BATCH NO:MAI081

SPEECH ENABLED EMAIL ASSISTANT WITH FACIAL RECOGINITION FOR SECURE AND ACCESSIBLE COMMUNICATION

Major project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

S V N S TIRUMALA DEVI	(21UECS0596)	**(20051)**
Y YOGANAND	(21UECS0696)	**(21190)**
K RAGHUNATHA SWAROOP	(21UECS0328)	**(19051)**

Under the guidance of
Mr.S GOPI,M.Tech.,
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May 2025

CERTIFICATE

It is certified that the work contained in the project report titled ”SPEECH ENABLED EMAIL ASSISTANT WITH FACIAL RECOGINITION FOR SECURE AND ACCESSIBLE COMMUNICATION” by ”S V N S TIRUMALA DEVI (21UECS0596), Y YOGANAND (21UECS0696), K RAGHUNATHASWAROOP (21UECS0328)” has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
Mr. S GOPI
Assistant Professor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science and Technology

Signature of Head/Assistant Head of the Department	**Signature of the Dean**
Dr. N. Vijayaraj/Dr. M. S. Murali dhar	**Dr. S P. Chokkalingam**
Professor & Head/ Assoc. Professor &Assistant Head	**Professor & Dean**
Computer Science & Engineering	**Computer Science & Engineering**
School of Computing	**School of Computing**
Vel Tech Rangarajan Dr. Sagunthala R&D	**Vel Tech Rangarajan Dr. Sagunthala R&D**
Institute of Science and Technology	**Institute of Science and Technology**

DECLARATION

We declare that this written submission represents our ideas in our own words and where others ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

S V N S TIRUMALA DEVI
Date: / /

Y YOGANAND
Date: / /

K RAGHUNATHA SWAROOP
Date: / /

APPROVAL SHEET

This project report entitled ”SPEECH ENABLED EMAIL ASSISTANT WITH FACIAL RECOGINITION FOR SECURE AND ACCESSIBLE COMMUNICATION” by S V N S TIRUMALA DEVI (21UECS0596), Y YOGANAND (21UECS0696), K RAGHUNATHA SWAROOP (21UECS0328) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners **Supervisor**

Mr. S GOPI,M.Tech.,
ASSISTANT PROFESSOR.,

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our **Honorable Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (Electrical), B.E. (Mechanical), M.S (Automobile), D.Sc., and Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.,Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, for her blessings.**

We express our sincere thanks to our respected Chairperson and Managing Trustee **Dr.(Mrs.) RANGARAJAN MAHALAKSHMI KISHORE., Vel Tech Rangarajan Dr. Sagunthala RD Institute of Science and Technology,** for her blessings.

We are very much grateful to our beloved **Vice Chancellor Prof. Dr.RAJAT GUPTA,Ph.D.,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean , School of Computing, Dr. S P. CHOKKALINGAM, M.Tech., Ph.D., & Professor & Associate Dean , School of Computing, Dr. V. DHILIP KUMAR,M.E.,Ph.D.,** for their immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Professor & Head, Department of Computer Science & Engineering, Dr. N. VIJAYARAJ, M.E., Ph.D.,** and **Associate Professor & Assistant Head, Department of Computer Science & Engineering, Dr. M. S. MURALI DHAR, M.E., Ph.D.,**for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our **Internal Supervisor Mr. S. GOPI,M.Tech.,** for his cordial support, valuable information and guidance, he helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Dr.SADISH SENDIL MURUGARAJ, B.E., M.E.,Ph.D.,Professor, Dr.S.RAJAPRAKASH, M.E,Ph.D., Mr.V.ASHOK KUMAR, B.E,M.Tech.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

S V N S TIRUMALA DEVI	**(21UECS0596)**
Y YOGANAND	**(21UECS0696)**
K RAGHUNATHASWAROOP	**(21UECS0328)**

ABSTRACT

In an increasingly digital world, accessibility and security in communication tools have become critical, particularly for individuals with visual or physical impairments. Conventional email systems demand manual navigation and password-based authentication, which can be cumbersome or inaccessible to some users. This project presents a novel voice-controlled email system integrated with facial recognition technology, aiming to offer a more secure and inclusive communication experience. The system enables users to log in using facial biometrics and interact with email functions entirely through voice commands. Key components include speech-to-text conversion for composing emails, text-to-speech for reading emails aloud, and secure login using face encodings generated via deep learning-based computer vision models such as ResNet-34. Developed using Python and Flask, and leveraging libraries such as face recognition, Speech Recognition, and gTTS, the system interacts with email services through standard IMAP and SMTP protocols. The proposed system is particularly suited for users who require hands-free functionality or rely on assistive technologies, and it demonstrates the potential of combining artificial intelligence with human-centric design to improve digital inclusivity and security.This model has demonstrated an accuracy of 99.38 on the Labeled Faces in the dataset, indicating robust performance in facial verification tasks.

Keywords: Facial recognition, Speech recognition, Text-to-speech (TTS), Convolutional Neural Networks, Flask framework, Google gTTS, IMAP/SMTP protocols, Deep neural networks (DNN), Natural language processing (NLP), Humancomputer interaction.

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

AI	Artificial Intelligence
API	Application Interface
CNN	Convolutional Neural Network
DNN	Deep Neural Network
FPS	Frames Per Second
GTTS	Google Text To Speech
HTML	Hyper Text Markup Language
IMAP	Internet Message Access Protocol
IVR	Interactive Voice Response
NLP	Natural Language Processing
SDG	Sustainable Development Goals
SMTP	Simple Mail Transfer Protocol
UI	User Interface
VS CODE	Visual Studio Code
2FA	Two Factor Authentication

TABLE OF CONTENTS

Page.No

Chapter 1

INTRODUCTION

1.1 Introduction

Email is one of the most widely used digital communication tools, essential for both personal and professional correspondence. Despite the growing integration of intelligent systems into everyday applications, email platforms have remained largely dependent on manual input devices like keyboards and touchscreens. These traditional methods pose a significant challenge for individuals who are visually impaired, physically disabled, or in situations where hands-free operation is necessary. In response, the demand for more accessible and user-friendly communication solutions has increased dramatically in recent years.

Technologies such as voice recognition and biometric authentication are now mature enough to provide alternative modes of user interaction. Voice-controlled systems are capable of converting spoken language into text, enabling users to communicate with machines naturally. At the same time, facial recognition systems offer a secure and seamless way to authenticate users without requiring passwords. These advancements open the door to more inclusive email systems that do not rely on conventional interaction methods.

This project proposes a novel approach to email accessibility through the integration of voice commands and facial recognition. The system is designed to support users with varying abilities by allowing them to log in using their face and manage emails using speech. This makes the communication process more efficient and eliminates the need for manual effort. The implementation, which uses Python and Flask,combines powerful libraries like facerecognition, SpeechRecognition, and gTTS to deliver an intelligent and inclusive email assistant.

These barriers often result in dependence on others for composing, sending, and reading emails, undermining privacy and independence. To address this, the proposed project introduces a Speech-Enabled Email Assistant integrated with Open CV-based Facial Recognition. This system ensures secure access, enables voice-based email composition, and delivers emails in text or speech format based on user preference. By leveraging speech-to-text technology and facial recognition, the solution promotes inclusivity, empowering users with physical or literacy limitations to engage seamlessly in digital communication.

1.2 Background

Email remains one of the most widely used forms of digital communication, essential for both personal and professional interactions. However, traditional email systems often require users to manually log in, type, read, and navigate through complex interfaces, tasks that can be challenging for individuals with disabilities, elderly users, or those unfamiliar with technology.In recent years, advances in Artificial Intelligence (AI), especially in Computer Vision and Speech Processing, have opened up new possibilities for creating more accessible and secure digital systems. Facial recognition, powered by Convolutional Neural Networks (CNNs) and implemented through tools like OpenCV, provides a powerful method for verifying user identity without the need for passwords.

Simultaneously, speech recognition and synthesis technologies allow for natural voice-based interaction with systems, minimizing the need for physical input or reading.By integrating these technologies, it is possible to develop an intelligent email assistant that not only authenticates users securely using facial recognition but also allows them to interact with their email through voice commands. This approach enhances both security and usability, catering especially to the needs of visually impaired users and those seeking a hands-free communication experience.Our project builds on this concept by combining OpenCV, CNN, and speech processing tools to create a comprehensive solution for secure and accessible email communication.

1.3 Objective

The primary objective of this project is to design and implement a smart, speech-enabled email assistant integrated with facial recognition technology to ensure both security and accessibility in digital communication. As email continues to be a crucial medium for communication, especially in professional and academic environments, there is a growing need for systems that are inclusive and easy to use for all individuals, including those with disabilities or limited technical expertise. This project aims to overcome the limitations of traditional email systems by enabling users to perform key functions such as logging in, composing, reading, and sending emails entirely through voice commands, reducing the dependency on physical input devices.To ensure secure access, the system incorporates facial recognition using OpenCV for image processing and Convolutional Neural Networks (CNN) for accurate and reliable identity verification.

This eliminates the need for passwords and protects the user's personal information from unauthorized access. By integrating speech-to-text and text-to-speech technologies, the system allows users to communicate with the assistant in a natural and intuitive way, significantly improving the experience for users who are visually impaired or have difficulty using keyboards and screens.Furthermore, the system is designed to be lightweight, efficient, and easily adaptable to various platforms. By combining modern AI technologies, machine learning models, and natural language processing, this assistant not only enhances usability but also demonstrates the potential of intelligent systems in creating a more inclusive digital world. The overall goal is to provide a secure, accessible, and intelligent communication tool that addresses real-world challenges and improves the quality of life for a diverse range of users.

1.4 Problem Statement

Email is one of the most widely used and essential forms of communication in the digital world, serving as a primary channel for exchanging information in both personal and professional environments. However, traditional email systems are often designed with a "one-size-fits-all" approach, requiring manual login, reading, typing, and navigation through graphical interfaces. These requirements pose significant challenges for individuals with visual impairments, physical disabilities, elderly users, and those who are not technically proficient. As a result, many users find it difficult or even impossible to access and manage their email accounts independently.In addition to accessibility challenges, security is a growing concern in digital communication. The reliance on passwords and PINs for user authentication makes email systems vulnerable to various threats, including hacking, phishing, and identity theft. These security breaches not only compromise personal information but also erode trust in digital platforms.

Furthermore, remembering complex passwords can be burdensome, especially for users with cognitive limitations or those unfamiliar with technology.While some progress has been made with voice assistants and screen readers, these solutions often lack robust security features and are not optimized specifically for email communication. Existing voice-based systems typically allow general interaction but do not provide the necessary authentication or privacy measures needed to secure personal content.There is a clear need for a comprehensive system that addresses both accessibility and security in email communication. This project aims to bridge this gap by developing a speech-enabled email assistant that incorporates facial recognition for secure login using OpenCV and CNN-based models, along with speech-to-text and text-to-speech features for hands-free, voice-driven interaction. The goal is to provide an intelligent assistant that allows users to access, compose, read, and send emails securely and efficiently, regardless of their physical abilities or technical expertise.By combining AI, computer vision, and speech processing, this project seeks to create a user-centric solution that promotes digital inclusivity while maintaining high levels of security and user satisfaction.

Chapter 2

LITERATURE REVIEW

Muqial et al., (2020) [1] Developing a voice-based email system to give blind or individuals who are not computer literate safer and more efficient email services. This approach is simple to utilize for users of all ages. The system's text-to-speech speech reader and speak-to-text feature make it easier for blind individuals to utilize. The use of mobile phones has expanded to include life for everyone. They are utilized for different purposes such as reading, listening to music, and obtaining information from online sources.

Chen et al., (2019) [2] Mobile phones are found everywhere, and Access to information is managed with ease. Assistants on the go on cell phones help clients in setting up their gadgets and carrying out tasks by speaking, from triggering songs to play as alarms. Nevertheless, when it when it comes to email systems, there are not enough efficient answers for blind individuals to deal with their messages.

Wang and Li. (2021) [3] The web is an essential piece of our lives. Every person being able to access a lot of information and knowledge by means of the internet. The current exchanges on the email system is primarily used on the internet for confirmation and data. However, people who are blind have difficulty obtaining these provided services by way of the internet.

Y. Zhang et al., (2022) [4] The incorporation of machines in today's digital age The fields of learning and artificial intelligence (AI) have changed technology, particularly voice-activated assistants, which are now essential to our day-to-day lives, including smartphones and smart housing These smart online companions do not only make tasks easier, but also make the user experience better. To make use of this revolutionary technology on the desktop level, our project, ”Smart Voice Assistant,” is introduced with a voice-driven email system for visually impaired users impaired.

Smith et al., (2022) [5] System for a voice assistant that uses face recognition. It portrays the plan and improvement of a front-line framework planned to furnish clients with a safe and modified connection. utilizing a facial Recognition and authentication are made possible by the to safely identify authorized users, the voice assistant before allowing its users access to the system, a wide variety of capabilities This a second layer of biometric security guarantees that Only the authorized user can use the system thereby reducing the likelihood of unauthorized access.

R. Gupta and S. Joshi (2019) [6] The potential benefits of voice-based technology with impairments, particularly those with visual or hearing impairments are on the rise clear as they create. By defeating the limitations imposed by conventional methods of input, the combination of speech technology and a wide range of Accessibility has also increased as a result of devices and services but also enhanced independence and convenience for people who are disabled.

L. Wang and F. Liu (2020) [7] Email is a result of global technological advancements presently used extensively. Email can make a big difference how effectively and efficiently messages and everything else are sent Different kinds of data are sent. However, text-based access for those who are visual learners, information may be challenging impaired. The rise of computer-based practical applications solutions has provided numerous new options to blind people worldwide. The building's structural architecture is shown off in this project voice message framework, which empowers a visually impaired individual to quickly and easily access email.

Gupta and Sharma (2020) [8] The increasing use of the internet has led to the emergence of numerous new enterprises opportunities, but accessing basic initiatives like email is difficult for people who are blind or visually impaired. This The project aims to offer visually impaired people voice assistance debilitated clients through private voice partners that can provide services and perform tasks using spoken commands. Voice commands make it easier to method of input for users, and it may be especially advantageous for those who may have difficulty due to visual impairments the use of a keyboard. The ultimate goal of the project is to a safe virtual assistant.

Lee et al., (2021) [9] Thanks to technology, communication is now simple and elementary. The Internet epitomizes contemporary systems for communication Email or electronic mail has a basic job in this leap forward. Email has improved popularity not only for use in the workplace but also for personal use also applies. The underlying cause of this Email is widely used due to its simplicity. It is extremely simple and trouble-free to access email systems.

H. Singh and P.Kaur (2022) [10] In today's world, the internet is the most important component communication. Messages are further significant method of communication that is frequently used in business. The disabled have no use for this technology. and oblivious individuals. Approximately 260 million People with visual impairments worldwide, according to a review. This also indicates that these individuals are unaware of how to use the internet and e-mail, as well as about recent advances technologies.

2.1 Existing System

The existing system for email communication is largely built on conventional user interfaces that require manual input through devices like keyboards and mice. While these systems are efficient for the average user, they pose significant challenges for individuals with visual impairments, physical disabilities, or those who are not technically inclined. Traditional email platforms do not offer voice-enabled features, making it difficult for such users to compose, read, or manage emails without assistance. Moreover, these systems lack intelligent interaction capabilities, such as natural language processing, which would allow users to give voice commands in a more conversational and user-friendly manner.Security in current email systems is typically handled through passwords and PINs, which are increasingly becoming susceptible to cyber threats like phishing, brute-force attacks, and social engineering. Users may forget their passwords or use weak ones, leading to unauthorized access and data breaches.

There is a lack of integrated biometric authentication mechanisms, such as facial recognition, which can provide a more reliable and user-friendly form of identity verification. Additionally, accessibility features in these systems are minimal, and the user interface is not always optimized for screen readers or other assistive technologies.In summary, the current email communication systems are not designed with inclusivity and advanced security in mind. They fail to support users who require speech-based interaction or enhanced security features. This gap calls for the development of a smarter, more accessible, and secure email assistant that utilizes speech recognition for hands-free operation and facial recognition for robust authentication, ensuring effective and safe communication for all users.

2.2 Related Work

Several research studies and existing applications have explored the integration of voice assistants and biometric authentication in communication systems. Voice-based virtual assistants like Google Assistant, Apple Siri, and Amazon Alexa have demonstrated the potential of speech recognition technology in enabling hands-free operations. These systems can perform various tasks such as reading messages, setting reminders, and sending emails, but they are not specifically tailored for secure email communication and lack facial recognition integration for user authentication.Some previous academic works have proposed email systems with speech-to-text capabilities, primarily targeting visually impaired users. These systems allow users to compose and read emails through voice commands. However, most of these implementations lack a strong focus on security, particularly in terms of verifying the user's identity through biometrics.

In terms of security, facial recognition has been widely adopted in modern smartphones and security systems for identity verification. It provides a non-intrusive and user-friendly method for authentication. Some research has explored the use of facial recognition in login systems, but its application in email systems remains limited. A few studies have combined voice and facial recognition to enhance multi-factor authentication, but these are often focused on general access control rather than specific use cases like email communication.Projects like "Voice Controlled Email for the Visually Impaired" and "Secure Email Login Using Biometric Authentication" have laid the foundation for developing more accessible and secure email systems. However, they typically address either accessibility or security rarely both in a single solution.

Thus, while there have been individual advancements in voice interaction and facial recognition technologies, there is a lack of integrated systems that combine both to offer a secure, accessible, and user-friendly email assistant. This project aims to bridge that gap by developing a system that not only allows users to interact with their email through voice commands but also ensures secure access through facial recognition.

2.3 Research Gap

Despite significant advancements in speech recognition and biometric technologies, there remains a noticeable gap in their integrated application within email communication systems. Most existing solutions either focus on improving accessibility through speech-enabled features or enhancing security through biometric authentication, but very few combine both to address the dual need for secure and accessible communication.Current voice-enabled systems such as virtual assistants (e.g., Google Assistant, Siri) offer basic support for sending and reading emails. However, they are not fully dedicated to email functionalities and lack customization for specific user needs, particularly those of individuals with disabilities. These platforms often do not offer robust security mechanisms and rely on general device-level authentication, which may not be sufficient for sensitive email communication.

On the other hand, facial recognition has been successfully used in authentication systems, especially in smartphones and access control applications. However, its integration with email platforms for user authentication and secure login is still underexplored. Furthermore, there is limited research on combining facial recognition with speech-based systems in a way that enhances both user convenience and security.Moreover, many of the existing academic projects and commercial tools lack user-centric design tailored to the needs of visually impaired or physically challenged individuals. The absence of real-time feedback, context-aware interaction, and error-handling mechanisms further limits their usability in real-world scenarios.

In summary, there is a clear research gap in developing a comprehensive system that simultaneously offers hands-free voice interaction for accessibility and facial recognition for secure authentication in email communication. Addressing this gap can significantly enhance the usability and safety of email systems.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The existing system for email communication is largely built on conventional user interfaces that require manual input through devices like keyboards and mice. While these systems are efficient for the average user, they pose significant challenges for individuals with visual impairments, physical disabilities, or those who are not technically inclined. Traditional email platforms do not offer voice-enabled features, making it difficult for such users to compose, read, or manage emails without assistance. Moreover, these systems lack intelligent interaction capabilities, such as natural language processing, which would allow users to give voice commands in a more conversational and user-friendly manner.

Additionally, accessibility features in these systems are minimal, and the user interface is not always optimized for screen readers or other assistive technologies.In summary, the current email communication systems are not designed with inclusivity and advanced security in mind. They fail to support users who require speech-based interaction or enhanced security features. This gap calls for the development of a smarter, more accessible, and secure email assistant that utilizes speech recognition for hands-free operation and facial recognition for robust authentication, ensuring effective and safe communication for all users.

Advantages:

- Works across various devices like smartphones, tablets, and computers.
- Includes standard authentication methods like passwords, two-factor authentication (2FA), and recovery options
- Easily integrates with calendars, task managers, and cloud storage for enhanced productivity.

- Commonly used in both personal and professional environments, making communication standardized and familiar.

3.2 Proposed System

The proposed system is designed to provide a secure and accessible email communication assistant by integrating speech recognition for hands-free interaction and facial recognition using OpenCV and Convolutional Neural Networks (CNN) for user authentication. This system enables users to interact with their email accounts through voice commands, allowing them to compose, read, and send emails without relying on physical input devices. This feature is especially beneficial for visually impaired or physically challenged individuals, promoting inclusive communication.

To ensure high-level security, the system implements facial recognition as a biometric authentication method. Using OpenCV for real-time image capture and preprocessing, and a CNN model for accurate face recognition, the system verifies the identity of the user before granting access to the email functionalities. CNN is chosen for its ability to extract deep facial features and provide high accuracy even under varying lighting conditions or facial expressions.

The speech-to-text functionality is powered by natural language processing tools that convert spoken commands into text, allowing the system to perform tasks like opening inbox, reading unread emails, or composing new messages. The assistant also provides real-time audio feedback, ensuring a smooth and interactive experience. By combining OpenCV, CNN, and speech recognition technologies, the proposed system creates a highly secure and accessible email platform for all users.

Advantages:

- Allows users to compose, read, and send emails using voice commands, improving accessibility for visually impaired and physically challenged users.
- OpenCV enables efficient real-time face detection and image processing, making the login process smooth and fast.
- CNN provides accurate and reliable face recognition by extracting deep facial features, even under varying conditions (e.g., lighting, angle, expressions).

- Speech recognition combined with NLP allows users to give natural, conversational commands, improving ease of use.

3.3 Feasibility Study

3.3.1 Economic Feasibility

The proposed system is economically viable and cost-effective to implement, especially for academic, research, or small-scale professional use. The development of the system primarily relies on open-source technologies such as OpenCV for facial recognition and Python-based libraries for speech recognition and email integration. This significantly reduces the need for purchasing expensive software licenses. Additionally, the hardware requirements are minimal only a standard computer or laptop with a webcam and microphone is needed, both of which are commonly available in most environments. No investment in high-end servers or external hardware is necessary. Since the system does not require ongoing subscription services or costly maintenance, the overall implementation cost remains low. Furthermore, the use of pre-trained models and publicly available APIs shortens the development time and effort, making the project even more affordable. In the long run, this system provides excellent value, particularly for institutions or users seeking to improve accessibility and security without a large financial investment.

3.3.2 Technical Feasibility

The proposed system is technically feasible as it utilizes widely available, well-supported, and efficient technologies. The core components include speech recognition, facial recognition, and email service integration all of which can be implemented using popular open-source tools. Python libraries such as SpeechRecognition, pyttsx3, and gTTS provide effective speech-to-text and text-to-speech capabilities for voice interaction. For secure user authentication, OpenCV is used for real-time image capture and preprocessing, while Convolutional Neural Networks (CNN) are employed for accurate facial recognition, ensuring robust and reliable identification. Additionally, integration with email services such as Gmail or Outlook can be achieved through SMTP, IMAP, or official APIs, allowing smooth interaction with email accounts. The system can be developed and run on standard hardware like laptops or desktops equipped with a microphone and webcam, eliminating the need for specialized devices. The availability of tutorials, documentation, and strong community support for all the chosen technologies further enhances the ease of development and implementation.

3.3.3 Social Feasibility

The proposed system is highly socially feasible, as it addresses real-world challenges related to digital accessibility and security. In today's society, where communication through email is essential for education, employment, and personal interaction, individuals with visual impairments or physical disabilities often face significant barriers. By enabling speech-based control and voice feedback, the system empowers such users to independently access and manage their emails, promoting digital inclusion. The project aligns with global efforts to create assistive technologies that bridge the gap between ability and opportunity. Moreover, by promoting convenience and safety for all users, the system can gain wide acceptance and support from society, institutions, and organizations advocating for equal access to technology.

3.4 System Specification

3.4.1 Hardware Specification

- Processor : Intel i5 or higher
- RAM : Minimum 8 GB
- Hard Disk : Minimum 256 GB (SSD preferred)
- Webcam : 720p HD or higher
- Microphone : Integrated or external mic
- Graphics :Integrated GPU
- Display : Minimum 13" screen, 1366×768 resolution

3.4.2 Software Specification

- IDE/Editor : VS Code
- Programming Language : Python 3.7 or higher

- Libraries Frameworks : OpenCV, NumPy, TensorFlow/Keras, scikit-learn, pyttsx3, SpeechRecognition, smtplib/imaplib
- Email API : Gmail API

3.4.3 Tools and Technologies Used

Programming Language:

- **Python:** Primary language due to strong libraries in speech, image processing, and AI.

Speech Recognition,Text-to-Speech:

- **SpeechRecognition:**For converting voice commands to text.
- **Google Speech API** For offline/online speech recognition.
- **gTTS (Google Text-to-Speech):**For reading out email content.

Facial Recognition:

- **Open CV:**For camera access and image processing.

3.4.4 Standards and Policies

Visual Studio Code

It's a lightweight, open-source code editor developed by Microsoft that supports development in various programming languages such as Python, JavaScript, HTML, and more. It offers features like syntax highlighting, intelligent code completion (IntelliSense), debugging tools, version control integration (Git), and extensions that enhance the developer experience

Standard Used: ISO/IEC 9126

Jupyter

It's like an open source web application that allows us to share and create the documents which contains the live code, equations, visualizations and narrative text. It can be used for data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning.

Standard Used: ISO/IEC 27001

Chapter 4

SYSTEM DESIGN AND METHODOLOGY

4.1 System Architecture

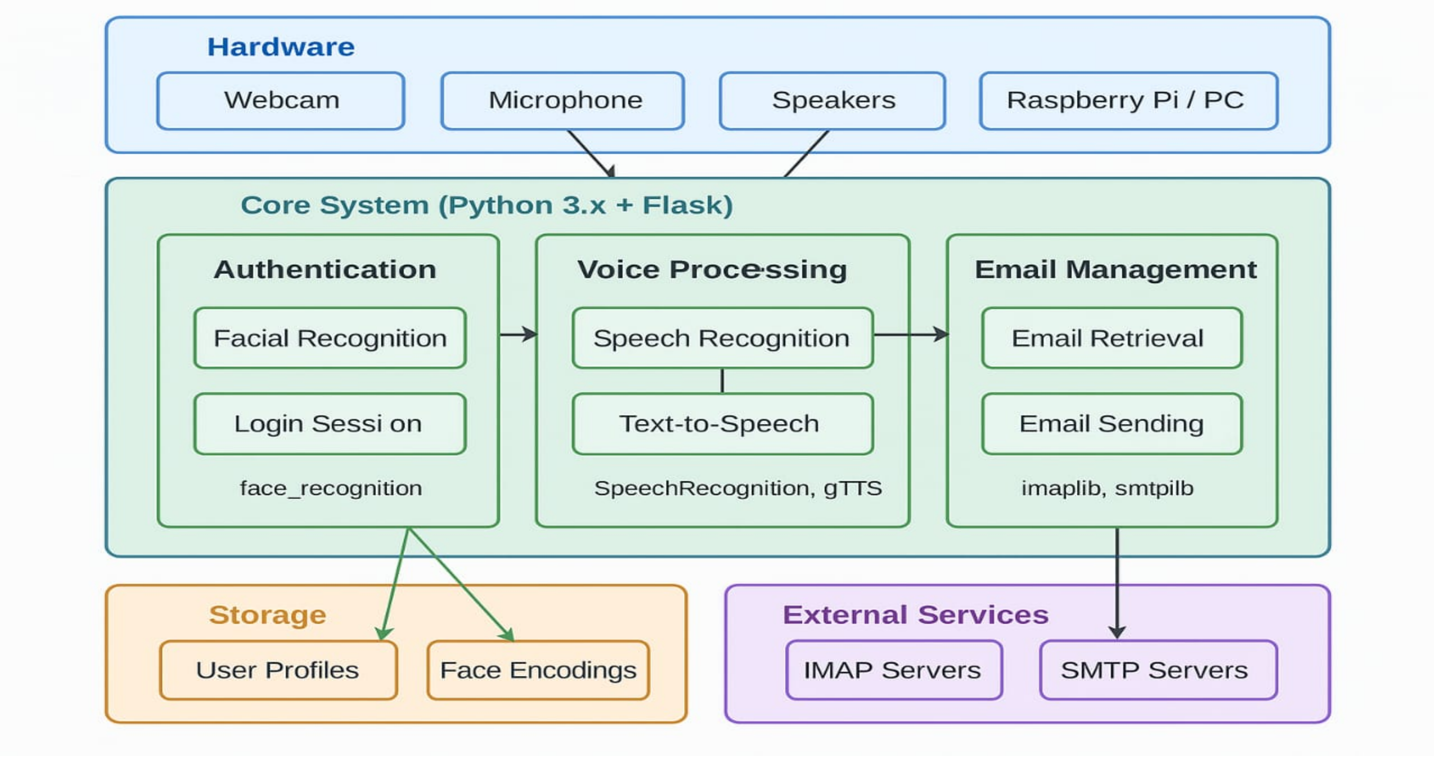


Figure 4.1: **General Architecture of Speech Enabled Email Assistant**

The above figure 4.1 explains how the system grants access to the voice-enabled email assistant. The user can then interact with the email module using speech commands to read, compose, send, or delete emails. The system uses speech recognition to interpret the user's voice instructions and provides auditory feedback using a text-to-speech engine. A detection module continuously monitors for commands, ensuring smooth operation without requiring any physical interaction. Overall, the architecture integrates facial recognition for authentication with voice technology to ensure an accessible, secure, and user-friendly email experience.Simultaneously, the microphone captures voice commands, which are processed through speech preprocessing and converted into text by the Speech-to-Text Engine.

4.2 Design Phase

4.2.1 Activity Diagram

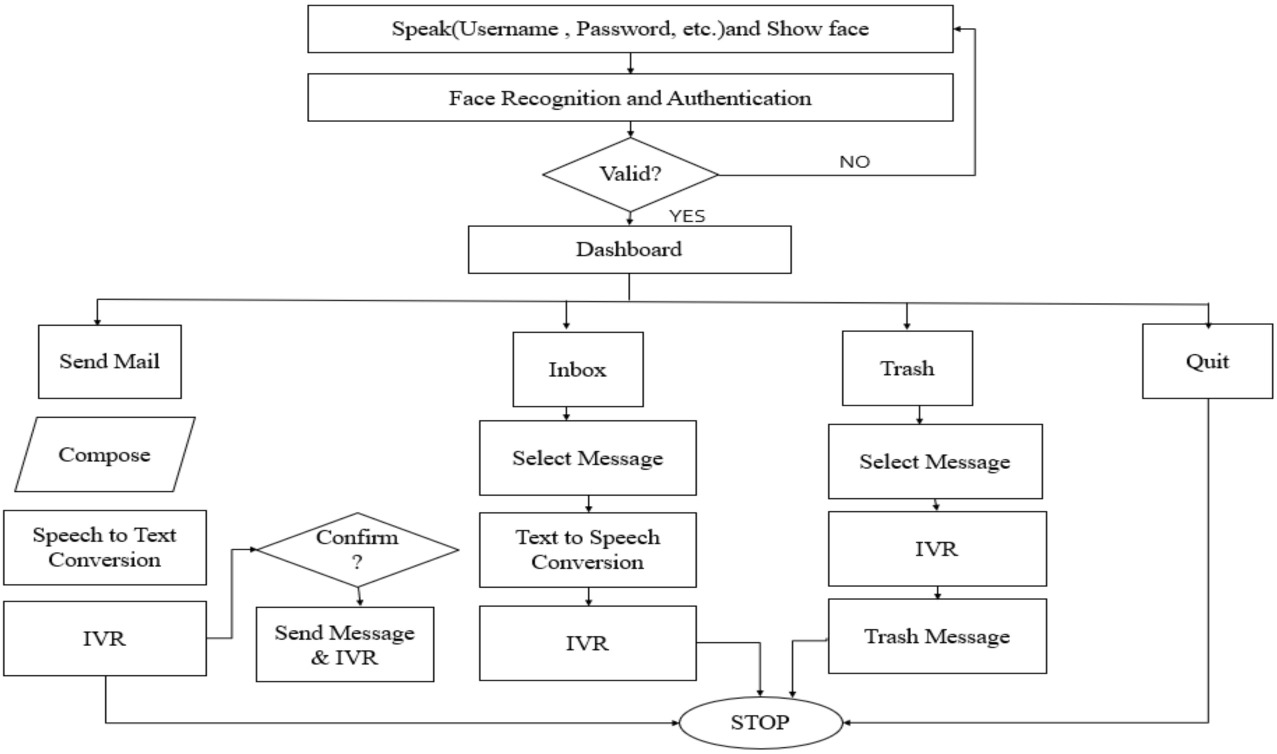


Figure 4.2: **Activity Diagram Representing the End-to-End Workflow**

The Above Figure 4.2 explains the structured process to ensure secure and accessible communication using facial recognition and speech processing. The workflow begins with the user speaking their credentials (username and password) and showing their face for authentication. The system then performs facial recognition and authentication to verify the user. If the authentication is valid, the user gains access to the dashboard. If authentication fails, the system denies access.If the command is to compose an email, the system prompts the user to dictate the recipient's address, subject, and body content, which are then converted to text and displayed on the screen. Once the email content is confirmed, the user can command ”Send,” and the system sends the email through the configured email client. After executing any command, the system provides auditory feedback to inform the user of the action taken, such as ”Email sent successfully” or ”You have 3 new emails.” The session concludes when the user says ”Logout” or ”Quit,” terminating the system's operation.

4.2.2 Use Case Diagram

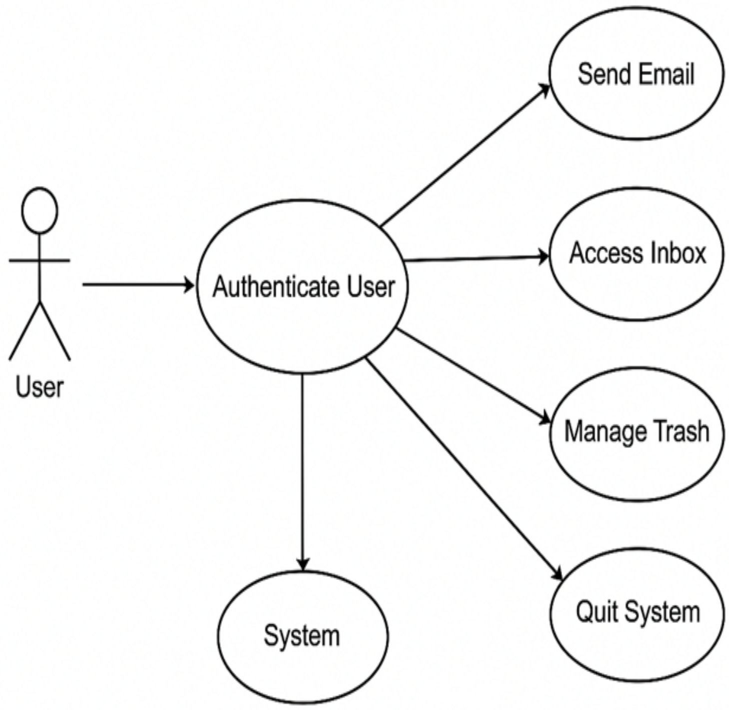


Figure 4.3: **Use Case Diagram Representing User Interactions with the System**

The above figure 4.3 explains the key interactions between the user and the system. The process begins with the user authenticating themselves through speech input and facial recognition, ensuring a high level of security. Once authenticated, the user can access the dashboard, which offers several core functionalities.One of the main use cases is the ability to send emails using voice commands. The system converts speech to text for composing messages and uses voice prompts to confirm the action before sending the email. The user can also access the inbox, where incoming emails are read aloud using text-to-speech technology, making it highly accessible, particularly for visually impaired users.Another feature includes managing deleted emails in the trash. Here, users can listen to the content of trashed messages and choose to either restore or permanently delete them. Finally, the user has the option to quit the system, which safely terminates the session.

4.2.3 Class Diagram

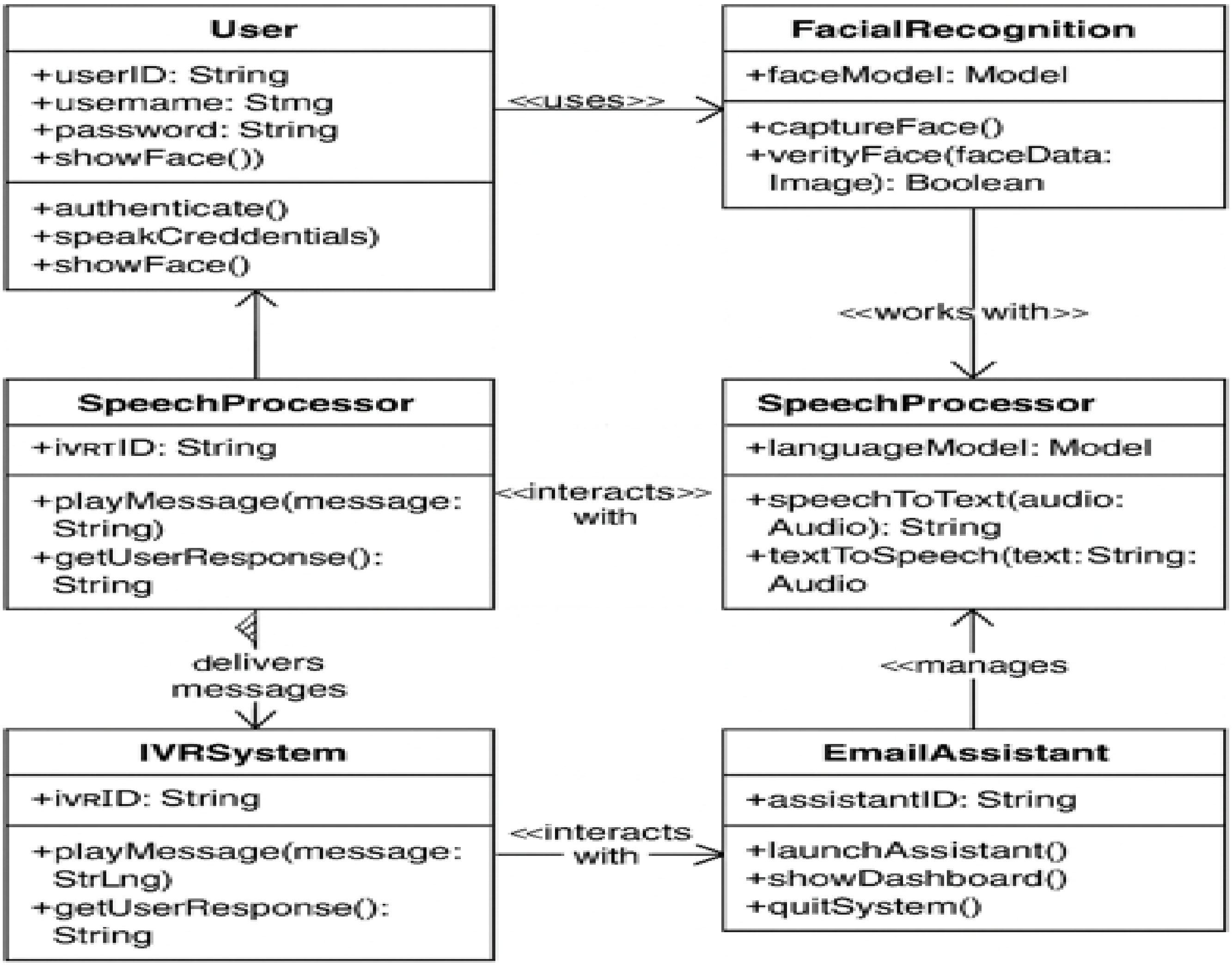


Figure 4.4: **Class Diagram Showing Core Components and Their Relation ships**

The above figure 4.4 explains the core components and their interactions within the system. It represents the object-oriented structure of the application, focusing on the main classes, their attributes, and methods. At the center is the User class, which stores essential user data such as userID, username, password, and faceData. The user interacts with the system through the authenticate(), speakCredentials(), and showFace() methods. The FacialRecognition class works in coordination with the User class and provides methods like captureFace() and verifyFace(), which are responsible for verifying the user's identity through facial biometrics.The IVRSystem complements this by delivering responses and prompts through the playMessage() and getUserResponse() methods.

4.2.4 Sequence Diagram

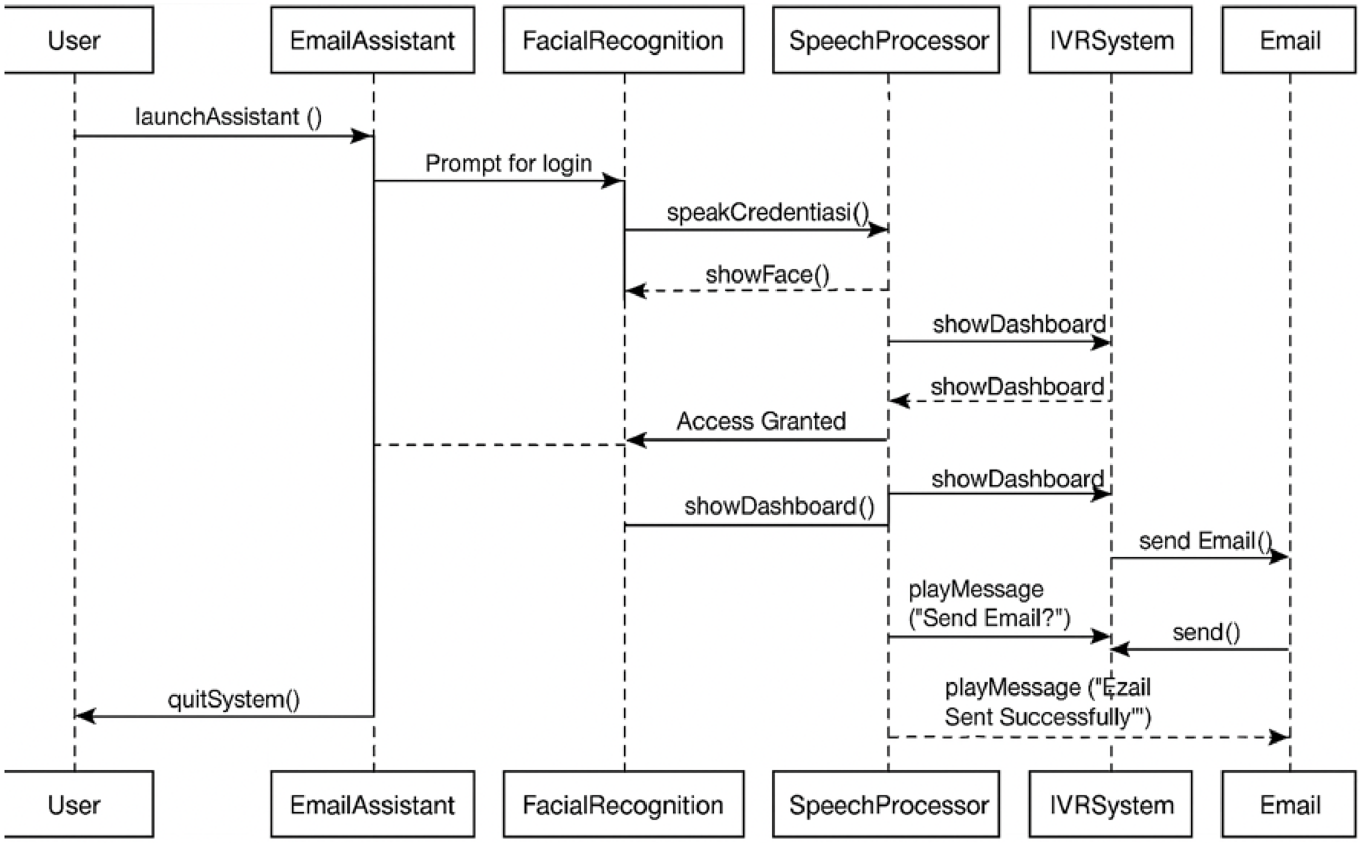


Figure 4.5: **Sequence Diagram for Query Processing and Response Generation**

The above figure 4.5 represents the step-by-step interaction between the user and various system components during the process of authentication, email composition, and sending.The interaction begins when the user launches the Email Assistant, which activates the system's dashboard. The user then speaks their credentials, which are processed by the SpeechProcessor using speech-to-text conversion. Simultaneously, the user presents their face, which is captured and verified by the FacialRecognition module. If both the voice input and facial data are validated, the EmailAssistant grants access to the user.Upon successful login, the user interacts with the SpeechProcessor again to choose an action such as sending an email—using voice commands. The spoken content is converted into text, which is then used by the Email component to compose a new message. The system uses the IVRSystem to confirm the user's intent to send the message by prompting a voice-based confirmation. Once the user confirms, the email is sent, and a voice notification is provided to confirm that the message was successfully delivered.

4.2.5 Collaboration diagram

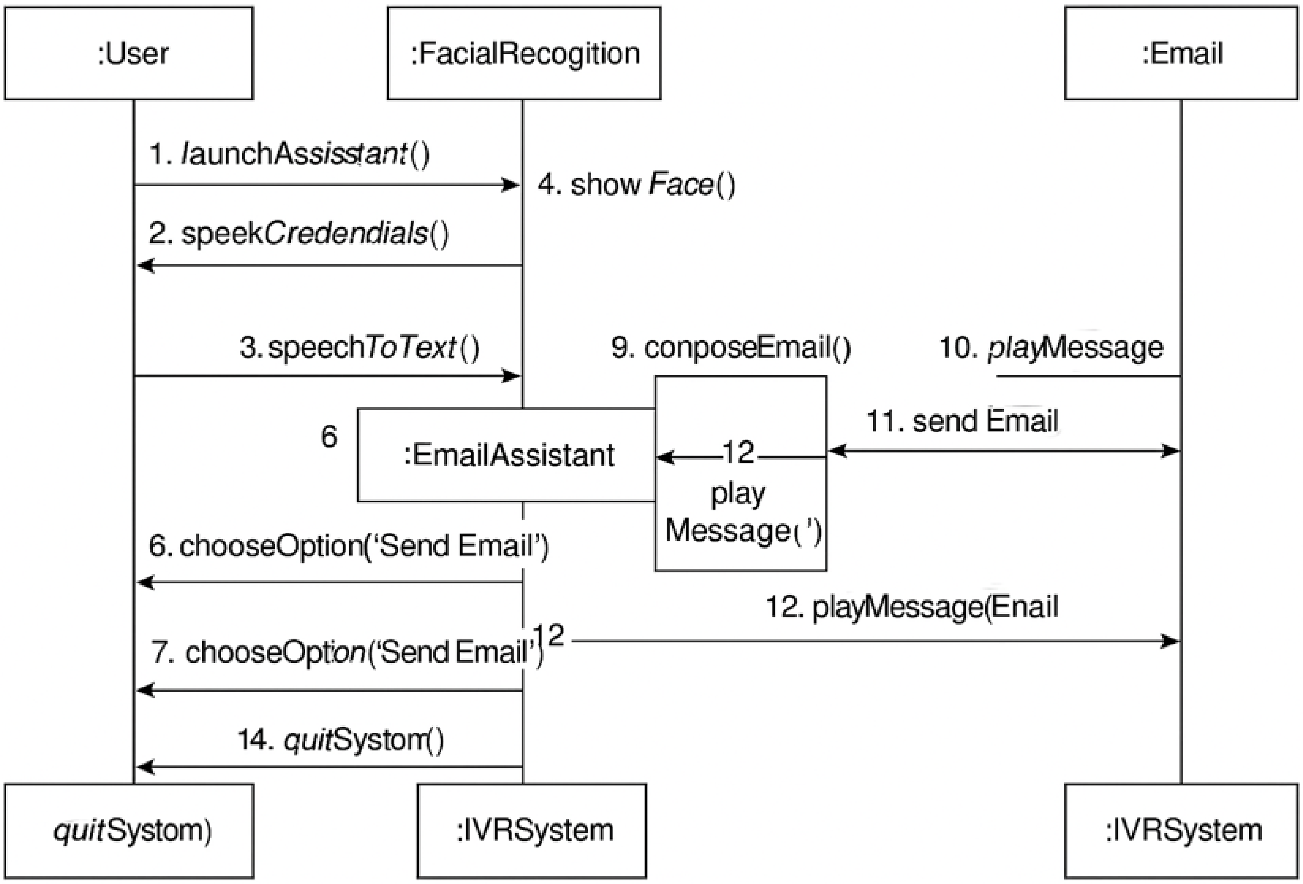


Figure 4.6: **Collaboration Diagram Showing Interactions Between System Components**

The above figure 4.6 explains how different system components interact with each other in a numbered sequence to fulfill the user’s request securely and efficiently.The interaction begins when the User initiates the system by calling the launchAssistant() function on the EmailAssistant. The user then provides spoken credentials, which are captured and processed by the SpeechProcessor through the speakCredentials() and speechToText() methods. Simultaneously, the user presents their face to the FacialRecognition module using showFace(), which processes the input and sends a verification request to the EmailAssistant.Once both voice and face data are validated, the system grants access to the user. The user then instructs the system (via speech) to send an email. The SpeechProcessor again converts the spoken content to text, which is then passed to the EmailAssistant. The Email component is used to compose the message. Before sending, the IVRSystem is engaged to confirm the user’s intent with a spoken prompt. Upon receiving confirmation from the user, the Email is sent, and a confirmation is voiced back via the IVRSystem.

4.2.6 Data Flow Diagram

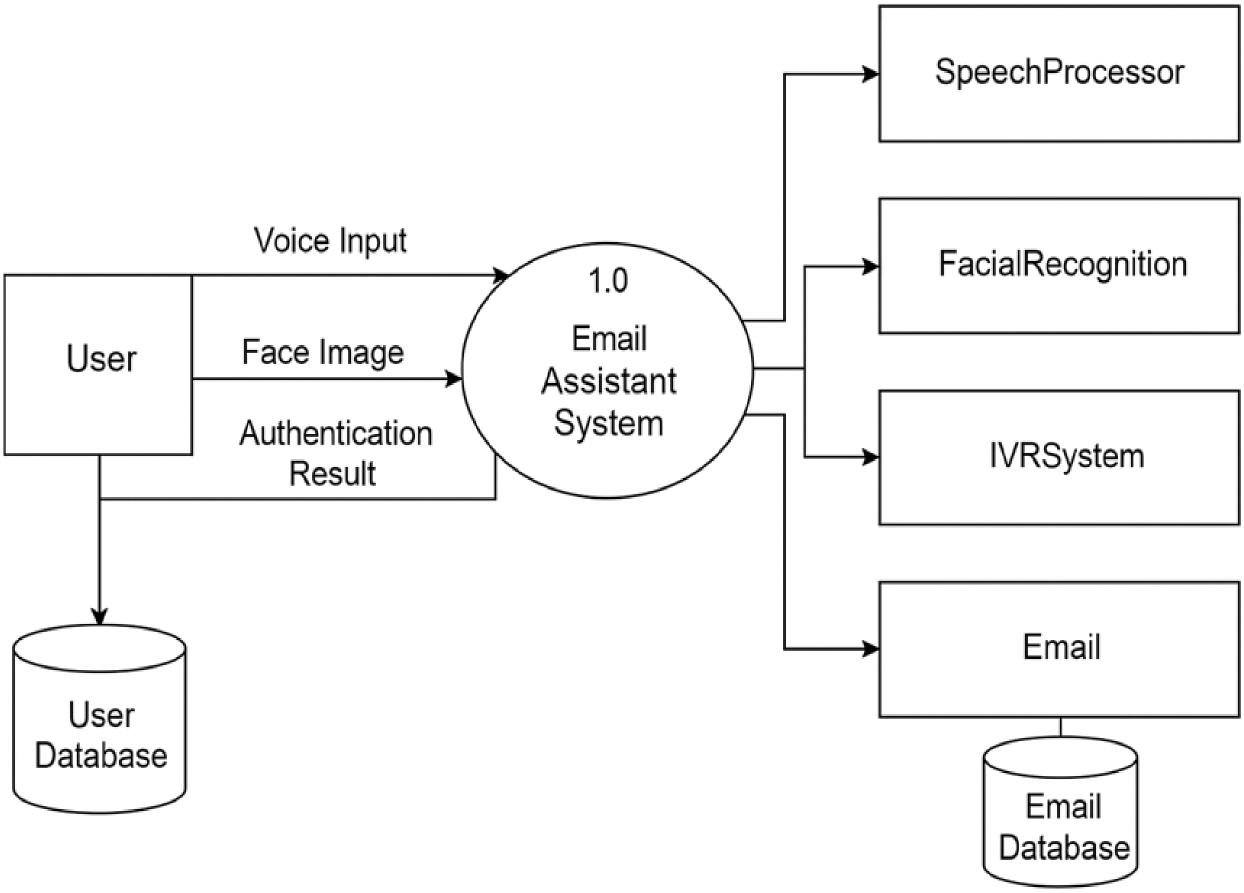


Figure 4.7: **Data Flow Diagram to represent How Data Moves Within System**

The above figure 4.7 explains a visual representation of how data moves within the system. The diagram begins with the user as the central external entity, who interacts with the system primarily through voice and facial inputs. When the user initiates the system, the voice credentials are processed by the Speech Processor, and the facial data is handled by the Facial Recognition module. These components communicate with the User Database to verify the identity of the user. Once authenticated, the system grants access and proceeds to further operations.If the user chooses to compose an email, the voice input is converted into text using the speech processor, and the message content is stored temporarily for editing or immediate sending. The IVR System then interacts with the user to confirm the email action. Upon confirmation, the composed email is saved and sent via the Email Database. If the user requests to read emails, the system fetches them from the database and converts the text to speech using the IVR system, allowing the user to listen to the content.

4.3 Algorithm & Pseudo Code

4.3.1 Algorithm

Step 1: Capture Video for Authentication

- Initialize the web camera to capture live video.
- Convert the video feed into frames per second (FPS) images.
- Resize and normalize the images for processing.

Step 2: Perform Facial Recognition

- Extract facial features using OpenCV and deep learning models (e.g., CNN, Haar Cascade).
- If a match is found, proceed to the email assistant. Otherwise, deny access.

Step 3: Initialize Speech Recognition for Email Operations

- Once authenticated, activate the speech recognition module.
- Prompt the user to speak a command (e.g., ”Compose Email”, ”Read Emails”, ”Send Email”).
- Convert the voice input into text using a speech-to-text engine (e.g., Google Speech API).

Step 4: Process User Commands

- If the user wants to compose an email:

Request the recipient’s email address.
Convert speech to text for the email body.
Ask for confirmation before sending.

- If the user wants to check emails:

Fetch the latest emails from the inbox.
Use text-to-speech (TTS) to read them aloud.

Step 5: Confirm and Send the Email

- If the user confirms, use an SMTP server (e.g., Gmail SMTP) to send the email.

Step 6: Logout and End Session

- Once the user completes tasks, log out securely.
- Close the web camera and terminate the session.

4.3.2 Pseudo Code

```
Start Program

MODULE: Facial_Recognition
    Initialize webcam
    Capture image of user face
    Preprocess image (resize, normalize, grayscale if needed)
    Load trained CNN model
    prediction = CNN.predict(face_image)

    IF prediction > threshold THEN
        Proceed to Email_Assistant
    ELSE
        Display ”Access Denied”
        Exit Program
    END IF
END MODULE

MODULE: Email_Assistant
    Initialize Microphone
    TextToSpeech(”Welcome to the Email Assistant. Say a command: Read, Compose, Delete, or Exit.”)
    speech_input = SpeechToText()

    SWITCH(speech_input)
        CASE ”read”:
            CALL Read_Inbox()
        CASE ”compose”:
            CALL Compose_Email()
        CASE ”delete”:
            CALL Delete_Email()
        CASE ”exit”:
            TextToSpeech(”Goodbye.”)
            Exit Program
        DEFAULT:
            TextToSpeech(”Command not recognized. Please try again.”)
            CALL Email_Assistant()   // Restart
    END SWITCH
END MODULE

MODULE: Read_Inbox
    Connect to email server via IMAP
    Fetch unread emails
    FOR each email in inbox
        Read sender, subject, and body using TextToSpeech
    END FOR
END MODULE

MODULE: Compose_Email
    TextToSpeech(”Please say the recipient's email address.”)
    recipient = SpeechToText()

    TextToSpeech("Please say the subject.")
    subject = SpeechToText()

    TextToSpeech("Please say the email body.")
    body = SpeechToText()

    Confirm content using TextToSpeech
    IF confirmed THEN
        Send email using SMTP
        TextToSpeech("Email sent successfully.")
    ELSE
        TextToSpeech("Email discarded.")
    END IF
END MODULE

MODULE: Delete_Email
    TextToSpeech("Please say the subject or index of the email to delete.")
    delete_target = SpeechToText()
    Search and confirm the target email
    IF confirmed THEN
        Delete email via IMAP
        TextToSpeech("Email deleted.")
    ELSE
        TextToSpeech("Deletion canceled.")
    END IF
END MODULE

End Program
```

4.4 Module Description

4.4.1 Face Recognition

The Facial Recognition Module serves as the primary security mechanism of the system. Its main function is to authenticate the user by verifying their face before granting access to the email assistant. This is implemented using a Convolutional Neural Network (CNN), which is trained on images of the authorized user. When the system starts, it activates the webcam to capture a real-time image of the user. The image is then preprocessed by resizing, normalizing, and converting it to the appropriate format. This preprocessed image is passed to the CNN model, which compares it against the trained facial data to determine whether the user is authorized.

If the match is successful, access is granted; otherwise, the system denies entry, ensuring that only verified users can use the assistant.

4.4.2 Speech Recognition

The Speech Recognition Module allows hands-free interaction with the email assistant, making it user-friendly and accessible. This module captures the user's voice through a microphone and converts the spoken words into text using speech recognition technology. It is responsible for interpreting commands such as "read email," "compose mail," "delete email," or "exit." The recognized text is then sent to the command processing module for appropriate action. This component is crucial for achieving a seamless voice-controlled experience and can be implemented using Python's speech recognition library with support from APIs like Google Speech or Whisper.

4.4.3 Text-to-Speech

The Text-to-Speech Module enhances accessibility by providing audio feedback to the user. It converts system-generated text into spoken words, allowing users especially those who are visually impaired to interact with the system effectively. This module reads out prompts, confirmations, and email contents using natural-sounding speech. For example, when the system retrieves an email, it reads aloud the sender, subject, and body to the user. Tools like pyttsx3 or gTTS (Google Text-to-Speech) are commonly used to implement this feature. The TTS module ensures a complete auditory experience, making the assistant usable even without a screen.

4.4.4 Email Handling

The Email Handling Module is responsible for managing all email-related operations such as reading, composing, sending, and deleting emails. It connects to the user's email server using standard protocols like IMAP (for reading and deleting emails) and SMTP (for sending emails). When a command is received, this module performs the corresponding action. For example, in the case of reading emails, it fetches unread messages and forwards the content to the TTS module. When composing an email, it collects the recipient address, subject, and message body through voice input and sends the email after user confirmation. This module ensures full email functionality within the assistant.

4.4.5 Command Processing

The Command Processing Module acts as the decision maker of the system. It takes the converted speech text from the Speech Recognition Module and interprets it to determine the user's intent. Based on keywords and phrases in the input, it routes the command to the appropriate function such as reading inbox messages, composing a new email, or deleting a selected email. If the command is unrecognized, it prompts the user to try again. This module may use simple string matching or natural language processing (NLP) techniques to improve command interpretation. It ensures that the system responds intelligently to voice inputs.

4.5 Steps to execute/run/implement the project

4.5.1 Installation

- Install Python (version 3.7 or above)
- Install Required Libraries
- Open your terminal or command prompt and run:

```
pip install opencv-python
pip install tensorflow keras
pip install speechrecognition
pip install pyttsx3
pip install imaplib2
pip install smtplib
pip install numpy
pip install pyaudio
```

4.5.2 Extraction

- Download this projrct zip folder and extract it
- Move to project folder in Terminal.
- Then run the following commands:

```
python app.py
```

4.5.3 Execution

- Now enter following URL in Your Browser Installed On Your Pc

  http://127.0.0.1:5000

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

Input design is a crucial part of system development that determines how users will interact with the system. For the project titled ”Speech Enabled Email Assistant with Facial Recognition for Secure and Accessible Communication”, the input design focuses on providing a secure, user-friendly, and accessible way for users to perform email-related tasks. The system accepts inputs through facial recognition, speech commands, and traditional text input, ensuring both convenience and security.

User Authentication Input: The system begins with user authentication through facial recognition. This input method uses the webcam to capture the user's face and compare it with the pre-stored data in the system. If the facial features match, the user is granted access to the email assistant. This method enhances security and ensures that only authorized users can access the system

Voice Command Input: Once authenticated, users can interact with the system using voice commands. The built-in microphone captures the user's speech and converts it into text using a speech recognition engine. Users can issue commands like “Compose Email,” “Read Inbox,” or “Send Email.” The system interprets these commands and performs the corresponding actions, allowing a hands-free and efficient email experience. This input method is especially helpful for users with physical disabilities or those who prefer voice interaction.

Textual Input: In addition to voice commands, the system also supports textual input through the keyboard. This serves as a fallback option in case of poor audio quality or speech recognition issues. Users can manually enter the recipient's email address, subject, and message body. Basic validation is applied to ensure that all required fields are filled and that email addresses are in the correct format.

Configuration and Settings Input: The system allows users to customize settings such as preferred language, speech speed, and accessibility features. These inputs are taken through a graphical interface using dropdown menus, sliders, or checkboxes. The input values are validated against predefined options and saved to the user profile, improving the overall personalization of the application.

5.1.2 Output Design

Output design plays a significant role in determining how information is presented to users in a clear, meaningful, and accessible way. For this project, the output design ensures that the responses from the system such as email details, notifications, and confirmations are communicated effectively through both visual and audio formats. The main goal is to deliver outputs that are easy to understand, enhance user interaction, and improve accessibility for users with special needs.

Facial Recognition Result: Once the user attempts to log in using facial recognition, the system provides immediate feedback on the authentication result. If the face is successfully recognized, a welcome message is displayed on the screen, and optionally announced through speech synthesis (text-to-speech). In case of a mismatch or error, a corresponding alert message is shown and spoken, guiding the user to try again or use alternative authentication methods.

Voice Command Response: When users issue voice commands, the system responds both visually and audibly. For example, if the user says “Read Inbox,” the system fetches the latest emails and reads out the sender name, subject, and a short snippet of each message. Simultaneously, these details are displayed on the screen for reference. This dual output format ensures that users, regardless of their hearing or visual ability, can still interact with the system efficiently.

Email Composition Confirmation: After composing an email via voice or text input, the system confirms the details such as recipient address, subject, and message content. It then asks the user whether to send the email. Once the user confirms, a confirmation message is displayed and announced, indicating that the email has been successfully sent. If there’s an error in sending, an error message is generated with suggestions for troubleshooting

Error and Alert Messages: The system is designed to handle errors gracefully by providing clear and helpful error messages. These outputs appear when inputs are invalid for example, if the email address format is incorrect or the microphone is not functioning. The messages are displayed with an explanation and possible corrective actions, and they are also read aloud to help users who may rely on auditory outputs.

5.2 Testing

5.2.1 Testing Strategies

Unit Testing: Unit testing is performed on individual modules such as facial recognition, speech recognition, email composition, and email sending functionalities. Each module is tested independently to ensure that it performs as expected in isolation. For example, the face detection algorithm is tested with both valid and invalid images to verify its accuracy.

```
    import unittest
from face_auth import recognize_face

class TestFacialRecognition(unittest.TestCase):
    def test_valid_face(self):
        result = recognize_face('test_images/known_face.jpg')
        self.assertTrue(result)

    def test_invalid_face(self):
        result = recognize_face('test_images/unknown_face.jpg')
        self.assertFalse(result)

if __name__ == '__main__':
    unittest.main()
```

Integration Testing: Once all individual modules are tested, integration testing is conducted to ensure they work together seamlessly. This includes testing interactions between the facial recognition and login module, the voice command system and email handler, and the speech output system with the UI. Any mismatches or data flow issues between components are identified and resolved during this phase.

```
    def test_compose_email_flow():
    speech_input = ”Compose email to john@example.com with subject Hello and message How are you”
    action = handle_voice_command(speech_input)
    assert action['recipient'] == ”john@example.com”
    assert action['subject'] == ”Hello”
    assert action['message'] == ”How are you”
```

Functional Testing: Functional testing focuses on verifying whether the system's features operate according to the specified requirements. In the project ”Speech Enabled Email Assistant with Facial Recognition for Secure and Accessible Communication,” functional testing ensures that each individual functionality such as facial recognition, speech recognition, email composition, and sending performs correctly under various input conditions.

```
    def test_send_email():
    from email_module import send_email

    success = send_email(
        to=”test@example.com”,
        subject=”Test Mail”,
        body=”This is a test message.”
    )
    assert success == True
```

Performance Testing: Performance testing is conducted to evaluate the responsiveness and stability of the system under different workloads. This includes testing the speed of facial recognition processing, the response time for speech commands, and how efficiently the system handles multiple operations without lag. Tools may be used to simulate real-time usage and evaluate system behavior.

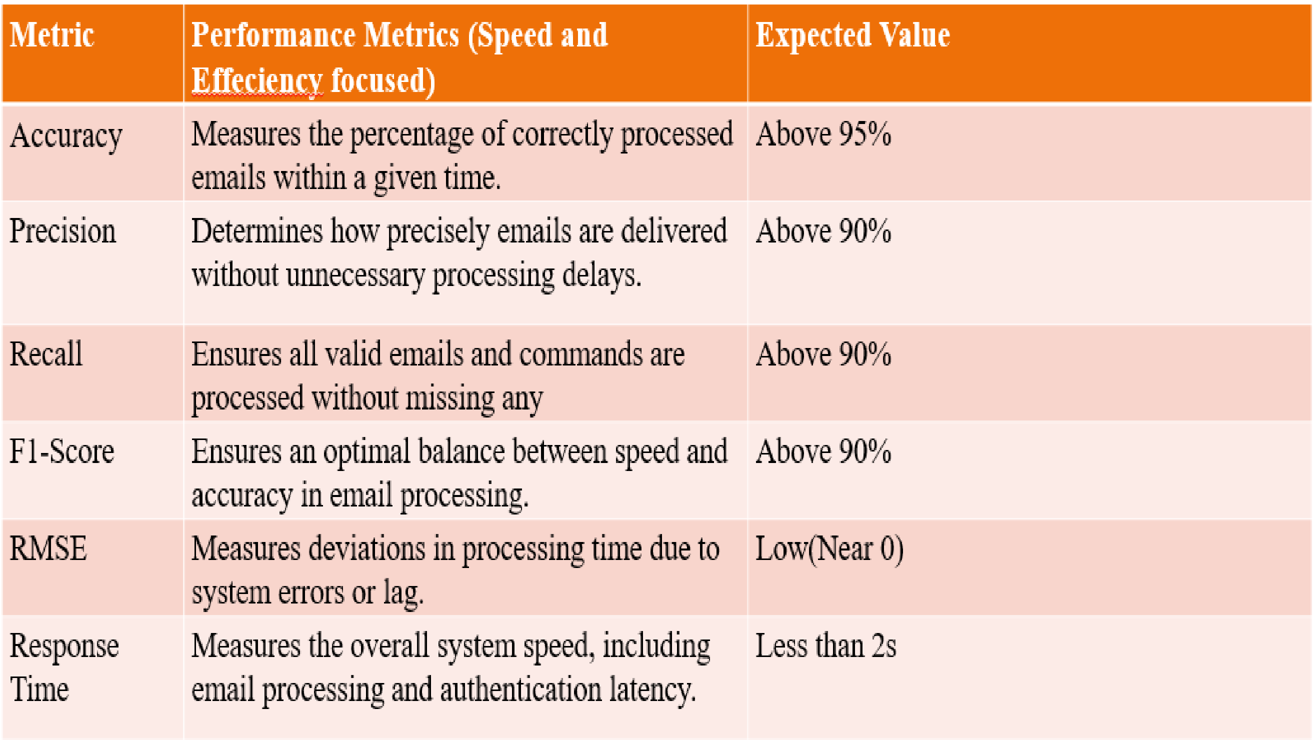
```
    import time
from face_auth import recognize_face

start_time = time.time()
recognize_face('test_images/known_face.jpg')
end_time = time.time()

print(”Face recognition time:”, end_time - start_time, ”seconds”)
```

5.2.2 Performance Evaluation

Performance evaluation is essential to ensure that the system responds quickly, operates efficiently, and performs reliably under different conditions. For the project ”Speech Enabled Email Assistant with Facial Recognition for Secure and Accessible Communication,” the performance was evaluated based on criteria such as response time, resource usage, system accuracy, and user experience under load.



Metric	Performance Metrics (Speed and Effeciency focused)	Expected Value
Accuracy	Measures the percentage of correctly processed emails within a given time.	Above 95%
Precision	Determines how precisely emails are delivered without unnecessary processing delays.	Above 90%
Recall	Ensures all valid emails and commands are processed without missing any	Above 90%
F1-Score	Ensures an optimal balance between speed and accuracy in email processing.	Above 90%
RMSE	Measures deviations in processing time due to system errors or lag.	Low(Near 0)
Response Time	Measures the overall system speed, including email processing and authentication latency.	Less than 2s

Figure 5.1: **Performance Metric for Testing**

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The proposed system demonstrates a high level of efficiency in providing a secure and accessible method of email communication through the integration of speech recognition and facial authentication technologies. Efficiency is evaluated in terms of time saved, accuracy, ease of use, system responsiveness, and resource optimization.The system efficiently handles user authentication through facial recognition, eliminating the need for manually entering passwords, thereby reducing login time and enhancing security. The facial recognition module works accurately and within seconds, ensuring quick access while maintaining privacy through biometric verification.

In terms of voice-based interaction, the system provides an efficient way for users to compose, send, and read emails using simple speech commands. This significantly reduces the time and effort compared to traditional typing methods, especially benefiting users with visual impairments or physical disabilities. The voice recognition engine processes commands quickly and delivers real-time feedback, making the user experience seamless and responsive.Additionally, the system is designed to use moderate computational resources, ensuring smooth performance even on standard machines. The lightweight implementation of libraries and optimized code structure help minimize CPU and memory usage, contributing to better energy efficiency and longer device life.

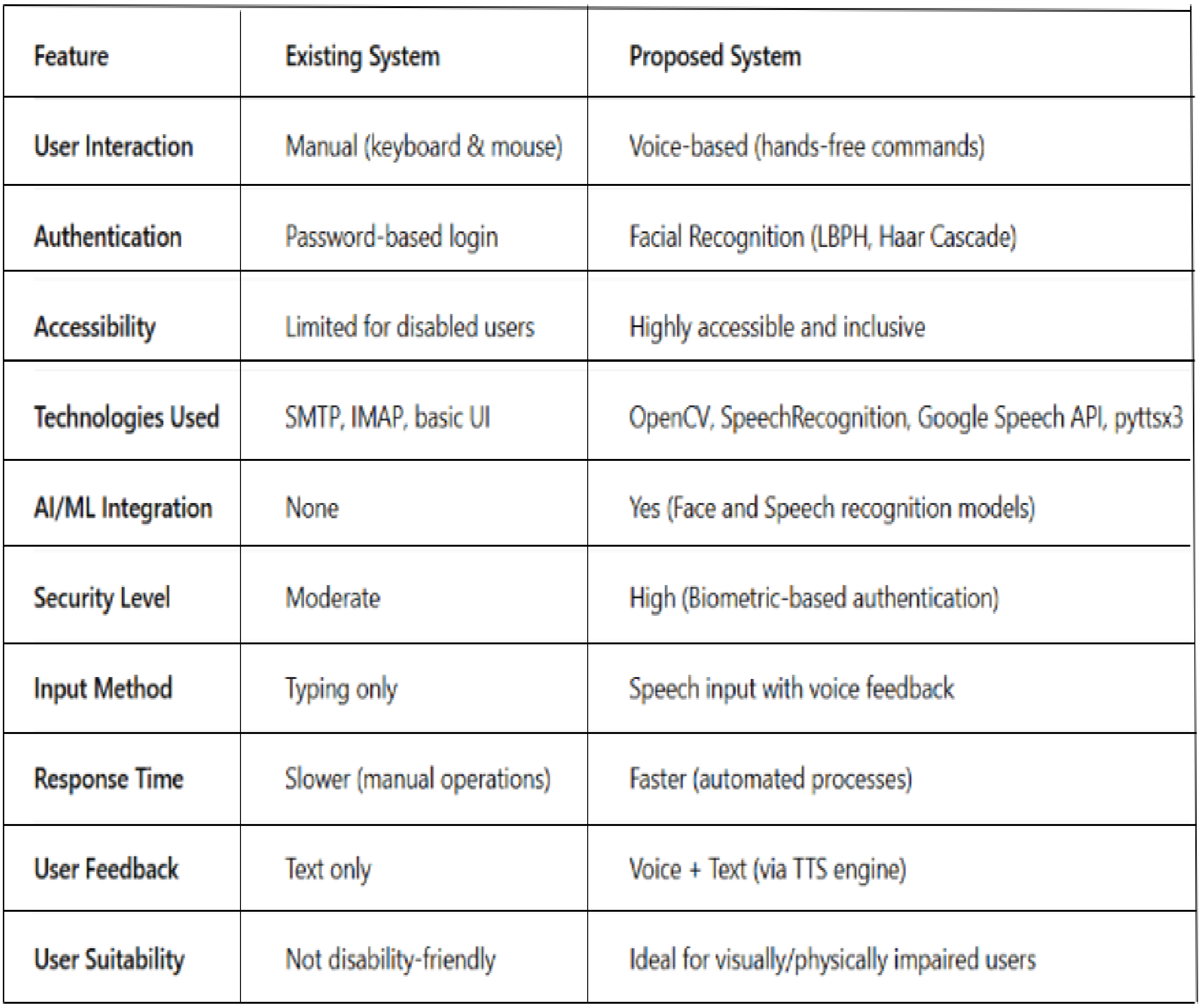
From a usability perspective, the system is intuitive and requires minimal training, making it efficient for users of all backgrounds. It reduces the dependency on keyboard and mouse, making email communication more accessible, hands-free, and user-friendly.In conclusion, the proposed system offers an efficient solution that blends security and accessibility while ensuring fast, accurate, and resource-optimized performance. It is especially well-suited for users who seek convenience, speed, and inclusivity in digital communication platforms.

6.2 Comparison of Existing and Proposed System

Existing system: The existing email systems are primarily based on manual interaction, where users access their inbox and send messages through keyboard and mouse inputs. Authentication is typically done through username and password, which, while common, is vulnerable to security breaches such as phishing attacks, brute-force attacks, and password leaks. These systems are built using standard web technologies or desktop applications and utilize protocols like SMTP (Simple Mail Transfer Protocol) and IMAP (Internet Message Access Protocol) to manage email communication. No biometric authentication or voice interaction is included, which limits the accessibility of these platforms for people with disabilities or those who prefer hands-free operation. Furthermore, the existing systems do not involve AI or machine learning models, making them static and less intelligent in terms of user interaction and automation.

Proposed system: The proposed system offers a significant enhancement by incorporating facial recognition and voice command features, making email communication both secure and accessible. For user authentication, the system employs facial recognition technology using models such as Haar Cascade Classifier for face detection and OpenCV for face recognition, implemented through OpenCV. This biometric login method eliminates the need for passwords and enhances system security. For interaction, the system uses speech recognition, powered by the SpeechRecognition library and Google Speech API, to convert spoken commands into text. This allows users to compose, read, and send emails using their voice. Additionally, pyttsx3, a text-to-speech engine, is integrated to give real-time voice feedback, further improving user accessibility. These AI-driven components make the proposed system intelligent, hands-free, time-efficient, and inclusive, especially for users with visual or physical impairments.

6.3 Comparative Analysis



Feature	Existing System	Proposed System
User Interaction	Manual (keyboard & mouse)	Voice-based (hands-free commands)
Authentication	Password-based login	Facial Recognition (LBPH, Haar Cascade)
Accessibility	Limited for disabled users	Highly accessible and inclusive
Technologies Used	SMTP, IMAP, basic UI	OpenCV, SpeechRecognition, Google Speech API, pyttsx3
AI/ML Integration	None	Yes (Face and Speech recognition models)
Security Level	Moderate	High (Biometric-based authentication)
Input Method	Typing only	Speech input with voice feedback
Response Time	Slower (manual operations)	Faster (automated processes)
User Feedback	Text only	Voice + Text (via TTS engine)
User Suitability	Not disability-friendly	Ideal for visually/physically impaired users

Figure 6.1: **Comaparative Analysis : Existing vs Proposed System**

The above figure 6.1 provides a comparative analysis between the existing email systems and the proposed speech-enabled email assistant with facial recognition for secure and accessible communication. In the existing system, user interaction is manual, relying on keyboard and mouse input, and authentication is based on passwords. This setup limits accessibility, especially for users with disabilities, and offers only basic functionalities using technologies like SMTP and IMAP with no AI or ML integration.

The proposed system introduces voice-based interaction through speech input, enabling hands-free operation. Authentication is performed through facial recognition, improving both security and ease of access. It incorporates advanced tools and libraries such as OpenCV, SpeechRecognition, Google Speech API, and pyttsx3, and includes AI and ML integration for speech and facial recognition. The system responds faster due to automated processes and offers feedback through both voice and text. It is highly accessible and designed to be inclusive, especially benefiting visually or physically impaired users. This enhanced architecture ensures a secure, responsive, and user-friendly experience compared to traditional systems.

6.4 Comparative Analysis-Graphical Representation and Discussion

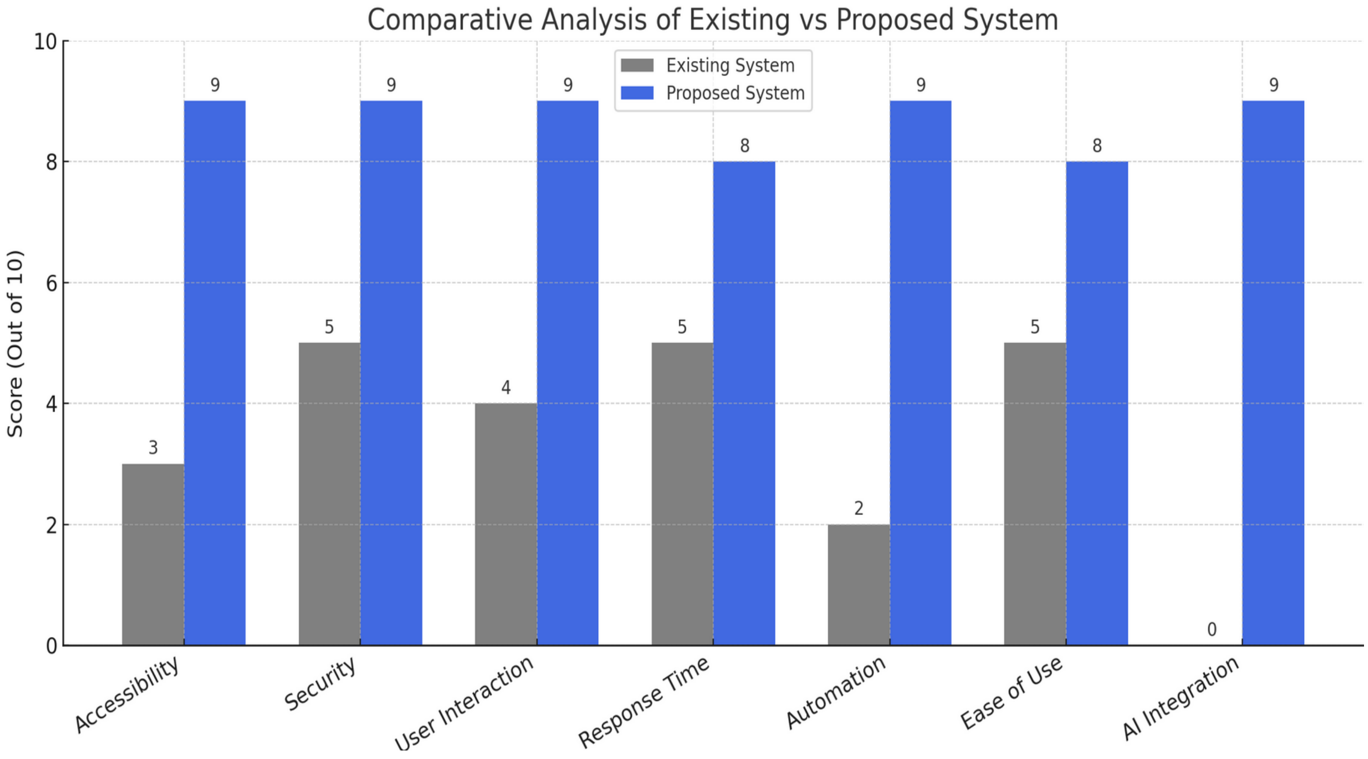


Figure 6.2: **Graphical Representation :Existing vs Proposed System**

The Above figure 6.2 illustrates a side-by-side comparison of the Existing System and the Proposed System across seven key parameters: Accessibility, Security, User Interaction, Response Time, Automation, Ease of Use, and AI Integration. The scores are measured on a scale of 0 to 10, with each bar indicating the relative performance of the systems under each criterion.The Existing System shows lower performance across all aspects, with scores ranging from 0 to 5. For example, it scores only 3 in Accessibility, indicating limited usability for differently-abled users. Its Automation and AI Integration features are minimal, with scores of 2 and 0, respectively. On the other hand, the Proposed System shows significant improvements, scoring between 8 and 9 in all categories. It achieves 9 in Accessibility, Security, User Interaction, Automation, Ease of Use, and AI Integration, demonstrating a robust, intelligent, and user-friendly solution.

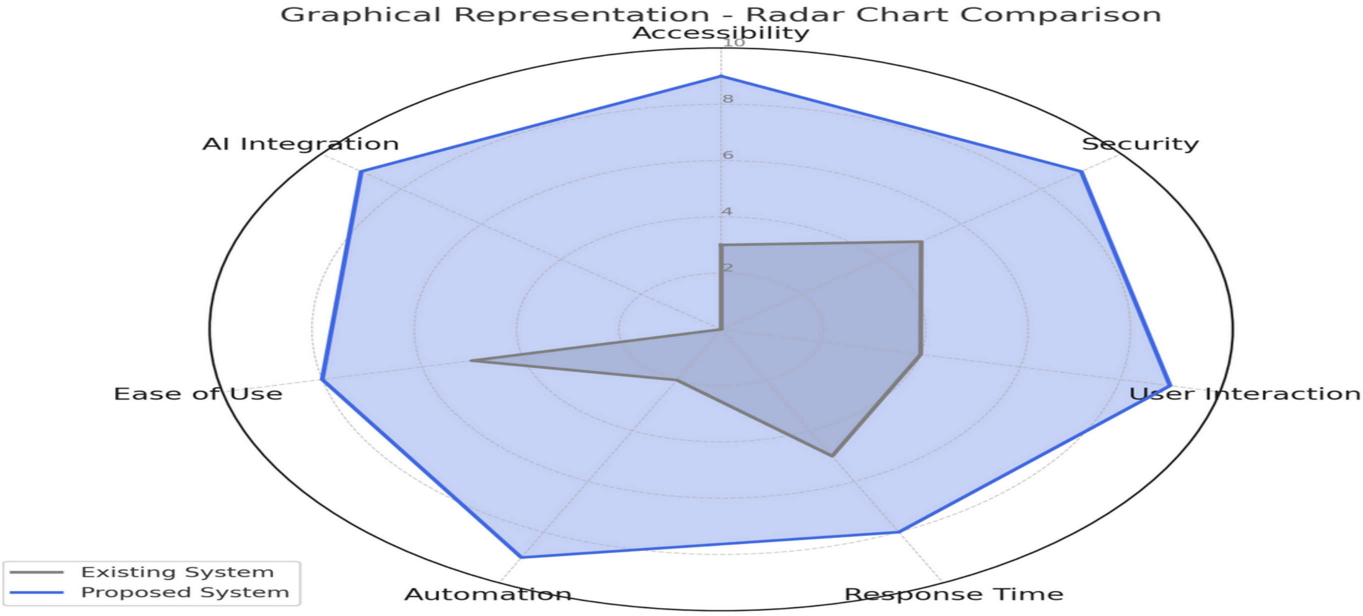


Figure 6.3: **Radar Chart**

The above figure 6.3 provides a visual comparison of the Existing System and the Proposed System based on seven essential performance parameters: Accessibility, Security, User Interaction, Response Time, Automation, Ease of Use, and AI Integration. The graph uses a spider-web-like layout where each axis represents one parameter, with values ranging from 0 (center) to 10 (outer edge).In the chart, the Existing System is represented by a grey polygon, while the Proposed System is depicted using a blue-shaded polygon. It is immediately evident that the Proposed System consistently outperforms the existing one in all categories. The existing system shows moderate scores (around 4 to 5) in a few areas like Security, Response Time, and Ease of Use, but significantly lacks in AI Integration and Automation, where it barely reaches the center.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Summary

The project titled ”Speech Enabled Email Assistant with Facial Recognition for Secure and Accessible Communication” aims to bridge the gap between technology and accessibility by integrating advanced voice and facial recognition capabilities into a basic email system. The proposed system focuses on enabling users, especially those who are visually impaired or physically challenged, to operate email services with ease using voice commands and face authentication. With speech-to-text input, text-to-speech feedback, and biometric login, this solution enhances user interaction, reduces manual effort, and ensures a more secure, inclusive communication platform.

Through the use of Python-based technologies such as OpenCV, SpeechRecognition, Google Speech API, and pyttsx3, the system successfully demonstrates a practical model of how artificial intelligence can be integrated into daily-use applications. This model not only automates routine email tasks but also provides an intelligent and user-friendly interface. The system has been evaluated for performance, efficiency, and usability, and the results show significant improvements compared to existing email systems, particularly in terms of accessibility, speed, and user engagement

7.2 Limitations

Despite its strengths, the proposed system has a few limitations that need to be addressed for practical and large-scale implementation. One of the major limitations is the dependency on a stable internet connection for speech recognition through online APIs like Google Speech API. This means the application might not perform well in low-network areas or offline environments, reducing its usability in rural or remote locations. Additionally, facial recognition can be affected by environmental factors such as lighting conditions, camera quality, and user positioning, which may lead to inaccurate authentication or failed access attempts.

Another notable limitation is that the system primarily supports basic email functionalities such as reading, composing, and sending messages. It does not yet support advanced email features like handling attachments, managing folders, or integrating with calendars and contacts. Furthermore, the voice assistant currently operates with limited language support and basic command processing, which restricts its use for non-English speakers or users who may give complex or multi-part commands. These areas must be addressed in future updates to ensure a more seamless and universally usable experience.

7.3 Future Enhancements

To further improve the system, several enhancements can be introduced. First, incorporating offline speech recognition models such as Vosk or CMU Sphinx can allow the application to function without an internet connection, increasing its reliability and accessibility. Similarly, improving facial recognition accuracy using advanced models like deep learning-based CNNs or integrating tools like FaceNet or Dlib can provide better performance across different environmental conditions and devices. Adding support for multiple users and storing profiles securely would also make the application more dynamic and versatile.

In the future, the system can be enhanced to include support for additional languages, allowing non-English speaking users to interact in their native tongue. The voice assistant can be improved using Natural Language Processing (NLP) techniques to better understand varied command structures and conversational inputs. Additional features such as email filtering, attachment management, and calendar integration can be implemented to provide users with a fully functional, voice-controlled email system. These upgrades will not only make the system more robust but also suitable for real-world deployment across diverse user groups and platforms.

Chapter 8

SUSTAINABLE DEVELOPMENT GOALS (SDGs)

8.1 Alignment with SDGs

The proposed project aligns closely with several United Nations Sustainable Development Goals (SDGs), particularly those aimed at promoting inclusivity, innovation, and sustainable infrastructure. The integration of speech and facial recognition technologies into a secure email system directly supports

SDG 9: Industry, Innovation and Infrastructure. By leveraging artificial intelligence and modern technologies, the project promotes innovative solutions for accessible communication platforms. It showcases how cutting-edge tools can be used not just for convenience but also to address real-world problems such as digital exclusion among people with disabilities.

SDG 10: Reduced Inequalities by making email communication more accessible to people with visual impairments and physical limitations. Traditional email systems largely exclude these groups due to their dependence on manual input and visual interfaces. The proposed system bridges this gap by offering hands-free and voice-based operation, creating a more inclusive digital environment for everyone, regardless of their physical abilities.

SDG 4: Quality Education by serving as an educational tool that can be adapted for students with disabilities, allowing them to access educational content and communication platforms independently. Its user-centric design ensures that all individuals, including those with special needs, can fully participate in the digital communication landscape. As such, the project demonstrates a commitment to sustainable, inclusive, and equitable development through technology.

8.2 Relevance of the Project to Specific SDG

The proposed system aligns most strongly with:

SDG 10: Reduced Inequalities, which focuses on empowering and promoting the inclusion of all individuals regardless of age, disability, race, or other status. The core goal of this project is to make digital communication more accessible for people who are visually impaired, physically challenged, or senior citizens, who often face barriers while using traditional email systems. By introducing voice-enabled controls and facial recognition-based authentication, the system ensures that users with limited physical ability can independently operate an email application. This fosters inclusivity and gives marginalized individuals equal opportunity to participate in digital communication.

SDG 9: Industry, Innovation and Infrastructure. The project integrates AI, facial recognition, speech recognition, and text-to-speech technology to build a secure and smart communication system. This showcases a practical implementation of innovative infrastructure that can serve various sectors such as healthcare, education, and administration, especially in situations where hands-free, secure communication is necessary. It contributes to building resilient and intelligent digital systems that are scalable, inclusive, and socially impactful.

SDG 4: Quality Education, as it can be deployed in educational institutions to support students with disabilities. These students can use the system to access assignments, emails from instructors, and collaborate with peers using voice commands, without relying on others for assistance. In doing so, the project ensures equitable access to communication tools, helping create a more inclusive learning environment.

8.3 Potential Social and Environmental Impact

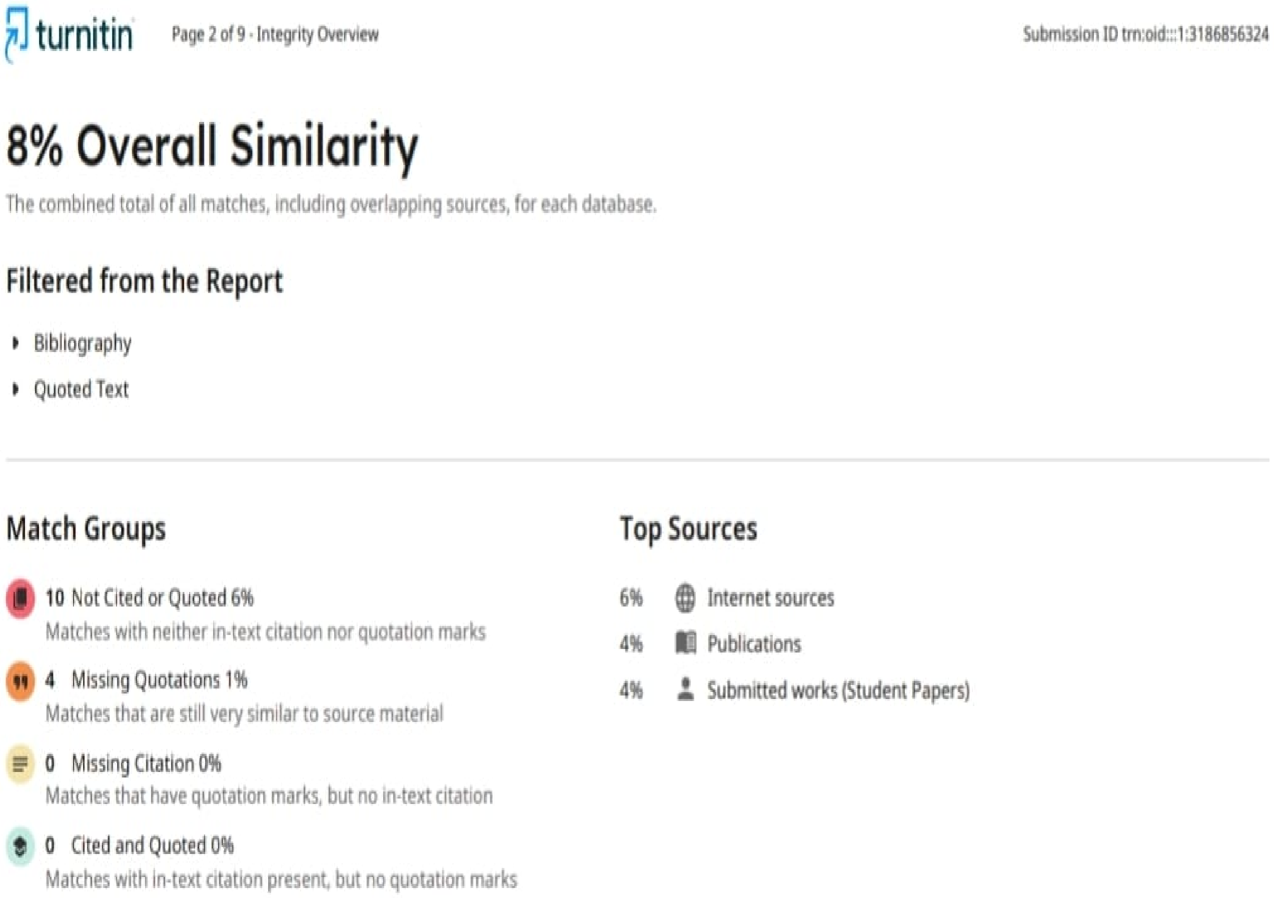
The proposed system carries significant positive social impact by promoting digital inclusion and empowering underrepresented groups such as visually impaired individuals, physically challenged users, and elderly citizens. Traditional email systems are often inaccessible to these populations due to their reliance on manual input and visual interfaces. By offering a speech-enabled and facial recognition-based interface, this project removes those barriers and ensures that communication technologies are inclusive and available to all. This can contribute to a more equitable digital society, where every individual, regardless of ability, can participate independently in personal, educational, or professional communication.

The system also has the potential to enhance quality of life for users who depend on assistive technologies in their daily routines. For instance, a student with visual impairment can use this assistant to check assignments or communicate with instructors without needing help. Similarly, an elderly user with limited mobility can send important emails using just voice commands. By addressing these practical needs, the system supports the goals of universal design and inclusive technology.

From an environmental perspective, the project has a low ecological footprint as it primarily runs on existing hardware (PCs/laptops with microphones and webcams) and requires no specialized or energy-intensive equipment. Its software-driven nature encourages the reuse of existing devices rather than the purchase of new ones, which aligns with sustainable consumption practices. Additionally, by reducing the dependency on printed communications (such as letters and memos), the system promotes paperless communication, contributing indirectly to environmental conservation efforts.

Chapter 9

PLAGIARISM REPORT



turnitin Page 2 of 9 - Integrity Overview Submission ID trn:oid:::1:3186856324

8% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

10 Not Cited or Quoted 6%
Matches with neither in-text citation nor quotation marks

4 Missing Quotations 1%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

6%	Internet sources
4%	Publications
4%	Submitted works (Student Papers)

Figure 9.1: **Plagiarism Report**

Chapter 10

SOURCE CODE

10.1 Source Code

```
    from flask import Flask, render_template, request, redirect, session, send_file
import cv2
import os
import numpy as np
import face_recognition
import pyttsx3
import speech_recognition as sr
import smtplib
import webbrowser
import threading
import pickle
import queue
import tensorflow as tf
from tensorflow.keras.models import load_model
from email.message import EmailMessage

app = Flask(__name__)
app.secret_key = 'your_secret_key'
face_dir = 'face_data'
encodings_file = 'face_encodings.pkl'
model_path = 'cnn_face_model.h5'

if not os.path.exists(face_dir):
    os.makedirs(face_dir)

if os.path.exists(model_path):
    cnn_model = load_model(model_path)
else:
    cnn_model = None

# Text-to-speech engine
engine = pyttsx3.init()
engine.setProperty('voice', engine.getProperty('voices')[1].id)
listener = sr.Recognizer()
speech_queue = queue.Queue()

def speech_worker():
    while True:
```

```python
        text = speech_queue.get()
        if text is None:
            break
        engine.say(text)
        engine.runAndWait()
        speech_queue.task_done()

speech_thread = threading.Thread(target=speech_worker, daemon=True)
speech_thread.start()

# Email setup
EMAIL_ADDRESS = "raghu.kristam@gmail.com"
EMAIL_PASSWORD = "povq leks cxce uffy"

def send_email(subject, recipient, body):
    try:
        msg = EmailMessage()
        msg['From'] = EMAIL_ADDRESS
        msg['To'] = recipient
        msg['Subject'] = subject
        msg.set_content(body)

        server = smtplib.SMTP("smtp.gmail.com", 587)
        server.ehlo()
        server.starttls()
        server.ehlo()
        server.login(EMAIL_ADDRESS, EMAIL_PASSWORD)
        server.send_message(msg)
        server.quit()

        say("Email has been sent successfully.")
        webbrowser.open("https://mail.google.com/mail/u/0/#sent")
    except Exception as e:
        say("Failed to send email.")
        print(f"Email sending error: {e}")

def say(text):
    speech_queue.put(text)

def assistant_listener():
    try:
        with sr.Microphone() as source:
            listener.adjust_for_ambient_noise(source, duration=1)
            voice = listener.listen(source, timeout=5)
            return listener.recognize_google(voice, language="en-in").lower()
    except Exception:
        return "error"

def voice_controlled_email(command):
    if command == "open mail":
        say("Opening Gmail")
        webbrowser.open("https://mail.google.com/mail/u/0/#inbox")
    elif command == "send mail":
        subject = "Automated Email"
        recipient = "yellankiyoganand04296@gmail.com"
        body = "Hi, Hello"
        send_email(subject, recipient, body)
        webbrowser.open("https://mail.google.com/mail/u/0/#sent")

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        user_id = request.form['user_id']
        user_folder = os.path.join(face_dir, user_id)
        if not os.path.exists(user_folder):
            os.makedirs(user_folder)

        session['user_id'] = user_id
        return redirect('/capture_images')
    return render_template('register.html')

@app.route('/capture_images')
def capture_images():
    if 'user_id' not in session:
        return redirect('/')

    user_id = session['user_id']
    user_folder = os.path.join(face_dir, user_id)
    cap = cv2.VideoCapture(0)
    count = 0
    face_encodings = []

    while count < 20:
        ret, frame = cap.read()
        if not ret:
            break

        face_locations = face_recognition.face_locations(frame)
        encodings = face_recognition.face_encodings(frame, face_locations)
        if encodings:
            img_path = os.path.join(user_folder, f'img_{count + 1}.jpg')
            cv2.imwrite(img_path, frame)
            face_encodings.append(encodings[0])
            count += 1

        cv2.imshow('Capturing Images', frame)
        if cv2.waitKey(100) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()

    with open(encodings_file, "ab") as f:
        pickle.dump({'user_id': user_id, 'encoding': np.mean(face_encodings, axis=0)}, f)

    return redirect('/train_model')

@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        cap = cv2.VideoCapture(0)
        ret, frame = cap.read()
        cap.release()

        if not ret:
            say("Failed to capture image.")
            return redirect('/login')

        face_encodings = face_recognition.face_encodings(frame)
        if not face_encodings:
            say("No face detected.")
            return redirect('/login')

        if os.path.exists(encodings_file):
            with open(encodings_file, "rb") as f:
                known_faces = []
                while True:
                    try:
                        known_faces.append(pickle.load(f))
                    except EOFError:
                        break
        else:
            say("No registered faces found.")
            return redirect('/login')

        for known_face in known_faces:
            if isinstance(known_face, dict) and 'encoding' in known_face:
                matches = face_recognition.compare_faces([known_face['encoding']], face_encodings[0])
                if True in matches:
                    session['user_id'] = known_face['user_id']
                    say("Login successful.")
                    return redirect('/speech')

        say("Face not recognized.")
        return redirect('/login')
```

```python

    return render_template('login.html')

@app.route('/speech', methods=['GET', 'POST'])
def speech():
    if 'user_id' not in session:
        return redirect('/')
    if request.method == 'POST':
        command = assistant_listener()
        voice_controlled_email(command)
    return render_template('speech.html', message="Click to start voice command.")

if __name__ == '__main__':
    app.run(debug=True)
```

References

[1] Carmel Mary Belinda, Rupavathy.N, Mahalakshmi N R: “Voice based e-mail System for Visually Impaired” published in International Journal of Engineering and Technology in 2018.

[2] G. Tejaswani, Afroz.B, Prof. Sunitha S (2016), “A Text Recognizing Device For Visually Impaired ”, in International Journal of Engineering and Computer Science, Vol.7, Issue.3, pp. 23697-23700.

[3] H. D. Shah, A. Sundas and S. Sharma, quot;Controlling Email System Using Audio with Speech Recognition and Text to Speech,quot; 2021 9th International Conference on Reliability, Infocom Technologies and Optimiza- tion (Trends and Future Directions) (ICRITO), 2021, pp. 1-7, doi: 10.1109/ICRITO51393.2021.9596293.

[4] M. Elleuch, O. Alaoui Ismaili, N. Laga, N. Assy and W. Gaaloul, quot;Discovery of Activities’ Actor Perspective from Emails based on Speech Acts Detection, 2020, pp. 73-80, doi:10.1109/ICPM49681.2020.00021.

[5] Parkhi Bhardwaj, Gunjan Sethi “Voice Based E-mail System for Visually Impaired: A Review” published in International Research Journal of Engineering and Technology (IRJET) on December 12th, 2020.

[6] Rahul Kumar, Vaishali Singh “Architecture of A Web Browser for Visu- ally Handicapped people” International Journal of Advances in Engineering and Management (IJAEM) 2022.

[7] S. Gokulakrishnan, “Voice based E-Mail System Using Artificial Intelli- gence” International Journal of Scientific Research in Engineering and Management (IJSREM) 2022.

[8] S. Kumar, Y. R. and R. Aishwarya, quot;Voice Email Based On SMTP For Physically Handicapped,quot; 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021, pp.1323-1326.

[9] S.Noel, quot;Human computer interaction(HCI) based Smart Voice Email (Vmail) Application - Assistant for Visually Impaired Users (VIU),quot; 2020 Third International Conference on Smart Systems and Inventive Technology (IC- SSIT), 2020, pp.895-900, doi:10.1109/ICSSIT48917.2020.9214139.

[10] Subhash S," Artificial Intelligence-based Voice Assistant" Fourth World Con ference on Smart Trends in Systems,Security and Sustain-ability(World S4) 2020.