3.	16/10/18	Approval of synopsis	
4.	16/10/18	Discussion about presentation	
5.	17/10/18	Presentation approval	
6.	27/10/18	Discussion about second presentation	
7.	29/10/18	Changes in project report	
8.	11/11/18	7^{th} semester report approval	
9.	10/03/19	Modules implementation	
10.	20/03/19	Modules integration	
11.	25/03/19	System testing	
12.	12/04/19	Report approval	

APPENDIX C

PROJECT SNAPSHOTS

APPENDIX C: PROJECT SNAPSHOTS

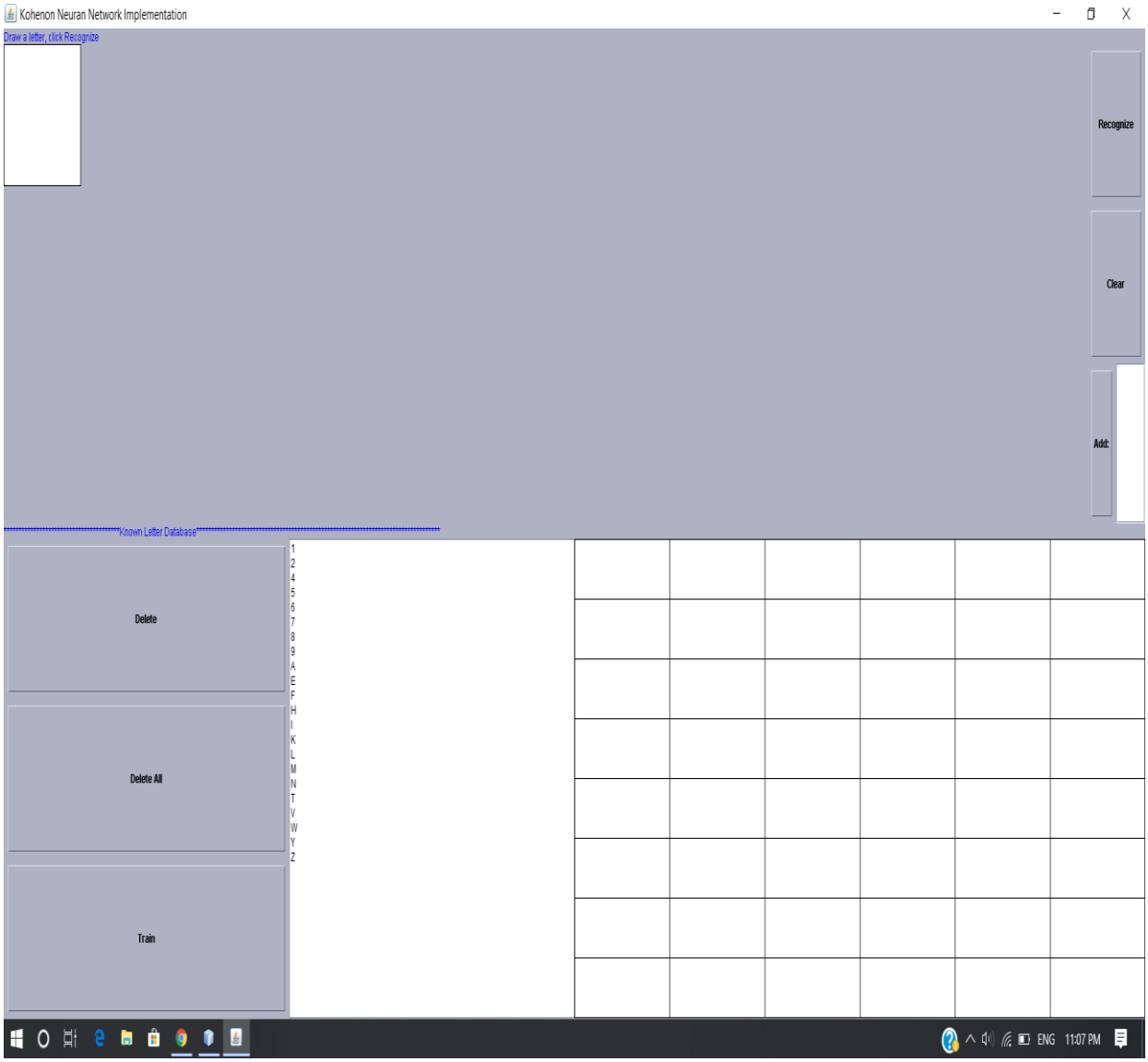

Figure: Front - end

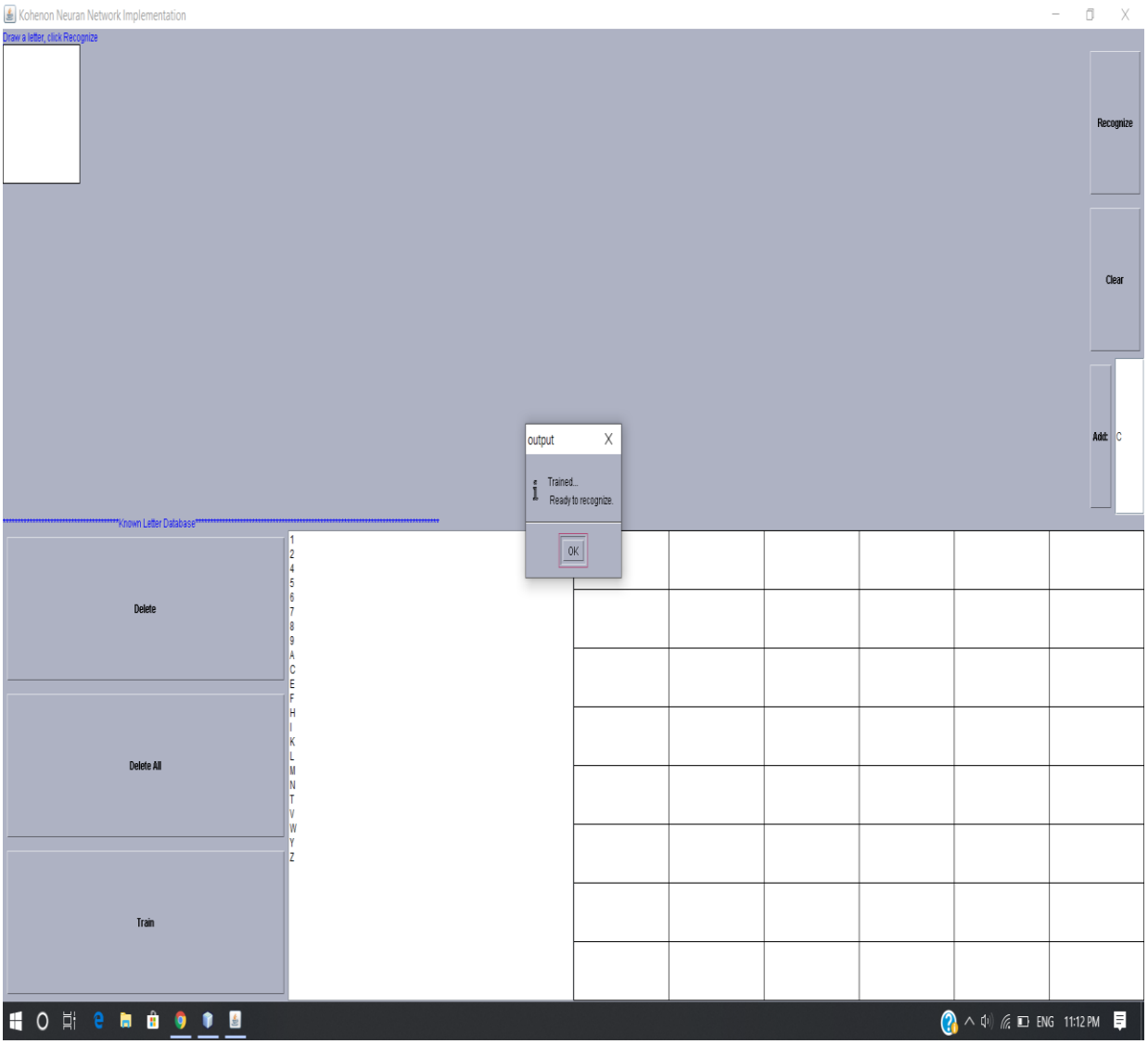

Figure: Training character

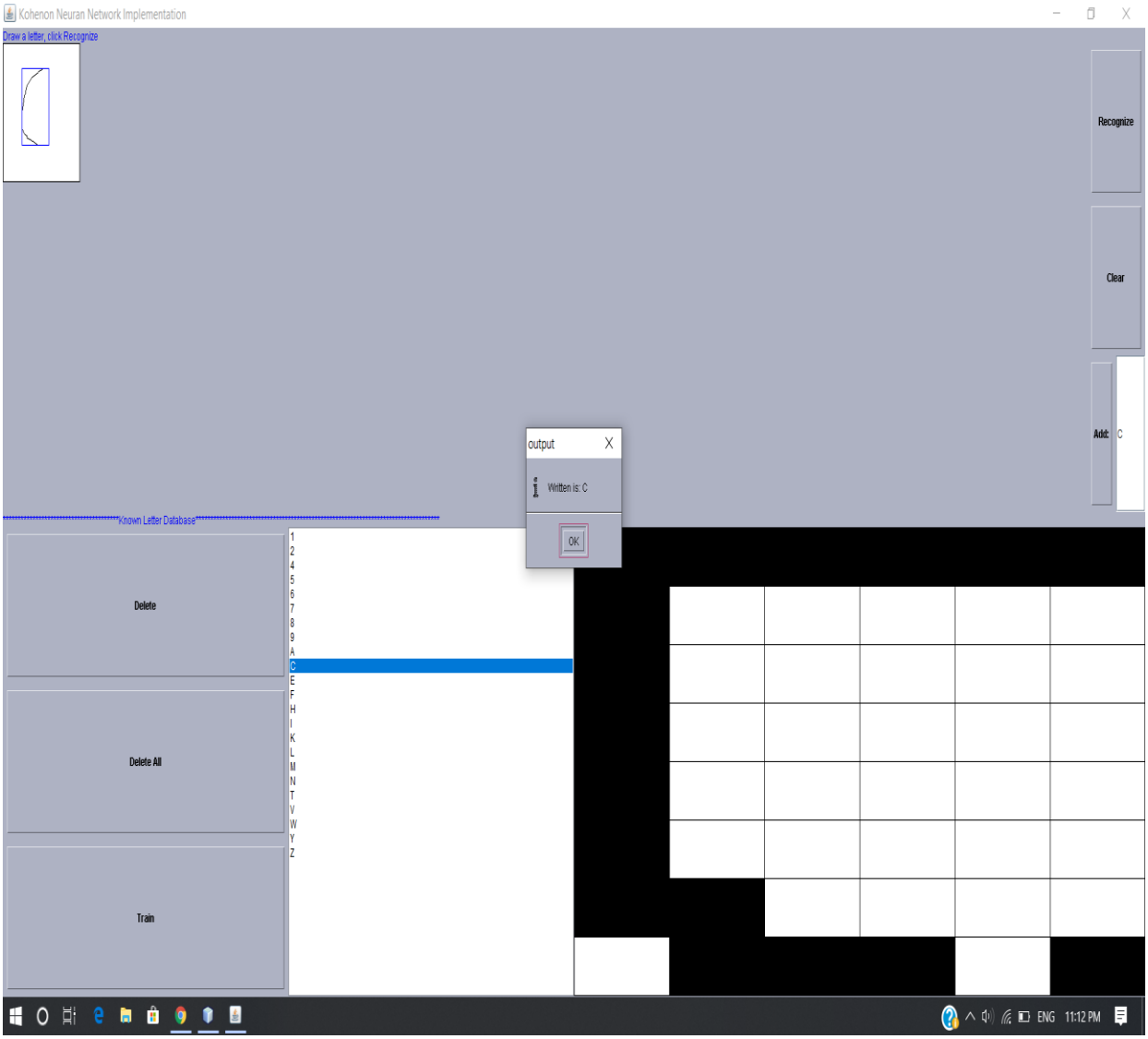

Figure: Recognizing character