**HAND WRITTEN CHARACTER AND NUMERICAL RECOGNITION USING MACHINE LEARNING**

***A PROJECT REPORT***

***Submitted in partial fulfillment of the requirements for the award of the***

***degree of***

# BACHELOR OF TECHNOLOGY

IN

## COMPUTER SCIENCE & ENGINEERING

*Submitted by*

|  |  |
| --- | --- |
| ***J. MAMATHA***  ***Roll No. 16551A0536*** | ***V. SARITHA***  ***Roll No. 16551A05A9*** |
| ***D. YOBU HOSSANNA***  ***Roll No. 16551A0563*** | ***P. CHANDRAKANTH GOUD***  ***Roll No. 16551A05B6*** |

### Under the Supervision of

### *Dr.N. LEELAVATHY*

***Professor***

***Department of CSE***



## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING GODAVARI INSTITUTE OF ENGINEERING & TECHNOLOGY

### CHAITANYA KNOWLEDGE CITY, NH-16, RAJAMAHENDRAVARAM, AP

Jawaharlal Nehru Technological University, Kakinada, AP, India

***APRIL 2020***

## GODAVARI INSTITUTE OF ENGINEERING & TECHNOLOGY

(Autonomous)

CHAITANYA KNOWLEDGE CITY, NH-16, RAJAMAHENDRAVARAM 533296, AP

## BONAFIDE CERTIFICATE

Certified that this project report **“**EMOTION ANALYSIS USING DEEP

### LEARNING” is the bonafide work of “J. MAMATHA (16551A0536), V. SARITHA

**(16551A05A9), D. YOBU HOSSANNA (16551A0563)** and **P.**

**CHANDRAKANTH(16551A05B6)”**, who carried out the project work under my supervision during the year 2019 to 2020, towards partial fulfillment of the requirements of the Degree of Bachelor of Technology in Computer Science & Engineering as administered under the Regulations of Godavari Institute of Engineering & Technology, Rajamahendravaram, AP, India and award of the Degree from Jawaharlal Nehru Technological University,Kakinada. The results embodied in this report have not been submitted to any other University for the award of any degree.

Signature of the Head of the Department Signature of the Supervisor

### Dr. B. SUJATHA Dr. N. LEELAVATHY HEAD OF THE DEPARTMENT SUPERVISOR

Department of Computer Science & Engineering Professor

,

External Viva voce conducted on

**Internal Examiner External Examiner**

## GODAVARI INSTITUTE OF ENGINEERING & TECHNOLOGY

(Autonomous)

CHAITANYA KNOWLEDGE CITY, NH-16, RAJAMAHENDRAVARAM 533296, AP **CERTIFICATE OF AUTHENTICATION**

We solemnly declare that this project report **“**EMOTION ANALYSIS USING DEEP LEARNING**”** is the bonafide work done purely by us, carried out under the supervision of Dr. N. LEELAVATHY, towards partial fulfillment of the requirements of the Degree of Bachelor of Technology in Computer Science & Engineering as administered under the Regulations of Godavari Institute of Engineering & Technology, Rajamahendravaram, AP, India and award of the Degree from Jawaharlal Nehru Technological University, Kakinada during the year 2019-2020.

We also declare that no part of this document has been taken up verbatim from any source without permission from the author(s)/publisher(s). Wherever few sentences, findings, images, diagrams or any other piece of information has been used for the sake of completion of this work, we have adequately referred to the document source. In the event of any issue arising here after about this work, we shall be personally responsible.

It is further certified that this work has not been submitted, either in part of in full, to any other department of the Jawaharlal Nehru Technological University Kakinada, or any other University, institution or elsewhere, in India or abroad or for publication in any form.

Signature of the Student(s)

**J. MAMATHA(16551A0536)**

**V. SARITHA(16551A05A9)**

**D. YOBUHOSSANNA(16551A0563)**

**P. CHANDRAKANTH GOUD (16551A05B6)**

## ACKNOWLEDGEMENT

We feel immense pleasure to express my sincere thanks and profound sense of gratitude to all those people who played a valuable role for the successful completion of our project.

We would like to express my gratitude to **Dr. K. V. V. SATYANARAYANA RAJU**, Chairman, Chaitanya group of Institutions, **Sri. K. SASI KIRAN VARMA**, Vice Chairman, GIET group of Institutions for providing necessary infrastructure.

We are thankful to Principal **Dr. P. M. M. S. SUBRAHMANYA SHARMA** for permitting and encouraging us in doing this project.

Our special thanks to **Prof. N. Leelavathy**, vice principal and professor, department of CSE, for his content guidance and motivation and also for providing an excellent environment. We are truly grateful for his valuable suggestions and advice.

We are proudly grateful to express my deep sense of gratitude and respect towards **Dr. B. SUJATHA**, Professor and Head of the Department, Computer Science and Engineering, whose motivation and constant encouragement has led to pursue a project in the field of software development. We are very fortunate, for having her to enlighten us in all the aspects of life and transforming us to be an ideal individual in this field.

We are much obliged and thankful to our internal guide **Dr. N. LEELAVATHY**, Professor, Computer Science and Engineering, for providing this opportunity and constant encouragement given by him during the course. We are grateful to his valuable guidance and suggestions during my project work.

Our parents have put us ahead of themselves, because of their hard work and dedication, we have had opportunities beyond our wildest dreams. Our heart full thanks to them for giving us all support we needed to be successful student and individual.

Finally, we express our thanks to other professors, classmates, friends, neighbors and nonteaching staff who helped us for the completion of our project and without infinite love and patience. .

## ABSTRACT

Handwriting differs from person to person. Sometimes it is very difficult for a person to identify what is written from the handwritten document by other person. In this situation Handwritten Character and numerical Recognition System will brings us the reading of various combined style of writing a character. Digitization of machine printed or handwritten text documents have become very popular with the advancements in computing and technology. Humans have tried to automatized their work by replacing themselves with machines. The transformation from manual to automatization gave rise to several research areas and text recognition is one among them. machine learning techniques have been proved to be very suitable for optical character recognition.

**CHAPTER 1**

## INTRODUCTION

### 1.1 Introduction of project

## Every individual has different handwriting as unique as the personality traits; even when a similar sentence is written twice by the same person the handwriting may not appear exactly the same [1]. Handwritten characters differ by 12 considerable characteristics: line quality, spacing (line or spaces between character and word), height, width and size of letters, pen lifts and separations, connection strokes, beginning and ending strokes, unusual letter formation, shading (pen pressure), slant, baseline habits, flourishment and embellishments and diacritic placement. External conditions also play a role in affecting the style of handwriting such as the types and colours of ink, pen tip type, smoothness of paper, table surface quality and material, personal emotions, age, gender and speed of the writing process.

Jindal et al. [10] have proposed a segmentation method for handwritten lines and words, which is based on midpoint detection technique, and accuracy of the results is 95%. Mehdi et al. [11] have improved the accuracy of word segmentation of cursive handwritten texts and also done the comparative analysis of the algorithms works with both bitmap and bitmap data. Jain et al. [12] have proposed a word segmentation technique for the OCR systems. The technique estimated the textual area as a large window. Then after, large window is alienated into small subwindows of distinct lines and these lines further segmented into the words as small sub–subwindows.

The designed framework is divided into three phases: registration, training and segmentationframework comprises eight distinct subsections: key-point detection, extraction of text regions (TRs), feature extraction using speeded-up robust features (SURF), super-pixels imposition, cost optimization function and labels imposition, structural learning, training, filtered features projection and word segmentation using structural support vector machine (S-SVM).

Intelligence of humans makes them different from computers. The human can do various tasks that are still impossible for machine to do by their own. One of such tasks is handwritten text recognition. Even though, Text recognition in the handwritten documents has been studied as one of the prominent research areas by different researchers during the last few decades [1] and because of that many automatic handwritten systems are developed by different researchers in past [2-12]. However, the recognition algorithm and its efficiency is still an open research issue. Due to the vast inconsistency in handwriting styles, frequently the state-of-the-art handwriting recognition systems gets fail to provide satisfactory performance on various types of handwriting samples.

Available approaches to handwriting recognition usually consist of various steps which mainly include 1.preprocessing, 2.feature extraction, 3.classification, 4. post processing. However, feature extraction and classifier design are the two major steps of any recognition system [3, 5]. Many researchers made different type of handwritten text recognition systems for different languages such as English [4, 8, 10], Chinese [11], Arabic [9], Japanese [12] Bangla [6], Malyalam [7] etc. Still the recognition problems of these scripts cannot be considered to be entirely solved.

.

## CHAPTER 2

## LITERATURE SURVEY

The main aim of this research work is to classify the emotional expression from the mouth region of the human face. As the initial task is to extract the mouth region from the facial image, a survey on various existing research works to segment the face expression images is reviewed and discussed.

|  |  |  |
| --- | --- | --- |
| Reference and year | Approach and Method | Performance |
| Chirag Patel, Ripal Patel, 2011 | Hand Written Character Recognition using Neural Network | The main aim of this project is to design expert system for , “HCR(English) using Neural Network”. that can effectively recognize a particular character of type format using the Artificial Neural Network approach. |
| Pradeep Jayabala, E Srinivasan, 2011 | Neural network based handwritten character recognition system without feature extraction | Handwriting recognition has been one of the active and challenging research areas in the field of image processing and pattern recognition. |
| R. Plamondon, 2000 | On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey | Handwriting has continued to persist as a means of communication and recording information in day-to-day life even with the introduction of new technologies. |
| M. Riedmiller, H. Braun, 1993 | A Direct Adaptive Method for Faster Backpropagation Learning | A learning algorithm for multilayer feedforward networks, RPROP (resilient propagation), is proposed. |
| M. Riedmiller, 1994 | Advanced Supervised Learning in Multi-layer Perceptrons - From Backpropagation to Adaptive Learning Algorithms, | Since the presentation of the backpropagation algorithm [1] a vast variety of improvements of the technique for training the weights in a feed-forward neural network have been proposed. |

## CHAPTER 3

## SYSTEM ANALYSIS

### 3.1 EXISTING SYSTEM

The existing recognition systems are implemented on hand written digits and handwritten character seperately. The system can be extended to recognize system that can be able to recognize both digits and characters with more accuracy.

**Disadvantages:**

* Line segmentation method for the historical documents which are affected by repeated use and aging problem describing the historical documents which are affected by problems of aging and repeated use by imposing text blocks over the text lines and words

**3.2 PROPOSED SYSTEM**

Hand Written Character and Numerical Recognition works in stages as preprocessing, feature extraction and recognition using neural network. Preprocessing includes series of operations to be carried out on document image to make it ready for further process. Then feature extraction technique is applied on image. Finally feature vector is presented to the selected algorithm for recognition. Here this extracted features are provided to Neural Network for recognition of character and digit.

**Advantage:**

## The experimental setup of the designed framework has two distinct phases of operations: In the first handwritten text script, image is segmented into word regions (WRs). Therefore, a speeded-up robust features (SURFs) algorithm

## is applied to extract the SURF descriptors (SDs) of text regions (TRs). Then, the SURF detects the key point and extracts their analogous SURF descriptors (SDs) along with their analogous scales and orientations (SOs). While the training process, the SD codebook is constructed by clustering the SDs of training samples.

**CHAPTER-4**

## SYSTEM STUDY

### 4.1 FEASIBILITY STUDY

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

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILLITY

## 4.1.1 ECONOMICAL FEASIBILITY

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

## 4.1.2 TECHNICAL FEASIBILITY

The technical issue usually raised during the feasibility stage of the investigation includes the following:

* Does the necessary technology exist to do what is suggested?
* Do the proposed equipments have the technical capacity to hold the data required to use the new system?
* Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
* Can the system be upgraded if developed?
* Are there technical guarantees of accuracy, reliability, ease of access and data security? Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’.

The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

## 4.1.3 SOCIAL FEASIBILITY

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

* Is there sufficient support for the management from the users?
* Will the system be used and work properly if it is being developed and implemented?
* Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

**CHAPTER 5**

## SYSTEM DESIGN

### 5.1 System Design

System design is the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. Systems design implies a systematic approach to the design of a system. It may take a bottom-up or top-down approach.

### 5.2 System Architecture

A system architecture or systems architecture is the conceptual model that architecture is the conceptual model that defines the structure, behavior and more views of a system. An architecture description is a formal description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. System architecture can comprise system components, the externally visible properties of those components, the relationships between them.

Fig-5.2 System Architecture

#### 5.2.1 INPUT DESIGN

Input design is a part of overall system design. The main objective during the input design is as given below:

* To produce a cost-effective method of input.
* To achieve the highest possible level of accuracy.
* To ensure that the input is acceptable and understood by the user.

### INPUT STAGES:

The main input stages can be listed as below:

* Data recording
* Data transcription
* Data conversion
* Data verification
* Data control
* Data transmission
* Data validation
* Data correction

### INPUT TYPES:

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

* External inputs, which are prime inputs for the system.
* Internal inputs, which are user communications with the system.
* Operational, which are computer department’s communications to the system?
* Interactive, which are inputs entered during a dialogue.

### INPUT MEDIA:

At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to;

* Type of input
* Flexibility of format
* Speed
* Accuracy
* Verification methods
* Rejection rates
* Ease of correction
* Storage and handling requirements
* Security
* Easy to use
* Portability

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive. As

Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

### 5.2.2 OUTPUT DESIGN

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provides a permanent copy of the results for later consultation. The various types of outputs in general are:

* External Outputs, whose destination is outside the organization
* Internal Outputs whose destination is within organization and they are the
* User’s main interface with the computer.
* Operational outputs whose use is purely within the computer department.
* Interface outputs, which involve the user in communicating directly.

### OUTPUT DEFINITION

##### The outputs should be defined in terms of the following points:

* Type of the output
* Content of the output
* Format of the output
* Location of the output
* Frequency of the output
* Volume of the output
* Sequence of the output

It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

##### 5.3 UML DIAGRAMS

The Unified Modeling Language is a standard language for specifying, visualizing, constructing and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objectsoriented software and the software development projects. UML provides the users with a ready-to-use, expressive visual modeling so they can develop and exchange meaningful words. It provides extensibility and specialization mechanisms to extend the core concepts. UML is independent of programming languages and development processes.

UML is designed to let developers and customers view a software system from a different perspective and in varying degrees of abstraction. UML diagrams commonly created in visual modeling tools. The unified modeling language allows the software engineers to expand an analysis model using the modeling notation that is governed by a set of syntactic, semantic pragmatic rules.

###### 5.3.1Use Case Diagram

A Use case diagram at its simplest is a representation of a user interaction with the system that shows the relationship between the user and different use cases in which the user is involved

Use case diagrams are a set of use cases, actors, and their relationships. They represent the use case view of a system. A use case represents a particular functionality of a system. Hence, use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as **actors**.

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

**Actors:** System, User

**Functionalities:** Real time video capture frame, Preprocessing, Segmentation, Feature Extraction, Classification, Result.



###### Fig-5.2.1 Use Case Diagram

5.3.2Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

In software engineering, 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, and the relationships among the classes.

**Class names:** Video input, Preprocess, Segmentation, Emotion result

**Video input Operations:** Open camera frame.

**Preprocess Operations:** Color segmentation and image pixel segmentation

**Segmentation Operations:** Eye local, Mouth Local, Black and white.

**Emotion result Operations:** emotion, Compare frame, Display result.



**Fig-5.3.2 Class Diagram**

5.3.3 Activity Diagram

Activity diagram describes the flow of control in a system. Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc. Activities are nothing but the functions of a system. Numbers of activity diagrams are prepared to capture the entire flow in a system. Activity diagrams are used to visualize the flow of controls in a system. This is prepared to have an idea of how the system will work when executed.



###### Fig-5.3.3ActivityDiagram

5.3.4 Sequence Diagram

UML Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent.

A sequence diagram is an interaction diagram. From the name, it is clear that the diagram deals with some sequences, which are the sequence of messages flowing from one object to another. Interaction among the components of a system is very important from implementation and execution perspective.

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



Fig 5.3.4 Sequence Diagram

###### 5.4 Dataset Design

A data set is a collection of data. In other words, a data set corresponds to the contents of a single database table, or a single statistical data matrix, where every column of the table represents a particular variable, and each row corresponds to a given member of the data set in question. Several characteristics define a data set's structure and properties. These include the number and types of the attributes or variables, and various statistical measures applicable to them, such as standard deviation.

5.4.1 Building Dataset

**Steps to Construct Dataset:**

To construct an efficient dataset

* Collect the raw data.
* Identify feature and label sources.
* Select a sampling strategy.
* Split the data.
* Apply the algorithm •Grab the results

**CHAPTER-6**

## SYSTEM IMPLEMENTATION

### 6.1 Modules

**Preprocessing:**

The aim of preprocessing is an improvement of the image data that suppresses unwanted distortions or enhances some image features important for further processing.

**Create the model:**

* Now we will create our model. A KNN model.
* It works better for data that are represented as grid structures, this is the reason why KNN works well for image classification problems.

**Train the model:**

* The functions of keras will start the training of the module.

**Evaluate the model:**

* We have images in our dataset which will be used to evaluate how good our model works.
* The MNIST dataset is well balanced so we can get around 99% accuracy.

#### 6.1.1 DataCollection

Data collection is a very basic module and the initial step towards the project. It generally deals with the collection of the right dataset. According to the accuracy, using the model with the data to analyze the predictions accurately. Data collection is the process of gathering and measuring information on variables of interest, in an established systematic fashion that enables one to answer stated research questions, test hypotheses, and evaluate outcomes. The data collection component of research is common to all fields of study including physical and social sciences, humanities, business, etc. While methods vary by discipline, the emphasis on ensuring accurate and honest collection remains the same.

**The importance of ensuring accurate and appropriate data collection**  Regardless of the field of study or preference for defining data (quantitative, qualitative), accurate data collection is essential to maintaining the integrity of research. Both the selection of appropriate data collection instruments (existing, modified, or newly developed) and clearly delineated instructions for their correct use reduce the likelihood of errors occurring.

### Consequences from improperly collected data include

* inability to answer research questions accurately
* inability to repeat and validate the study
* distorted findings resulting in wasted resources
* misleading other researchers to pursue fruitless avenues of investigation
* compromising decisions for public policy
* causing harm to human participants and animal subjects

While the degree of impact from faulty data collection may vary by discipline and the nature of investigation, there is the potential to cause disproportionate harm when these research results are used to support public policy recommendations.

**Issues related to maintaining integrity of data collection** The Primary rationale for preserving data integrity is to support the detection of errors in the data collection process, whether they are made intentionally (deliberate falsifications) or not (systematic or random errors).

Most, Craddick, Crawford, Redican, Rhodes, Rukenbrod, and Laws (2003) describe ‘quality assurance’ and ‘quality control’ as two approaches that can preserve data integrity and ensure the scientific validity of study results. Each approach is implemented at different points in the research timeline (Whitney, Lind, Wahl, 1998):

1. Quality assurance - activities that take place *before* data collection begins
2. Quality control - activities that take place *during* and *after* data collection

### Quality Assurance

Since quality assurance precedes data collection, its main focus is **'prevention'** (i.e., forestalling problems with data collection). Prevention is the most cost-effective activity to ensure the integrity of data collection. This proactive measure is best demonstrated by the standardization of protocol developed in a comprehensive and detailed procedures manual for data collection. Poorly written manuals increase the risk of failing to identify problems and errors early in the research endeavour. These failures may be demonstrated in a number of ways:

* Uncertainty about the timing, methods, and identify of person(s) responsible for reviewing data
* Partial listing of items to be collected
* Vague description of data collection instruments to be used in lieu of rigorous stepby-step instructions on administering tests
* Failure to identify specific content and strategies for training or retraining staff members responsible for data collection
* Obscure instructions for using, making adjustments to, and calibrating data collection equipment (if appropriate)
* No identified mechanism to document changes in procedures that may evolve over the course of the investigation.

An important component of quality assurance is developing a rigorous and detailed recruitment and training plan. Implicit in training is the need to effectively communicate the value of accurate data collection to trainees (Knatterud, Rockhold, George, Barton, Davis, Fairweather, Honohan, Mowery, O'Neill, 1998). The training aspect is particularly important to address the potential problem of staff who may unintentionally deviate from the original protocol. This phenomenon, known as **‘drift’**, should be corrected with additional training, a provision that should be specified in the proceduresmanual.

Given the range of qualitative research strategies (non-participant/participant observation, interview, archival, field study, ethnography, content analysis, oral history, biography, unobtrusive research) it is difficult to make generalized statements about how one should establish a research protocol in order to facilitate quality assurance. Certainly, researchers conducting non-participant/participant observation may have only the broadest research questions to guide the initial research efforts. Since the researcher is the main measurement device in a study, many times there are little or no other data collecting instruments. Indeed, instruments may need to be developed on the spot to accommodate unanticipated findings.

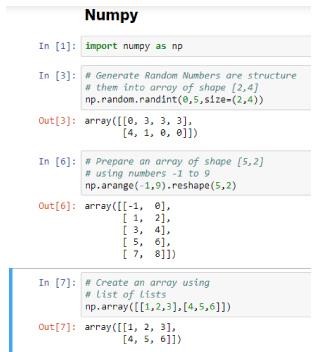
### Quality Control

While quality control activities (detection/monitoring and action) occur during and after data collection, the details should be carefully documented in the proceduresmanual. A clearly defined communication structure is a necessary pre-condition for establishing monitoring systems. There should not be any uncertainty about the flow of information between principal investigators and staff members following the detection of errors in data collection. A poorly developed communication structure encourages lax monitoring and limits opportunities for detecting errors.

Detection or monitoring can take the form of direct staff observation during site visits, conference calls, or regular and frequent reviews of data reports to identify inconsistencies, extreme values or invalid codes. While site visits may not be appropriate for all disciplines, failure to regularly audit records, whether quantitative or quantitative, will make it difficult for investigators to verify that data collection is proceeding according to procedures established in the manual. In addition, if the structure of communication is not clearly delineated in the proceduresmanual, transmission of any change in procedures to staff members can be compromised. Quality control also identifies the required responses, or ‘actions’ necessary to correct faulty data collection practices and also minimize future occurrences. These actions are less likely to occur if data collection procedures are vaguely written and the necessary steps to minimize recurrence are not implemented through feedback and education.

### Numpy

Python has a strong set of data types and data structures. Yet it wasn’t designed for Machine Learning per say. Enter numpy (pronounced as num-pee). Numpy is a data handling library, particularly one which allows us to handle large multi-dimensional arrays along with a huge collection of mathematical operations. The following is a quick snippet of numpy in action.



Numpy isn’t just a data handling library known for its capability to handle multidimensional data. It is also known for its speed of execution and vectorization capabilities. It provides MATLAB style functionality and hence requires some learning before you can get comfortable. It is also a core dependency for other majorly used libraries like pandas, matplotlib and so on. It’s documentation itself is a good starting point.

### Pandas

Think of relational data, think pandas. Yes, pandas is a python library that provides flexible and expressive data structures (like dataframes and series) for data manipulation. Built on top of numpy, pandas is as fast and yet easier to use.



Pandas provides capabilities to read and write data from different sources like CSVs, Excel, SQL Databases, HDFS and many more. It provides functionality to add, update and delete columns, combine or split dataframes/series, handle datetime objects, impute null/missing values, handle time series data, conversion to and from numpy objects and so on. If you are working on a real-world Machine Learning use case, chances are, you would need pandas sooner than later. Similar to numpy, pandas is also an important component of the SciPy or Scientific Python Stack .

### 3. Scipy

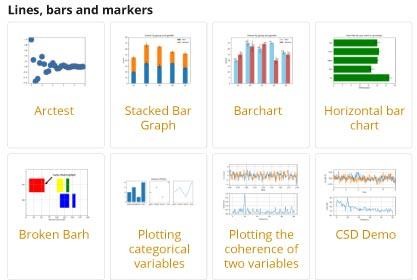
Pronounced as Sigh-Pie, this is one of the most important python libraries of all time. Scipy is a scientific computing library for python. It is also built on top of numpy and is a part of the Scipy Stack.



This is yet another *behind the scenes* library which does a whole lot of heavy lifting. It provides modules/algorithms for linear algebra, integration, image processing, optimizations, clustering, sparse matrix manipulation and many more. .

### . Matplotlib

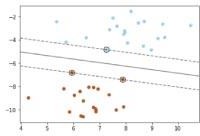
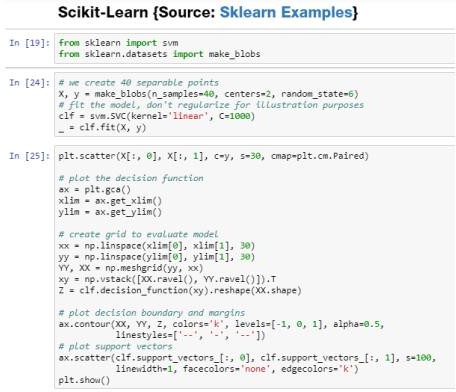
Another component of the SciPy stack, matplotlib is essentially a visualization library. It works seamlessly with numpy objects (and its high-level derivatives like pandas). Matplotlib provides a MATLAB like plotting environment to prepare high-quality figures/charts for publications, notebooks, web applications and so on.



Matplolib is a high customizable low-level library that provides a whole lot of controls and knobs to prepare any type of visualization/figure. Given its low-level nature, it requires a bit of getting used to along with plenty of code to get stuff done. Its well documented and extensible design has allowed a whole list of high-level visualization libraries to be built on top. Some of which, we will discuss in the coming sections. :

### . Scikit-Learn

Designed as an extension to the SciPy library, scikit-learn has become the de-facto standard for many of the machine learning tasks. Developed as part of Google Summer of Code project, it has now become a widely contributed open source project with over 1000 contributors.

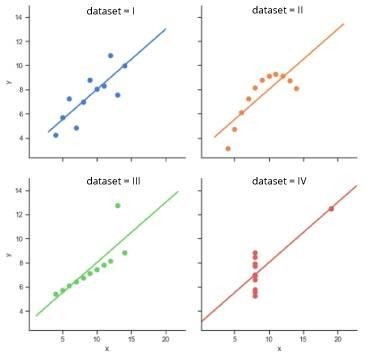
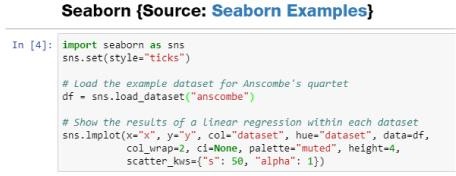


Scikit-learn provides a simple yet powerful fit-transform and predict paradigm to learn from data, transform the data and finally predict. Using this interface, it provides capabilities to prepare classification, regression, clustering and ensemble models. It also provides a multitude of utilities for preprocessing, metrics, model evaluation techniques, etc.

### Visualization

**6. Seaborn**

Built on top of matplotlib, seaborn is a high-level visualization library. It provides sophisticated styles straight out of the box (which would take some good amount of effort if done using matplotlib).



Sample plots using seaborn.

Apart from styling prowess and sophisticated color pallets, seaborn provides a range of visualizations and capabilities to work with multivariate analysis. It provides capabilities to perform regression analysis, handling of categorical variables and aggregate statistics.

### CHAPTER 7

### SYSTEM REQUIREMENTS

#### 7.1 Functional Requirements

Outputsfromcomputersystemsarerequiredprimarilytocommunicatetheresultsofprocessingto users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs in generalare:

* External Outputs, whose destination is outside theorganization.
* Internal Outputs whose destination is within organization and they arethe
* User’s main interface with thecomputer.
* Operational outputs whose use is purely within the computerdepartment.
* Interface outputs, which involve the user in communicatingdirectly.
* Understanding user’s preferences, expertise level and his business requirements.

.

#### 7.2 Non-Functional Requirements

NON-FUNCTIONAL REQUIREMENT (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of nonfunctional requirement, *“how fast does the website load?”* Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non- functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users are> 10000. Description of non-functional requirements is just as critical as a functionalrequirement.

* Usabilityrequirement
* Serviceabilityrequirement
* Manageabilityrequirement
* Recoverabilityrequirement
* Securityrequirement
* Data Integrity requirement
* Capacityrequirement
* Availabilityrequirement
* Scalability requirement
* Interoperability requirement
* Reliability requirement
* Maintainability requirement
* Regulatory requirement
* Environmental requirement

### 7.3 Hardware Requirements:

* System : Intell I-3, 5, 7 Processor.
* Hard Disk : 500 GB.
* Floppy Drive : 1.44 Mb.
* Monitor : 14’ Colour Monitor.
* Mouse : Optical Mouse.
* Ram : 2Gb.

### 7.4 Software Requirements:

* Operating system **:** Windows 7,8,10 Ultimate, Linux, Mac.
* Front-End **:** Python.
* Coding Language **:** Python.
* Software Environment **:** Anaconda(jupyter or spyder).

#### 7.5 Overview of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Small Talk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

**7.5.2 Features of Python**

#### Python's features include

* **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain** − Python's source code is fairly easy-to- maintain.
* **A broad standard library** − Python's bulk of the library is very portable and crossplatform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable** − User can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases** − Python provides interfaces to all major commercial databases.
* **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable**− Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

* It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* It supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

##### 7.5.3 Installing Python

###### 7.5.3.1 Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python<https://www.python.org/>

User can download Python documentation from[https://www.python.org/doc/.](https://www.python.org/doc/) The documentation is available in HTML, PDF, and PostScript formats.

###### 7.5.3.2 Installing Python

Python distribution is available for a wide variety of platforms. User need to download only the binary code applicable for the platform and install Python.

If the binary code for the platform is not available, user need a C compiler to compile the source code manually. Compiling the source code offers more flexibility in terms of choice of features that require in installation.

###### 7.5.3.3 Windows Installation

Here are the steps to install Python on Windows machine.

* Open a Web browser and go to[https://www.python.org/downloads/.](https://www.python.org/downloads/)
* Follow the link for the Windows installer *python-XYZ.msi* file where XYZ is the version that need to install.
* To use this installer *python-XYZ.msi*, the Windows system must support Microsoft Installer 2.0. Save the installer file to local machine and then run it to find out if user machine supports MSI.
* Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished.

###### 7.5.3.4 Setting up PATH

Programs and other executable files can be in many directories, so operating systems provide a search path that lists the directories that the OS searches for executables.

The path is stored in an environment variable, which is a named string maintained by the operating system. This variable contains information available to the command shell and other programs.

The path variable is named as PATH in Unix or Path in Windows (Unix is case sensitive; Windows is not).

To add the Python directory to the path for a particular session in Windows −

**At the command prompt** − type path %path%; C:\Python and press Enter.

**Note** − C:\Python is the path of the Python directory

#### 7.6 OPENCV

OpenCV is a cross-platform library using which can develop real-time **computer vision applications**. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection.

Let’s start the chapter by defining the term "Computer Vision".

Computer Vision

Computer Vision can be defined as a discipline that explains how to reconstruct, interrupt, and understand a 3D scene from its 2D images, in terms of the properties of the structure present in the scene. It deals with modeling and replicating human vision using computer software and hardware.

Computer Vision overlaps significantly with the following fields −

* **Image Processing** − It focuses on image manipulation.
* **Pattern Recognition** − It explains various techniques to classify patterns.
* **Photogrammetry** − It is concerned with obtaining accurate measurements from images.

Computer Vision Vs Image Processing

**Image processing** deals with image-to-image transformation. The input and output of image processing are both images.

**Computer vision** is the construction of explicit, meaningful descriptions of physical objects from their image. The output of computer vision is a description or an interpretation of structures in 3D scene.

##### 7.6.1 Applications of Computer Vision

Here user have listed down some of major domains where Computer Vision is heavily used.

Robotics Application

* Localization − Determine robot location automatically
* Navigation
* Obstacles avoidance
* Assembly (peg-in-hole, welding, painting)
* Manipulation (e.g. PUMA robot manipulator)
* Human Robot Interaction (HRI) − Intelligent robotics to interact with and serve people

Medicine Application

* Classification and detection (e.g. lesion or cells classification and tumor detection)
* 2D/3D segmentation
* 3D human organ reconstruction (MRI or ultrasound)
* Vision-guided robotics surgery

Industrial Automation Application

* Industrial inspection (defect detection)
* Assembly
* Barcode and package label reading
* Object sorting
* Document understanding (e.g. OCR)

Security Application

* Biometrics (iris, finger print, face recognition)
* Surveillance − Detecting certain suspicious activities or behaviors

Transportation Application

* Autonomous vehicle
* Safety, e.g., driver vigilance monitoring

##### 7.6.2 Features of OpenCV Library

Using OpenCV library, user can −

* Read and write images
* Capture and save videos
* Process images (filter, transform)
* Perform feature detection
* Detect specific objects such as faces, eyes, cars, in the videos or images.
* Analyze the video, i.e., estimate the motion in it, subtract the background, and track objects in it.

OpenCV was originally developed in C++. In addition to it, Python and Java bindings were provided. OpenCV runs on various Operating Systems such as windows, Linux, OSX, FreeBSD, Net BSD, Open BSD, etc.

This tutorial explains the concepts of OpenCV with examples using Java bindings.

##### 7.6.3 OpenCV Library Modules

Following are the main library modules of the OpenCV library.

Core Functionality

This module covers the basic data structures such as Scalar, Point, Range, etc., that are used to build OpenCV applications. In addition to these, it also includes the multidimensional array **Mat**, which is used to store the images. In the Java library of OpenCV, this module is included as a package with the name **org.opencv.core**.

Image Processing

This module covers various image processing operations such as image filtering, geometrical image transformations, color space conversion, histograms, etc. In the Java library of OpenCV, this module is included as a package with the name **org.opencv.imgproc**.

Video

This module covers the video analysis concepts such as motion estimation, background subtraction, and object tracking. In the Java library of OpenCV, this module is included as a package with the name **org.opencv.video**.

Video I/O

This module explains the video capturing and video codecs using OpenCV library. In the Java library of OpenCV, this module is included as a package with the name **org.opencv.videoio**.

calib3d

This module includes algorithms regarding basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence and elements of 3D reconstruction. In the Java library of OpenCV, this module is included as a package with the name **org.opencv.calib3d**.

features2d

This module includes the concepts of feature detection and description. In the Java library of OpenCV, this module is included as a package with the name **org.opencv.features2d**.

Obj detect

This module includes the detection of objects and instances of the predefined classes such as faces, eyes, mugs, people, cars, etc. In the Java library of OpenCV, this module is included as a package with the name **org.opencv.objdetect**.

High gui

This is an easy-to-use interface with simple UI capabilities. In the Java library of OpenCV, the features of this module is included in two different packages namely, **org.opencv.imgcodecs** and **org.opencv.videoio**.

##### 7.6.4 A Brief History of OpenCV

OpenCV was initially an Intel research initiative to advise CPU-intensive applications. It was officially launched in 1999.

* In the year 2006, its first major version, OpenCV 1.0 was released.
* In October 2009, the second major version, OpenCV 2 was released.
* In August 2012, OpenCV was taken by a nonprofit organization OpenCV.org.

##### 7.7 Numpy

NumPy and SciPy are open-source add-on modules to Python that provide common mathematical and numerical routines in pre-compiled, fast functions. These are growing into highly mature packages that provide functionality that meets, or perhaps exceeds, that associated with common commercial software like MatLab. The NumPy (Numeric Python) package provides basic routines for manipulating large arrays and matrices of numeric data. The SciPy (Scientific Python) package extends the functionality of NumPy with a substantial collection of useful algorithms, like minimization, Fourier transformation, regression, and other applied mathematical techniques.

##### 7.8 Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.

In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data.

Prior to Pandas, Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, It can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyze.

Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

###### 7.8.1 Key Features of Pandas

* Fast and efficient Data Frame object with default and customized indexing.
* Tools for loading data into in-memory data objects from different file formats.
* Data alignment and integrated handling of missing data.
* Reshaping and pivoting of date sets.
* Label-based slicing, indexing and sub setting of large data sets.
* Columns from a data structure can be deleted or inserted.
* Group by data for aggregation and transformations.
* High performance merging and joining of data.
* Time Series functionality.

#### 7.9 KERAS

KERAS runs on top of open source machine libraries like TensorFlow, Theano or Cognitive Toolkit (CNTK). Theano is a python library used for fast numerical computation tasks. TensorFlow is the most famous symbolic math library used for creating neural networks and deep learning models. TensorFlow is very flexible and the primary benefit is distributed computing. CNTK is deep learning framework developed by Microsoft. It uses libraries such as Python, C#, C++ or standalone machine learning toolkits. Theano and TensorFlow are very powerful libraries but difficult to understand for creating neural networks.

KERAS is based on minimal structure that provides a clean and easy way to create deep learning models based on TensorFlow or Theano.it is designed to quickly define deep learning models. Well, it is an optimal choice for deep learning applications.

##### 7.9.1 Features of KERAS

KERAS leverages various optimization techniques to make high level neural network API easier and more performant. It supports the following features −

* Consistent, simple and extensible API.
* Minimal structure - easy to achieve the result without any frills.
* It supports multiple platforms and backends.
* It is user friendly framework which runs on both CPU and GPU.
* Highly scalability of computation.

##### 7.9.2 Benefits of using KERAS

KERAS is highly powerful and dynamic framework and comes up with the following advantages −

* Larger community support.
* Easy to test.
* KERAS neural networks are written in Python which makes things simpler.
* It supports both convolution and recurrent networks.
* Deep learning models are discrete components, so that, user can combine into many ways.

##### 7.10 TensorFlow

TensorFlow is a software library or framework, designed by the Google team to implement machine learning and deep learning concepts in the easiest manner. It combines the computational algebra of optimization techniques for easy calculation of many mathematical expressions.

The official website of TensorFlow is mentioned below − [www.tensorflow.org](https://www.tensorflow.org/)

Let us now consider the following important features of TensorFlow −

* It includes a feature of that defines, optimizes and calculates mathematical expressions easily with the help of multi-dimensional arrays called tensors.
* It includes a programming support of deep neural networks and machine learning techniques.
* It includes a high scalable feature of computation with various data sets.
* TensorFlow uses GPU computing, automating management. It also includes a unique feature of optimization of same memory and the data used.

**7.10.1 Why Tensor flow is so Popular?**

TensorFlow is well-documented and includes plenty of machine learning libraries. It offers a few important functionalities and methods for the same.

TensorFlow is also called a “Google” product. It includes a variety of machine learning and deep learning algorithms. TensorFlow can train and run deep neural networks for handwritten digit classification, image recognition, word embedding and creation of various sequence models.

##### 7.11 Anaconda

Anaconda is the data science platform for data scientists, IT professionals and business leaders of tomorrow. It is a distribution of[Python,](https://www.edureka.co/blog/introduction-to-python/)[R,](https://www.edureka.co/blog/r-programming-language) etc. With more than 300 packages for[data science,](https://www.edureka.co/blog/learn-python-for-data-science/) it becomes one of the best platforms for any project.

Anaconda is an open-source distribution for python and R. It is used for[datascience,](https://www.edureka.co/blog/how-to-learn-data-science/)[machine learning,](https://www.edureka.co/blog/introduction-to-machine-learning/)[deep learning,](https://www.edureka.co/blog/deep-learning-with-python/) etc. With the availability of more than 300 libraries for data science, it becomes fairly optimal for any programmer to work on anaconda for data science.



Anaconda helps in simplified package management and deployment. Anaconda comes with a wide variety of tools to easily collect data from various sources using various machine learning and AI algorithms. It helps in getting an easily manageable environment setup which can deploy any project with the click of a single button.

##### 7.12 JUPYTER Notebook

IPYTHON notebook was developed by Fernando Perez as a web based front end to IPYTHON kernel. As an effort to make an integrated interactive computing environment for multiple language, Notebook project was shifted under Project JUPYTER providing front end for programming environments JUILA and R in addition to Python.

A notebook document consists of rich text elements with HTML formatted text, figures, mathematical equations etc. The notebook is also an executable document consisting of code blocks in Python or other supporting languages.

JUPYTER notebook is a client-server application. The application starts the server on local machine and opens the notebook interface in web browser where it can be edited and run from. The notebook is saved as IPYNB file and can be exported as html, pdf and LATEX files

**7.6 Code:**

**import time**

**import os**

**from flask import Flask, render\_template, request**

**# import our OCR function**

**from ocr\_core import ocr\_core**

**from model import hdr\_predition**

**from model import hdr\_accuracy**

**from model import hdr\_img**

**# define a folder to store and later serve the images**

**UPLOAD\_FOLDER = '/static/uploads/'**

**# allow files of a specific type**

**ALLOWED\_EXTENSIONS = set(['png', 'jpg', 'jpeg'])**

**app = Flask(\_\_name\_\_)**

**# function to check the file extension**

**def allowed\_file(filename):**

**return '.' in filename and \**

**filename.rsplit('.', 1)[1].lower() in ALLOWED\_EXTENSIONS**

**# route and function to handle the home page**

**@app.route('/')**

**def home\_page():**

**return render\_template('index.html')**

**# route and function to handle the upload page**

**@app.route('/upload', methods=['GET', 'POST'])**

**def upload\_page():**

**start = time.time()**

**if request.method == 'POST':**

**# check if there is a file in the request**

**if 'file' not in request.files:**

**return render\_template('upload.html', msg='No file selected')**

**file = request.files['file']**

**# if no file is selected**

**if file.filename == '':**

**return render\_template('upload.html', msg='No file selected')**

**if file and allowed\_file(file.filename):**

**# call the OCR function on it**

**extracted\_text = ocr\_core(file)**

**end = time.time()**

**predition\_time =round(end-start,3)**

**# extract the text and display it**

**return render\_template('upload.html',**

**msg='Successfully processed',**

**predition\_time=predition\_time,**

**extracted\_text=extracted\_text,**

**img\_src=UPLOAD\_FOLDER + file.filename)**

**elif request.method == 'GET':**

**return render\_template('upload.html')**

**# route and function to handle the digit page**

**@app.route('/digit', methods=['GET', 'POST'])**

**def digit\_page():**

**start = time.time()**

**if request.method == 'POST':**

**index = request.form['index']**

**predition = hdr\_predition(index)**

**accuracy = hdr\_accuracy()**

**img\_src=hdr\_img(index)**

**end = time.time()**

**predition\_time =round(end-start,3)**

**# extract the text and display it**

**return render\_template('digit.html',**

**predition=predition,**

**accuracy=accuracy,**

**predition\_time=predition\_time,**

**img\_src=img\_src)**

**elif request.method == 'GET':**

**return render\_template('digit.html')**

**if \_\_name\_\_ == '\_\_main\_\_':**

**app.debug = True**

**app.run()**

## CHAPTER 8

## SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

Testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing is the process where that test data is prepared and is used for testing the modules individually and later the validations given for the fields. Then the testing takes place, which make sure that all components of the system properly function as a unit by actually trying to force the system to fail. Testing is the best performed when employee departments are asked to assist in identifying all possible situations that might arise.

1. Testing is a process of executing a program with the intent of finding errors.
2. A good test is one that has high probability of finding an undiscovered error.
3. A successful test is one that uncovers yet undiscovered error.

### TYPES OF TESTS

All tests should be planned long before. Testing should begin “in the small” and progress towards testing “in the large”. The first tests planned and executed generally, focus on individual program modules. As testing progresses, testing shift focus in an attempt to find errors in integrated clusters modules of ultimately in the entire system. Testing of debugged program is one of the most critical aspects of computer programming because without that work, the system will never produce the output for which it is designed. The philosophy behind testing is to find error. Hence, a successful test is one that finds errors. Actuallytesting is the state of implementation that is aimed at ensuring that the system works accurately and efficiently before live operation commences. Any engineered product can be tested in one of the two ways.

#### 8.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

#### 8.2 Integration testing

Integration testing is a systematic technique for constructing the program structures while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested modules and build a program structure that has been dictated by the design.

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

#### 8.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

**Functional testing is centered on the following items** Valid Input: identified classes of valid input must be accepted.

Invalid Input: identified classes of invalid input must be rejected.

Functions: identified functions must be exercised.

Output: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

#### 8.4 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configurationoriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

Once individual module testing is completed, modules are assembled and integrated to perform as a system. Then the top-down testing begins from upper-level to lower-level module testing was carried out to check whether the entire system is performing satisfactorily. This is the test done for the whole system integration after completion of the module testing.

#### Test Data and Output

For all types to test dummy data were keyed in and after the testing was over live data were keyed in from the source documents and concerned Department people were followed to key in and work on the system.

#### 8.5Module Testing

To locate errors, focus is given on the modules, independent of other modules. This is known as module testing. This enables us to detect error and correct it without affecting any other modules. Whenever the program was not executing the required function, it was corrected to get required results. Thusall the modules are tested from bottom to top starting with the smallest and lowest level modules and proceeding to next level. Individual like purchase order rising was done successfu11y for connection to the next modules.

#### 8.5Specification Testing

Executing the specification starting what the program should do and how it should perform under various conditions. Test cases for various situations and combination of condition in all the modules were tested.

#### 8.7 Code Testing

This examines the logic of the program. For example, the logic for calculating profit, rate taking the percentages entered by employee was tested and verified with the data that was carried out by manual methods

#### 8.8 User Interface Testing

The data that is given by the user should re-lie on the particular range. When the employee gives wrong data, the system will give error message. According to this, the program was developed. The value of the data should fall on the given range of value. The following are some of the testing strategies that are carried out during the testing period.

#### 8.9 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

All tests should be planned long before. Testing should begin “in the small” and progress towards testing “in the large”. The first tests planned and executed generally, focus on individual program modules. As testing progresses, testing shift focus in an attempt to find errors in integrated clusters modules of ultimately in the entire system. Testing of debugged program is one of the most critical aspects of computer programming because without that work, the system will never produce the output for which it is designed. The philosophy behind testing is to find error. Hence, a successful test is one that finds errors. Actuallytesting is the state of implementation that is aimed at ensuring that the system works accurately and efficiently before live operation commences. Any engineered product can be tested in one of the two ways.

#### 8.10 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. User cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Knowing the specified function that has been designed to perform, tests can be conducted that demonstrate each function is operational, at the same time searching for errors in each function. When computer software is considered, black box testing alludes to tests that are conducted at the software interface. Although they are designed to uncover error blackboxtestsareusedtodemonstratethatsoftwarefunctionareoperational properly accepted and output is correctly produced; and that the integrity of external information (data files) is maintained. A black box tests examined some fundamental aspect of a system with little regard for the internal logic structure of the software.

Black Box Testing attempts to find errors in following categories,

1. In-correct or missing errors
2. Errors in data structures or external database access.
3. Performance errors that Initialization and termination errors.

#### 8.11 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Acceptance test is conducted to enable the customer to validate all requirements. It can range from an informal test drive to a planned and systematically executed series of tests they are Alpha testing and beta testing at the developer site. The software is used in a natural setting with the developer looking over the shoulder of the employee and recording errors and usage problems. These tests are conducted sites by the software in an environment, which cannot be controlled by the developer.

Testing is a process which reveals errors in the program. It is the major quality measure employed during software development. During testing, the program is executed with a set of test cases, output of the program for the test cases is evaluated to determine if the program is perform as it is expected to perform and order to make sure that the system does not have error

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

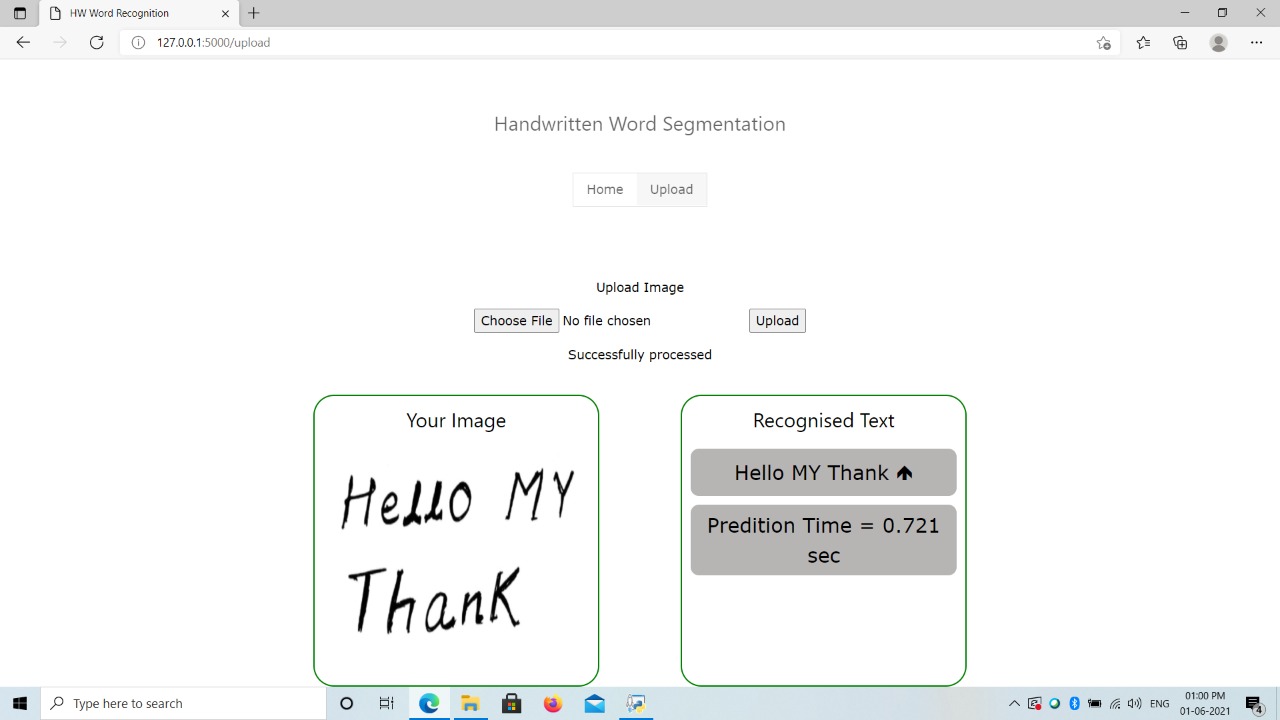
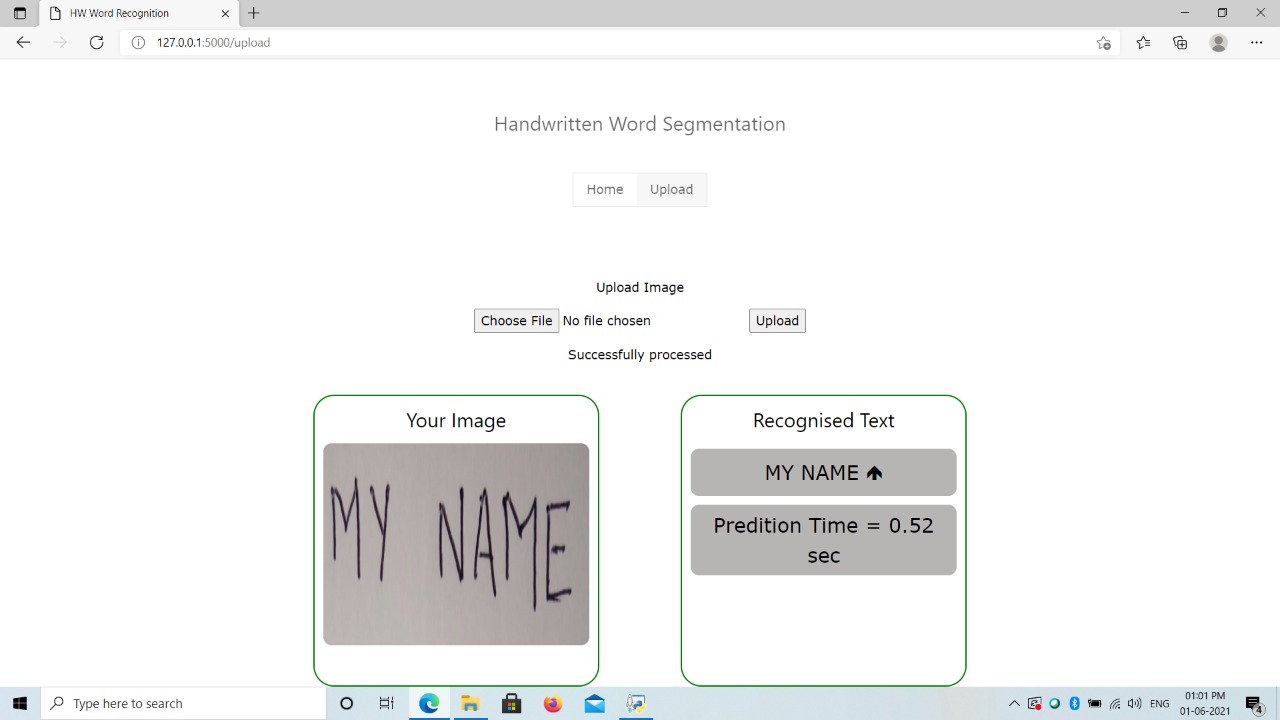
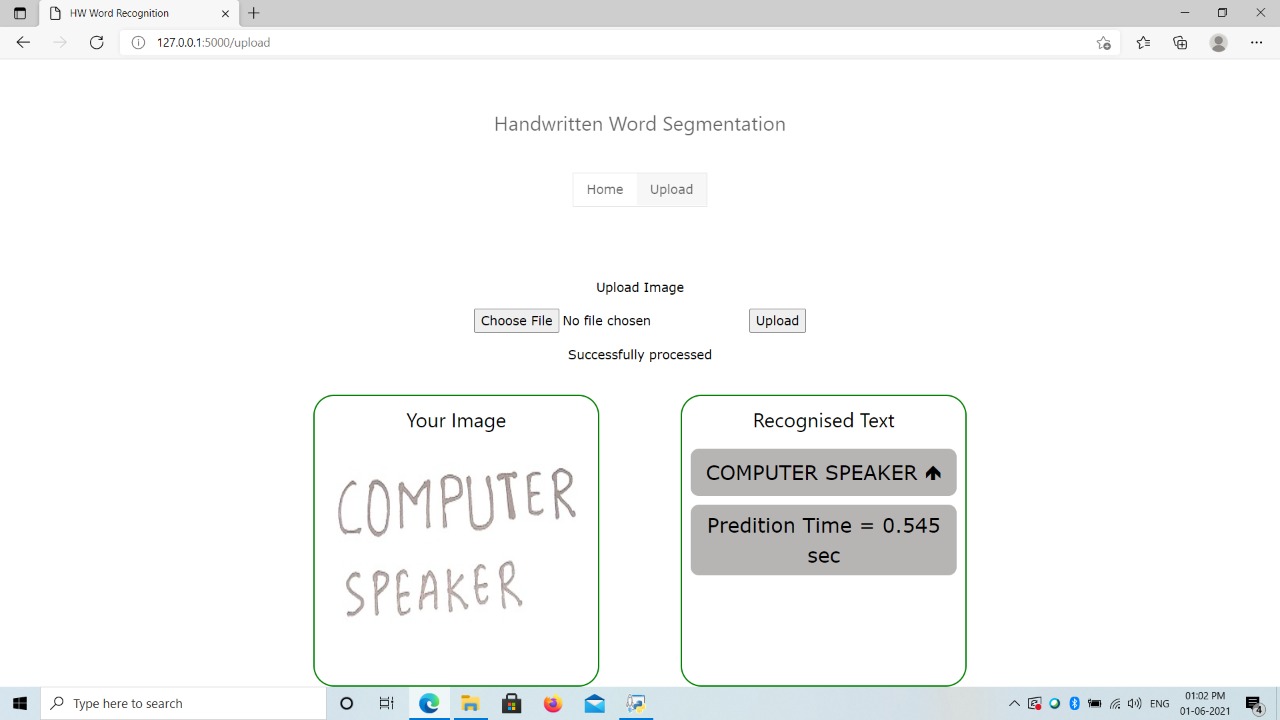
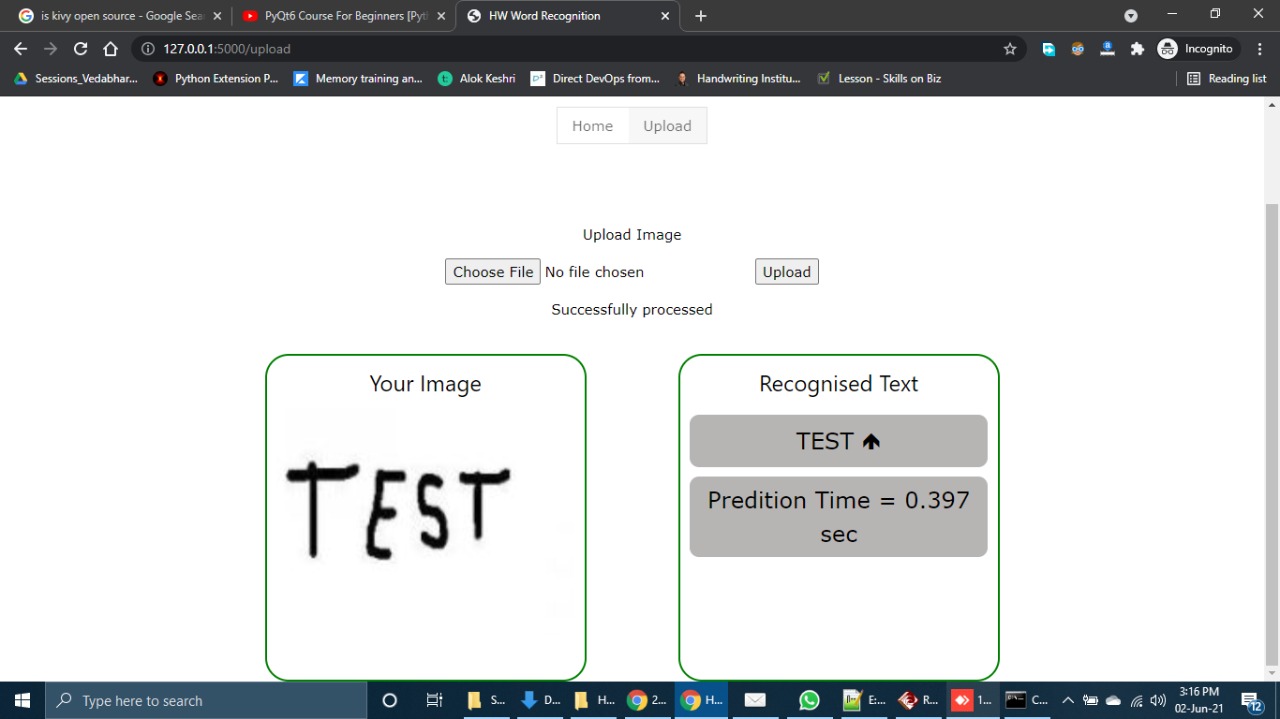
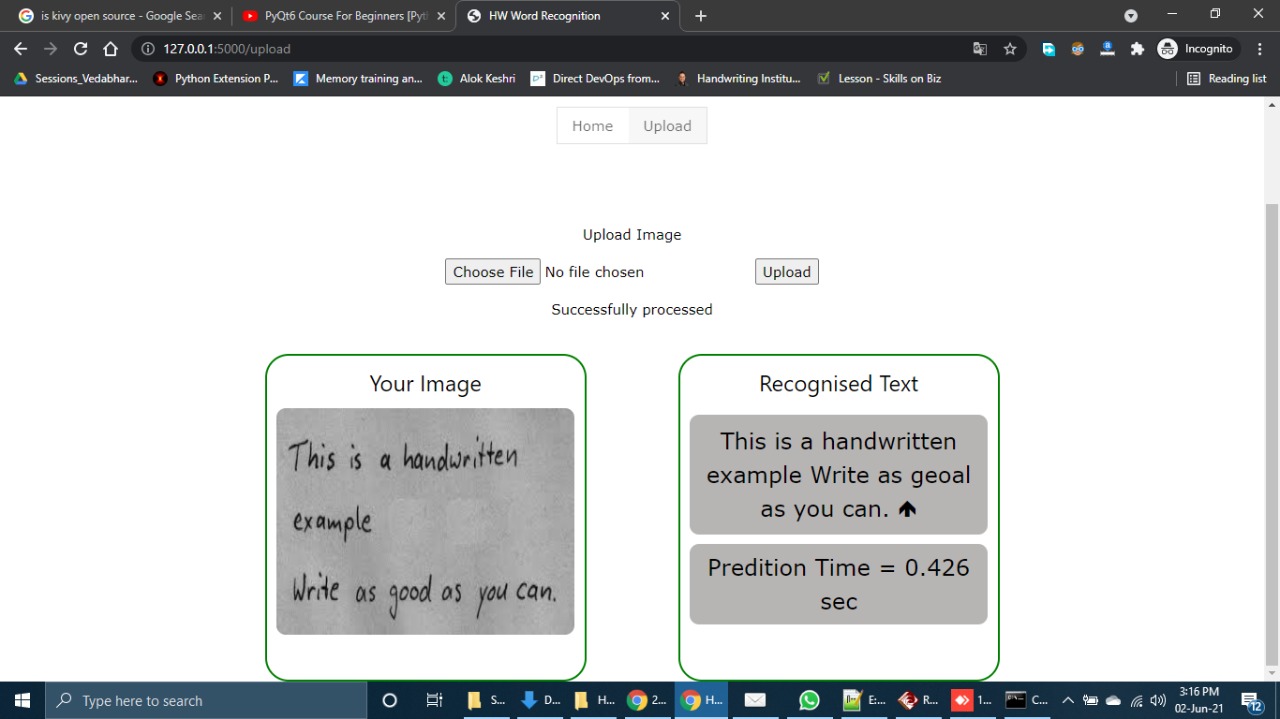
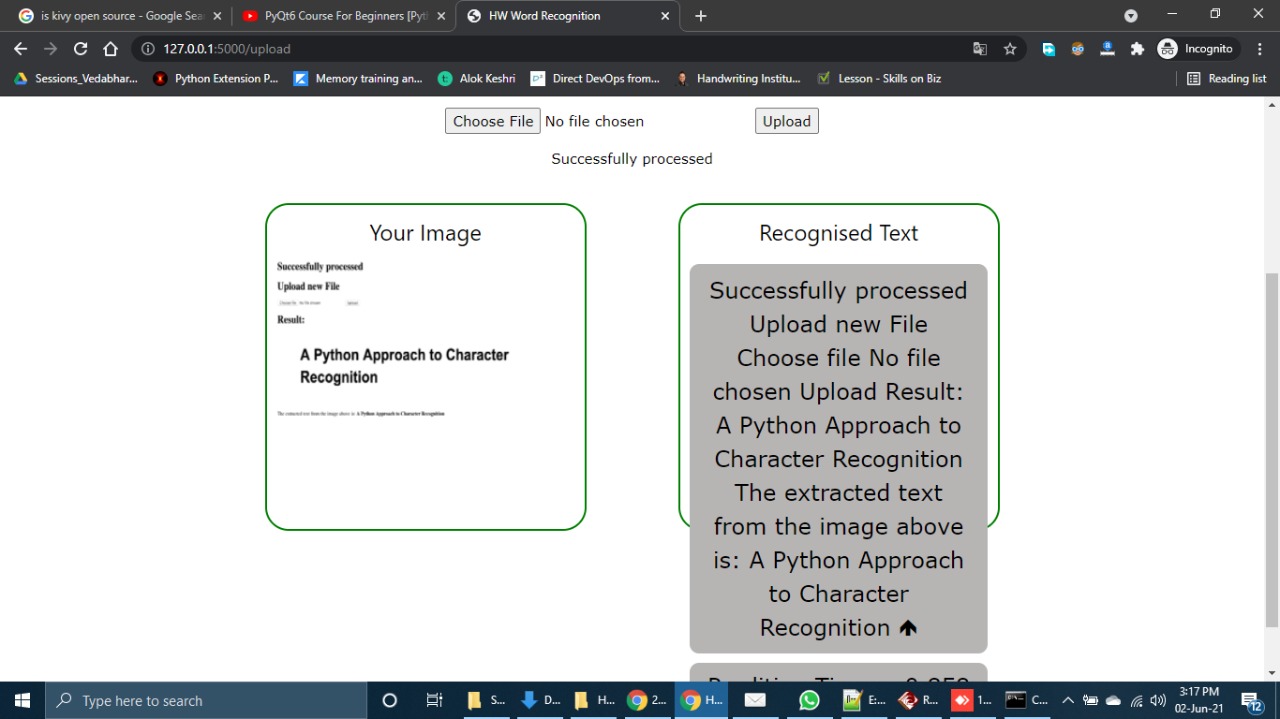
#### 8.12 Validation Testing

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected, and a final series of software testing validation testing may begin. Validation succeeds when the software function in a manner that can be reasonably accepted by a customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirements. After each validation testing, one of the two possible conditions exit.

The function or performance characteristics conform to the specification and are accepted. A deviation from specification is uncovered and deficiency list is created. The validation test was carried out on all modules. Most of the characteristics conformed to the specifications and where a deviation was encountered, the appropriated corrective measure was taken. In all validations test was executed successfully**.**

**CHAPTER 9**

## RESULTS



**CHAPTER 10**

## CONCLUSION AND FUTURE SCOPE

In conclusion, the outcome of this experiment has achieved the objective and the hypothesis was proven to be accurate. The significant increase in number of pixels used in data input had further increased the accuracy of down sampled handwritten data. The advantage of relying on huge number of pixels was to produce sharper and clearer handwritten data, which aids the recognition process and outcome. As a future work, the number of pixels can be further increased to a significant amount in order to increase character recognition accuracy. With a very large number of pixels, the recognition capability may be improved to the extent whereby the identification of handwritten data could be distinguished between the respondents.

**APPENDIX**

import time

import os

from flask import Flask, render\_template, request

# import our OCR function

from ocr\_core import ocr\_core

from model import hdr\_predition

from model import hdr\_accuracy

from model import hdr\_img

# define a folder to store and later serve the images

UPLOAD\_FOLDER = '/static/uploads/'

# allow files of a specific type

ALLOWED\_EXTENSIONS = set(['png', 'jpg', 'jpeg'])

app = Flask(\_\_name\_\_)

# function to check the file extension

def allowed\_file(filename):

return '.' in filename and \

filename.rsplit('.', 1)[1].lower() in ALLOWED\_EXTENSIONS

# route and function to handle the home page

@app.route('/')

def home\_page():

return render\_template('index.html')

# route and function to handle the upload page

@app.route('/upload', methods=['GET', 'POST'])

def upload\_page():

start = time.time()

if request.method == 'POST':

# check if there is a file in the request

if 'file' not in request.files:

return render\_template('upload.html', msg='No file selected')

file = request.files['file']

# if no file is selected

if file.filename == '':

return render\_template('upload.html', msg='No file selected')

if file and allowed\_file(file.filename):

# call the OCR function on it

extracted\_text = ocr\_core(file)

end = time.time()

predition\_time =round(end-start,3)

# extract the text and display it

return render\_template('upload.html',

msg='Successfully processed',

predition\_time=predition\_time,

extracted\_text=extracted\_text,

img\_src=UPLOAD\_FOLDER + file.filename)

elif request.method == 'GET':

return render\_template('upload.html')

# route and function to handle the digit page

@app.route('/digit', methods=['GET', 'POST'])

def digit\_page():

start = time.time()

if request.method == 'POST':

index = request.form['index']

predition = hdr\_predition(index)

accuracy = hdr\_accuracy()

img\_src=hdr\_img(index)

end = time.time()

predition\_time =round(end-start,3)

# extract the text and display it

return render\_template('digit.html',

predition=predition,

accuracy=accuracy,

predition\_time=predition\_time,

img\_src=img\_src)

elif request.method == 'GET':

return render\_template('digit.html')

if \_\_name\_\_ == '\_\_main\_\_':

app.debug = True

app.run()

import time

start = time.time()

import numpy as np

import matplotlib.pyplot as pt

import pandas as pd

from sklearn.neighbors import KNeighborsClassifier

data =pd.read\_csv("train.csv").values

clf=KNeighborsClassifier(n\_neighbors=3)

#trianing dataset

xtrain =data[0:40000,1:]

train\_lebel=data[0:40000,0]

clf.fit(xtrain,train\_lebel)

#test dataset

xtest=data[40000:,1:]

actual\_lebel=data[40000:,0]

def hdr\_predition(index):

index=int(index,10)

predition = clf.predict([xtest[index]])

return predition

#Accuracy

def hdr\_accuracy():

p=clf.predict(xtest)

count=0

for i in range(0,1000):

count+=1 if p[i]==actual\_lebel[i] else 0

accuracy = (count/1000)\*100

return accuracy

def hdr\_predition\_time():

end = time.time()

predition\_time = end - start

return predition\_time

def hdr\_img(index):

index=int(index,10)

d=xtest[index]

d.shape=(28,28)

pt.imshow(255-d,cmap='gray')

seq=str(index)

pt.savefig('static/uploads/pic'+seq+'.png')

return 'static/uploads/pic'+seq+'.png'

## REFERENCES

/

[1] Harralsonet. al. Handwriting Examination: Theory, Proficiency, and Methods. A Survey of Forensic Handwriting Examination Research in Response to the Nas Report.Retrieved 2013.

[2] Jaeger, S., Manke, S., Reichert, J., &Waibel, A. (2001). Online Handwriting Recognition: The Npen++ Recognizer. International Journal on Document Analysis and Recognition, 3(3), 169-180.

[3] Graves &Schmidhuber (2009). Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks.Neural Information Processing Systems Conference.

[4] Verma B, Lu J, Ghosh M and Ghosh R 2004 A Feature Extraction Technique for Online Handwriting Recognition Neural Networks 2004 Proceedings 2004 IEEE Int. Jt. Conf. 2 1337-1341. [5] Jayakumar, A., Babu, G. S., Raman, R., &Nedungadi, P. (2015). Integrating Writing Direction and Handwriting Letter Recognition in Touch-Enabled Devices (Vol. 2, Ser. 380).Springer India.

[6] Ahmed et al. (2015). OCR System for Poor Quality Images Using Chain-Code Representation. The 1st International Conference on Advanced Intelligent System and Informatics (Aisi2015), 407, 151-161.

[7] Bouillon, M., Anquetil, E., &Almaksour, A. (2013). Decremental Learning of Evolving Fuzzy Inference Systems: Application to Handwritten Gesture Recognition.

[8] Sarkar, M., Roy, S., Das, A., &Nath, A. (2017). International Journal of Advance Research in Computer Science and Management Studies.Recognition of Handwritten Characters Using Binarization, Image Segmentation and Correlation Matching Technique, 5(6).

[9] Zhang X Y, Bengio Y and Liu C L 2017 Online and Offline Handwritten Chinese Character Recognition: A Comprehensive Study and New Benchmark Pattern Recognition. 61 348-360

[10] O. Samanta et al (2018). An HMM framework based on spherical-linear features for online cursive handwriting recognition, Information Sciences, Volume 441, Pages 133-151, ISSN 0020-0255