

Voice Based Image Editor



Internship Project in association with **Computer Society of India, Hyderabad** submitted
in partial fulfillment of the requirement for the award of the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Under the esteemed guidance of

Mr. Krishna Teja
Trainer, Packet Prep Team
Mentor
Dr. Sesham Anandh
Professor
MVSR Engineering College

By

G VARUN KUMAR (18R11A0511)

K NITESH SAI (18R11A0515)

L JAYA KRISHNA (18R11A0523)



Department of Computer Science and Engineering
Accredited by NBA

Geethanjali College of Engineering and Technology
(UGC Autonomous)

(Affiliated to J.N.T.U.H, Approved by AICTE, New Delhi)

Cheeryal (V), Keesara (M), Medchal Dist.-501 301
April 2020

Geethanjali College of Engineering and Technology
(UGC Autonomous)

(Affiliated to J.N.T.U.H, Approved by AICTE, New Delhi)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Accredited by NBA

CERTIFICATE

This is to certify that the B.Tech Major Project report entitled “**Voice based Image Editor**” is a bonafide work done by G.VARUN KUMAR(18R11A0511),K.NITESH SAI (18R11A0515), L.JAYA KRISHNA(18R11A0523) in partial fulfillment of the requirement of the award for the degree of Bachelor of Technology in “Computer Science and Engineering” from Jawaharlal Nehru Technological University, Hyderabad during the year 2019-2020.

Dr.Sesham Anandh,
Professor
MVSR College of Engineering

Mr.Krishna Teja
Pakcet Prep Team, Head Internal Guide

ACKNOWLEDGEMENT

We are greatly indebted to the Management of Geethanjali College of Engineering and Technology, Cheeryal, Hyderabad, for providing us the necessary facilities to successfully carry out this mini project work titled “**Voice based Image Editor**”.

Firstly, we thank and express our solicited gratitude to **Dr. DSR MURTHY, HOD, CSE department, Geethanjali College of Engineering and Technology** and **Sunil Sir, Computer Society of India , Hyderabad**, for his invaluable help and support which helped us a lot in successfully completing our Internship.

Moreover, we also express our gratitude to **Professor Dr.Sesham Anandh, MVSR College of Engineering and Trainer Mr.Krishna Teja, Packet Prep Team** our guide and patron, for his continued support throughout our endeavour to make our project successfully done.

We would like to express our sincere gratitude to our **Principal Dr.S. UDAY KUMAR** for providing the necessary infrastructure to complete our project.

We convey our gratitude to our **Chairman, Mr. G. RAVINDER REDDY**, for his invaluable support and encouragement for propelling our innovation forward.

Finally, we would like to express our heartfelt gratitude to our parents and all our peers who were very supportive and for their encouragement to achieve our goals.

G.VARUN KUMAR(18R11A0511),

K.NITHESH SAI(18R11A0515),

L.JAYA KRISHNA(1R11A0523)

ABSTRACT

The project aims in dealing with Images that can be edited using voice commands. Voice recognition is one of the hands-free technologies which can make a manual task much easier. The project has a voice recognition feature through which images can be edited. User must give the voice command as instructed in the application and get the desired outputs. An Authorization System is provided where the user must register his details and later login. After a successful login, User can edit the images using voice commands. Images can be converted to black and white or grayscale. They can also be cropped, rotated, resized and flipped by using their respective voice commands. Images are displayed and also saved. User can then logout by using the voice command “logout” and exit the application. Microphone is necessary to use so as to execute the program.

LIST OF FIGURES

Figure no.	Figure Description	Page no.
1.1	Working of the project	2
3.1	MVC Architecture	8
4.1	System Architecture	11
4.2	Use case Diagram	13
4.3	Class Diagram	17
4.4	State Diagram	18
4.5	Component Diagram	19
4.6	Deployment Diagram	20
4.7	System Design	21
5.1	Basic Levels of Testing	24
5.2	Black Box Testing	25

LIST OF ABBREVIATIONS

S.no	Abbreviation	Full form
1	IDE	Integrated Development Environment
2	UML	Unified Modeling Language
3	DFD	Data-flow diagram
4	PIL	Python image library

LIST OF IMAGES

S.no	Name	Page no.
1	Login	31
2	Registration	32
3	Main Layout	33
4	Color Window	34
5	Modify Window	35
6	Black and White	36

7	Grayscale	37
8	Image Rotated	38
9	Image resized	39
10	Crop	40
11	Flip	41

TABLE OF CONTENTS

S.No	Contents	Page no
	Abstract	ii
	List of Figures	iii
	List of Abbreviations	iv
	List of Screenshots	iv
1.	Introduction.....	1
	1.1 Objective	2
2.	System Analysis.....	3
	2.1 Existing System	3
	2.2 Proposed System	3
	2.2.1 Details	3
	2.2.2 Impact of Environment	4
	2.2.3 Feasibility Study	4
	2.3 Modules Description	6
3.	Literature Overview.....	7
	3.1 Project Literature	7
	3.2 Introduction to Python	7
	3.2.1 Python Technology	8
	3.2.2 MVC Architecture	9
	3.2.3 Library Specific To Project	10
4.	System Design.....	11

4.1 System Architecture	12
4.2 UML Diagrams	13
4.3 System Design	21
5. Testing.....	23
5.1 Testing	23
5.2 Test cases	29
6. Output Screens	31
7. Conclusion	42
7.1 Further Scope	43
8. Bibliography... ..	44
8.1 Website References	44
8.2 Technical Publication References	44
9. Appendices.....	45
9.1 SW used	45
9.2 Methodologies used	48
9.3 Testing Methods used	49
9.4 Sample code	49
10. Plagiarism Report.....	56

CHAPTER – 1

INTRODUCTION

AIM: To build a Voice based image editor application which can optimize, edit, and color the images. This project deals with a hands free technology which is Voice Recognition.

In today's world, the importance of images on e-commerce, travel, and media websites has been increasing over the time. Optimizing web images is a process of delivering high-quality images in the right format, dimension, size, and resolution while keeping the smallest possible size. Image editor can be done in different ways, be it by resizing the images, caching, or by compressing the size. Image editor helps in improving page load speed, boosts websites' SEO ranking, and improves user experience. The file format used for an image affects how large the file is. Most images for the web should be in JPEG format, not PNG or GIF. This is because it's easiest to adjust the quality (which affects the file size) with JPEG files. JPEG files lose visual information when they are compressed. As a result, compression can shrink JPEG files to a fraction of their original sizes, which is usually not possible with GIF and PNG files (both are lossless). Image search engine editor and image editor are separate, but related. In this project voice recognition technology enables users to edit images quickly and efficiently. Voice recognition is a computer software program or hardware device with the ability to decode the human voice. Hands-free technology like voice recognition makes the task easier.

Voice recognition is a computer software program or hardware device with the ability to decode the human voice. Voice recognition is commonly used to operate a device, perform commands, or write without having to use a keyboard, mouse, or press any buttons. Today, this is done on a computer with ASR (automatic speech recognition) software programs. Many ASR programs require the user to "train" the ASR program to recognize their voice so that it can more accurately convert the speech to text. For example, you could say "open Internet" and the computer would open the Internet browser. This Voice recognition is implemented in the project.

The user can interact with the application by giving specific commands and optimize or edit the image accordingly.

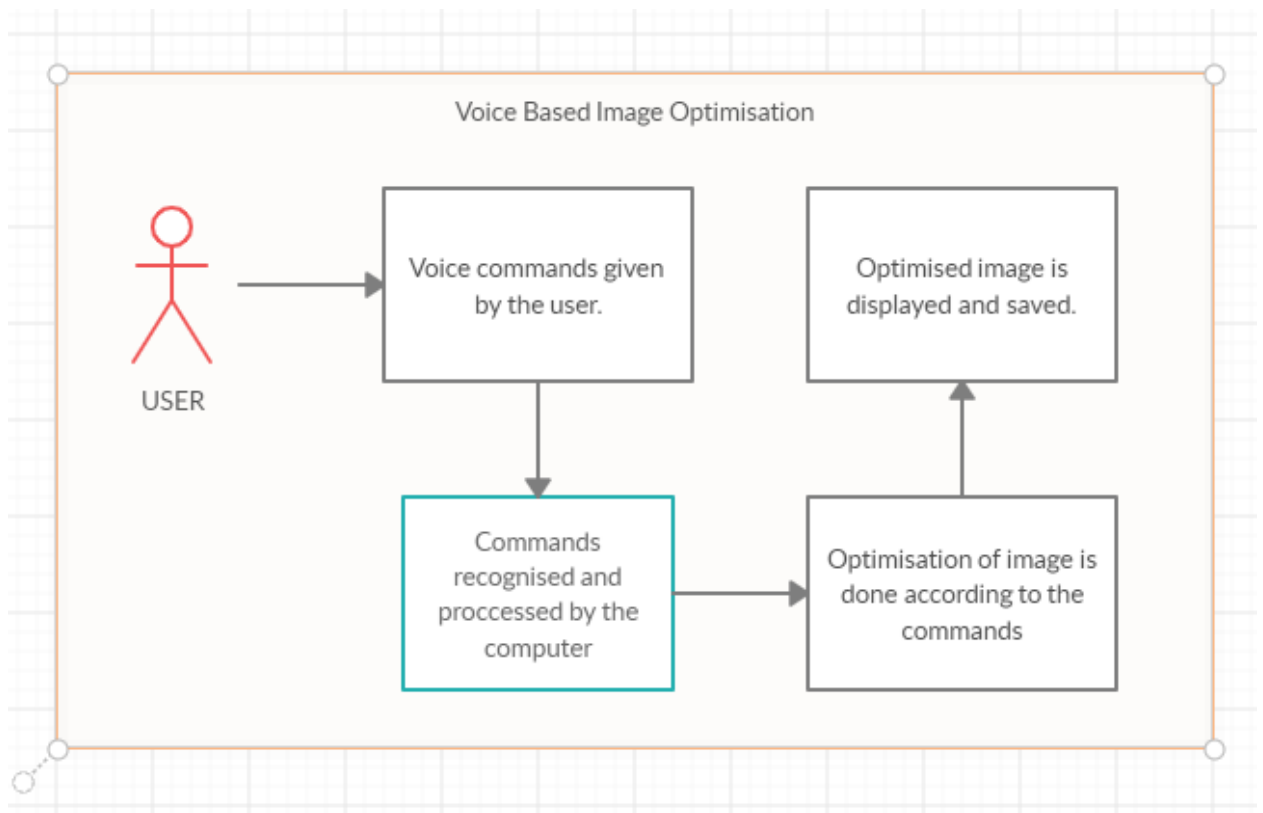


Fig 1.1 Working of the project

In the above figure 1.1, it explains how the user can interact with the system. User must give voice commands through microphone; the commands are recognized by the computer and are converted to text. Then the images are edited or optimized. Edited Images are then displayed and saved.

1.1 OBJECTIVES

- Voice based image editor is used to modify and color images.
- Images can be cropped, resized, rotated, flipped, and converted to black and white, grayscale using voice.
- Security is ensured to the designed application.

CHAPTER – 2

SYSTEM ANALYSIS

2.1 Existing System

The existing system in an Image editor software is basic editing of any image file without voice recognition technology. For example, user can edit images using online software like Optimizilla, PIXLR manually through mouse or keyboard. This software is not equipped with voice recognition technology.

Disadvantages:

- No security feature. Anyone can have access to the app.
- Hands-free technology like voice recognition is not equipped.

2.2 Proposed System

The proposed system is more robust and need an Authorization to access the application. Voice recognition is enabled so that user can interact with the application and edit the images quickly.

Advantages:

- Image file can be converted to required size.
- Users who register can have access to the app.
- Enhancement of the image is possible.
- Features like crop, rotate are available.
- Both mouse and Voice recognition can be used to control the application.

2.2.1 Details

Software requirements:

Operating System	:	Windows family
Technology	:	Python 3.8
IDE	:	PyCharm, Visual Studio

2.2.2 Impact on Environment

Image editor helps in improving page load speed, boosts websites' SEO ranking, and improves user experience. Improves Page Load Speed, Page load speed is the amount of time taken by a web page to load completely. It depends on many factors ranging from your website host to website layout and design. The websites having less than 2 seconds load speed are most loved by its users. So, if you are optimizing 64% of your website's weight, which is images, you will be improving your website speed. Using Voice Recognition, this enables hands free technology. There are many people with visual impairments who rely on screen readers and text-to-speech dictation systems. And converting audio into text can be a critical communication tool for the hearing-impaired.

2.2.3 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 FEASIBILITY

ECONOMICAL FEASIBILITY

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

TECHNICAL FEASIBILITY

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

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

2.3 Module Description

1 Image Editor

This Module is used to optimize images. Features like crop, rotate, flip, resize are available.

2 Database Module

This Module is used to grant or deny user access to the app. Authorization is given to the users after they register.

3 Tkinter Module

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps –

- Import the Tkinter module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.

Enter the main event loop to take action against each event triggered by the user.0

CHAPTER – 3

LITERATURE OVERVIEW

3.1 PROJECT LITERATURE

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

3.2. INTRODUCTION TO PYTHON

[1][1]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, Smalltalk, 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.

3.2.1 Python Technology

[1][5]Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

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 cross-platform 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** – you 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.

3.2.2 MVC Architecture:

There are two major Python versions: **Python 2** and **Python 3**. Both are quite different.

Model View Controller is the most commonly used design pattern. Developers find it easy to implement this design pattern.

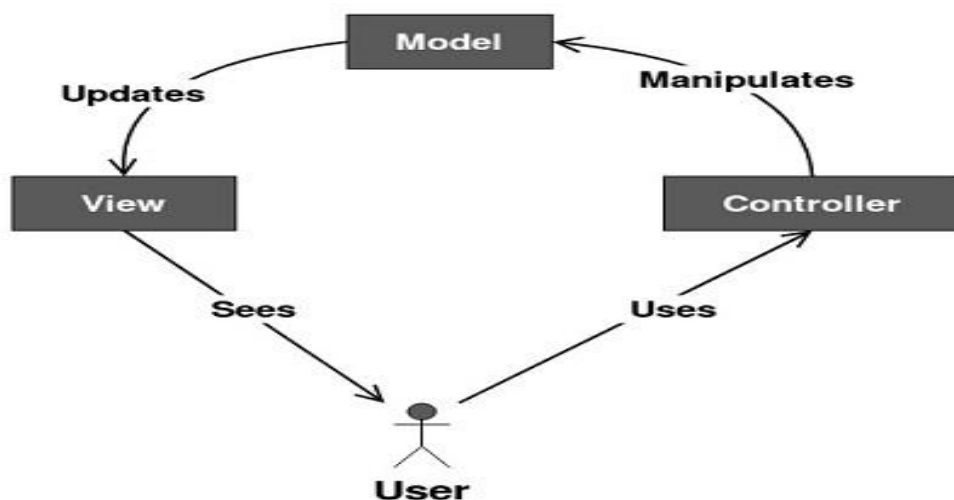


Fig 3.1 MVC architecture

Model

It consists of pure application logic, which interacts with the database. It includes all the information to represent data to the end user.

View

View represents the HTML files, which interact with the end user. It represents the model's data to user.

Controller

It acts as an intermediary between view and model. It listens to the events triggered by view and queries model for the same.

3.2.3 Libraries Specific to project:

3.2.3.1 Pillow library: Pillow is the friendly PIL fork by Alex Clark and Contributors. PIL is the Python Imaging Library by Fredrik Lundh and Contributors.

Installation: Install Pillow with **pip**:

```
pip install Pillow
```

3.2.3.2 Speech Recognition: Library for performing speech recognition, with support for several engines and APIs, online and offline.

Speech recognition engine/API support:

- CMU Sphinx (works offline)
- Google Speech Recognition
- Google Cloud Speech API
- Wit.ai
- Microsoft Bing Voice Recognition
- Houndify API
- IBM Speech to Text
- Snowboy Hotword Detection (works offline)

Installation: Install Speech Recognition with **pip**:

```
pip install SpeechRecognition
```

3.2.3.3 PyAudio module: [1][4] PyAudio provides Python bindings for PortAudio v19, the cross-platform audio I/O library. With PyAudio, you can easily use Python to play and record audio streams on a variety of platforms (e.g., GNU/Linux, Microsoft Windows, and Mac OS X).

Installation: Install Pillow with **pip**:

pip install PyAudio

CHAPTER – 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

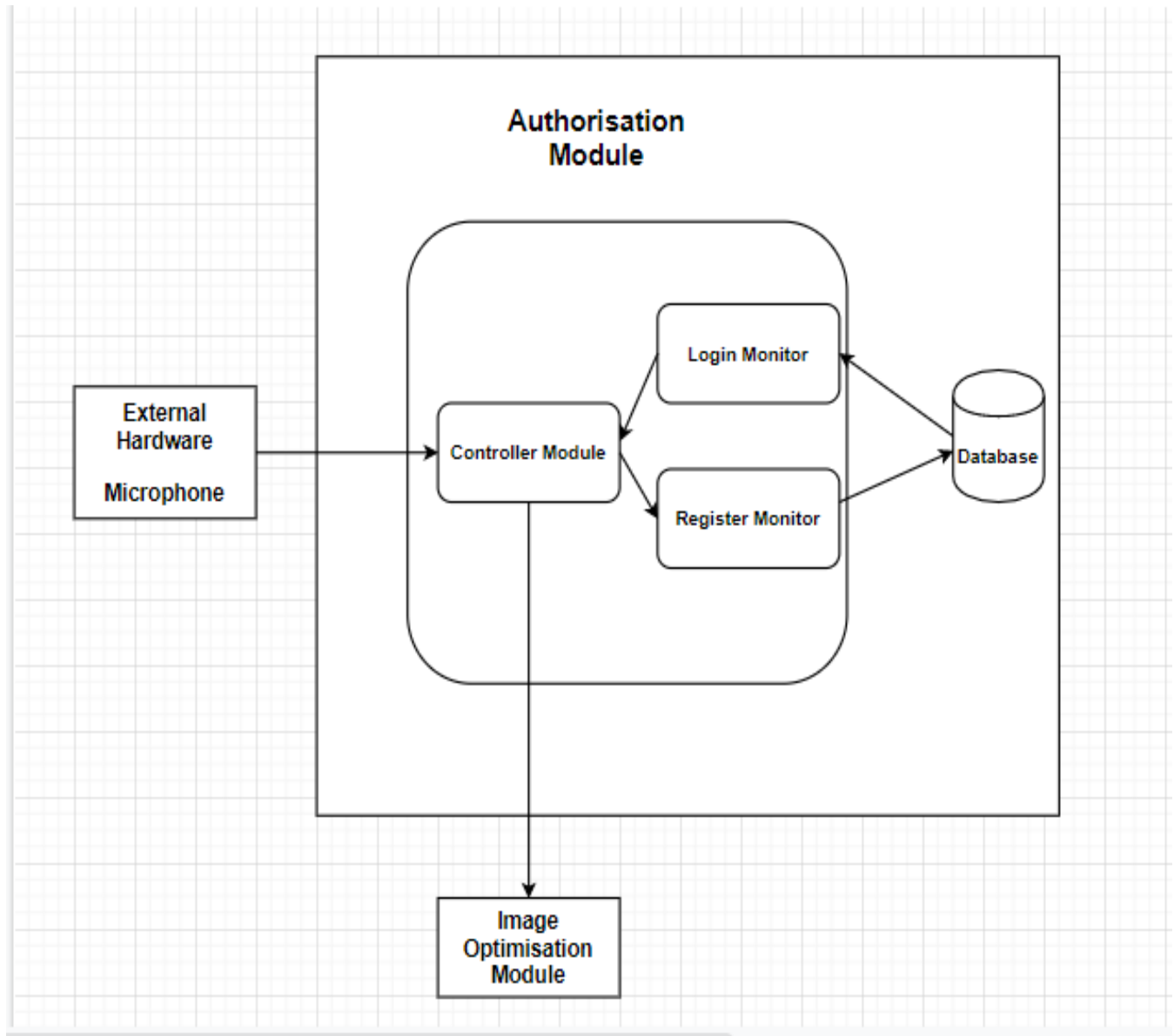


Fig 4.1 System Architecture

The above diagram explains the flow of System architecture which consists of some modules, an external hardware and a database. The modules are classified into

Authorization Module:

- **Controller module:** This module swaps the control to different modules as per the user commands.
- **Login module:** This module provides access to the application for the registered users.
- **Registration module:** Users can register their details in this module.

Database:

[1][2]The database used here is **SQLITE 3**.

- The database module is used to validate the data of the users during the process of login. The database consists of a table named users which has five fields.
- Name: This is a name field where the name of the user is stored in form of string.
- Username: This is a username field where the username of the user is stored in form of string.
- Id: A unique id is provided by default to the users in form of integers and auto increment option is enabled.
- Phone: Phone number provided by the user is stored in form of integers.
- Email: email id of the user is stored.
- Password: Password of the user is stored in the database in form of string. Characters are hidden in this field.
-

Image Optimizer Module:

The Images Optimizer module provides a simple way to optimize the images which are uploaded by the user

Every uploaded image is automatically optimized according to the given command. The original uploaded image will not be destroyed in the process.

Microphone as external hardware:

Microphone is an hardware source that is required for our project to listen to our command and recognize it.

By using this microphone our commands will be converted into text and given as an event to the image conversion.

4.2 UML DIAGRAMS

Use Case Diagram

Description

A usage case outline inside the Unified Modeling Language we utilized in our Project Development is Star (UML) can be a kind of behavioral chart portrayed out by way of and produced using a Use-case examination. Its thought is to give a graphical layout of the presence of thoughts given through a machine to the volume acting professionals, their objectives (addressed as use cases), and any conditions between those use instances. The best explanation at the back of a use case diagram is to reveal what shape limits are performed out of that on-display screen character. Parts of the entertainers in the gadget could be diagrams.

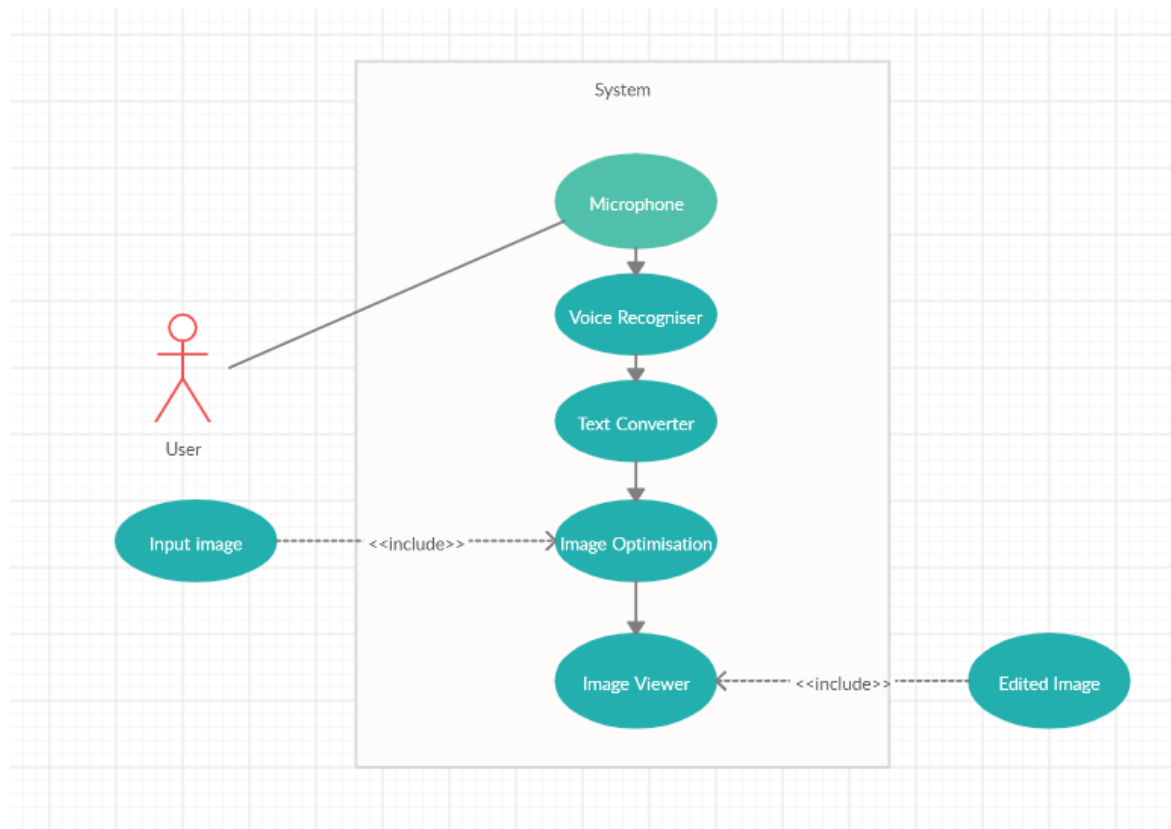


Fig 4.2 Use case diagram

The above figure explains the flow of the program execution. User must use microphone to give the commands. The commands are then recognized by the voice recognizer and are converted to text. The system prompts for input image. The user then must upload the image file path with the image name and extension. Image Editor Module then edits the image according to the voice commands given by the user. The images are then displayed to the user and saved.

Class Diagram

Description

In PC code making plans, a class plot within the Unified Modeling Language we utilized in our Project Development is Star (UML) could be a kind of static shape outline that delineates the shape of a device by using showing the system's characterizations, their characteristics, operations (or methodologies), and furthermore the associations the various groupings. It elucidates that elegance carries statistics.

Identification of analysis classes:

A class is a set of objects that share a common structure and common behavior (the same attributes, operations, relationships and semantics). A class is an abstraction of real-world items.

There are 4 approaches for identifying classes:

- a. Noun phrase approach:
- b. Common class pattern approach.
- c. Use case Driven Sequence or Collaboration approach.
- d. Classes, Responsibilities and collaborators Approach

1. Noun Phrase Approach

The guidelines for identifying the classes

- Look for nouns and noun phrases in the use cases.
- Some classes are implicit or taken from general knowledge.
- All classes must make sense in the application domain; Avoid computer implementation classes – defer them to the design stage.
- Carefully choose and define the class names After identifying the classes we have

to eliminate the following types of classes:

- Adjective classes.

2. Common class pattern approach

The following are the patterns for finding the candidate classes:

- Concept class.
- Events class.
- Organization class
- Peoples class
- Places class
- Tangible things and devices class.

3. Use case driven approach

We must draw the sequence diagram or collaboration diagram. If there is need for some classes to represent some functionality then add new classes which perform those functionalities.

4. CRC approach:

The process consists of the following steps:

- Identify classes' responsibilities (and identify the classes)
- Assign the responsibilities
- Identify the collaborators.

Guidelines for identifying the tentative associations:

- A dependency between two or more classes may be an association. Association often corresponds to a verb or prepositional phrase.
- A reference from one class to another is an association. Some associations are implicit or taken from general knowledge.

1. Top-down:

Look for noun phrases composed of various adjectives in a class name. Avoid excessive refinement. Specialize only when the sub classes have significant behavior.

2. Bottom-up:

Look for classes with similar attributes or methods. Group them by moving the common attributes and methods to an abstract class. You may have to alter the definitions a bit.

3. Reusability:

Move the attributes and methods as high as possible in the hierarchy.

4. Multiple inheritances:

Avoid excessive use of multiple inheritances. One way of getting benefits of multiple inheritances is to inherit from the most appropriate class and add an object of another class as an attribute.

Aggregation or a-part-of relationship:

It represents the situation where a class consists of several component classes. A class that is composed of other classes doesn't behave like its parts. It behaves very differently. The major properties of this relationship are transitivity and anti-symmetry.

- Does the part class belong to the problem domain?
- Is the part class within the system's responsibilities?
- Does the part class capture more than a single value?(If not then simply include it as an attribute of the whole class)
- Does it provide a useful abstraction in dealing with the problem domain?

There are three types of aggregation relationships. They are:

Assembly:

It is constructed from its parts and an assembly-part situation physically exists.

Container:

A physical whole encompasses but is not constructed from physical parts.

Collection member:

A conceptual whole encompasses parts that may be physical or conceptual. The container and collection are represented by hollow diamonds, but composition is represented by solid diamond.

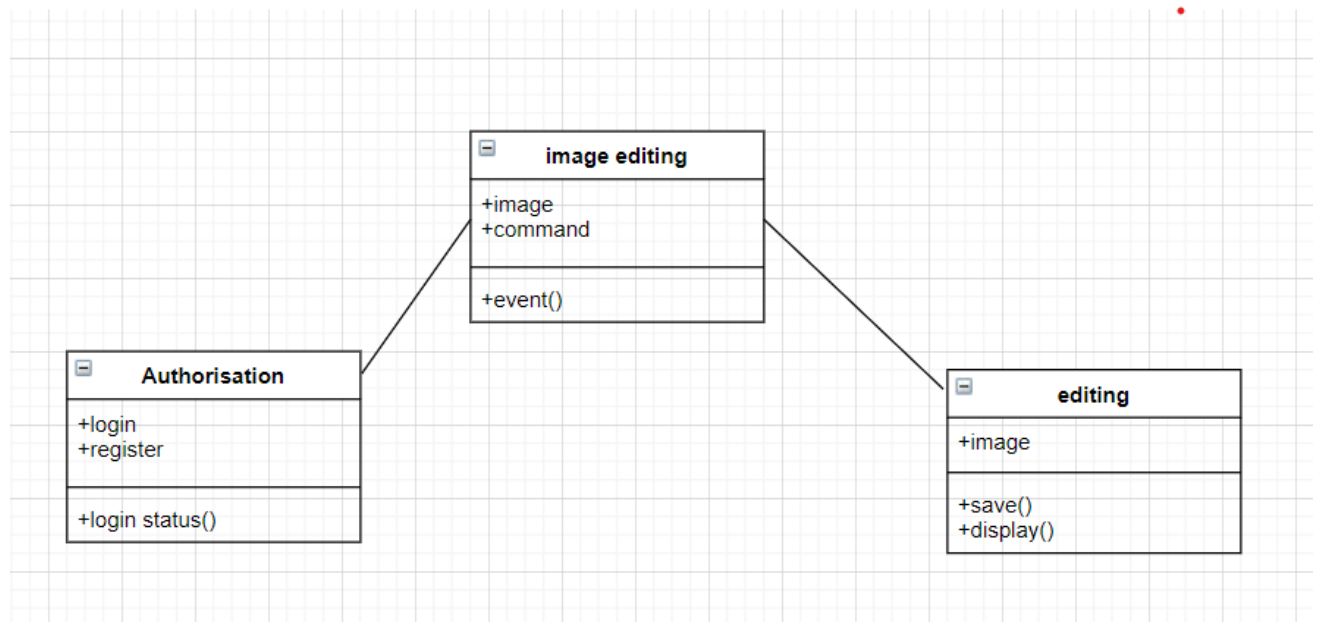


Fig 4.3 class diagram

Here the class diagram explains the three main steps in the project, they are :

- i. Authorization: Authorization: Here in authorization there is a process to get access, first we have to register then the login will be successful. The status of our login will be popped up as message
- ii. Voice command: We have to give the voice command according to the given options in the menu bar that the image will be sent to the controller according to the event name that has been given through voice command.
- iii. Image conversion: .Image will be converted to the given event. The converted image will be saved and displayed on the screen

C State Chart

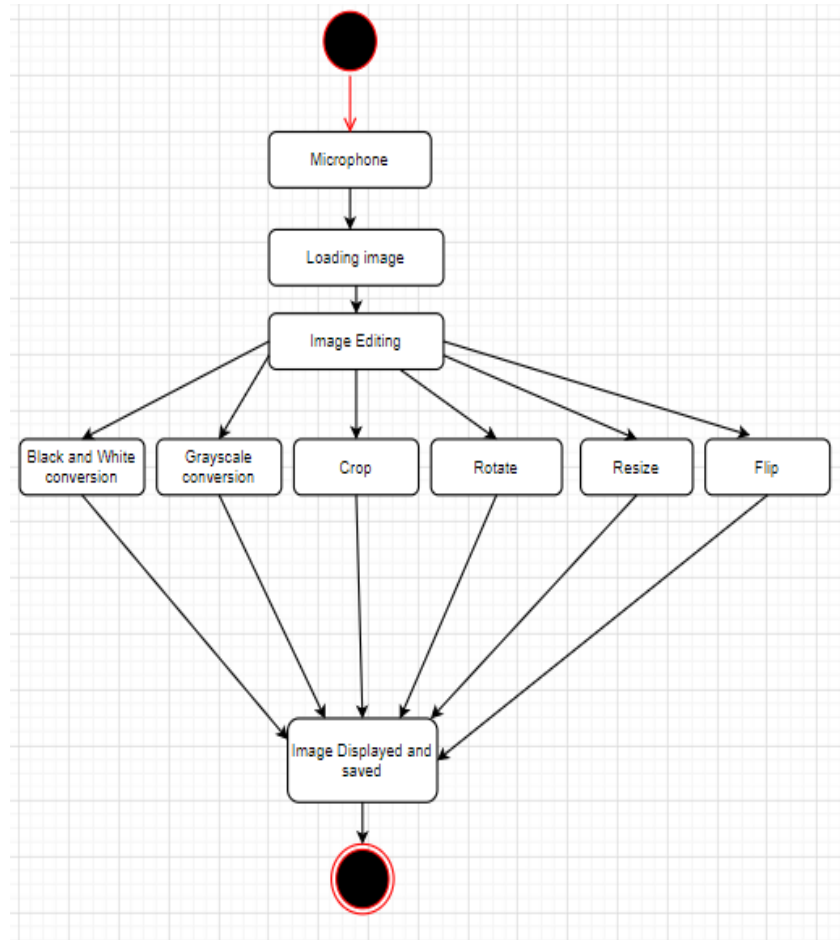


Fig 4.4 State Chart

Description

Kingdom graph (nation machine outline or state chart chart) A nation graph, likewise known as a kingdom system graph or state chart outline, will be a example of the states and problem will accomplish and additionally the advances among those states inside the Unified Modeling Language we used in our Project Development is Star (UML).

F Component Diagram

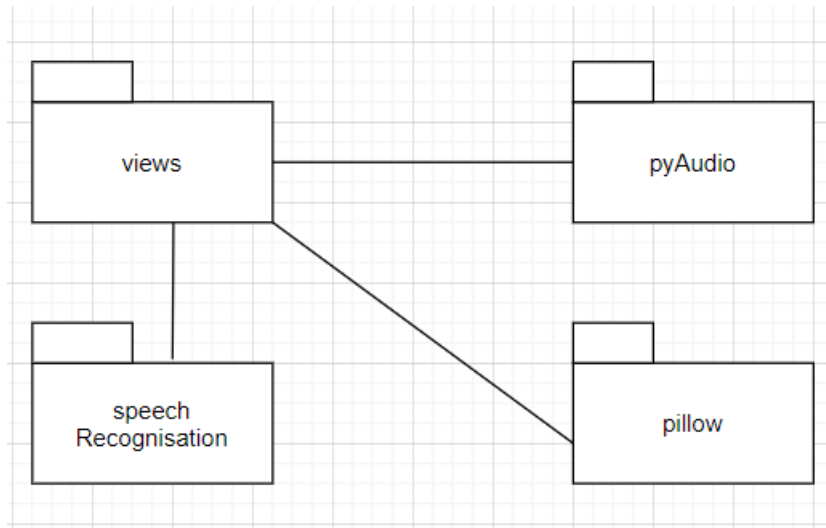


Fig 4.5 Component Diagram

Description

A part graph, otherwise known as AN UML section chart, depicts the affiliation and wiring of the bodily division in a completely framework. Phase graphs are usually interested by facilitate exhibit usage factors of hobby and twofold watch that every part of the framework's wanted capacity is secured by prepared development. Within the primary fashion of UML, factors incorporated into these charts had been bodily: data, records table, files, and executable, every bodily element with a place. Within the world of UML, a couple of, these elements aren't maximum bodily but rather several calculated stay solitary outline components, for example, an enterprise process that gives or expects interfaces to speak with completely unique develops inside the framework.

G Deployment Diagram



Fig 4.6 Deployment Diagram

Description

A UML deployment diagram is a diagram that shows the configuration of run time processing nodes and the components that live on them. Deployment diagrams are a kind of structure diagram used in modeling the physical aspects of an object-oriented system. They are often be used to model the static deployment view of a system (topology of the hardware)

4.3 SYSTEM DESIGN

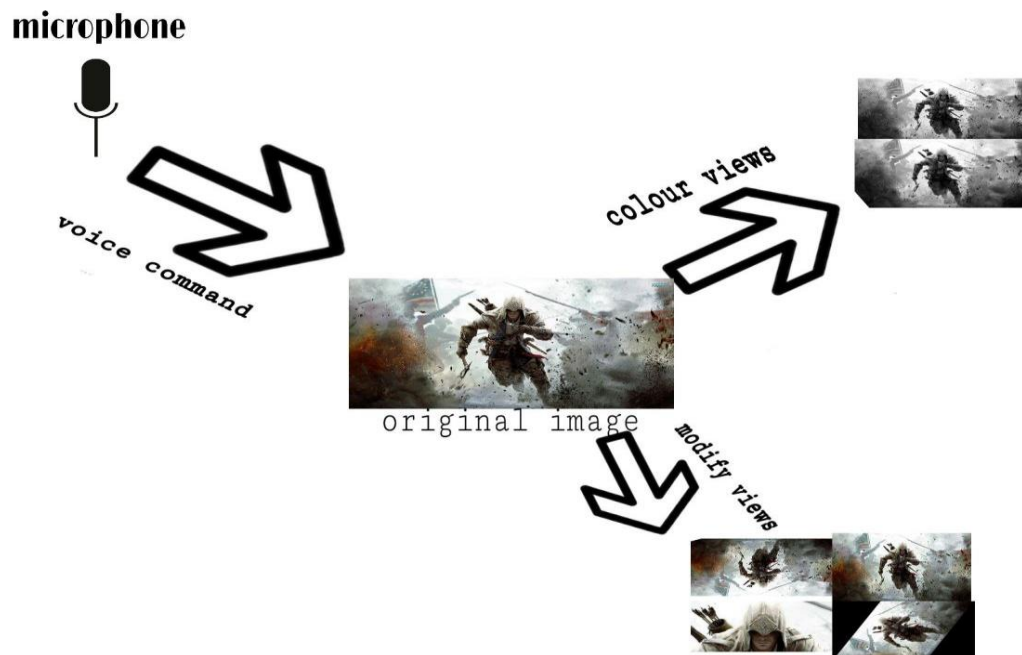


fig 4.7 System Design

The above figure explains the design of the system which consists of several sections. The microphone is used to take in the voice commands. Later according to the voice commands the uploaded image is transformed. The color view converts the image to black and white and the modify view edits the image. Image can be cropped, rotated, flipped, resized in modify view

CLASSIFICATION OF SYSTEM DESIGN

- After the authorization process, the next task is to load the image and edit the image accordingly.
- Initially the photo or the image can be uploaded by giving the path.

Load

- In this phase, the path of the image should be provided in the empty box. The button load must be clicked and voice command “load” must be given so as to load the image.

Color View

- After loading the image, user can open the color window by using the voice command “color” and convert the image to black and white, grayscale.
- If the voice is not recognized, the user must try the process again.

Modify View

- The user can then exit the color window by giving the voice command “exit” after clicking the button exit.
- The user now return to main window and can open the modify window by giving the command “modify”.
- Operations like crop, rotate, flip, resize can now be performed.

CHAPTER – 5

TESTING

5.1 Testing

Testing is the debugging program is one of the most critical aspects of the computer programming triggers, without programming that works, the system would never produce an output of which it was designed. Testing is best performed when user development is asked to assist in identifying all errors and bugs. The sample data are used for testing. It is not quantity, but quality of the data used the matters of testing. Testing is aimed at ensuring that the system was accurately an efficiently before live operation commands.

Testing objectives:

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say, testing is a process of executing a program with intent of finding an error.

- A successful test is one that uncovers an as yet undiscovered error.
- A good test case is one that has probability of finding an error, if it exists.
- The test is inadequate to detect possibly present errors.
- The software more or less confirms to the quality and reliable standards.

Levels of Testing:

In order to uncover present in different phases we have the concept of levels of testing.

The Basic Levels of Testing

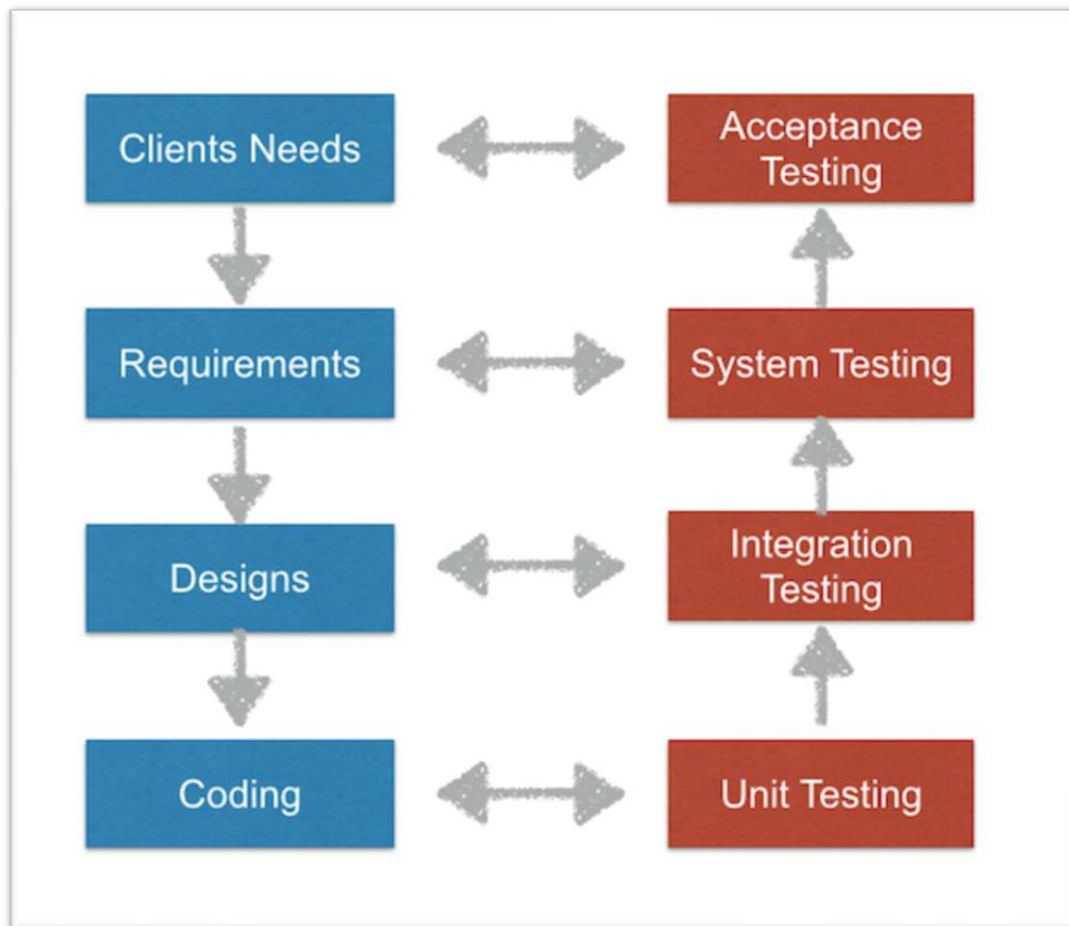


Fig 5.1 Basic levels of Testing

Unit testing:

In the unit testing we test each module individually and integrate with the overall system. Unit testing focuses verification efforts on the smallest unit of software design in the module. This is also known as module testing. The module of the system is tested separately. This testing is carried out during programming stage itself. In the testing step each module is found to work satisfactorily as regard to expected output from the module. There are some validation checks for fields also. For example, the validation check is done for varying the user input given by the user which validity of the data entered. It is very easy to find error debut the system.

Each Module can be tested using the following two Strategies:

1. Black Box Testing
2. White Box Testing

BLACK BOX TESTING

Black box testing is a software testing technique in which functionality of the software under test (SUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on the software requirements and specifications.

In Black Box Testing we just focus on inputs and output of the software system without bothering about internal knowledge of the software program.



Fig 5.2

The above Black Box can be any software system you want to test. For example: an operating system like Windows, a website like Google, a database like Oracle or even your own custom application. Under Black Box Testing, you can test these applications by just focusing on the inputs and outputs without knowing their internal code implementation.

Black box testing - Steps

Here are the generic steps followed to carry out any type of Black Box Testing.

- Initially requirements and specifications of the system are examined.
- Tester chooses valid inputs (positive test scenario) to check whether SUT processes them correctly. Also some invalid inputs (negative test scenario) are chosen to verify that the SUT is able to detect them.
- Tester determines expected outputs for all those inputs.

- Software tester constructs test cases with the selected inputs.
- The test cases are executed.
- Software tester compares the actual outputs with the expected outputs.
- Defects if any are fixed and re-tested.

Types of Black Box Testing

There are many types of Black Box Testing but following are the prominent ones -

- **Functional testing** – This black box testing type is related to functional requirements of a system; it is done by software testers.
- **Non-functional testing** – This type of black box testing is not related to testing of a specific functionality, but non-functional requirements such as performance, scalability, usability.
- **Regression testing** – Regression testing is done after code fixes , upgrades or any other system maintenance to check the new code has not affected the existing code.

WHITE BOX TESTING

White Box Testing is the testing of a software solution's internal coding and infrastructure. It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability. White box testing is also known as clear, open, structural, and glass box testing.

It is one of two parts of the "box testing" approach of software testing. Its counterpart, black box testing, involves testing from an external or end-user type perspective. On the other hand, Whitebox testing is based on the inner workings of an application and revolves around internal testing. The term "white box" was used because of the see-through box concept. The clear box or white box name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "black box testing" symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested

What do you verify in White Box Testing?

White box testing involves the testing of the software code for the following:

- Internal security holes
- Broken or poorly structured paths in the coding processes
- The flow of specific inputs through the code
- Expected output

- The functionality of conditional loops
- Testing of each statement, object and function on an individual basis

The testing can be done at system, integration and unit levels of software development. One of the basic goals of white box testing is to verify a working flow for an application. It involves testing a series of predefined inputs against expected or desired outputs so that when a specific input does not result in the expected output, you have encountered a bug. To give a simplified explanation of white box testing, we have divided it into two basic steps. This is what testers do when testing an application using the white box testing technique:

STEP 1) UNDERSTAND THE SOURCE CODE

The first thing a tester will often do is learn and understand the source code of the application. Since white box testing involves the testing of the inner workings of an application, the tester must be very knowledgeable in the programming languages used in the applications they are testing. Also, the testing person must be highly aware of secure coding practices. Security is often one of the primary objectives of testing software. The tester should be able to find security issues and prevent attacks from hackers and naive users who might inject malicious code into the application either knowingly or unknowingly.

Step 2) CREATE TEST CASES AND EXECUTE

The second basic step to white box testing involves testing the application's source code for proper flow and structure. One way is by writing more code to test the application's source code. The tester will develop little tests for each process or series of processes in the application. This method requires that the tester must have intimate knowledge of the code and is often done by the developer. Other methods include manual testing, trial and error testing and the use of testing tools as we will explain further on in this article.

Integration Testing:

Data can be lost across an interface, one module can have an adverse effect on the other sub functions, when combined, may not produce the desired major functions. Integrated testing is the systematic testing for constructing the uncover errors within the interface. The testing was done with sample data. The developed system has run successfully for this sample data. The need for integrated test is to find the overall system performance.

System testing:

Once the individual module testing is completed, modules are assembled and integrated to perform as a system. The top down testing, which began from upper level to lower level module, was carried out to check whether the entire system is performing satisfactorily.

There are three main kinds of System testing:

- i. Alpha Testing
- ii. Beta Testing
- iii. Acceptance Testing

Alpha Testing:

This refers to the system testing that is carried out by the test team with the Organization.

Beta Testing:

This refers to the system testing that is performed by a selected group of friendly customers.

Acceptance Testing:

This refers to the system testing that is performed by the customer to determine whether or not to accept the delivery of the system.

Code testing:

This examines the logic of the program. For example, the logic for updating various sample data and with the sample files and directories were tested and verified.

Specification Testing:

Executing this specification starting what the program should do and how it should performed under various conditions. Test cases for various situation and combination of conditions in all the modules are tested.

Output testing:

After performance of the validation testing, the next step is output testing. The output displayed or generated by the system under consideration is tested by asking the user about the format required by system.

5.2 Test Cases

Code testing, Output testing were implemented and executed for this project and the test cases are shown below.

Test Case 1	
Test Case Name	Image conversion to black and white
Description	After giving the voice command “black and white” image is converted to black and white.
Output	Converted image is displayed and saved.

Test Case 2	
Test Case Name	Image conversion to gray scale
Description	After giving the voice command “grayscale” image is converted to grayscale.
Output	Converted image is displayed and saved.

Test Case 3	
Test Case Name	Image resized to 50%
Description	After giving the voice command “resize” image is resized to 50% of the original size.
Output	Resized image is displayed and saved.

Test Case 4	
Test Case Name	Image rotation
Description	After giving the voice command “rotate” image is rotated to the angle according to the given voice command.
Output	Rotated image is displayed and saved.

Test Case 5	
Test Case Name	Image flip
Description	After giving the voice command “flip” image is inverted.
Output	Inverted image is displayed and saved.

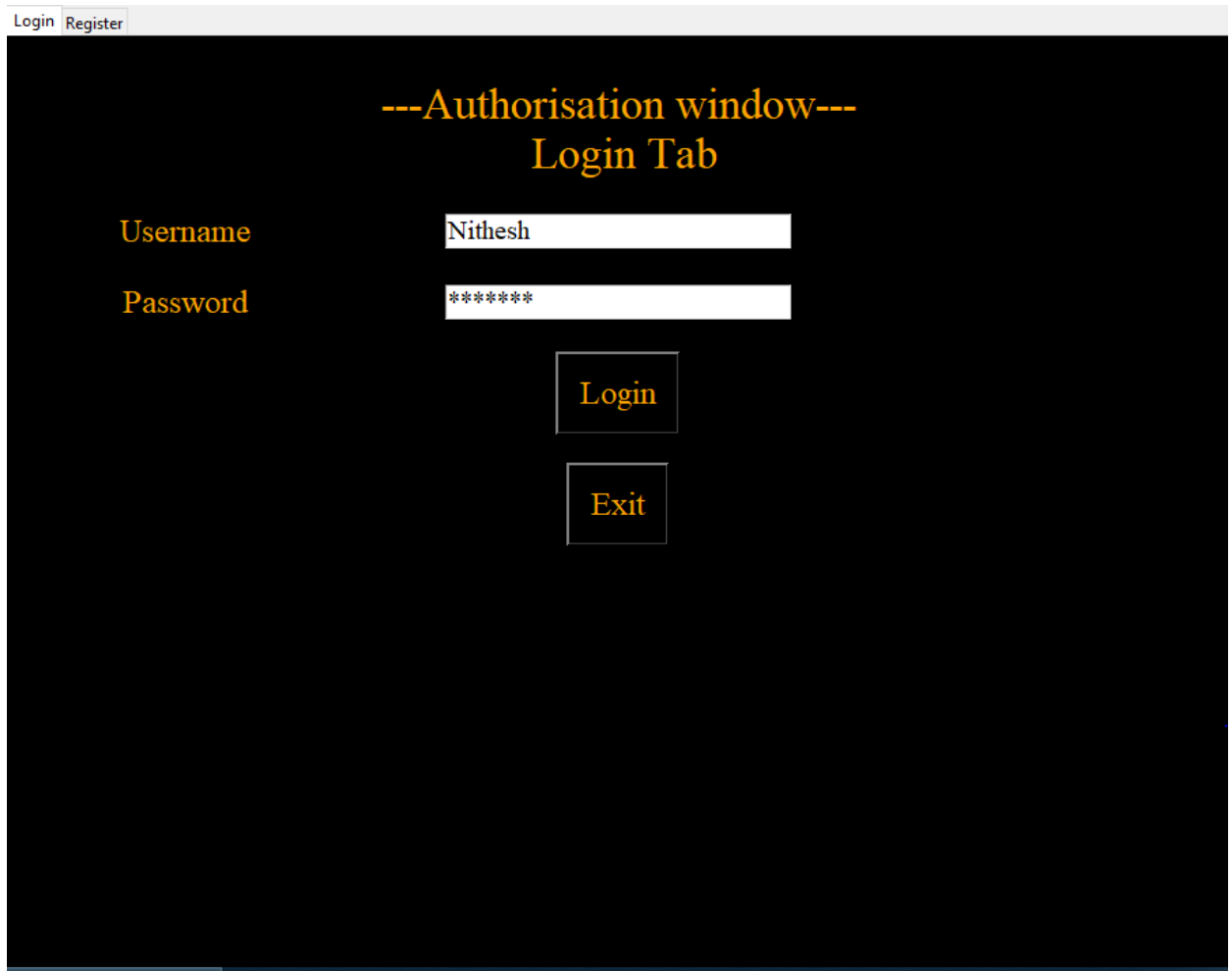
Test Case 6	
Test Case Name	Image cropping
Description	After giving the voice command “crop” image is cropped according to the given voice command. Either 50% crop or 25 % crop.
Output	Cropped image is displayed and saved.

CHAPTER – 6

OUTPUT

SCREENS

Login



The screenshot shows a web application interface for login. At the top, there are two tabs: 'Login' (active) and 'Register'. The main content area has a black background with orange text. The title is '---Authorisation window---' followed by 'Login Tab'. Below this, there are two input fields: 'Username' with the value 'Nithesh' and 'Password' with masked characters '*****'. At the bottom, there are two buttons: 'Login' and 'Exit'.

Login Register

---Authorisation window---
Login Tab

Username Nithesh

Password *****

Login

Exit

User should enter his login credentials in order to access the app. If user enter the wrong details then it prompts an error saying user not found.

Registration

[Login](#) [Register](#)

---Authorisation window---

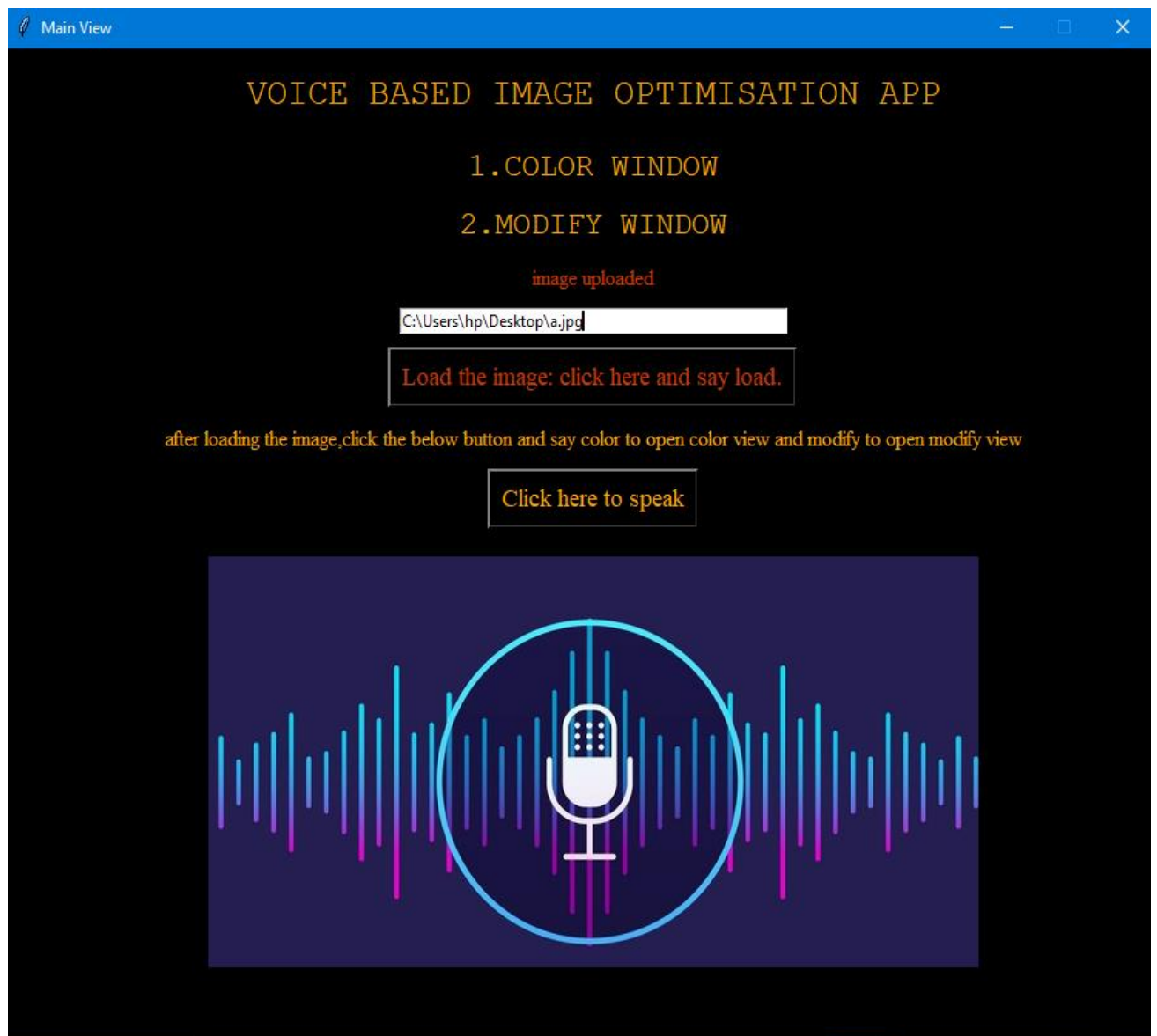
Register Tab

Name	<input type="text" value="Nithesh"/>
Username	<input type="text" value="Nithesh"/>
Password	<input type="password" value="*****"/>
Email	<input type="text" value="nitheshk@gmail.com"/>
Phone	<input type="text" value="912374819280"/>

Register

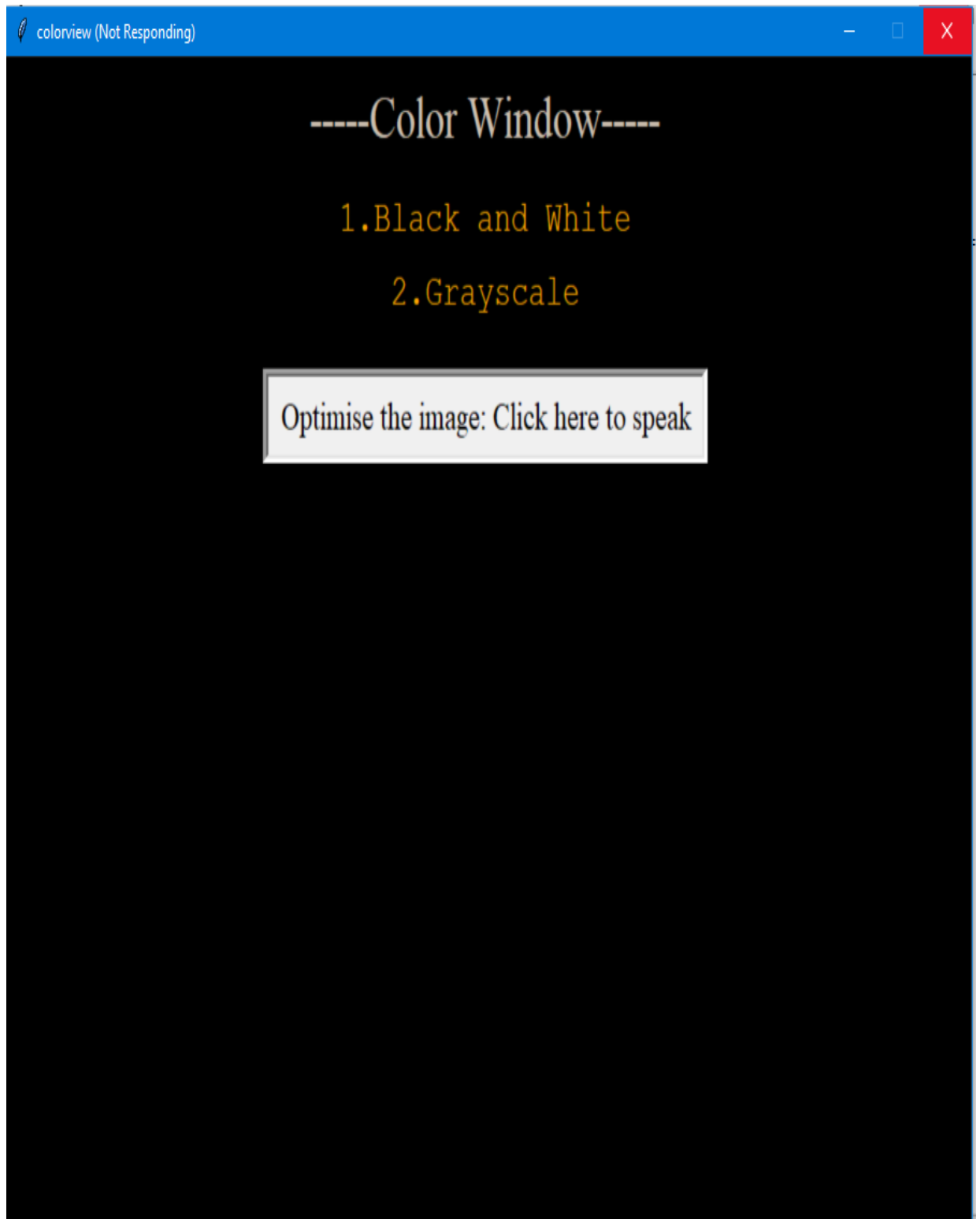
must register his details in this tab. User can navigate to login tab or register tab by clicking the tabs. In register tab user must register his/her details and then login. A message is popped up displaying “you can login now”.

Home Page:



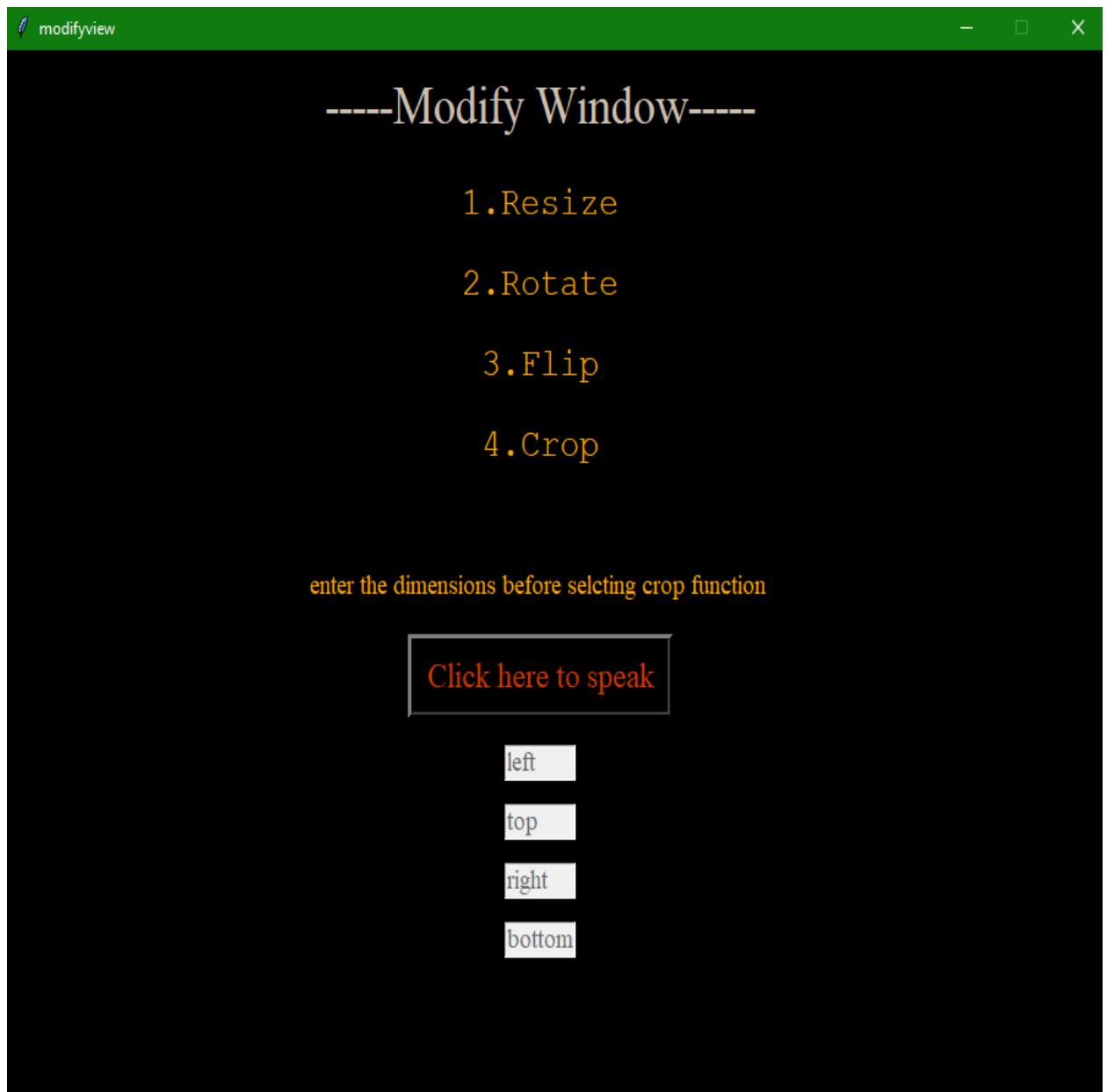
After a successful login, user must upload the image path in white box along with the image file name and extension. User then must click the “load the image” button and say load. The image will be uploaded in few seconds. Now the user can edit the image by opening color window or modify window. User must speak color or modify to open the window after clicking the “click here to speak” button.

Color Window:



This screen will appear once the user gives the voice command “Black and White”. In this window the user can convert the image to black and white or grayscale . The button must be clicked before giving the voice command.

Modify Window



In this window, the user can resize, rotate, flip and crop by using the voice commands resize, rotate, resize and flip respectively. Before cropping the image the user must enter the dimensions and call crop function. The dimensions must be given perfectly according to the image. Use paint in order to find the dimensions of the part of the image.

Black and White:



After giving the voice command “Black and White”, the image is converted to black and white. It is then displayed and saved

Grayscale:



After giving the voice command “Grayscale”, the image is converted to grayscale.

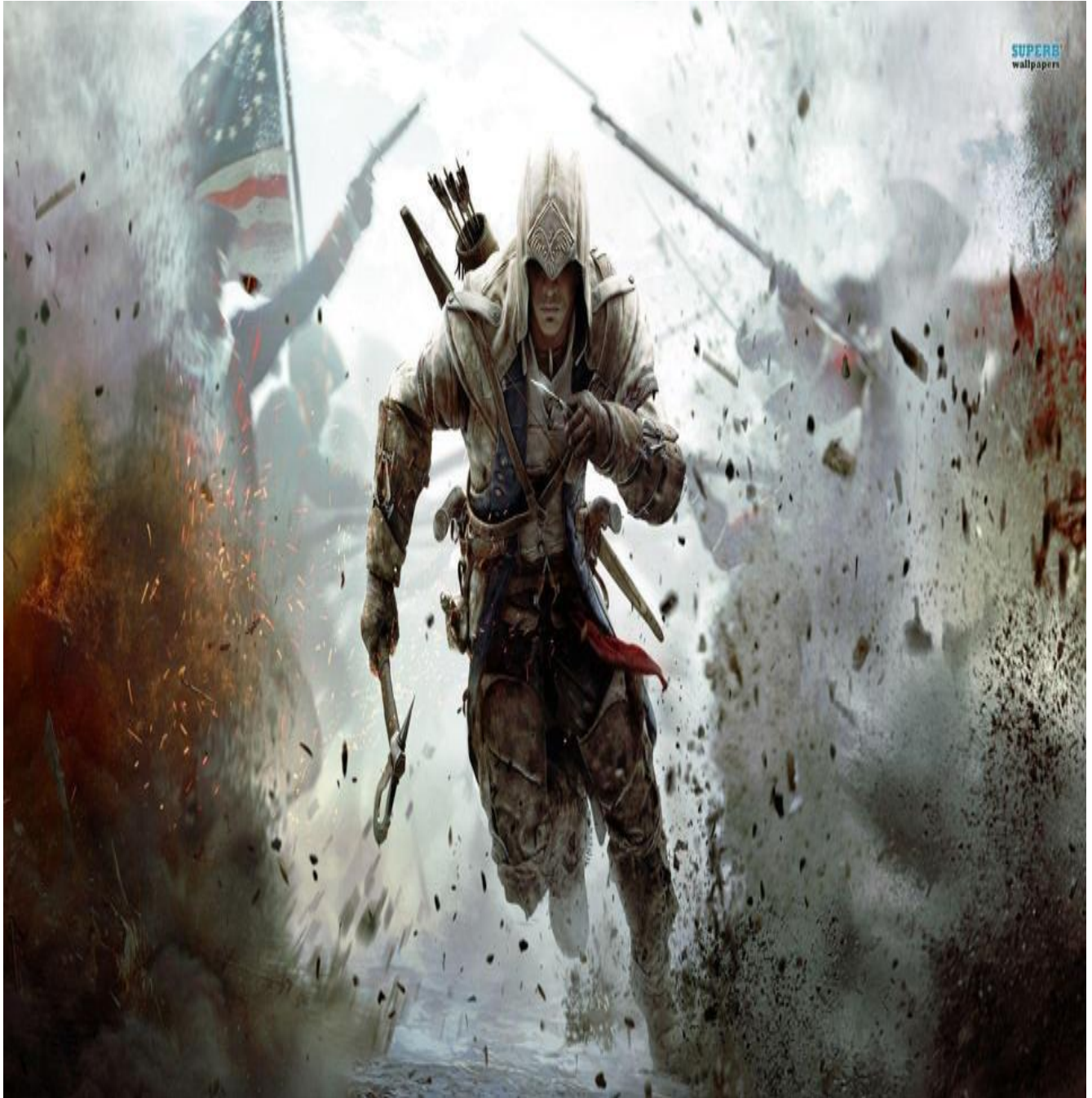
It is then displayed and saved

Image rotated:



After giving the voice command “Rotate”, the image is rotated to an angle mentioned by the user. This image is then displayed and saved.

Image resizes:



After the voice command resize, the image is resized to half of its size. The image pixels size is also reduced to half of its original size.

Image Crop:



The dimensions must be provided before cropping the image. The image is cropped according to the dimensions given by the user after saying “crop”. Image is then displayed and saved.

Image Flip



Image is flipped after saying the voice command “flip” by the user. Image is then displayed and saved.

CHAPTER –7

CONCLUSION

This project is a low cost, voice based image editor application has been proposed based on Voice Recognition. It can detect user voice in a rapid manner .Using the voice commands given by the user, images can be optimized accordingly. Feature like crop, rotate, resize, flip, black and white and grayscale conversion are available. The system works well even in case of disturbance or noise in our surroundings. An adaptive thresholding technique has been developed to detect voice command and optimize images in real time. . Because of this, the system can decide if the voice is recognized or not. If the voice is not recognized by any of our event, then it shows a message saying “invalid command, try again”. Using Voice based Image Optimizer; it can be implemented without any physical work. This is a hands-free technology.

BENEFITS

- Image file can be converted to required size.
- Users who register can have access to the app.
- Enhancement of the image is possible.
- Features like crop, rotate are not available.
- Images can be saved.
- Improves page load speed and user experience.

LIMITATIONS

- Image loading takes time and uploading an image needs a path or file location.
- Images with the same file name will be replaced.
- Multiple Images cannot be optimized at the same time.
- Voice command must be loud and clear with a proper accent.
- Conversion of image to Black and White and Greyscale increases the image size.

7.1 FUTURE SCOPE OF THE PROJECT

[2][1]This voice based image optimizer does contain few more features than the required ones, like conversion of an image to black and white and grayscale. This Image Optimizer poses a problem in Color conversion as it increases the image size instead of reducing it. In today's world large image files cannot be shared through emails, so this app can be used to reduce the image file size and can be shared. Web pages Page load speed is the amount of time taken by a web page to load completely. It will be depended on many factors from your website host to website layout and design. The websites which have less than 2 seconds load speed are most loved by its users. So, if you want to optimize 64% of your website's weight, which is images, you will be improving your speed of the website. Many studies stated that users leave those websites which are slow. The web pages which load in 2.5 seconds experience 25% bounce rate.. Therefore, user satisfaction and happiness depends on page load speed, which can be improved by optimizing web images.

CHAPTER – 8

BIBLIOGRAPHY

8.1 Website References

- i. Python Documentation - [https://www.python.org/doc\[1\]\[1\]](https://www.python.org/doc[1][1])
- ii. MySQL Documentation - [https://dev.mysql.com/doc\[1\]\[2\]](https://dev.mysql.com/doc[1][2])
- iii. [https://www.tutorialspoint.com/python/index.htm\[1\]\[3\]](https://www.tutorialspoint.com/python/index.htm[1][3])
- iv. [https://www.pyimagesearch.com/\[1\]\[4\]](https://www.pyimagesearch.com/[1][4])
- v. [https://www.javatpoint.com/python-tutorial\[1\]\[5\]](https://www.javatpoint.com/python-tutorial[1][5])

8.2 Technical Publication References

- i. [https://ieeexplore.ieee.org/document/255190\[2\]\[1\]](https://ieeexplore.ieee.org/document/255190[2][1])
- ii. [https://ieeexplore.ieee.org/document/1051901\[2\]\[2\]](https://ieeexplore.ieee.org/document/1051901[2][2])

CHAPTER – 9

APPENDICES

9.1 Software Used

- **PyCharm**

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated together to create a convenient environment for productive Python, web, and data science development.

PyCharm is available in three editions: Professional, Community, and Edu. The Community and Edu editions are open-source projects and they are free, but they have less features. PyCharm Edu provides courses and helps you learn programming with Python. The Professional edition is commercial and provides an outstanding set of tools and features. In addition, in the Professional edition, one can develop Django, Flask, and Pyramid applications. Also, it fully supports HTML (including HTML5), CSS, JavaScript, and XML: these languages are bundled in the IDE via plugins and are switched on for you by default. Support for the other languages and frameworks can also be added via plugins.

Plugins Used in PyCharm

PyAudio:

Classes

PyAudio, Stream

Host Specific Classes

PaMacCoreStreamInfo

Stream Conversion Convenience Functions

get_sample_size(), get_format_from_width()

PortAudio version

get_portaudio_version(), get_portaudio_version_text()

Portaudio Sample Formats

paFloat32, paInt32, paInt24, paInt16, paInt8, paUInt8, paCustomFormat

PortAudio Host APIs

paInDevelopment, paDirectSound, paMME, paASIO, paSoundManager, paCoreAudio, paOSS, paALSA, paAL, paBeOS, paWDMKS, paJACK, paWASAPI, paNoDevice

PortAudio Error Codes

paNoError, paNotInitialized, paUnanticipatedHostError, paInvalidChannelCount, paInvalidSampleRate, paInvalidDevice, paInvalidFlag, paSampleFormatNotSupported, paBadIODeviceCombination, paInsufficientMemory, paBufferTooBig, paBufferTooSmall, paNullCallback, paBadStreamPtr, paTimedOut, paInternalError, paDeviceUnavailable, paIncompatibleHostApiSpecificStreamInfo, paStreamIsStopped, paStreamIsNotStopped, paInputOverflowed, paOutputUnderflowed, paHostApiNotFound, paInvalidHostApi, paCanNotReadFromACallbackStream, paCanNotWriteToACallbackStream, paCanNotReadFromAnOutputOnlyStream, paCanNotWriteToAnInputOnlyStream, paIncompatibleStreamHostApi

PortAudio Callback Return Codes

paContinue, paComplete, paAbort

PortAudio Callback Flags

paInputUnderflow, paInputOverflow, paOutputUnderflow, paOutputOverflow, paPrimingOutput.

Pillow:

Python Imaging Library (abbreviated as PIL) (in newer versions known as Pillow) is a free and open-source additional library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats. It is available for Windows, Mac OS X and Linux. The latest version of PIL is 1.1.7, was released in September 2009 and supports Python 1.5.2–2.7, with Python 3 support to be released "later".

Development appears to be discontinued, with the last commit to the PIL repository coming in 2011. Consequently, a successor project called Pillow has forked the PIL repository and added Python 3.x support. This fork has been adopted as a replacement for the original PIL in Linux distributions including Debian and Ubuntu.

Pillow offers several standard procedures for image manipulation. These include:

- per-pixel manipulations,
- masking and transparency handling,
- image filtering, such as blurring, contouring, smoothing, or edge finding,
- image enhancing, such as sharpening, adjusting brightness, contrast or color,

- adding text to images and much more.

Speech Recognition:

Library for performing speech recognition, with support for several engines and APIs, online and offline.

Speech recognition engine/API support:

- CMU Sphinx (works offline)
- Google Speech Recognition
- Google Cloud Speech API
- Wit.ai
- Microsoft Bing Voice Recognition
- Houndify API
- IBM Speech to Text
- Snowboy Hotword Detection (works offline)

Quickstart: pip install SpeechRecognition. See the “Installing” section for more details.

To quickly try it out, run `python -m speech_recognition` after installing.

Project links:

- [PyPI](#)
- [Source code](#)
- [Issue tracker](#)

MYSQL

MySQL is an open-source relational database management system and used to store and retrieve the computed values in the database.

Using MySQL Connector, we establish the connection between the program and MySQL database. After establishing connection, we compute the training set values by visual behavior and store them in the database.

During Machine learning Phase we retrieve the database values and store them locally in csv format and we use machine learning algorithms to predict the Driver Drowsiness.

Library Reference

The library reference documents every publicly accessible object in the library. This document is also included under `reference/library-reference.rst`.

See Notes on using PocketSphinx for information about installing languages, compiling PocketSphinx, and building language packs from online resources. This document is also included under `reference/pocketsphinx.rst`.

9.2 Methodologies Used

Pillow library: Pillow is the friendly PIL fork by Alex Clark and Contributors. PIL is the Python Imaging Library by Fredrik Lundh and Contributors. Pillow is a Python Imaging Library (PIL), which adds support for opening, manipulating, and saving images. The current version identifies and reads a large number of formats. Write support is intentionally restricted to the most commonly used interchange and presentation formats.

Speech recognition: Speech recognition has its roots in research done at Bell Labs in the early 1950s. Early systems were limited to a single speaker and had limited vocabularies of about a dozen words. Modern speech recognition systems have come a long way since their ancient counterparts. They can recognize speech from multiple speakers and have enormous vocabularies in numerous languages. The first component of speech recognition is, of course, speech. Speech must be converted from physical sound to an electrical signal with a microphone, and then to digital data with an analog-to-digital converter. Once digitized, several models can be used to transcribe the audio to text. Most modern speech recognition systems rely on what is known as a Hidden Markov Model (HMM). This approach works on the assumption that a speech signal, when viewed on a short enough timescale (say, ten milliseconds), can be reasonably approximated as a stationary process—that is, a process in which statistical properties do not change over time. In a typical HMM, the speech signal is divided into 10-millisecond fragments. The power spectrum of each fragment, which is essentially a plot of the signal's power as a function of frequency, is mapped to a vector of real numbers known as cepstral coefficients. The dimension of this vector is usually small—sometimes as low as 10, although more accurate systems may have dimension 32 or more. The final output of the HMM is a sequence of these vectors. To decode the speech into text, groups of vectors are matched to one or more phonemes—a fundamental unit of speech. This calculation requires training, since the sound of a phoneme varies

from speaker to speaker, and even varies from one utterance to another by the same speaker. A special algorithm is then applied to determine the most likely word (or words) that produce the given sequence of phonemes. One can imagine that this whole process may be computationally expensive. In many modern speech recognition systems, neural networks are used to simplify the speech signal using techniques for feature transformation and dimensionality reduction *before* HMM recognition. Voice activity detectors (VADs) are also used to reduce an audio signal to only the portions that are likely to contain speech. This prevents the recognizer from wasting time analyzing unnecessary parts of the signal

9.3 Testing Methods

- **Microphone processing:**

At first, the voice command is given through microphone. The microphone will be the internal hardware of the system. From the voice commands, image modification will be done. The edited or optimized image will be saved and displayed on the screen.

9.4 SAMPLE CODE:

Mainlayout.py:

```
from tkinter import *
from tkinter import messagebox
from PIL import ImageTk, Image
import speech_recognition as sr
import time
import threading
from views.colorlayout import colorlayout
from views.modifylayout import modifylayout
from views.AuthView import AuthView

class MyApp:

    def load(self):
        av = AuthView()
        av.transfer_control = self.mainlayout
        av.load()

    def mainlayout(self):
        cl = colorlayout()
        ml = modifylayout()
        self.main = Tk()
        self.main.title("Main View")
```

```

self.main.resizable(0,0)
self.main['background']='black'
w = 900
h = 750
ws = self.main.winfo_screenwidth()
hs = self.main.winfo_screenheight()
x = (ws/2) - (w/2)
y = (hs/2) - (h/2)

self.main.geometry('%dx%d+%d+%d' % (w, h, x, y))

l0 = Label(self.main,text = "VOICE BASED IMAGE EDITOR APP",fg="Orange", bg="black",font=('Courier', 20))
l0.grid(row=2, column=5, padx=170, pady=15)
l1 = Label(self.main,text = "1.COLOR WINDOW",fg="Orange", bg="black",font=("Courier", 18))
l1.grid(row=4, column=5, padx=110, pady=5)
l2 = Label(self.main,text = "2.MODIFY WINDOW",fg="Orange", bg="black",font=("Courier", 18))
l2.grid(row=6, column=5, padx=110, pady=5)
r=sr.Recognizer()
l3 = Label(self.main,text="after loading the image,click the below button\nsay color to open color view and modify to open modify view\n say logout to logout the application",fg="Orange", bg="black",font=("Times", 12))
l3.grid(row=14, column=5, padx=110, pady=5)
l4 = Label(self.main,text = "Enter the image path below to load the image\nClick the button and say load.",fg="OrangeRed3", bg="black",font=('Times', 12))
l4.grid(row=8, column=5, padx=110, pady=5)
var1= StringVar()
e1 = Entry(self.main,width=50,textvariable=var1).grid(row = 10,column = 5,padx=110, pady=5)

def button():
    with sr.Microphone() as source:
        try:
            audio=r.listen(source,phrase_time_limit=4)
            #a=l3.config(text="waiting for the command")
            # tc = threading.Thread(target=a,args=())
            # tc.start()
            text = r.recognize_google(audio)
            if text == 'color':
                a = l3.config(text="Recent window opened: color window")
                t = threading.Thread(target=a,args=())
                t.start()
                cl.color()
            elif text == 'modify':
                a = l3.config(text="Recent window opened: modify window")
                t = threading.Thread(target=a,args=())
                t.start()

```

```

        m1.modifyview()
    elif text == 'logout':
        MsgBox = messagebox.askquestion ('Logout Application','Are you
sure you want to Logout the application',icon = 'warning')
        if MsgBox == 'yes':
            self.main.destroy()
            exit()
        else:
            messagebox.showinfo('Return','You will now return to the a
pplication screen')
    else:
        raise sr.UnknownValueError

except sr.UnknownValueError:
    l3.config(text="voice is not audible.try again")
    print('voice is not audible')

def button2():
    r1=sr.Recognizer()
    with sr.Microphone() as source:
        try:
            print('waiting for the command')
            audio = r1.listen(source,phrase_time_limit=2)
            l3.config(text="waiting for the command")
            l3.config(text="waiting for the command")
            text = r1.recognize_google(audio)
            if text=='load':
                a = l4.config(text="uploading the image")
                t = threading.Thread(target=a,args=())
                t.start()
                time.sleep(4)
                im = Image.open((str(var1.get())))
                print("image loaded")
                im.save("image.jpg")
                b = l4.config(text="image uploaded")
                t1 = threading.Thread(target=b,args=())
                t1.start()
            else:
                raise sr.UnknownValueError

        except sr.UnknownValueError:
            l3.config(text="voice is not audible.Try again")
            print('voice is not audible')

```

```

        b2 = Button(self.main,text="Load the image: click here and say load.",fg="OrangeRed3", bg="black",bd = 3,highlightthickness=4,font=("Times", 14),command=button2)
        b2.grid(row=12, column=5, padx=110, pady=5)

        b = Button(self.main,text="Click this button",fg="Orange", font=("Times", 14),
        bg="black",bd = 3,highlightthickness=4,command=button)
        b.grid(row=16, column=5, padx=110, pady=5)

        path = "mainpic.png"
        img = ImageTk.PhotoImage(Image.open(path))
        panel = Label(self.main, image=img,width=600,height=300)
        panel.photo = img
        panel.grid(row=18, column=5, padx=160, pady=15)

app = MyApp()
app.load()

```

Authview:

```

from tkinter import *
from tkinter import ttk
from PIL import ImageTk
import os
from tkinter import messagebox
from controllers.AuthController import AuthController

class AuthView:

    next = None
    def load(self):
        self.window = Tk()
        self.window.resizable(0,0)
        self.window['background']='black'
        w = 900
        h = 750
        ws = self.window.winfo_screenwidth()
        hs = self.window.winfo_screenheight()
        x = (ws/2) - (w/2)
        y = (hs/2) - (h/2)

        self.window.geometry('%dx%d+%d+%d' % (w, h, x, y))
        self.window.overrideredirect(True)
        self.window.title("Authorisation")
        tab_control = ttk.Notebook(self.window,width=900, height=750)

        self.login_tab = Frame(tab_control,bg="black",padx=20,pady=20)
        self.register_tab = Frame(tab_control,bg="black",padx=20,pady=20)

```

```

tab_control.add(self.login_tab, text="Login")
tab_control.add(self.register_tab, text = "Register")

self.login()
self.register()
tab_control.grid()

self.window.mainloop()

def register(self):
    window = self.register_tab
    l0 = Label(window, text = "VOICE BASED IMAGE EDITOR APP", fg="Orange", bg="black", font=('Courier', 23))
    l0.grid(row=0, column=1, padx=1, pady=5)
    t1 = Label(window, text="---Authorisation window---\n Register Tab", fg="Orange", bg="black", font=("Times", 20))
    t1.grid(row=1, column=1, padx=1, pady=10)

    # Name label and entry
    n1 = Label(window, text="Name", fg="Orange", bg="black", font=("Times", 18))
    n1.grid(row=2, column=0, padx=30, pady=10)

    ne = Entry(window, width=25, fg="black", bg="white", font=("Times", 15))
    ne.grid(row=2, column=1, padx=2, pady=10)

    # username label and entry
    ul = Label(window, text="Username", fg="Orange", bg="black", font=("Times", 18))
    ul.grid(row=3, column=0, padx=30, pady=10)

    ue = Entry(window, width=25, fg="black", bg="white", font=("Times", 15))
    ue.grid(row=3, column=1, padx=2, pady=10)

    # password name and entry
    pl = Label(window, text="Password", fg="Orange", bg="black", font=("Times", 18))
    pl.grid(row=4, column=0, padx=30, pady=10)

    pe = Entry(window, show="*", width=25, fg="black", bg="white", font=("Times", 15))
    pe.grid(row=4, column=1, padx=2, pady=10)

    # email label and entry
    el = Label(window, text="Email", fg="Orange", bg="black", font=("Times", 18))
    el.grid(row=5, column=0, padx=30, pady=10)

    ee = Entry(window, width=25, fg="black", bg="white", font=("Times", 15))
    ee.grid(row=5, column=1, padx=2, pady=10)

    # Phone label and entry
    ph1 = Label(window, text="Phone", fg="Orange", bg="black", font=("Times", 18))

```

```

        ph1.grid(row=6, column=0,padx=30, pady=10)

        phe = Entry(window, width=25,fg="black", bg="white",font=("Times", 15))
        phe.grid(row=6, column=1,padx=2, pady=10)

        b1 = Button(window, text="Register",fg="Orange",bd = 3,highlightthickness=4,bg="black",font=("Times", 18),command = lambda: self.registerControl(ne.get(), phe.get(), ee.get(), ue.get(), pe.get()), padx=5,pady=5)
        b1.grid(row=7, column=1,padx=0,pady=5)

    def login(self):
        window = self.login_tab

        l0 = Label(window,text = "VOICE BASED IMAGE EDITOR APP",fg="Orange", bg="black",font=('Courier', 23))
        l0.grid(row=0, column=1, padx=1, pady=5)

        ti = Label(window, text="---Authorisation window---\n Login Tab",fg="Orange", bg="black",font=("Times", 20))
        ti.grid(row=1, column=1,padx=30, pady=10)

        ul = Label(window,text="Username",fg="Orange", bg="black",font=("Times", 18))
        ul.grid(row=2,column=0,padx=30, pady=10)

        ue = Entry(window,width=25,fg="black", bg="white",font=("Times", 15))
        ue.grid(row=2,column=1,padx=2, pady=10)

        pl = Label(window, text="Password",fg="Orange", bg="black",font=("Times", 18))
        pl.grid(row=3, column=0,padx=30, pady=10)

        pe = Entry(window, show="*", width=25,fg="black", bg="white",font=("Times", 15))
        pe.grid(row=3, column=1,padx=2, pady=10)

        b1 = Button(window,text="Login",fg="Orange", bg="black",bd = 3,highlightthickness=4,font=("Times", 18),command=lambda: self.loginControl( ue.get() ,pe.get()), padx=5,pady=5)
        b1.grid(row=4,column=1,padx=30, pady=10)

        b2 = Button(window,text="Exit",fg="Orange", bg="black",bd = 3,highlightthickness=4,font=("Times", 18),command=lambda: self.exit(),padx=5,pady=5)
        b2.grid(row=5,column=1,padx=30, pady=10)

```

```

        linfo= Label(window,text ="Register and Login to acess the App",fg="Orange", bg="black",font=('Times', 16))
        linfo.grid(row=6, column=1, padx=0, pady=10)

        linfo1= Label(window,text ="Click the Register tab above to register",fg="Orange", bg="black",font=('Times', 16))
        linfo1.grid(row=7, column=1, padx=0, pady=10)

    def loginControl(self,username,password):
        ac = AuthController()
        message = ac.login(username,password)
        if message == 1:
            self.window.destroy()
            self.transfer_control()
            #self.next()
        else:
            messagebox.showinfo('Message',message)

    def registerControl(self,name,phone,email,username,password):

        ac = AuthController()
        message = ac.register(name,phone,email,username,password)
        messagebox.showinfo('Message', message)

    def exit(self):
        MsgBox = messagebox.askquestion ('Exit Application','Are you sure you want to exit the application',icon = 'warning')
        if MsgBox == 'yes':
            exit()
        else:
            messagebox.showinfo('Return','You will now return to the application screen')

av = AuthView()

```


CHAPTER – 10

PLAGIARISM REPORT



PLAGIARISM SCAN REPORT

Date	June 12, 2020	Words	185
------	---------------	-------	-----

Exclude URL:

96%
Unique

4%
Plagiarized

Content Checked for Plagiarism:

This voice based image optimizer does contain few more features than the required ones, like conversion of an image to black and white and grayscale. This Image Optimizer poses a problem in Color conversion as it increases the image size instead of reducing it. In today's world large image files cannot be shared through emails, so this app can be used to reduce the image file size and can be shared. Web pages Page load speed is the amount of time taken by a web page to load completely. It will be depended on many factors from your website host to website layout and design. The websites which have less than 2 seconds load speed are most loved by its users. So, if you want to optimize 64% of your website's weight, which is images, you will be improving your speed of the website. Many studies stated that users leave those websites which are slow. The web pages which load in 2.5 seconds experience 25% bounce rate.. Therefore, user satisfaction and happiness depends on page load speed, which can be improved by optimizing web images.

Report Generated on **June 12, 2020** by prepostseo.com

PLAGIARISM SCAN REPORT

Words 142 Date June 18,2020

Characters 860 Exclude Url

0%

Plagiarism

100%

Unique

0

Plagiarized
Sentences

10

Unique Sentences

Content Checked For Plagiarism

The project aims in dealing with Images that can be edited using voice commands. Voice recognition is one of the hands-free technologies which can make a manual task much easier. The project has a voice recognition feature through which images can be edited. User must give the voice command as instructed in the application and get the desired outputs. An

Authorization System is provided where the user must register his details and later login. After a successful login, User can edit the images using voice commands. Images can be converted to black and white or grayscale. They can also be cropped, rotated, resized and flipped by using their respective voice commands. Images are displayed and also saved. User can then logout by using the voice command "logout" and exit the application. Microphone is necessary to use so as to execute the program.

Sources

Similarity

