



MUKESH PATEL SCHOOL OF  
TECHNOLOGY MANAGEMENT  
& ENGINEERING

---

# **Software Requirements Specification**

**for**

## **Digit Recognition System**

**Prepared by,**

**E050 Nihal Shetty**

**E051 Kartikey Shrivastav**

**E071 Arya Shah**

**Mentor: Dr, Shubha Puthran**

# Table of Contents

<b>Table of Contents .....</b>	<b>ii</b>
<b>Revision History .....</b>	<b>ii</b>
<b>1. Introduction.....</b>	<b>1</b>
1.1 Purpose .....	1
1.2 Document Conventions .....	2
1.3 Intended Audience and Reading Suggestions .....	2
1.4 Product Scope.....	3
1.5 References .....	4
<b>2. Overall Description.....</b>	<b>5</b>
2.1 Product Perspective .....	5
2.2 Product Functions.....	6
2.3 Data Flow Diagrams .....	6
2.4 User Classes and Characteristics.....	8
2.5 Operating Environment.....	13
2.6 Design and Implementation Constraints .....	13
2.7 Assumptions and Dependencies.....	14
<b>3. Non Functional Requirements .....</b>	<b>14</b>
3.1 User Interfaces .....	14
3.2 Hardware Interfaces .....	16
3.3 Safety and Security Requirements-.....	17
3.4 Software Quality Attributes .....	17

## Revision History

Name	Date	Reason For Changes	Version

## 1. Introduction

### 1.1 Purpose

Digit recognition is one of the active research topics in digital image processing. It is a classic machine learning problem. The goal of this project is to take an input in the form of handwritten digits and determine what those digits are. The principal task in digit recognition is to extract HOG (Histogram of Oriented Gradients) features from the dataset of handwritten digits and to build a classifier on it. Then we predict the handwritten digit input with the help of predefined digits of the database using this classifier. To build the classifier we write python scripts.

**Feedback to the Customer**-This software requirement specification assures the project management stakeholders and client that the development team has really understood the business requirements documentation properly. This also provides confidence that the team will develop the functionality which has been detailed.

**Facilitating other Documentation**-The SRS forms the basis for a load of other important documents such as the Software Design Specification.

**Product Validation**-It basically helps in validating with the client that the product which is being delivered, meets what they asked for.

Which means that the product we have output is Equal to the standards of the documentation in the SRS which the client satisfied and agreed on.

Characteristics of a Software Requirement Specification:

#### 1.1.1 Accuracy

We will ensure the accuracy of the software and the data entered to the dataset used in the project.

#### 1.1.2 Clarity

This SRS will be clearly stating what the user wants in the software.

#### 1.1.3 Completeness

The software requirement specification contains all the requirements stated in the business requirements documentation that the user specified.

#### 1.1.4 Consistency

The document is consistent from beginning till the end. It helps the readers understand the requirements well.

### **1.1.5 Prioritizations of Requirements**

The requirements will be full filled according to the order of priority and preference.

### **1.1.6 Verifiability**

At the end of the project, the user/client will be able to verify that all that all the agreed deliverables have in fact been produced and meet the project management requirements specified.

### **1.1.7 Modifiability**

The SRS can be modified when the development team and user feel the need.

### **1.1.8 Traceability**

Each requirement stated in the SRS is uniquely associated to a source such as a use case or interaction document etc.

## **1.2 Document Conventions**

The document is prepared using Microsoft Word from the Microsoft Office 2019 Suite and has used the font type 'Times New Roman'. The fixed font size that has been used to type this document is 12pt with 1.5 line spacing. It has used the bold property to set the headings of the document. All pages except the cover page are numbered, the numbers appear on the upper right hand corner of the page. Every image and data table are numbered and referred to the in the main text. Standard IEEE template is the template used to organize the appearance of the document and its flow.

## **1.3 Intended Audience and Reading Suggestions**

Recognition of handwritten text could be a difficult task for the computer and this can be time consuming and frustrating. For example, teachers in particular have to face this issue most commonly while correcting test papers, homework assignments and examinations. A doctor's writing is also said to be very difficult to interpret. Our program seeks to minimize these obstacles by providing instant results and with considerable accuracy. The software also takes into consideration of being able to teach children number digits by letting them draw on the interface and then predicting what number they drew.

End User for the problem solution is the everyday man who wants to cut time in copying, and typing the printed books, documents, etc. OCR has a very wide range of applications and scope that each and everyone with a phone a working job can find it useful. From an industry making the roads safe to a company working in the field of data extraction. Everyone can find an application of OCR making it one of the most exciting projects for the near future.

## 1.4 Product Scope

There are many applications of character recognition software such as-

- 1. Data Entry-** It can be used to enter data in the datasets of businesses, hospitals, etc in a more efficient manner by removing the hassle of typing the characters.
- 2. Traffic Sign Recognition-** A moving vehicle can read the signposts on the roads and accordingly change its state. For example, a car is moving with speed 80km/h can sense the maximum speed limit by reading the maximum speed sign and change its speed accordingly.
- 3. Textual Books-** It can also be used to make textual versions of printed books, cutting considerable time in typing the printed characters.
- 4. Assistive text-** This technology can be used to scan and the printed documents and be able to 'read' the characters.
- 5. Test for CAPTCHA-** Character recognition software can be used to test the robustness and the difficulty of the CAPTCHA tests. Improvements can be made on the basis of these tests.

### Scope of Product in the above fields:

**1. Data Entry** – Data Entry is one of the if not the biggest application of OCR and has the most scope. Some of the major advantages in using OCR for data entry are as follows-

#### Higher Productivity:

OCR software helps businesses to achieve higher productivity by facilitating quicker data retrieval when required. The time and effort which the employees were required to put in for extracting relevant data can now be channelized to focus on core activities.

#### Cost Reduction:

OCR usage can help the companies and organisations cutting down on hiring employees for data extraction. This will help in eliminating costs of copying, printing, shipping, etc. OCR also will help

cut the miscellaneous cost of lost documents, misplacing, etc. and can help in getting more office space.

**High Accuracy:**

One of the major challenges of data entry is inaccuracy. Automated data entry tools such as OCR data entry result in reduced errors and inaccuracies, resulting in efficient data entry. Besides, problems like data loss can also be successfully tackled by OCR data entry. As there is no manpower involved, the issues such as keying in wrong information accidentally or otherwise can be eliminated.

**Increased Storage Space:**

OCR can scan, document, and catalogue information from enterprise-wide paper documents. This simply means that the data can now be stored in an electronic format in servers, eradicating the need for maintaining huge paper files. In this way, OCR data entry serves as one of the best tools to implement “Paperless” approach across the organization

**2. Traffic Sign Recognition-** Through the application of OCR, the New Smart Car industry can reap many benefits from this technology. Traffic Sign Recognition is an idea whereby a car can recognize the traffic sign(s) such as ‘speed limit’, ‘turn ahead’, ‘men at work’, ‘slippery road’, etc. and then modify the speed or the movement of the car accordingly. OCR will help immensely in this, making it one of the major scope of OCR.

## 1.5 References

Given the ubiquity of handwritten documents in human transactions, Optical Character Recognition (OCR) of documents have invaluable practical worth Optical character recognition is a science that enables to translate various types of documents or images into analyzable, editable and searchable data. During last decade, researchers have used artificial intelligence / machine learning tools to automatically analyze handwritten and printed documents in order to convert them into electronic format. The objective of this literature survey is to summarize research that has been conducted on character recognition of, handwritten documents and to provide research directions. In this literature survey we collected, synthesized and analyzed research articles on the topic of handwritten OCR (and closely related topics) which were published between year 2000 to 2018. We followed widely used electronic databases by following pre-defined review protocol. Articles were searched using

keywords, forward reference searching and backward reference searching in order to search all the articles related to the topic. After carefully following study selection process few articles were selected for this review. This review serves the purpose of presenting state of the art results and techniques on OCR and also provide research directions by highlighting research gaps.

#### Reference Links:

1. <http://yann.lecun.com/exdb/mnist/>
2. <https://arxiv.org/abs/2001.00139>
3. <https://www.researchgate.net/publication/326552575> Implementation of Optical Character Recognition Using Machine Learning
4. <https://www.researchgate.net/publication/326284496> Persian text classification via character-level convolutional neural networks
5. <https://www.researchgate.net/publication/234762329> OCR for printed Kannada text to machine editable format using database approach

## 2. Overall Description

### 2.1 Product Perspective

The product created by the group is by no means, a follow-on member of a product family or a replacement for certain existing systems. Instead, it is a new, self-contained product.

#### 1. Addressing the Ambiguities:

The term digit recognition is used in the form of Pattern recognition wherein handwritten input in the form of drawing by the user is given to the software and the predicted output is displayed with prediction accuracy. By no means will the software use scanning functionalities where users can scan a particular digit with and expect the software to interpret it.

**2. Addressing the Inconsistencies:** The software is not complex in nature and thus has very little chance of grave inconsistencies arising. As far as inconsistencies in the handwritten digit input by the user is concerned, the software is capable of displaying the percentage of accuracy of the number predicted after being trained and tested with the predefined dataset.

**3. Addressing the Incompleteness:** The software tries its best to predict the handwritten digits given as input by the user. The software is not capable of predicting alphabets as far as version 1.0 is considered. The software restricts itself to the dataset of predefined handwritten digits and assumes that the particular dataset fulfills all the requirements

## **2.2 Product Functions**

The user will be able to perform the following functions in the product:

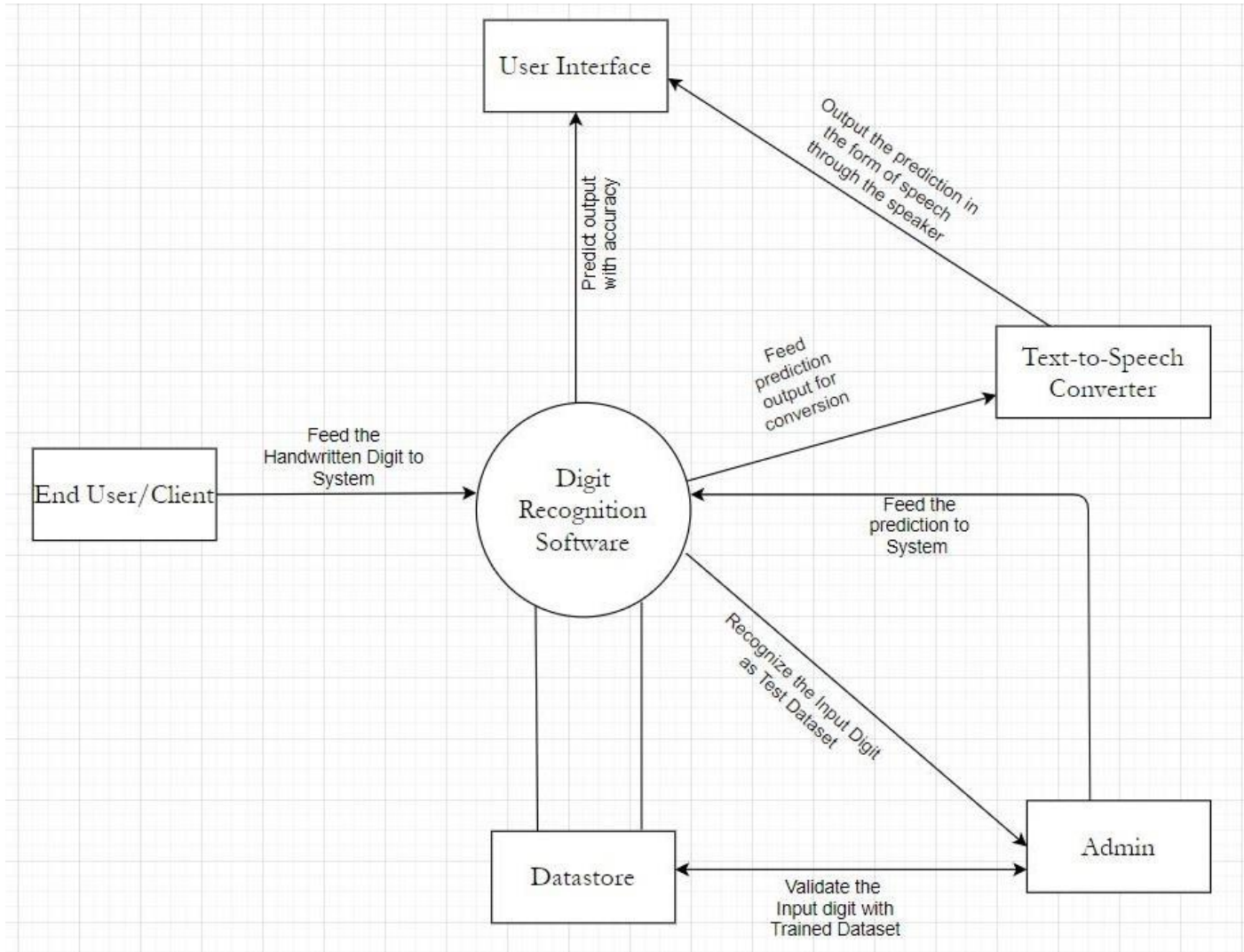
- Draw handwritten numerical digits on the GUI (Graphical User Interface) of the software.
- The user will be able to predict the handwritten digit drawn earlier on the interface.
- The user will be able to get the accuracy of the digit predicted by the software.
- If need be, the software will support Alphabet prediction as well.
- The software shall support text-to-speech functionality in future versions too.

## **2.3 Data Flow Diagrams**

### **2.3.1 Data Flow Diagram Level 0**

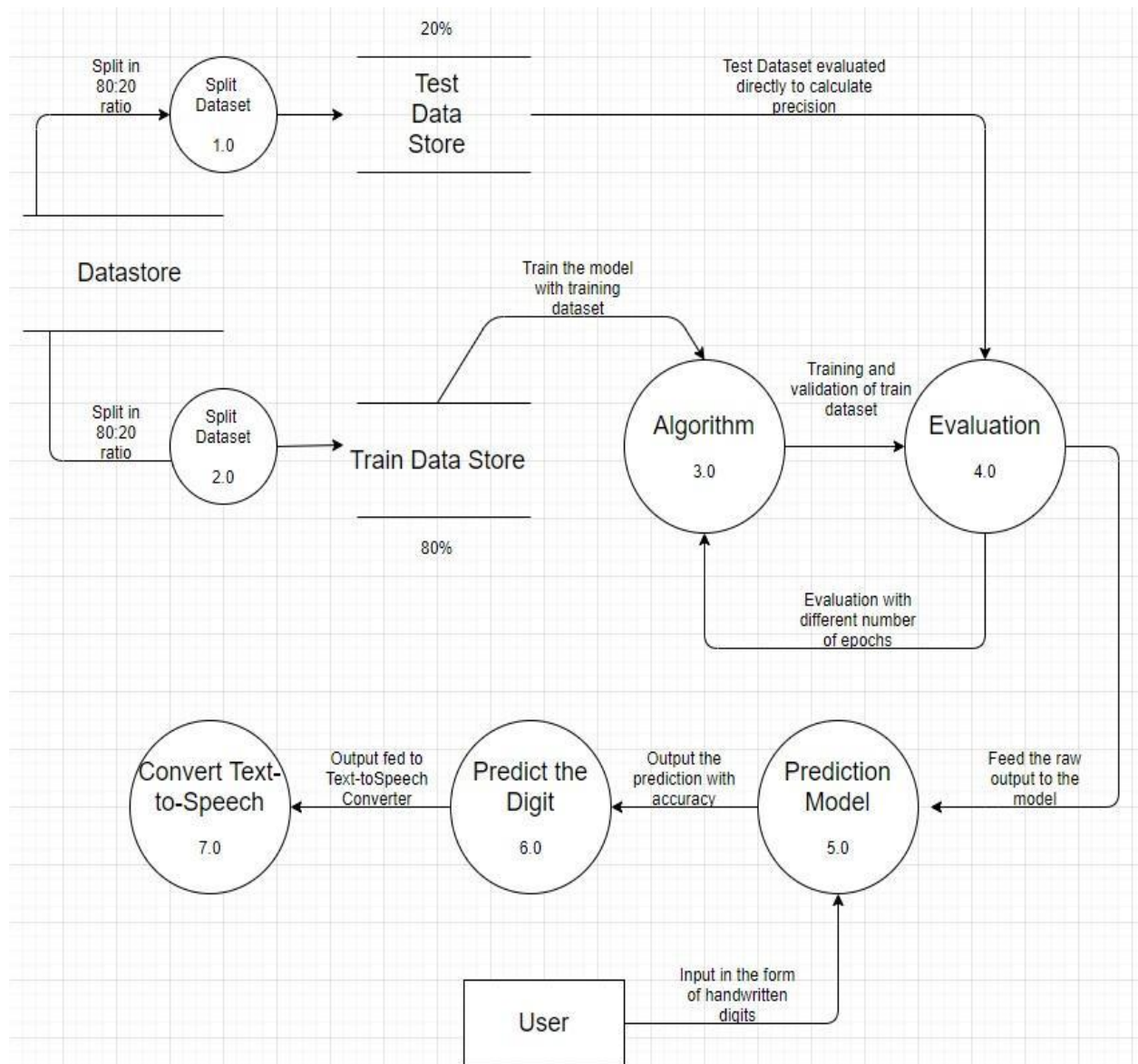
The following figure depicts the 0-Level Data Flow Diagram:





### 2.3.2 Data Flow Diagram Level 1

The following figure depicts the Level 1 Data Flow Diagram:



## 2.4 User Classes and Characteristics

### 1. End-User/Client:

The End User/Client is the sole user of the software. The user has a variety of functionalities that the software has to offer for use.

#### Key Functions:

1. Inputs the image containing handwritten text that he/she wants to be detected.
2. Checks for the accuracy of the Digit Recognition.
3. Copies the recognized digits to the clipboard.
4. Convert the output prediction from text to speech.
5. Submit feedback to the admin for improvement in predictions.

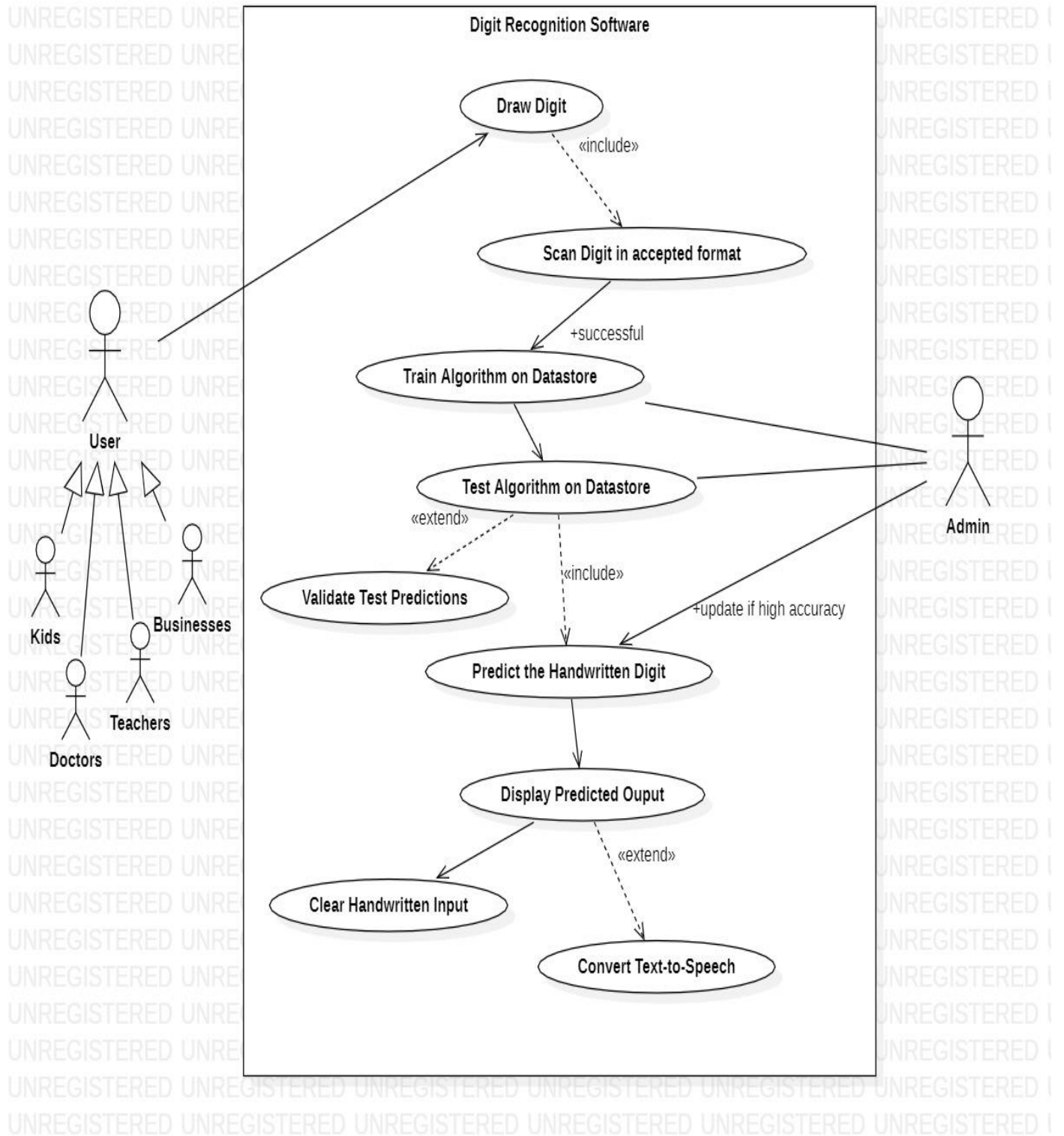
#### 2. Admin:

Admin has the full access to the system which means he is able to manage any activity with regard to the system. He is the highest privileged user who can access the system.

#### Key Functions:

1. Communicates with the Datastore of the Digit Recognition System.
2. Updates and modifies the database for better accuracy of predictions.
3. Manages the inputs and feedbacks of the End-User/Clients.
4. Update the libraries for text to speech conversion.

## 2.5 Use Case Diagram



**Use Case Scenarios**

<b>Name</b>	Draw Digit
<b>Description</b>	Draw the handwritten digit on the software interface
<b>Actors</b>	Teachers, Doctors, Children, Business Professionals
<b>Pre Condition</b>	The user should start the software and have access to mouse/digital trackpad for handwritten drawing.
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. User clicks on the .exe software</li><li>2. User navigates to the Drawing Area on the GUI</li></ol>
<b>Extension</b>	Users can maximize the software window for more drawing space.
<b>Post Condition</b>	Handwritten digit is recorded by the software

<b>Name</b>	Train/Test Algorithm on Datastore
<b>Description</b>	The dataset needs to be prepared and trained for the software to predict handwritten digits with sufficient accuracy.
<b>Actors</b>	Admin
<b>Pre Condition</b>	The admin should have sole access to the Software Datastore and rights to modify and update the same.
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. Admin accesses the software datastore</li><li>2. Admin Trains the datastore with previous handwritten digit data</li><li>3. Admin Tests the datastore with appropriate prediction algorithms</li></ol>

	4. Admin updates the datastore with relevant data
<b>Extension</b>	Admin can validate the datastore to retrieve the accuracy and precision of the predicted digits
<b>Post Condition</b>	The datastore is ready for the prediction of user drawn digits

<b>Name</b>	Predict Handwritten Digit
<b>Description</b>	User can predict the handwritten digit by clicking on the predict button with accuracy of the digit predicted
<b>Actors</b>	Teachers, Doctors, Children, Business Professionals
<b>Pre Condition</b>	The user must have entered a handwritten digit on the drawing area of the GUI
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. User draws the digit with mouse/trackpad</li><li>2. User clicks on the predict button to predict the digit with accuracy</li></ol>
<b>Extension</b>	User can clear the drawn digit and redraw it in case there is a change of numbers
<b>Post Condition</b>	The user can now display the predicted digit on the user interface

<b>Name</b>	Display Predicted Output
<b>Description</b>	User can now display the output by the software and display the prediction accuracy in percentage

<b>Actors</b>	Teachers, Doctors, Children, Business Professionals
<b>Pre Condition</b>	User must have drawn the digit on the GUI and pressed the predict button in order for the software to run its algorithm to predict and identify the digit
<b>Main flow of events</b>	<ol style="list-style-type: none"> <li>1. User draws the digit on the GUI</li> <li>2. User clicks on the predict button</li> <li>3. User finds the predicted output of the handwritten digit with accuracy mentioned in percentage</li> </ol>
<b>Extension</b>	<p>User can opt to get the output in Text-to-Speech format by an additional click of a button</p> <p>User</p>
<b>Post Condition</b>	The user gets the predicted output and can copy the output to the clipboard and paste it in their respective documents

## 1.1 2.6 Operating Environment

### System Requirements:

- Processor: 1 gigahertz (GHz) or faster processor
- RAM: 1GB and above
- OS: Windows 7/8/10 and later versions
- Display: 800 x 600
- Python 3.6 and later versions
- Graphics: DirectX 9.0 and later versions

## 1.2 2.7 Design and Implementation Constraints

- The software does not require any login or registration.

- Only the administrator has the access to the complete system along with the privilege to modify the Datastore.
- Should be optimized enough to use less RAM and Processing power.
- Each user should have access to a keyboard and mouse in order to interact with the software.
- Software only supports single digit number recognition.
- Text-to-speech functionality requires the user system to have a speaker for the verbal output.

### **1.3 2.8 Assumptions and Dependencies**

The key assumptions and dependencies for the project are as follows:

- Each user should have access to a mouse or trackpad in order to draw digits on the UI.
- No advertisements from the third party shall ever be part of the software.
- Datastore of the software will only be accessed by the admin and will not be available to modify by the User.
- Only the Admin shall have access to the Datastore and reserved the rights to manage and update the same.
- It is not mandatory for the user to have an active network connection in order to use the software standalone on a Windows machine.

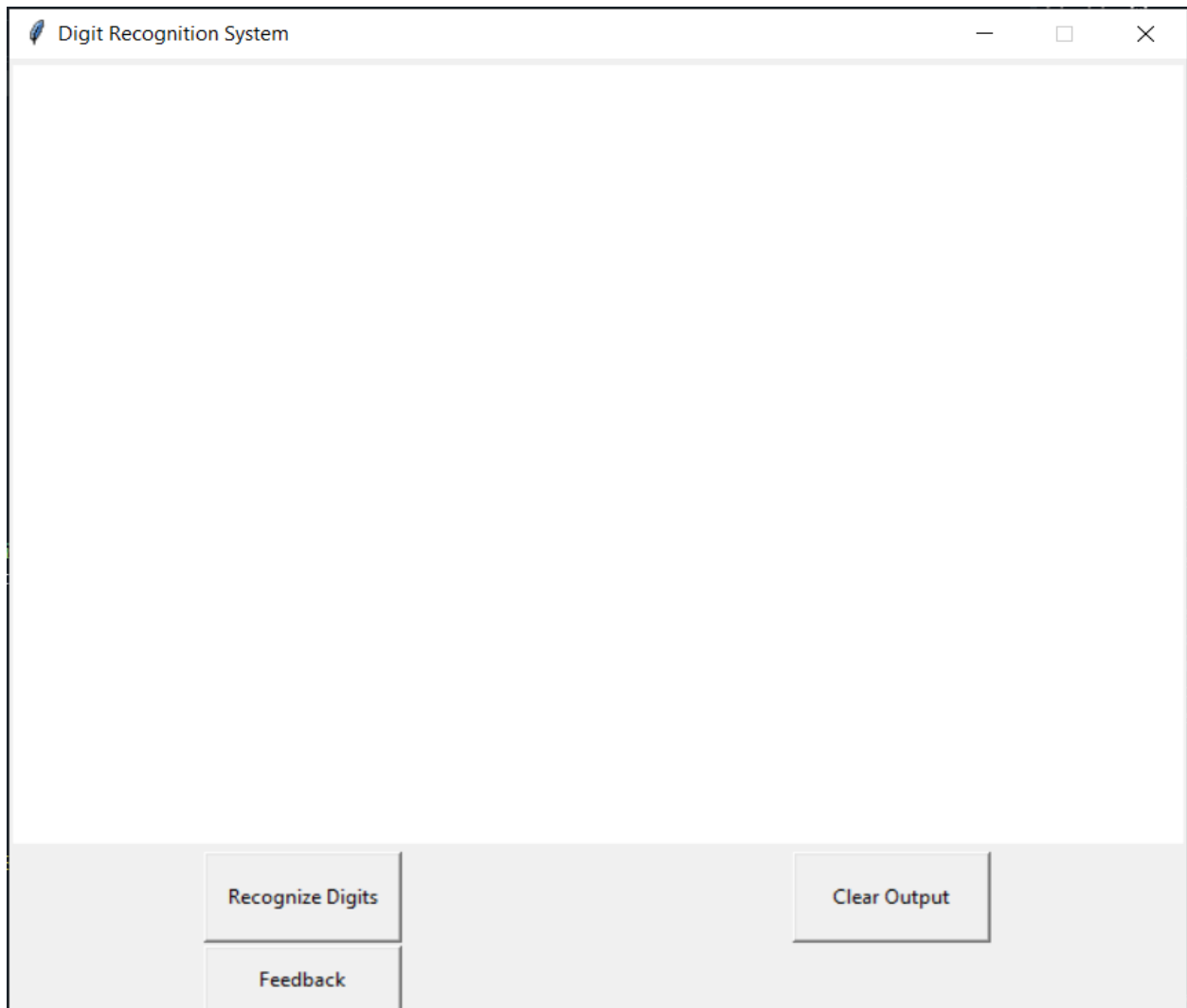
## **3. Non Functional Requirements**

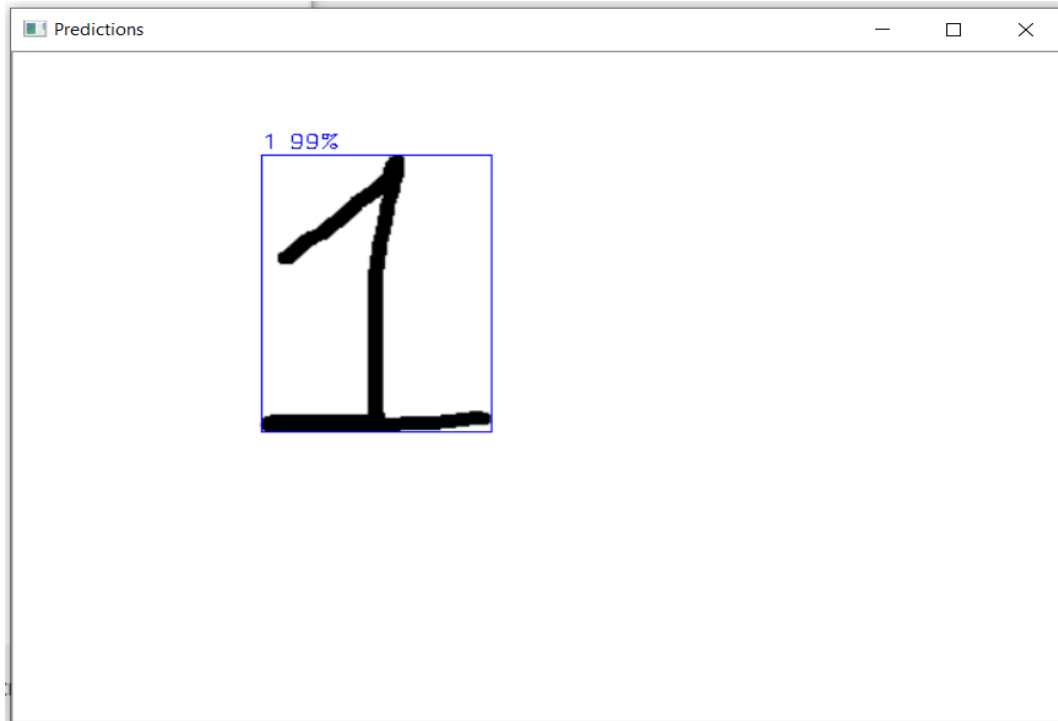
### **3.1 User Interfaces**

- Drawing box: The user interface will comprise a drawing box which can accept input via the mouse/laptop touchpad which makes it easier for the user to operate. This region will cover 70-80% of the application window. If the user chooses to enter input in an image format, he can load it using the "load image" button. The opened image will then be displayed instead of the drawing box since it is not in use.
- Display box: The machine learning algorithm will then recognize the inputted number and display it in the display region which will be present below the drawing box. This display box can also be used to provide instructions for the user.



- Predict button: A button to run the algorithm is also present. On pressing this button, the machine learning algorithm will use the inputted data to identify the number that was entered.
- TTS Button: A Text-to-Speech (TTS) option is provided via the TTS button which will read out the output as an audio. This is extremely useful for people with eye disabilities.
- Clear button: This button will clear the display region and the drawing region. In other words, it acts as a "RESET" switch in electronic circuits. It can accept fresh input for a new user or the same user who wants to run the program again.





## 3.2 Hardware Interfaces

### 1) Laptop/Personal Computer:

The hardware interface for the project should be a Personal Computer/Laptop with a Windows 10 Operating System. There are no specific hardware component requirements since our project is not a memory intensive or high storage consuming in nature. The purpose of this PC/Laptop is to provide information of the data entered by the user in handwritten format.

The minimum system requirements needed for the software to run are as follows:

- Processor: 1 gigahertz (GHz) or faster processor
- Hard Disk Space (HDD/SSD): 500 MB
- RAM: 1GB and above
- Display: 800 x 600
- Graphics: 512MB with DirectX 9.0 and later versions

### 2) Display Unit:

A Display unit is necessary for the input and output operations. It is used to display greetings to welcome the user, help him/her with the interface, display information about the software and to

accept the input and display the output that has been obtained based on it. It can also display video/audio media as a form of entertainment for the user.

### **3) Printer (Optional):**

The Printer can print the transcripts of the user interaction and final output that was obtained in a standard format.

## **3.3 Safety and Security Requirements-**

For extensive damages to the database due to unforeseen circumstances, the data collected through the constant usage (trial and error, elimination) and the already stored data is backed up to archival storage. The backup will be stored as Cloud storage, so that it can be accessed easily even when the physical database gets lost, destroyed, etc.

All the administrative and data entry operators have unique logins so the system can understand who logs in to the system at the moment so no intruders are allowed except system controlled administrative. No user or administrator can change or record valuable data.

## **3.4 Software Quality Attributes**

### **Reliability**

The product will be reliable enough to sustain in any condition. It will give consistently correct results. Product reliability is measured in terms of working of the project under different working environment and different conditions.

### **Maintainability**

Different versions of the product would be easy to maintain. While developing it would be easy to add more lines of code to the existing system, and will also be easy to upgrade for new features and new technologies from time to time. Maintenance will be cost-effective and easy.

**Usability**

This is measured in terms of ease of use i.e. The application would be user-friendly. Should be easy to learn for all kinds of users from all age groups and qualifications. Navigation will also be simple. It will also be easy to use for input preparation, operation, and interpretation of the output.

**Correctness**

The application should be correct in terms of its functionality, calculations used internally and the navigation should be correct. This means the application should adhere to functional requirements.

**Efficiency**

Time required to complete any task given to the system will be very less thereby rendering a very efficient product. The system utilizes processor capacity, disk space and memory efficiently.

**Testability**

The system will be easy to test and find defects and easy to divide into different modules for testing whenever necessary.

**Flexibility**

The project is flexible enough to modify and adaptable to other products with which an interaction is needed. It is easy to interface it with other standard 3rd party components