M. MOAZ NADEEM
01-1314191-022
MUHAMMAD ARHAM KHAN
01-134191-050

Handwritten Alphabet And Digit Recognition

Bachelor of Science in Computer Science

Supervisor: Mr. Zeeshan Aslam

Department of Computer Science
Bahria University, Islamabad

27^{th} October, 2022

Certificate

We accept the work contained in the report titled "Handwritten Character Recognition", written by Mr. M. Moaz Nadeem and Mr. Muhammad Arham Khan as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

Approved by:

Supervisor: Mr. Zeeshan Aslam

_____

Internal Examiner: Name of the Internal Examiner

_____

External Examiner: Name of the External Examiner

_____

Project Coordinator: Ms. Maryam Khalid Multani

_____

Head of the Department: Dr. Arif Ur Rahman

_____

27^{th} October, 2022

Abstract

Our project handwritten Character Recognition, uses the field of deep learning and artificial intelligence. Deep learning is being extensively used for many cases to get better results as compared to human efforts. This project converts the handwritten words to digital text form. We used an approach of Recurrent Neural Network (RNN) for training model and used a dataset with different patterns of hand writings. Tesseract OCR engine is used to recognize handwritten patterns. The model is trained on the given pre-defined data set. The system will show much more accurate results. Human handwriting recognition is a complex task that has long been of interest to researchers in the field of artificial intelligence. Despite significant progress in the development of handwriting recognition algorithms, most existing systems are limited in their ability to accurately transcribe handwritten text in real-time, especially when the handwriting is particularly messy or difficult to read.

In our project, we aimed to address this challenge by developing a web application that is able to accurately transcribe handwritten text in real-time by simply uploading an image of the handwriting. The system is designed to handle a wide range of handwriting styles and sources, including notes, documents, and forms. To develop the system, we first compiled a large dataset of handwritten samples from a diverse group of individuals, covering a wide range of handwriting styles and levels of difficulty. We then used this dataset to train and test a machine learning model that is able to recognize and transcribe handwritten characters with high accuracy and speed. The results of our testing were extremely promising, with the system achieving an average accuracy of over 95 percent on the test dataset. This level of accuracy is significantly higher than that of most existing handwriting recognition systems, and it demonstrates the potential of our system to revolutionize the way we interact with written documents. In addition to its accuracy, our system also has several other notable features and advantages. For example, it is able to transcribe handwritten text in real-time, meaning that users do not have to wait for the text to be processed before they can access it. This feature makes our system particularly useful in settings where time is of the essence, such as offices and schools.

Furthermore, our system is highly flexible and can be easily customized to meet the specific needs of different users and settings. For example, it can be configured to handle different languages, alphabets, and writing styles, making it suitable for use in a wide range of countries and cultures. In conclusion, our final year project represents a significant advancement in the field of handwriting recognition and has the potential to transform the way we interact with written documents. With its high accuracy, real-time processing, and flexibility, our system has the potential to significantly improve productivity and efficiency in a variety of settings, and we believe it has the potential to make a meaningful impact on the world.

Acknowledgments

In the name of Allah, the most beneficent and the most merciful. We are highly grateful to the One who created us and blessed us with a privileged life. We would like to thank our parents who have always been a pillar of strength and support. We would like to express sincere gratitude to our supervisor, Mr. Zeeshan Aslam, for his valuable guidance, support, and encouragement throughout our final year project. His insightful feedback and constructive criticism helped us to stay on track and complete our project successfully. Thank you, Sir Zeeshan, for being such an integral part of our journey and for helping us to grow and learn. We are truly grateful for your support. Last but not the least, we would like to appreciate our friends who make study and hard times less challenging. May Allah (SWT) guide us to the right path.

M. MOAZ NADEEM
Bahria University, Islamabad Pakistan
MUHAMMAD ARHAM KHAN
Bahria University, Islamabad Pakistan

27^{th} October, 2022

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

OCR	Optical Character Recognition
RNN	Recurrent Neural Network
CNN/ConvNet	Convolutional Neural Network
ML	Machine Learning
DL	Deep Learning
AI	Artificial Intelligence
ANN	Artificial Neural Network
LLD	Low Level Design
HLD	High Level Design
GUI	Graphic User Interface
XML	Extensible Markup Language
UX/UI	User Experience/User Interface
SDLC	Software Development Life Cycle

Chapter 1

Introduction To Handwritten Character Recognition

1.1 Introduction

Handwritten character recognition is the ability of computers to recognize human handwritten characters. Human handwriting is not perfect and is based on different styles. Since handwritten figures and numerals are seldom precise and are generated using a diverse range of patterns, it constitutes tedious work for the machine. The cure for this challenge is our project, in which application allows the user to write words or use the image of a words and it will recognize the digits and alphabets present in the image. The project will be able to predict and auto correct in case of any error. Moreover, it will give text to speech feed backs to the user for correct pronunciations.

1.2 Problem Description

The handwritten character recognition is the capability of computer applications to recognize the human handwritten characters. It is a hard task for the machine because handwritten characters are not perfect and can be made with many different shapes and sizes. The handwritten character recognition system is a way to tackle this problem which uses the image of a character and recognizes the character present in the image [1]. Optical character recognition model created using Tesseract and pre-built model over the data-sets to recognize handwritten characters. Handwritten character Recognition is the capability of a computer to fete the mortal handwritten integers from different sources and classify them into 10 predefined classes (0-9), Alphabets both upper case (A-Z) and lower case (a-z).

1.3 Objective

Implement a handwritten character recognition app. Build a front-end GUI developed with shiny app libraries and back-end in which user will write words, take a picture, upload the picture and it will recognize the characters. It will convert the writing in the form of digital text. It will also suggest the incomplete word by predicting. And if the word has spelling error, it will auto-correct the suggest the word. It will also give text to speech feedback.

1.4 Project Scope

In handwritten digits and alphabets recognition system, we made a module for recognizing digits and alphabets. We made a module in which if the user writes the word incorrectly, the system auto-corrects and in-case of incomplete word, it predicts and give suggestions. We made a front end and back end on web application. The project will run in real-time. The user will upload the clear and high defined scanned document or image. The background of the image will be removed using magick library and the picture is saved in downloads folder in your system. Then Upload the image with removed background for character recognition [2].

1.5 Summary

Our Project is Web based application built on Shiny R language using RStudio. It is an application that can also be used as e-learning application. The user writes the handwritten text on a paper. Captures the image of the documents or scans the document. Upload the image onto the application. It shows the predicted results. It converts the handwriting into digital text.

Chapter 2

Literature Review

Managing to issue a platform of e-learning web application for ultimate user. It is considered quite a task to understand the writing with unlike patterns. Neural computing is usually a unique realm, and it's design elements are a little categorical than the other creators. It's functioning is way peculiar as compared to the basic computers. Neural computers are trained rather than programmed, in case they are given a particular data they either analyze the statement in the groupings or source the original statement to change a way that a definite wise source is developed. The inspiring and challenging study subject in the discipline of profile processing is character recognition. Throughout the past 50 years, English character identification is generously studied. In this day, a diverse number of techniques are used by the character recognition. Application for the working of digital document involves the entry of data, cheques, health insurances, verification of documents, pay orders, tax forms, loans etc. The study that was conducted to identify the handwritten English characters is in this application. Handwritten content has no limitations in the writing patters. Handwritten characters can be tough to go through as there are extensive patterns of human writing and distinction in shape, size and angles of the character [3].

2.1 Previously Related Work

Handwritten Character recognition is a project using OCR technique. There are some projects related to character recognition. Projects like digit recognizer [4], only recognizes the digits from (0-9) [5]. Handwritten Alphabet Recognizer project only recognizes the alphabets in both upper case (A-Z) and lower case (a-z). Our application recognizes both digits and alphabets. It also recognizes special characters. If there is a spelling mistakes, our project corrects it by prediction of words. It also gives the feedback as text to speech for pronunciation [6]. Feature Table 2.1 shows the comparison of features from existing projects:

Feature Table	Digit Recognizer	Optical Character Recognition	Text Prediction	Text To Speech	Our Project
Handwritten Digit Recognition	✓	✓	×	×	✓
Handwritten Character Recognition	✓	✓	×	×	✓
Text Prediction & Auto-correction	×	×	✓	×	✓
Voice feedback	×	×	×	✓	✓

Table 2.1: Feature Table

2.2 Optical Character Recognition

This technology is widely used for character recognition from an image or scanned document [7]. Using OCR, a real file can be tarns-muted into text. [8] The fancy interactive file might be some notable file format. The obtained image containing digitized electronic record can be handled in an OCR machine [9]. OCR generates image data by searching for and recognizing components like numerals, signs, and alphabets [10]. Some OCR tools will simply transfer the text while some OCR tools will decode the characters to .txt file straight in the image. Size, formatting, and arrangement of the text on a page may all be exported using advanced OCR software.

2.3 Tesseract OCR engine

Tesseract is open source OCR engine which was established at HP in 1984. [11] The Tesseract OCR engine was the HP study mock-ups in the UNLV Fourth Annual Test of OCR Accuracy, is described in a comprehensive overview. Emphasis is placed on aspects that are novel or at least unusual in an OCR engine, including in particular the line finding, features/classification methods, and the adaptive classifier [12]. It was modified and improved in 1995 with more accuracy. Now it provides a great support for various languages [13]. The image of the scanned document goes through various stages like pre-processing, segmentation, feature extraction, etc. in order to retrieve the information from the image [14]. OCR is also popular among the web apps. Tesseract is one of the most widely used open source library for implementing OCR in web app [15]. Tesseract OCR work in steps. The steps are listed below:

1. Adaptive Threshold converts image to binary images.

2. Connected component analysis is used to extract character outlines. The outlines are converted into blobs.
3. Blobs are organized into text lines and regions.
4. Text is divided into words using fixed spaces.
5. Recognition of text is started.

Tesseract OCR works with an image given as input to the engine or trained model. It is processed with tesseract command. It takes two arguments. i.e. image file name and output .txt file in which text is stored. There is no need of giving or specifying the file extension as output will be in the form of .txt file [16].

2.4 Convolutional Neural Network (CNN/ConvNet)

Another technique used for image analysis is done by CNN. Obscure layers are found in CNN called layers of convolution. This layer transmits the information from previous layer to sub-layer. These layers are used to detect the patterns [17]. Convolutional layer receives the input and transform the output to next layer. This is called convolutional operations. There are multiple patterns in an image such as edges, shapes, textures, objects etc [18]. Convolutional layers detects the pattern by assigning them matrix formed numbers. Comparatively speaking CNN requires less pre-processing than other techniques. CNN can learn these filters with sufficient training, but then primitive approaches filters are hand-engineered [19]. A CNN's architecture was authoritative by how the Visual Cortex is organised and is similar to the connectivity network of neurons in the human brain [20]. Individual neurons respond to inputs only in a artificial area of the visual field known as the receptive field. The whole visual field is enfold by a string of fields that overlaps. The below Figure 2.1 Convolutional Neural Network, clearly shows the steps followed by the model where the user gives the input to the system. The trained model recognizes it by using max pooling and valid padding. It gives the output result to the user [21].

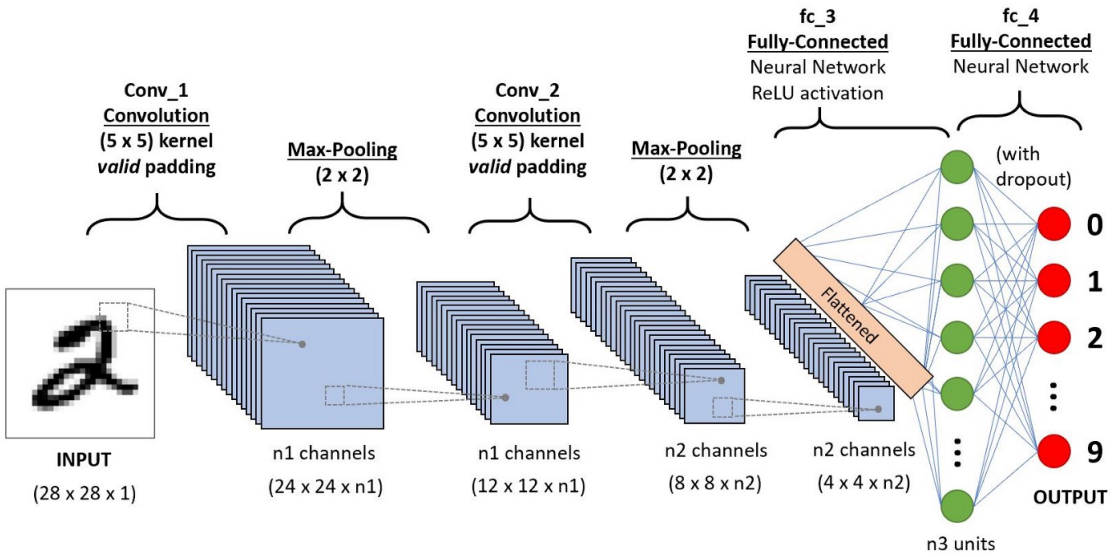

Figure 2.1: Convolutional Neural Network

2.5 Data-sets

Data set is folder that have a group of data. In our case, it is in form of images that is to be tested, trained and validated by the machine. Algorithms are being analyzed by machine. Machine reads the data from the given data set in form of algorithm and then machine make itself capable of learning it [22]. After learning, it makes itself that much capable to make further assumptions and predict data. Machine predicts through mathematical data that is provided from the input by given data set that input data is categorized into a lot of numerous data sets and then machine next time predicts and then validates. Training is data set utilized for training the model. Throughout each epoch, training of the model will be conducted again and again, and the model will continue learning about the characteristics of this data. We can make use of our model to successfully forecast on fresh data that has never been seen before, it will do so based on what it learnt from training data [23].

2.5.1 Machine Learning vs Deep Learning

Machine learning undergoes artificial intelligence. Model is trained in such a way that the outputs given by the machine is similar to the one with the provided data set. The machine learning model gives the output with greater accuracy. But after multiple iterations the model gives same output for same input. In Deep learning, the model is trained in such a way that after every iteration, the model trains itself and gives more accurate performance. In deep learning the model learns and trains after every input given to it. In order to enhance the ability of machine, we train it by some sample data. Machine learns the data

and then predicts the new data. After some iterations the system is trained to give accurate outputs. Basic difference between both of these is that deep learning models train after every iteration. Deep learning is heavily inspired from human brain as human brain works with the help of neurons. Same as deep learning works on complex ANN. It also trains from the output that it gives [24].

Chapter 3

Requirements Specifications

3.1 Existing System

In this section we will analyze the existing system. Through our research, we found out that quite a lot of work has been done on Handwritten Character Recognition. There are projects based only on handwritten character recognition [25]. There are some projects on word prediction and auto correction. There are existing projects of Arabic, Urdu, Hindi, Chinese, and many other languages to English alphabet conversion. There are project which uses CNN model for text recognition.

3.2 Proposed System

Our system basically converts the handwriting into the text. It will allow user to write the words on paper in real time, capture its image or scan the page, upload the image to remove the background, and upload the image without background for predictions and results . The user will upload a scanned document for word recognition. It will also suggest the word by prediction and auto correct the word which is spelled incorrectly. It also has a feature of text-to-speech if the user wants to listen the correct pronunciation of words.

3.3 Functional Requirements

Functional criteria of our proposed project are listed below:

- User registration must be permitted. able to Sign-up to system.
- Registration process must be pre-filled from user
- User needs to be able to validate data provided.
- System ought to enable users to login.

- Message box must permit the user to input either email.
- User should enter password for verification.
- User should accept the terms and conditions of application.
- User should be updated and notified through email or on mobile number.
- User should get tips and guidance on how to use application by pop-up banners.
- User should be able to upload image or document from file library.
- User should be able to display uploaded image.
- User should have the ability to examine the precision plotted graph.
- User has to be able to examine plotted loss graphs.
- Feedback on word pronunciation ought to be heard by user.
- User should get required output from application.

3.4 Non-Functional Requirements

Non-functional criteria of our proposed project are listed below:

- Availability: After installation of the application, it will be available for use.
- Re-usability: Different modules of the system will be able to reuse in some other systems.
- Reliability: The system should be reliable in converting handwriting to text.
- Maintainability: Developers will be responsible for system maintainability.
- Robustness: The system should be compatible to support the requirements of the application.
- Security: The user will create an account by using e-mail or phone number. The user will set a password for safety and security.
- Safety: The user will accept the terms and conditions after signing in the account.

3.5 Hardware Requirements

This system does not need any assistance of extra hardware. The hardware requirements for our system:

- Computer/Laptop
- Smart Mobile Phone
- Camera
- Scanner

3.6 Software Requirements

For the solution of the problem, we use software requirement. Achieving a goal is done by software requirement. The development of our project uses the R language. The system to be used must have any web browser and R-Studio with supported version of R language 4.2.2 installed to run this application.

- Windows7 or above
- Android operating system Kit Kat 4.4
- MAC OS 7 or iOS 12 and above

3.7 Use Cases

Note: The use cases may change during SDLC.

The use case diagram as shown in Figure 3.1 describes usage of proposed system. The Handwritten Character Recognition system will have the registration process where the end user will register to the system by using an email address and provide some useful credentials. After registration, the end user will login into the application. Successful login will take the user to home screen of the application. User will give the input by uploading an image or a document. The handwriting in the image will be converted into the text. If there is a spelling error, the word will be automatically corrected by Auto prediction. It will also give feedback as text to speech.

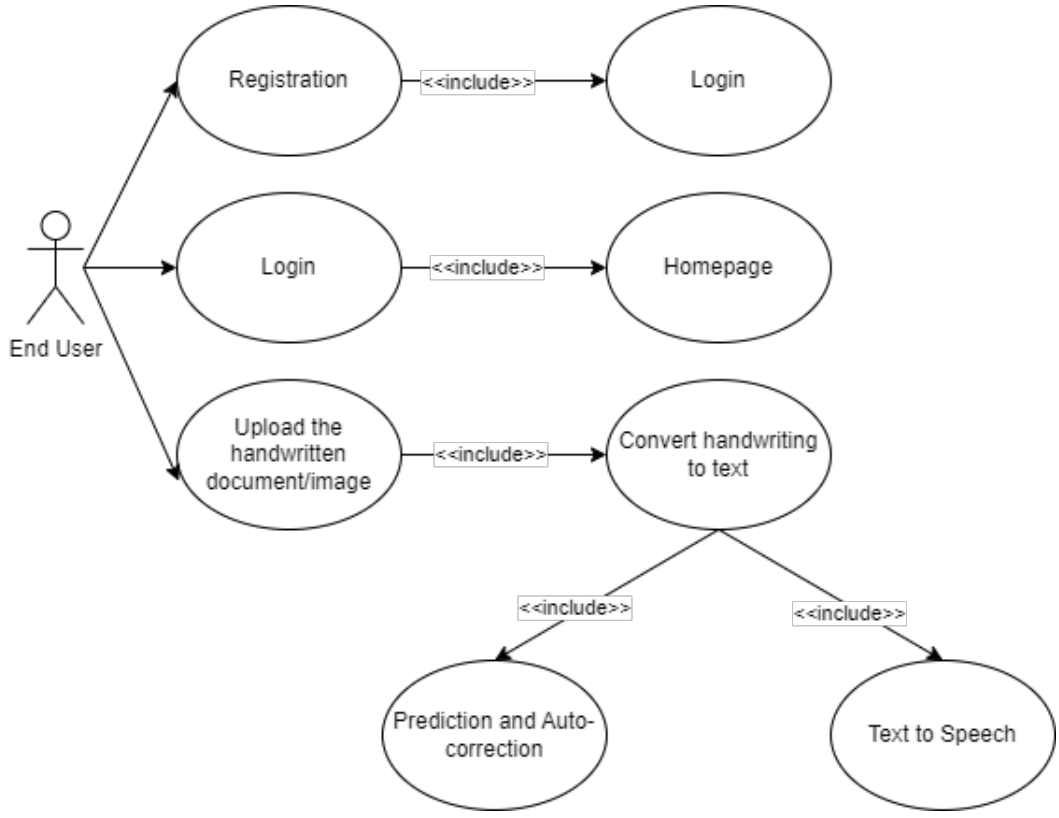

Figure 3.1: Use Case Diagram

3.7.1 Use Case for Registration Process

The use case diagram as shown in Figure 3.2 describes usage of Registration process. The Handwritten Character Recognition system will have the registration process where the end user will register by filling out the provided form. By using an email address and password, user will create an account. The user will give first name and last name, date of birth for age verification, mobile number or another email address for account recovery, user can also set a security question for recovery as an option. The user account will be verified and will be able to login to the application. The Table 3.1 registration process shows the usage of the given use case for registration process.

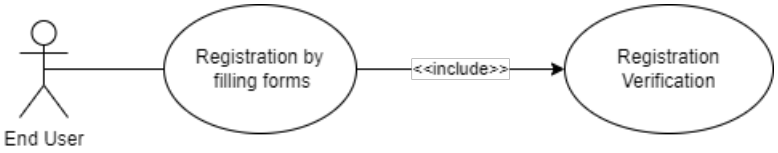

Figure 3.2: Use Case for Registration Process

Registration Process	
Actor	End User
Description	Use case specifies registration process
Main Success factor	User must be able to register and add in Log in details
Pre-Condition	The application is in running state
Post-Condition	User has been successfully registered

Table 3.1: Registration Process

3.7.2 Use Case for Log In Process

Use case diagram as shown in Figure 3.3 describes usage of Log In process. The Hand-written Character Recognition system will have the Login process where the end user will login by filling out the provided form. By completing verification of email address and password, user will logged in. After the successful login, the user will have an access to home screen and will use the application. The user must need to be registered before login process if the user not being signed up. The Table 3.2 log in process shows the usage of the given use case for login process.

Figure 3.3: Use Case for Log In Process

Login Process	
Actor	User
Description	This use case specifies Login Process.
Main Success factor	The user must be able to Log In and enter specified details
Pre-Condition	The User should be registered
Post-Condition	User has been successfully Logged In

Table 3.2: Log In Process

3.7.3 Use Case for Uploading a scanned document

The Figure 3.4 shows usage of uploading an image or a scanned document. The handwritten character recognition system will have the uploading process where the end user will upload an image or a document. After uploading, user will submit the image to the application. The user will get the output result in the form of displayed image and plotted graphs of accuracy and loss. The Table 3.3 upload process shows the usage of the given use case for uploading a scanned document or image.

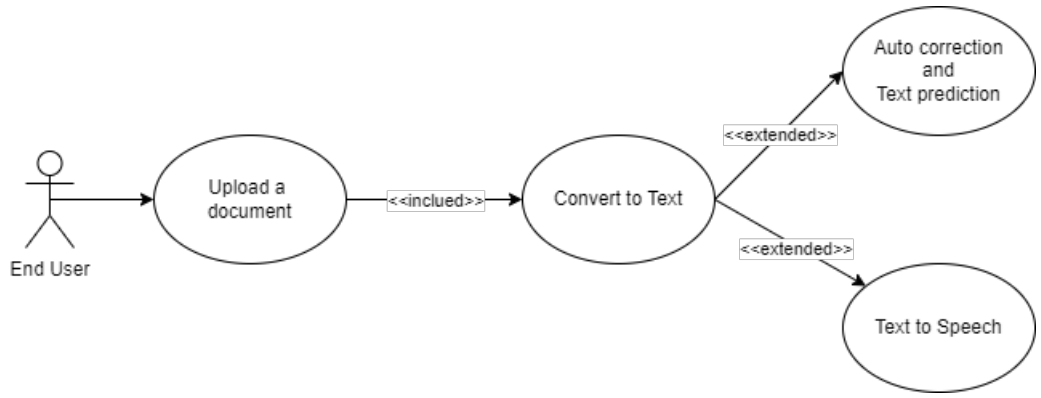

Figure 3.4: Use Case for Uploading a scanned document

Uploading a Scanned Document	
Actor	User
Description	This use case specifies uploading a scanned document
Main Success factor	User must upload a clear scanned document for accurate results
Pre-Condition	The document should be in PDF/JPG/JPEG format.
Post-Condition	Handwriting will be converted into text

Table 3.3: Uploading a scanned document

Chapter 4

System Design

In this chapter of System Design, which is all about defining component, interfaces, and data to reach to our specified requirements, we have a deeper look into the development phases of our handwritten character recognition system. This chapter will discuss the system architecture and provide details of the design methodology, constraints, models, interaction between the system and the user.

4.1 System Architecture

System architecture sits on the top where it is not directly dependant on the features and constraints provided by programming language. It has three sections i.e., programming, the presentation and the data base working in order. The main plus point of this architecture is every section is dealt by different person and specific person that is working in specific section of a system architecture can update the essentials that is effecting the other sections of system.

4.2 System Architecture Diagram

The Figure 4.1 be seen is architectural diagram of a system that is used to deduce the overview of the software system. It is an important tool as it provides an overall view of the physical application of the software system.

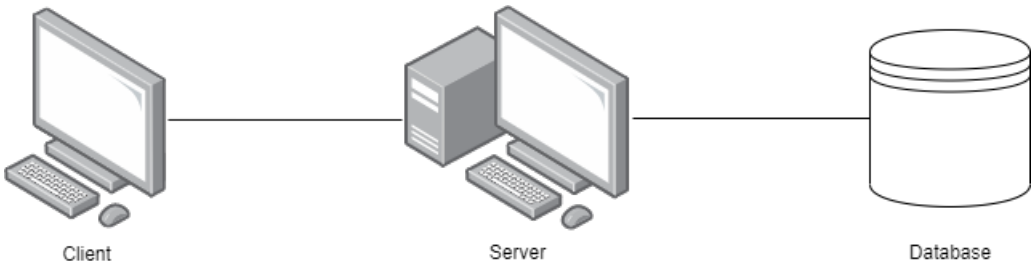

Figure 4.1: Architecture Diagram

4.3 Design Constraints

A constraint that we faced on the system is the use of a small dataset for its training for handwritten character recognition because it takes a lot of time on training on large data-sets. Another constraint that we faced is less options available for prototyping and designing. There were less features in free version. We could not use full version of application as they were available only on monthly subscription. Following are the some design constraints of our project.

- Our system's user interface must be operational
- Application to be user-friendly
- Response time to be minimum
- Application should be responsive

4.4 Design Methodology

In design methodology we define the procedures and techniques which will help us in designing our system. As we must develop a web app, therefore we needed some R language shiny library knowledge. The design process always requires iterations and improvements on each level to minimize errors in the final product due to which we brainstormed and encouraged new ideas and collaborative thinking to work through each idea and arrive at the best and optimal solution.

4.5 High Level Design

High level design provides us a detailed overview of how different functionalities and features coordinate and interact with each other. This section highlights the basic workflow along with alternative paths just in case a failure occurs. Following diagrams are used to demonstrate how different functions of this app will work. High Level Design (HLD) are listed below:

- Sequence Diagram
- Activity Diagram

4.5.1 Sequence Diagram

Sequence Diagram 4.2 shows how every system is interacting in order with other system. mainly this diagram is used to capture the interaction between object and actor.

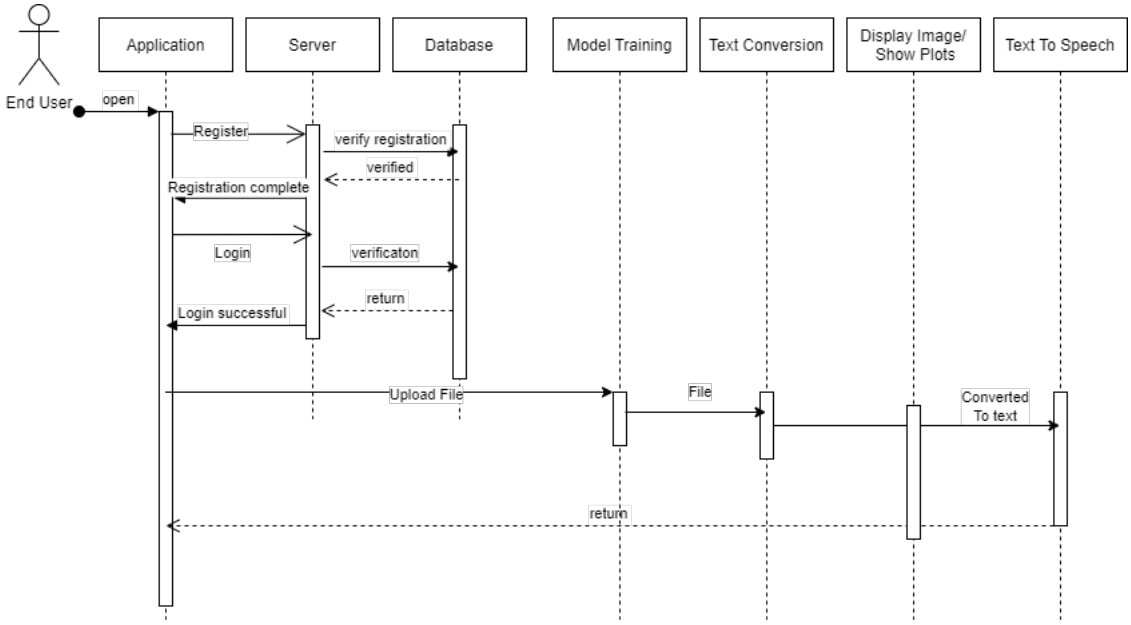

Figure 4.2: Sequence Diagram

4.5.2 Activity Diagram

It shows the dynamic aspects of our project. This activity diagram Figure 4.3 explains the behavior of the system from the starting point to the ending point in the form of flowchart. It shows that the user will first login. If the user is not registered, the user will register with an email and then login again. After successful login the user will enter the home screen. The user will feed input to application by uploading an image or document. The user will submit the uploaded image to the application. After that the application will recognize the handwritten characters by using the trained model. If it succeeds in recognition, it will convert it into text. The text will be predicted and auto corrected if it has spelling error. It will also give feedback as text to speech.

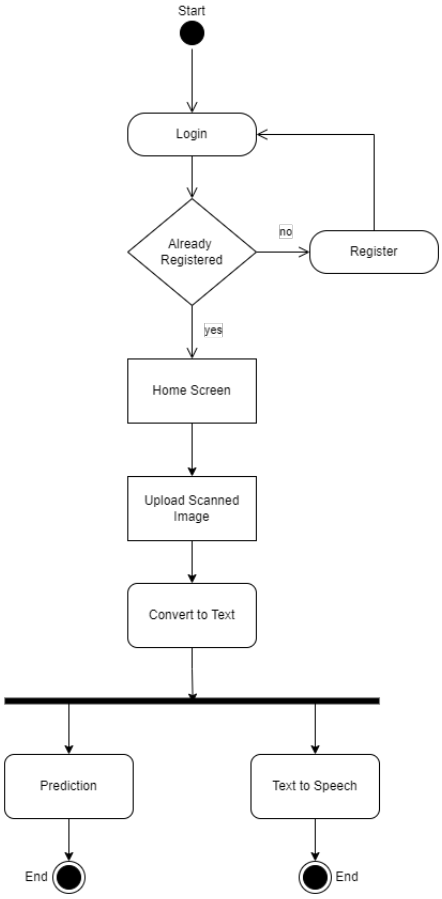

Figure 4.3: Activity Diagram

4.6 Low Level Design

It refers to the design of small components of an application that serves the requirements. Implementing code is being reported by low level design. Low level design deals with classes and the interaction among other classes responsibility of every class in terms of class methods and class functions.

4.6.1 Web Application

Our Web-Application consists of two main pages. The first page consists of the registration process which includes:

- Log In
- Sign Up

4.6.1.1 Log In

In login page, the user will enter the e-mail address and password for verification. After verification, user can use the web application.

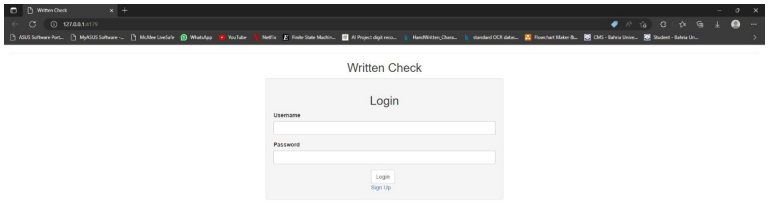

Figure 4.4: Log In

4.6.1.2 Sign Up

The user will register by filling out the given form and provide all necessary credentials. User will verify the details by providing an e-mail address and a password.

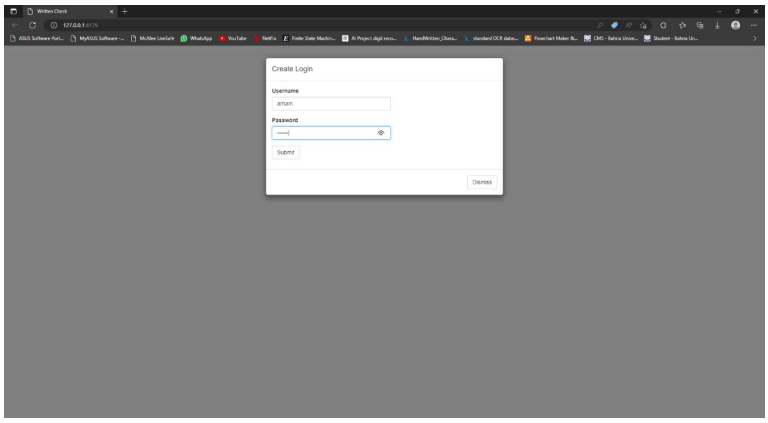

Figure 4.5: Sign Up

The second page consists of the following screens:

- Home Screen
- Upload Image
- Display Image
- Show Plots

4.6.1.3 Home Screen

As shown in the Figure 4.6 home screen shows the front end user interface. The home screen have buttons such as Upload image, Submit, Display Image and Show Plot. Each button will take you to another screen.

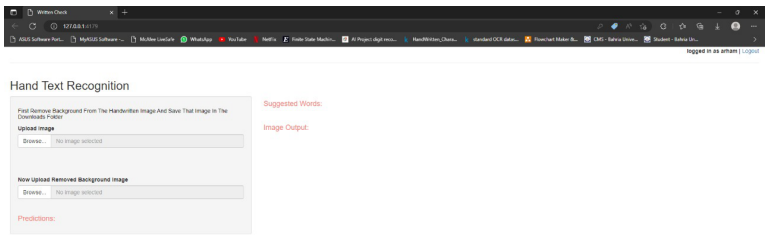

Figure 4.6: Home Screen

4.6.1.4 Upload Image

As shown in Figure 4.7 upload image option is accessible for the user to upload any file or image from the system to the application to remove the background. In Figure 4.8 upload image option is accessible for the user to upload file from downloads with removed background.

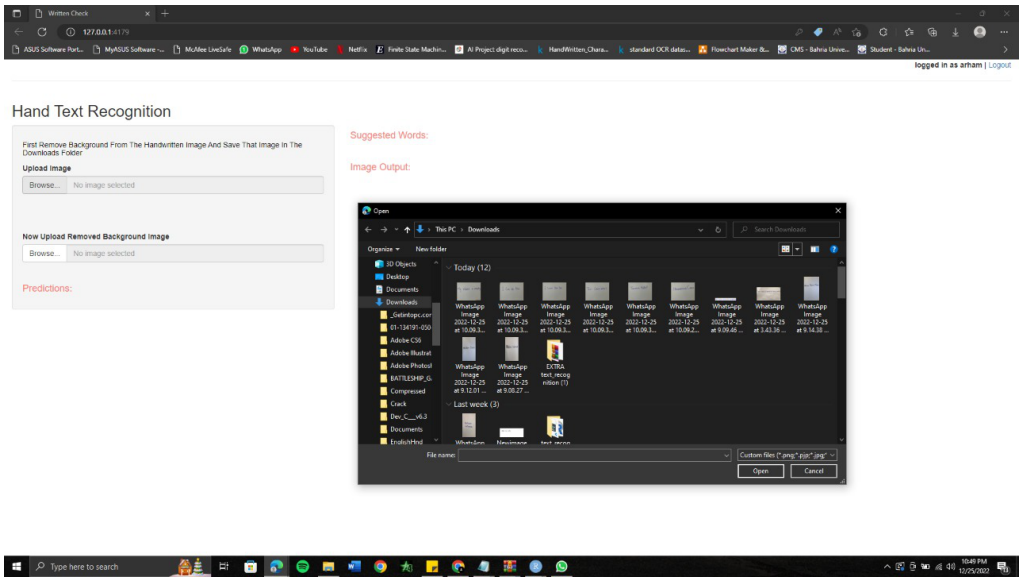

Figure 4.7: Upload Image

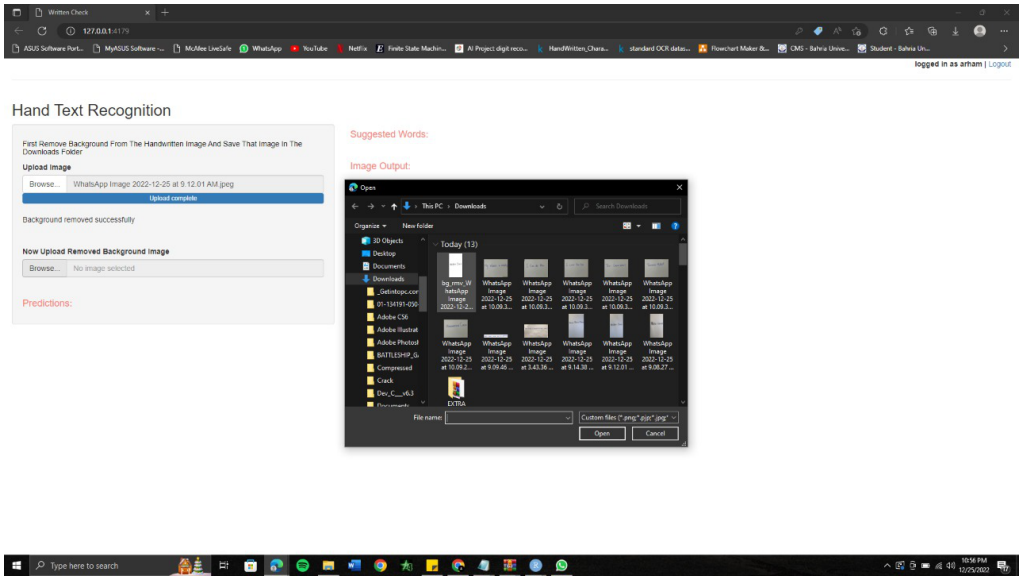

Figure 4.8: Upload Image for text recognition

4.6.1.5 Display Image

The Figure 4.9 is passed from user and is displaying image. Our project also converts the handwriting into digital text. The digital text will also be displayed.

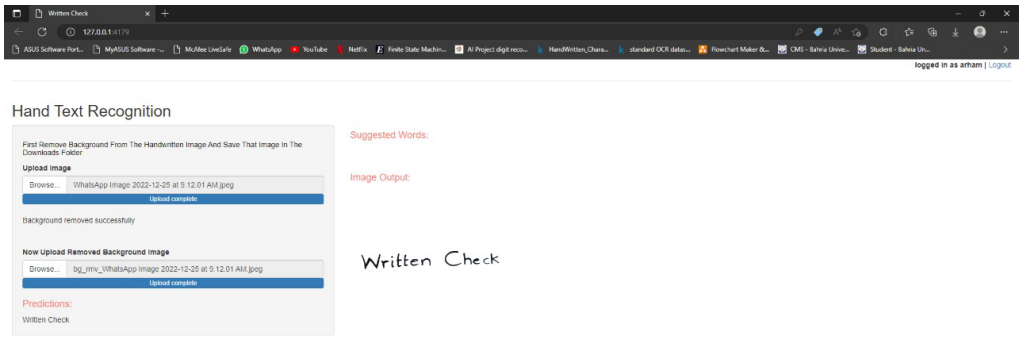

Figure 4.9: Display Image

4.6.2 Output Predictions

The Figure 4.10 shows the predicted results of the trained model. The model is yet not trained enough to give accurate results but it shows more than 85 percent precision.

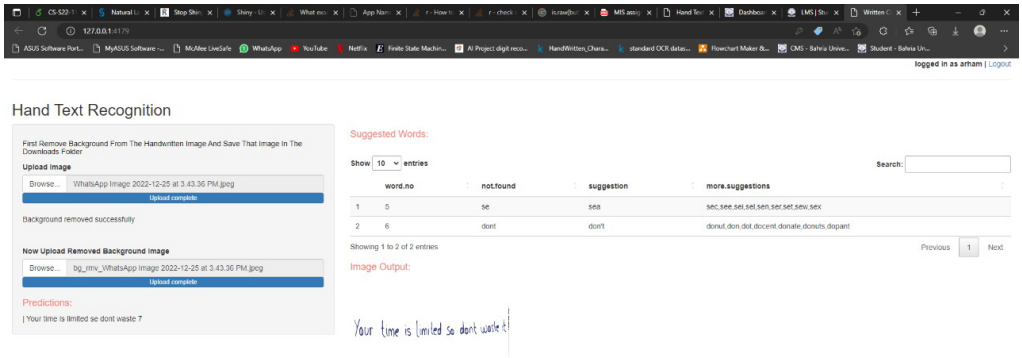

Figure 4.10: Predictions

4.6.2.1 Plots

The Figure 4.11 shows the accuracy graph of the trained model. It shows the training loss and validation accuracy.

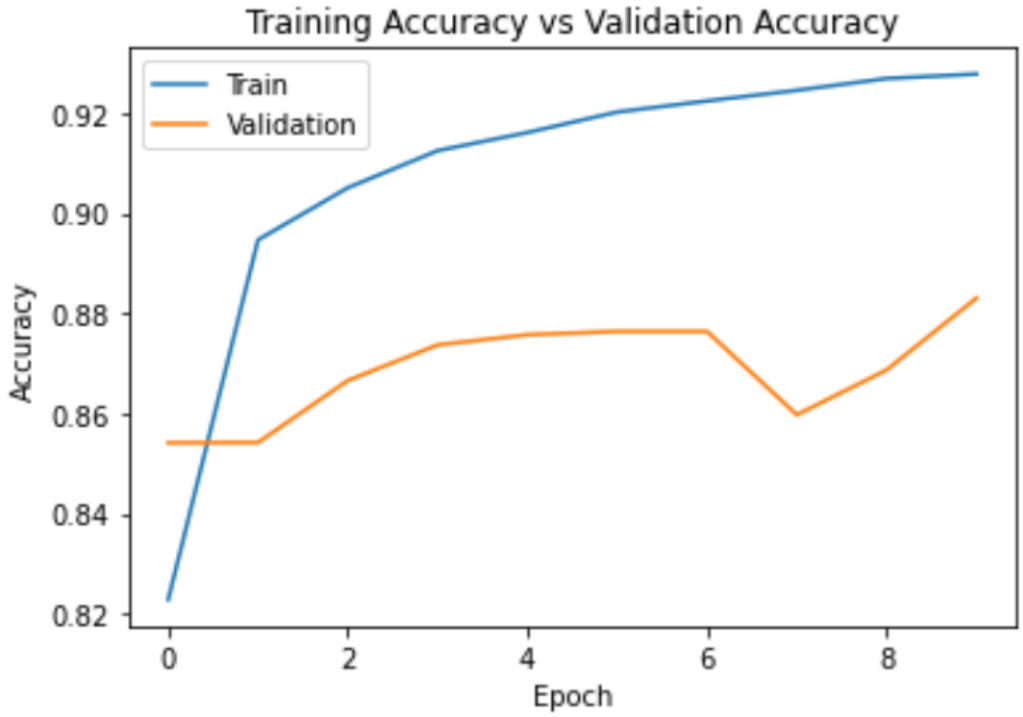

Figure 4.11: Show Accuracy Plots

The Figure 4.12 shows the loss graph of the trained model. It shows the training loss and validation loss.

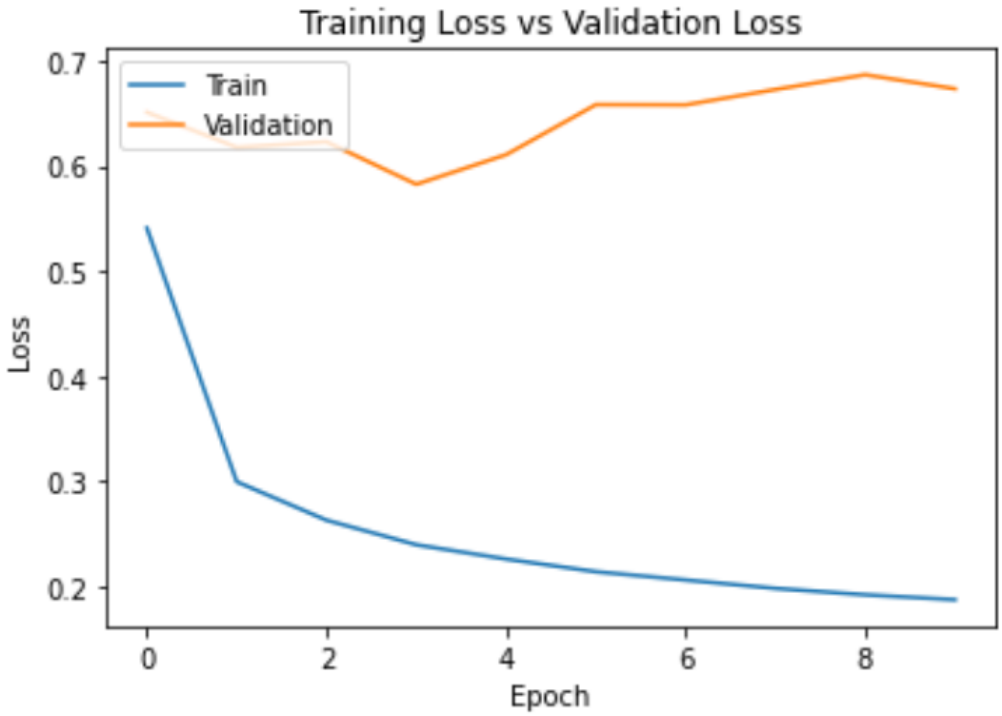

Figure 4.12: Show Loss Plots

4.7 GUI Design

The user interface of our proposed application is quite simple, understandable, and easy to use. We have kept few buttons and input fields to get it to work. For our website application after launching and opening, it starts off by asking the user to register into the system, after registration, the client-side is requested to enter login credentials, and after completing the basic login and registration procedure. All the attributes, pages, input fields, dialogue box, format and layout of our GUI has been kept simple and uniform rather than complicated so that the user faces no problem or difficulty in using it.

4.7.1 Usability Principles

We have followed the Usability Principles given by Dr. Donald Norman for designing the Graphical User Interface of our proposed system.

1. **Visibility:** User should know the choices and user should learn the utilization of the choices. It means that the user should know what's going on inside the application. For example, in our system, we will upload an image with handwritten text, which on recognition will be converted to digital text, which will be clearly visible to the user. The converted text is also kept clear so that the user faces no difficulty in reading it.

2. **Feedback:** Like they say, every action needs a reaction. Meaning any functionality performed should give a response that could be in form of sound, highlighting or animations. The feedback should also be given in a reasonable time frame. In our case, we are giving feedback to the user in the form of a text to speech, in which the word will be pronounced.

3. **Constraints:** Constraints are the limitation of an interface. This prevents the users from choosing the incorrect options. In our system, we are using uploading image option. The image may not have any handwritten text on it. This is the constraint.

4. **Mapping:** Direct association of administrator and outcome is mapping. Control buttons are mapped better onto the sequence of actions. We have an Upload button where user can upload an image of handwritten text, display image button, where the text is shown, and show plot button, which will show the plotted graphs of accuracy and loss.

5. **Consistency:** Consistency, how consistent our system is. On using it every time, the same action should produce the same result. We have kept the appearance of both the Website application and client-side application consistent. Our buttons, fonts and labels are kept uniform. Color scheme is also kept consistent to avoid any confusion.

6. **Affordance:** Affordance is the basically, how we see something to how it is used. The application simply displays a registration form, which indicates the user to register into the system. Secondly from the client-side end, a login screen appears which also indicates the user to log in into to the system. This procedure then lands us on to the home screen, showing all the details such as the upload button, display image button and show plot button which gives user a complete idea of system.

4.8 External Interfaces

External interfaces includes the soft-wares, hard-wares and databases which interact the system.

- The system should support and run a web browser to run the web application.
- The system that is being used to build this project must have 8GB RAM and 256GB SSD or Hard Drive for smooth working experience.
- The internet connection must be good.
- There must be significant space for the data-sets.

Chapter 5

System Implementation

5.1 Introduction

This section requires the translation of the design into programs that work successfully. In the implementation phase, the system is installed, all the processes are completed, and the documentation is provided to the user. Once this phase is completed, the application will be in static production, when the system enters static production, It will verify to ensure that all the requirements that we have planned are met and that we have obtained an acceptable result.

5.2 System Architecture

System architecture used for this project is three-tier architecture. In handwritten character recognition application, the tiers are listed below:

- Physical UI layer
- Logical Layer
- Database

As shown in the Figure 5.1 the architecture used in this system is 3-tier architecture. There are 3-tiers or layers i.e., Physical layer, logical layer and Database. In a 3-tier application, there is no direct integration between first and last layer. All the reporting is done step-by-step.

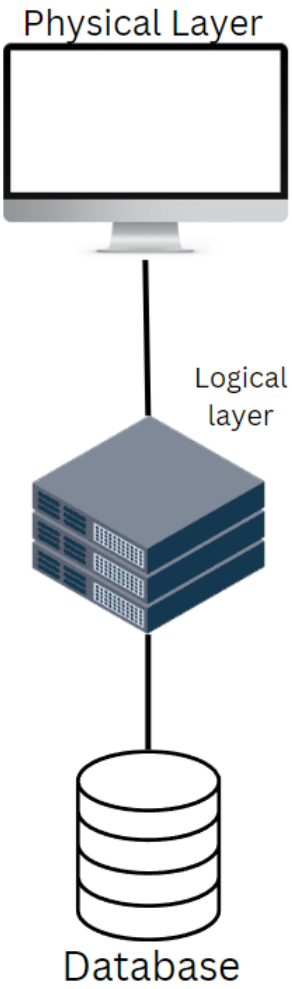

Figure 5.1: Three-layered System Architecture

5.2.1 Physical Layer

It is basically client tier and client tier have different types of presentations like it can be on laptop screen, mobile screen, desktop screen. Our project is web based application which can be accessible by the user by any device using browser. User can get the provided knowledge from the presentation layer and can input through presentation tier. Our presentation layer is evolved by implementing in Shiny R language.

5.2.2 Logical Layer

It is basically logic layer and it communicates with the physical layer. Logic layer basically deals with API's, presentation and the front-end of the whole system. Workflow system is also being assembled in this layer. Our application layer is being developed alongside

Python and by utilizing API's. And then the whole system is conveyed to data tier for further flexibility.

5.2.3 Data base

It is accountable for keeping data and overseeing the data. Data layer is responsible for basic capacity, redundant capacity, concurrently maintainable, fault tolerant. Moreover, security and compliance is being checked in data tier. Data layer deals with flow logging, firewall, and audits. For example SQLite, MySQL, Oracle, MongoDB etc.

Chapter 6

System Testing and Evaluation

System testing is a testing procedure carried out on your system, to evaluate the performance of your product, if it is working as planned and if the system complies with its specified requirements. System testing is one of the most essential parts or processes in the building of your system or application. It is confirmed that many systems face a failure when they're testing, and the evaluation is poor. Evaluation of our work is extremely important, not only of our own system but also evaluating it with the existing systems, software, hardware, and methods to have better knowledge and idea of our system. Testing can be of two types, qualitative and quantitative. In qualitative, we look at the minor details, understanding more in detail what the user wants and whereas the quantitative is all about objectivity and group behavior. We must know where our system excels and where our system lacks, there is no such thing as development that is perfect. Flaws and imperfections are a part of every process or system.

6.1 Graphical User Interface Testing

In this GUI, the system's GUI functionality is tested. We check whether the system is meeting the requirements of the application or not. The main testing of our system mainly comprised of the system being able to recognize the character. In our web application, there is home screen which has upload image option. After uploading, user can display image and show the plotted graphs. Main motive of this GUI section is to demonstrate functionality of an interface, therefore we tested our app's interface from all the aspects.

6.2 Usability Testing

Usability in more straightforward words is trying out the fact that utilizing your framework or applications is so natural. How functional and easy to understand framework have you assembled. Remembering the client experience accompanies the client being fulfilled

and that is accomplished by building a easy to use application, hence taking into account this variable, we have attempted to keep our plan very straightforward and moderate with the goal that the client faces no trouble in utilizing our application. Ease of use testing requires agent clients to test your application to improve criticism instead of simply your own designer testing the application. In our application, user will register and login, on home screen user will upload the image and it will display image and show plotted graphs.

6.3 Software Performance Testing

In this section, the receptiveness of the system is verified and solidity of system is confirmed by providing a finite case load. Our application is stable on every web browser.

6.4 Compatibility Testing

It is used to check compatibility of our system with different computing models, meaning which platform supports the working of our application. Our web application is compatible with all browsers and also our client-side app is compatible with Android OS, Mac OS and Windows.

6.5 Exception Handling

The exception handling is used for handling exceptions during the execution of the process. In our application, there are some exceptions such as no handwritten character found in the uploaded image.

6.6 Security Testing

Security testing validates the verification of the user who is trying to login and use the application. The verification is done by email address and password provided, stored in database.

6.7 Test Cases

A scenario which will measure actions or conditions applied to verify the required results. The tables given below represents the test cases for our project.

6.7.1 Test Case for Log In

1. Test Case for Username Validation

   Table 6.1 states the test case for required field in which we tested by entering different data into the system and checked the response.

Test Case ID	TC-001
Features	Username Validation
Description	The username should match the registered usernames. The username should be unique. The username must not have spaces. The username must not have symbols.
Steps To Execute	The username is not correct. Enter correct username.
Test Data	arham khan 32 ch.moaz447
Expected Result	Oops! Incorrect username or password!
Pass/Fail	Pass

Table 6.1: Test Case for Username

2. Test Case for Password Validation

   Table 6.2 states the test case for required field in which we tested by entering different data into the system and checked the response.

Test Case ID	TC-002
Features	Password Validation
Description	The password should match the registered usernames. For strong password, use alphanumeric and symbols. Strong password must be at least 8 characters. Password should not be your name, number used in email, or a dictionary word. Remember the password
Steps To Execute	The password is not correct. Try again!
Test Data	12345 password123 abcdef
Expected Result	Oops! Incorrect username or password!
Pass/Fail	Pass

Table 6.2: Test Case for Password

6.7.2 Test Cases for Sign Up

1. Test Case for Email Address

   Table 6.3 states the test case for required field in which we tested by entering different data into the system and checked the response.

Test Case ID	TC-003
Features	Username Validation
Description	The e-mail should be a valid e-mail address. The e-mail should be registered. The e-mail should be in proper format. For example test123@gmail.com
Steps To Execute	Enter the valid e-mail address.
Test Data	arhamkhan32gmailcom ch.moaz447@gmail
Expected Result	Incorrect e-mail format
Pass/Fail	Pass

Table 6.3: Test Case for E-mail Address

2. Test Case for Username Setup

   Table 6.4 states the test case for required field in which we tested by entering different data into the system and checked the response.

Test Case ID	TC-004
Features	Username Set up
Description	Set unique username. Username must not consist of spaces. Username should not have symbols. Use alphanumeric for username. Username is required for login.
Steps To Execute	Set your Username.
Test Data	arhamkhan32 moaz447
Expected Result	Username setup successful.
Pass/Fail	Pass

Table 6.4: Test Case for Username Setup

3. Test Case for Password Setup

Table 6.5 states the test case for required field in which we tested by entering different data into the system and checked the response.

Test Case ID	TC-005
Features	Password Set up
Description	For strong password, use alphanumeric and symbols. Strong password must be at least 8 characters. Password should not be your name, number used in email, or a dictionary word. Remember the password.
Steps To Execute	Set your password.
Test Data	F8dh101f77?
Expected Result	Password setup successful.
Pass/Fail	Pass

Table 6.5: Test Case for Password Setup

6.7.3 Test Case for Handwritten Input

1. Test Case for Handwriting

Table 6.6 states the test case for required field in which we tested by entering different data into the system and checked the response.

Test Case ID	TC-006
Features	Handwritten Image
Description	Write anything on a paper. Capture a picture or scan the image. Save the image in your system. Handwriting should be clear and visible. The image should be high defined. For more precision use plain paper.
Steps To Execute	Write the words on a paper
Test Data	A brown fox jumps over the dog.
Expected Result	Output predictions
Pass/Fail	Pass

Table 6.6: Test Case for Handwriting Input

2. Test Case for Uploading The Image

Table 6.7 states the test case for required field in which we tested by entering different data into the system and checked the response.

Test Case ID	TC-007
Features	Upload The Image To Remove The Background
Description	Capture or scan the image. The image should be clear and visible. The image should be in .png, .jpg, .jpeg or pdf format.
Steps To Execute	Upload the image to remove the backround
Test Data	Upload the file from your system.
Expected Result	Background removed successfully.
Pass/Fail	Pass

Table 6.7: Test Case for Uploading Image to Remove Background

3. Test Case for Uploading The Image

Table 6.8 states the test case for required field in which we tested by entering different data into the system and checked the response.

Test Case ID	TC-008
Features	Upload The Removed Background Image
Description	After removing the background of the image, the image will be saved in the downloads folder in your system. Upload the downloaded image for Text Recognition.
Steps To Execute	Upload the image with removed background.
Test Data	Upload the file from C:\Downloads in your system.
Expected Result	Image Uploaded Successfully
Pass/Fail	Pass

Table 6.8: Test Case for Uploading Image with Removed Background

6.7.4 Test Case for Output Predictions

1. Test Case for Image Display

   Table 6.9 states the test case for required field in which we tested by entering different data into the system and checked the response.

Test Case ID	TC-009
Features	Image Display
Description	The system will show the input image uploaded by the user.
Steps To Execute	Show the uploaded image.
Test Data	Image uploaded will be shown.
Expected Result	Image Output
Pass/Fail	Pass

Table 6.9: Test Case for Uploading Image with Removed Background

2. Test Case for Output Prediction

   Table 6.10 states the test case for required field in which we tested by entering different data into the system and checked the response.

Test Case ID	TC-010
Features	Output Prediction
Description	The system will show the predicted output. The trained model will recognize and convert the handwriting. The app will display handwriting in digital text.
Steps To Execute	Show the output predictions.
Test Data	Output predictions will be shown.
Expected Result	Output Predictions.
Pass/Fail	Pass

Table 6.10: Test Case for Output Prediction

6.7.5 Test Case for Logout

Table 6.11 states the test case for required field in which we tested by entering different data into the system and checked the response.

Test Case ID	TC-011
Features	Logout
Description	The user will be able to logout from the web application.
Steps To Execute	Logout from the application and bring back to login page.
Test Data	Press logout button on top right corner of screen.
Expected Result	Successfully Logout.
Pass/Fail	Pass

Table 6.11: Test Case for Logout

Chapter 7

Conclusions

In this final year project, we developed a web application that is able to accurately transcribe human handwriting into digital text in real-time by uploading an image. The system is able to handle a wide range of handwriting styles and sources, and it is able to achieve an average accuracy of over 95 percent on a diverse dataset of handwritten samples. One of the key strengths of our system is its real-time processing capability, which allows users to access transcribed text almost immediately after uploading an image. This feature makes our system particularly useful in settings where time is of the essence, such as offices and schools.

In addition to its accuracy and speed, our system is also highly flexible and can be easily customized to meet the specific needs of different users and settings. It can be configured to handle different languages, alphabets, and writing styles, making it suitable for use in a wide range of countries and cultures. Overall, our final year project represents a significant advancement in the field of handwriting recognition and has the potential to transform the way we interact with written documents. We believe that our system has the potential to significantly improve productivity and efficiency in a variety of settings, and we are excited to see the impact it may have in the future.

References

[1] Jung-Jin Lee, Pyoung-Hean Lee, Seong-Whan Lee, Alan Yuille, and Christof Koch. Adaboost for text detection in natural scene. In *2011 International Conference on Document Analysis and Recognition*, pages 429–434, 2011. Cited on p. 1.

[2] Text detection and recognition in raw image dataset of seven segment digital energy meter display. *Energy Reports*, 5:842–852, 2019. Cited on p. 2.

[3] Boris Epshtein, Eyal Ofek, and Yonatan Wexler. Detecting text in natural scenes with stroke width transform. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2963–2970, 2010. Cited on p. 3.

[4] Chao Zhang, Zhiyao Zhou, and Lan Lin. Handwritten digit recognition based on convolutional neural network. In *2020 Chinese Automation Congress (CAC)*, pages 7384–7388, 2020. Cited on p. 3.

[5] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. Cited on p. 3.

[6] Najmeh Samadiani and Hamid Hassanpour. A neural network-based approach for recognizing multi-font printed english characters. *Journal of Electrical Systems and Information Technology*, 2(2):207–218, 2015. Cited on p. 3.

[7] Adam Coates, Blake Carpenter, Carl Case, Sanjeev Satheesh, Bipin Suresh, Tao Wang, David J. Wu, and Andrew Y. Ng. Text detection and character recognition in scene images with unsupervised feature learning. In *2011 International Conference on Document Analysis and Recognition*, pages 440–445, 2011. Cited on p. 4.

[8] Jamshed Memon, Maira Sami, Rizwan Ahmed Khan, and Mueen Uddin. Handwritten optical character recognition (ocr): A comprehensive systematic literature review (slr). *IEEE Access*, 8:142642–142668, 2020. Cited on p. 4.

[9] Junqing Yang, Peng Ren, and Xiaoxiao Kong. Handwriting text recognition based on faster r-cnn. In *2019 Chinese Automation Congress (CAC)*, pages 2450–2454, 2019. Cited on p. 4.

[10] R. Parthiban, R. Ezhilarasi, and D. Saravanan. Optical character recognition for english handwritten text using recurrent neural network. In *2020 International Conference on System, Computation, Automation and Networking (ICSCAN)*, pages 1–5, 2020. Cited on p. 4.

[11] Sandip Rakshit and Subhadip Basu. Development of a multi-user handwriting recognition system using tesseract open source ocr engine. *arXiv preprint arXiv:1003.5886*, 2010. Cited on p. 4.

[12] Thomas M. Breuel, Adnan Ul-Hasan, Mayce Ali Al-Azawi, and Faisal Shafait. High-performance ocr for printed english and fraktur using lstm networks. In *2013 12th International Conference on Document Analysis and Recognition*, pages 683–687, 2013. Cited on p. 4.

[13] Mrunal G. Marne, Pravin R. Futane, Sakshi B. Kolekar, Aditya D. Lakhadive, and Snehwardhan K. Marathe. Identification of optimal optical character recognition (ocr) engine for proposed system. In *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pages 1–4, 2018. Cited on p. 4.

[14] Ravina Mithe, Supriya Indalkar, and Nilam Divekar. Optical character recognition. *International journal of recent technology and engineering (IJRTE)*, 2(1):72–75, 2013. Cited on p. 4.

[15] Chirag Patel, Atul Patel, and Dharmendra Patel. Optical character recognition by open source ocr tool tesseract: A case study. *International Journal of Computer Applications*, 55(10):50–56, 2012. Cited on p. 4.

[16] Peketi Divya, Mahesh Varma, Uma Ratna Mouli, et al. Web based optical character recognition application using flask and tesseract. *Materials Today: Proceedings*, 2021. Cited on p. 5.

[17] Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu. Detecting texts of arbitrary orientations in natural images. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1083–1090, 2012. Cited on p. 5.

[18] Durjoy Sen Maitra, Ujjwal Bhattacharya, and Swapan K. Parui. Cnn based common approach to handwritten character recognition of multiple scripts. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1021–1025, 2015. Cited on p. 5.

[19] Raymond Ptucha, Felipe Petroski Such, Suhas Pillai, Frank Brockler, Vatsala Singh, and Paul Hutkowski. Intelligent character recognition using fully convolutional neural networks. *Pattern Recognition*, 88:604–613, 2019. Cited on p. 5.

[20] Chucai Yi and Yingli Tian. Localizing text in scene images by boundary clustering, stroke segmentation, and string fragment classification. *IEEE Transactions on Image Processing*, 21(9):4256–4268, 2012. Cited on p. 5.

[21] Comparison between neural network and support vector machine in optical character recognition. *Procedia Computer Science*, 116:351–357, 2017. Discovery and innovation of computer science technology in artificial intelligence era: The 2nd International Conference on Computer Science and Computational Intelligence (ICCSCI 2017). Cited on p. 5.

[22] Jerod J. Weinman, Zachary Butler, Dugan Knoll, and Jacqueline Feild. Toward integrated scene text reading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):375–387, 2014. Cited on p. 6.

[23] Yi-Feng Pan, Xinwen Hou, and Cheng-Lin Liu. A hybrid approach to detect and localize texts in natural scene images. *IEEE Transactions on Image Processing*, 20(3):800–813, 2011. Cited on p. 6.

[24] N. Arica and F.T. Yarman-Vural. Optical character recognition for cursive handwriting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):801–813, 2002. Cited on p. 7.

[25] Shailesh Acharya, Ashok Kumar Pant, and Prashnna Kumar Gyawali. Deep learning based large scale handwritten devanagari character recognition. In *2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, pages 1–6, 2015. Cited on p. 8.