

A PROJECT REPORT ON
“Face Sketch Creation and Recognition”

Submitted in the partial fulfillment of the requirements for the
Award of the degree
Of
BACHELOR OF COMPUTER ENGINEERING

SUBMITTED BY

Sanket Bapu Kale

Exam No: B190704225

Mayur Sunil Shinde

Exam No: B190704234

Abhishek Ashok Gavhane

Exam No: B190704218

Pramod Ananda Rumale

Exam No: B190704248

UNDER THE GUIDANCE OF
Prof. Anoop Kushwaha



DEPARTMENT OF COMPUTER ENGINEERING
ALARD COLLEGE OF ENGINEERING AND MANAGEMENT,
MARUNJI, PUNE.

SAVITRIBAI PHULE PUNE UNIVERSITY

2023 – 2024.



CERTIFICATE

This is to certify that the project report entitles

“Face Sketch Creation and Recognition”

Submitted by

Sanket Bapu Kale

Exam No: B190704225

Mayur Sunil Shinde

Exam No: B190704234

Abhishek Ashok Gavhane

Exam No: B190704218

Pramod Ananda Rumale

Exam No: B190704248

is a Bonafide student of this institute and the work has been carried out by him/her under the supervision of **Prof. Anoop Kushwaha** and it is approved for the partial fulfilment of the requirement of Savitribai Phule Pune University, for the award of the degree of Bachelor of Engineering (Computer Engineering).

Prof Anoop Kushwaha
Project Guide
Dept. of Computer Engg.

Prof. Sayali Haldavanekar
Head of Department
Dept. of Computer Engg.

External Examiner

Dr. K.D Sapate
Principal

Acknowledgements

It is our privilege to express our sincerest regards to our project guide, **Prof. Anoop Kushwaha**, for his valuable input, guidance, encouragement, whole-hearted cooperation, and constructive criticism throughout the duration of our project.

We deeply express our sincere thanks to our Head of Department **Prof. Sayli M. Haldavanekar** for encouraging and allowing us to present the project on the topic “**Face Sketch Creation and Recognition**” at our department premises for the partial fulfilment of the requirements leading to the award of Bachelor degree in Computer Engineering.

We take this opportunity to thank all our faculty members who directly or indirectly helped our project. We pay our respect and love to our parents and all other family members and friends for their love and encouragement throughout our careers. Last but not least we thank our friends for their cooperation and support.

Abstract

In this project, we aim to revolutionize the way law enforcement identifies suspects in criminal investigations. Forensic face sketching plays a vital role in criminal investigations by aiding in suspect identification. We're developing an innovative system that combines deep learning techniques and cloud infrastructures with forensic sketching to create and recognize facial sketches quickly and accurately. Traditional forensic sketching has a rich history, but it's time to modernize this process. A proposed solution involves developing a standalone application for composite sketching, leveraging drag-and-drop features to bypass the need for forensic artists, using cutting-edge technology to not only create detailed facial sketches but also rapidly cross-reference them with law enforcement databases for swift suspect identification. Our goal is to improve the efficiency of the criminal identification process, making it faster and more effective in today's world, where crime rates are on the rise. This project has the potential to greatly assist investigatory organizations in their pursuit of justice.

Table of Contents

	Topic	Page No.
	Acknowledgement	
	Abstract	
1.	INTRODUCTION	1-2
	1.1. Overview	2
	1.2. Motivation	2
	1.3. Problem Statement	2
2.	LITERATURE SURVEY	3-6
3.	REQUIREMENT SPECIFICATIONS	7-10
	3.1. Hardware Requirements	8
	3.2. Software Requirements	8
	3.3. Functional Requirements	8
	3.4. Non-Functional Requirements	8
	3.5. System Requirements	9
	3.6. Analysis Model	9
4.	SYSTEM DESIGN	11-17
	4.1. System Architecture	12
	4.2. UML Diagrams	14
	4.2.1 Use case Diagram.	14
	4.2.2 Activity Diagram	15
	4.2.3 Sequence Diagram	16
	4.2.4 Class Diagram.	17

5.	IMPLEMENTATION DETAILS	18-27
5.1.	Description of Tools	19
5.2.	Programming Language Description	20
5.3.	Algorithm Details	22
5.4.	Output of Implementation (In form of Screen Shots)	24
6.	TESTING	28-31
6.1.	Types of Testing	29
6.2.	Test Cases	30
7.	RESULTS AND CONCLUSION	32-33
7.1.	Conclusion	33
7.2.	Future Scope	33

Appendices

Reference

List of Figures

3.1	Waterfall model	10
4.1	System flow of application (A)	12
4.1	Flow chart for creating sketch (B)	13
4.2.1	Use Case Diagram	14
4.2.2	Activity Diagram	15
4.2.3	Sequence Diagram	16
4.2.4	Class Diagram	17
5.3.1	Splash screen for standalone desktop application	24
5.3.2	Login Screen	24
5.3.3	Dashboard to create facial sketch	25
5.3.4	Complete face sketch in dashboard	25
5.3.5	Save face sketch in the system	26
5.3.6	Dashboard for recognition	26
5.3.7	Opened face sketch from the system	27
5.3.8	Face sketch matched to the database record	27

Acknowledgements

It is our privilege to express our sincerest regards to our project guide, **Prof. Anoop Kushawa**, for his valuable input, guidance, encouragement, whole-hearted cooperation, and constructive criticism throughout the duration of our project.

We deeply express our sincere thanks to our Head of Department **Prof. Sayli M. Haldavanekar** for encouraging and allowing us to present the project on the topic “**Face Sketch Creation and Recognition**” at our department premises for the partial fulfilment of the requirements leading to the award of Bachelor degree in Computer Engineering.

We take this opportunity to thank all our faculty members who directly or indirectly helped our project. We pay our respect and love to our parents and all other family members and friends for their love and encouragement throughout our careers. Last but not least we thank our friends for their cooperation and support.

Abstract

In this project, we aim to revolutionize the way law enforcement identifies suspects in criminal investigations. Forensic face sketching plays a vital role in criminal investigations by aiding in suspect identification. We're developing an innovative system that combines deep learning techniques and cloud infrastructures with forensic sketching to create and recognize facial sketches quickly and accurately. Traditional forensic sketching has a rich history, but it's time to modernize this process. A proposed solution involves developing a standalone application for composite sketching, leveraging drag-and-drop features to bypass the need for forensic artists, using cutting-edge technology to not only create detailed facial sketches but also rapidly cross-reference them with law enforcement databases for swift suspect identification. Our goal is to improve the efficiency of the criminal identification process, making it faster and more effective in today's world, where crime rates are on the rise. This project has the potential to greatly assist investigatory organizations in their pursuit of justice.

INTRODUCTION

1. INTRODUCTION

1.1 OVERVIEW

Increasing crime rates necessitate faster and more efficient processes for law enforcement departments to bring criminals to justice. Face recognition technology can be a valuable tool for identifying and verifying offenders. The traditional method of using hand-drawn face sketches by forensic artists has limitations and can be time-consuming. However, this approach faces limitations with current technologies and there are few skilled forensic sketch artists. There is a need for an application that provides a comprehensive set of facial features for creating face sketches. This application would expedite the identification of criminals in a faster and more efficient manner.

1.2 MOTIVATION

This project aims to make it easier for forensic department to catch criminals by improving the way they identify suspects. Right now, forensic departments often use hand-drawn sketches to help find suspects, but this can take a long time and there aren't always enough skilled artists available. We want to change that by creating a new computer program that can quickly make accurate sketches of suspects' faces. This program will use advanced technology to create detailed sketches and then compare them with databases of known criminals to find matches faster. By doing this, we hope to help police catch criminals more quickly and efficiently, especially at a time when crime rates are increasing.

1.3 PROBLEM STATEMENT

To develop an Application for Face Sketch Creation and Recognition.

LITERATURE SURVEY

2. LITERATURE SURVEY

2.1 STUDY OF RESEARCH PAPER -

1. Analyzing the Scientific Evolution of Face Recognition Research and Its Prominent Subfields:

Yahya Zennayi, This paper presents a science mapping approach to analyze the thematic evolution of face recognition research. For this reason, different bibliometric tools are combined (performance analysis, science mapping and Co-word analysis) in order to identify the most important, productive and the highest impact subfields. Moreover, different visualization tools are used to display a graphical vision of face recognition field to determine the thematic domains and their evolutionary behavior. Finally, this study proposes the most relevant lines of research for the face recognition field. Findings indicate a huge increase in face recognition research since 2014. Mixed approaches revealed a great interest compared to local and global approaches. In terms of algorithms, the use of deep learning methods is the new trend. On the other hand, the illumination variation impact on face recognition algorithms performances is nowadays, the most important and impacting challenge for the face recognition field.

2. Recent Advances in Deep Learning Techniques for Face Recognition:

Jakaria Rabbi; Mabrook S. Al-Rakhami, In recent years, researchers have proposed many deep learning (DL) methods for various tasks, and particularly face recognition (FR) made an enormous leap using these techniques. Deep FR systems benefit from the hierarchical architecture of the DL methods to learn discriminative face representation. Therefore, DL techniques significantly improve state-of-the-art performance on FR systems and encourage diverse and efficient real-world applications. In this paper, we present a comprehensive analysis of various FR systems that leverage the different types of DL techniques, and for the study, we summarize 171 recent contributions from this area. We discuss the papers related to different algorithms, architectures, loss functions, activation functions, datasets, challenges, improvement ideas, current and future trends of DL-based FR systems. We provide a detailed discussion of various DL methods to understand the current state-of-the-art, and then we discuss various activation and loss functions for the methods. Additionally, we summarize different datasets used widely for FR tasks and discuss challenges related to illumination, expression, pose variations, and occlusion. Finally, we discuss improvement ideas, current and future trends of FR tasks.

3. Face Recognition: A Convolutional Neural-Network Approach:

Steve Lawrence, C. Lee Giles, Ah Chung Tsoi, Andrew D. Back, This paper introduces a novel hybrid neural network solution for face recognition, offering a favourable comparison against other existing methods. The system employs a combination of local image sampling, a self-organizing map (SOM) neural network, and a convolutional neural network(CNN). By utilizing the SOM, the system achieves dimensionality reduction and invariance to minor image changes, while the convolutional neural network provides partial invariance to translation, rotation, scale, and deformation. Results from experiments using different approaches, such as the Karhunen–Loeve (KL) transform and a multilayer perceptron (MLP), are presented. The proposed method consistently outperforms the eigenfaces approach, particularly when considering databases with varying degrees of variability in expression, pose, and facial details. The paper also discusses computational complexity and the potential for adding new classes to the trained recognizer.

4. Face Sketch Creation and Identification:

T.D. Zope, H.L.Koli, Forensic face sketching is a technique used by Investigatory organizations to help identify suspects in criminal investigations. This paper will discuss the process of forensic sketch construction and identification. The paper will begin by providing an over view of the history of forensic sketching and its role in criminal investigations. It will then describe the process of creation of a face sketch, including the initial interview, sketching techniques, and finalizing the sketch. The paper will also discuss the challenges and limitations of forensic sketching and identify potential solutions to these challenges. The paper will conclude by giving a summary of the present status of the research in this field and proposing possible are as for further study.

5. Forensic Face Sketch Construction and Recognition:

A.Patil, H.Dinkar, P.R.Tambe, Prof. Diksha Bhawe, In forensic science, it is seen that hand-drawn face sketches are still very limited and time-consuming when it comes to using them with the latest technologies used for recognition and identification of criminals. In this paper, we present a standalone application that would allow users to create composite face sketches of the suspect without the help of forensic artists using the drag and drop feature in the application and can automatically match the drawn composite face sketch with the police database much faster and efficiently using deep learning and cloud infrastructure. Keywords: Forensic Face Sketch, Face Sketch Construction, Face Recognition, Criminal Identification, Deep Learning, Machine Learning, Two-Step Verification.

6. Face Sketch Synthesis and Recognition:

Xiaoou Tang and Xiaogang Wang, In this paper, we propose a novel face photo retrieval system using sketch drawings. By transforming a photo image into a sketch, we reduce the difference between photo

and sketch significantly, thus allow effective matching between the two. To improve the synthesis performance, we separate shape and texture information in a face photo, and conduct transformation on them respectively. Finally, a Bayesian classifier is used to recognize the probing sketch from the synthesized pseudo-sketches. Experiments on a data set containing 606 people clearly demonstrate the efficacy of the algorithm.

REQUIREMENT SPECIFICATIONS

3. REQUIREMENT SPECIFICATIONS

3.1. Hardware Requirements –

- Hardware: Intel core i5
- Speed: 2.80 GHz
- RAM: 8GB
- Hard Disk: 500 GB
- Key Board: Standard Windows K-Software

3.2. Software Requirements –

- Operating System: Windows 7(64 Bit) and Above
- IDE: NetBeans (Preferred)
- Programming Language: JavaFx, Java
- Database: SQLite
- Other: AWS

3.3. Functional Requirements –

The various functional requirements of the system can be summarized as: -

- User Authentication
- Facial Sketch Generation
- Proper Validation
- Criminal Database
- Cloud Infrastructure

3.4. Non-Functional Requirements –

3.4.1 Performance Requirements –

- The performance of the functions and every module must be well. The overall performance of the software will enable the forensic users to work. The performance of encryption of data should be fast. The performance of the virtual environment should be fast.
- The application is designed in modules where errors can be detected easily. This makes it easier to install and update new functionality if required.

3.4.2 Safety Requirement –

- The application is designed in modules where errors can be detected and fixed easily. This makes it easier to install and update new functionality if required.

3.4.3 Software Quality Attributes –

Our software has many qualities and attributes that are given below: -

Adaptability: This software is adaptable by some forensic users.

Maintainability: After the deployment of the project if any error occurs then it can be easily maintained by the software developer.

Reliability: The performance of the software is better which will increase the reliability of the Software.

User Friendliness: Since the software is a GUI application; the output generated is much user friendly in its behaviour.

Integrity: Integrity refers to the extent to which access to software or data by unauthorized persons can be controlled.

Security: Users are authenticated using many security phases so reliable security is provided.

Testability: The software will be tested considering all the aspects.

3.5. System Requirements –

Login is required from the login page.

Upload Sketch

Import Drawn Sketch

Make Rought Sketch

Criminal Database

3.6. Analysis Model –

Here for the development of this project, we have followed the **Waterfall Model**. The classical waterfall model is the basic software development life cycle model. It is very simple but idealistic. The classical waterfall model divides the life cycle into a set of phases. This model considers that one phase can be started after the completion of the previous phase. That is the output of one phase will be the input to the next phase. Thus, the development process can be considered as a sequential flow in the waterfall.

Waterfall model phases followed during this project are as follows:

- Requirement Analysis
- System Design

- Implementation
- System Testing
- System Deployment
- System Maintenance

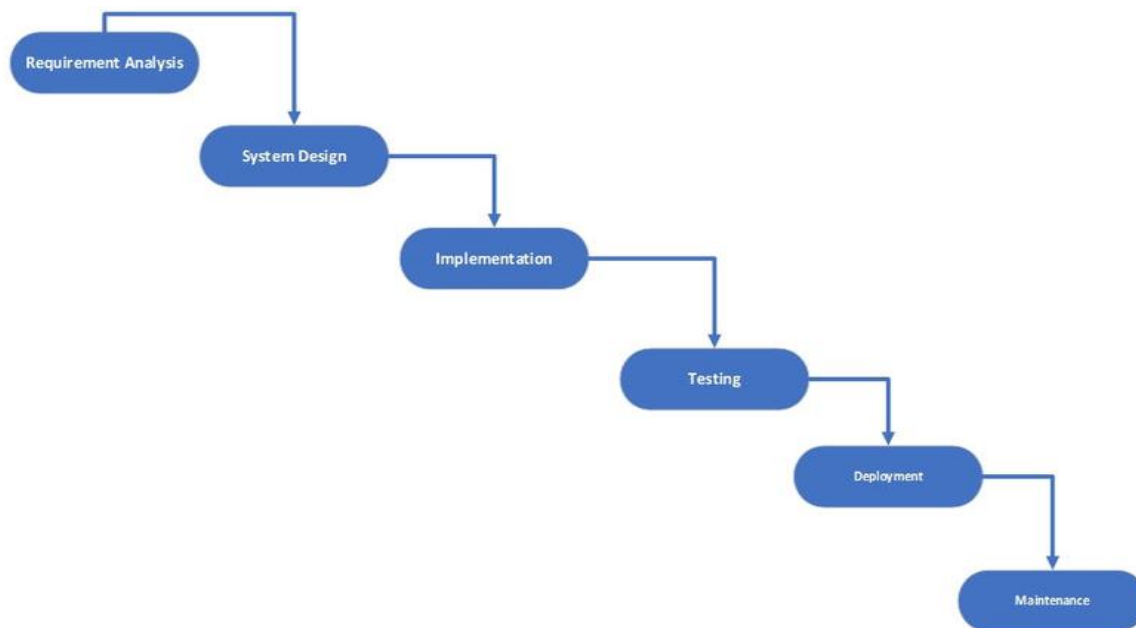


Fig 3.4.5.1: Waterfall Model

- **Requirement Analysis:** In this phase, you must define the requirements. You should explain business opportunities and plan the time and effort needed to build the project. Based on this information, you can evaluate technical and economic feasibility.
- **System Design:** When you have identified the project, work with stakeholders to define requirements. You can use the user flow diagram or the high-level UML diagram to show the work of new features and show how it will apply to your existing system.
- **Implementation:** When the team defines the requirements, the work begins. Designers and developers start working on their project, which aims to deploy a working product. The product will undergo various stages of improvement, so it includes simple, minimal functionality.
- **System Testing:** In this phase, the Quality Assurance team examines the product’s performance and looks for the bug.
- **System Deployment:** In this phase, the team issues a product for the user’s work environment.
- **System Maintenance:** Maintenance is the most important phase of a software life cycle. The effort spent on maintenance is 60% of the total effort spent to develop a full software.

SYSTEM DESIGN

4. SYSTEM DESIGN

4.1. System Architecture –

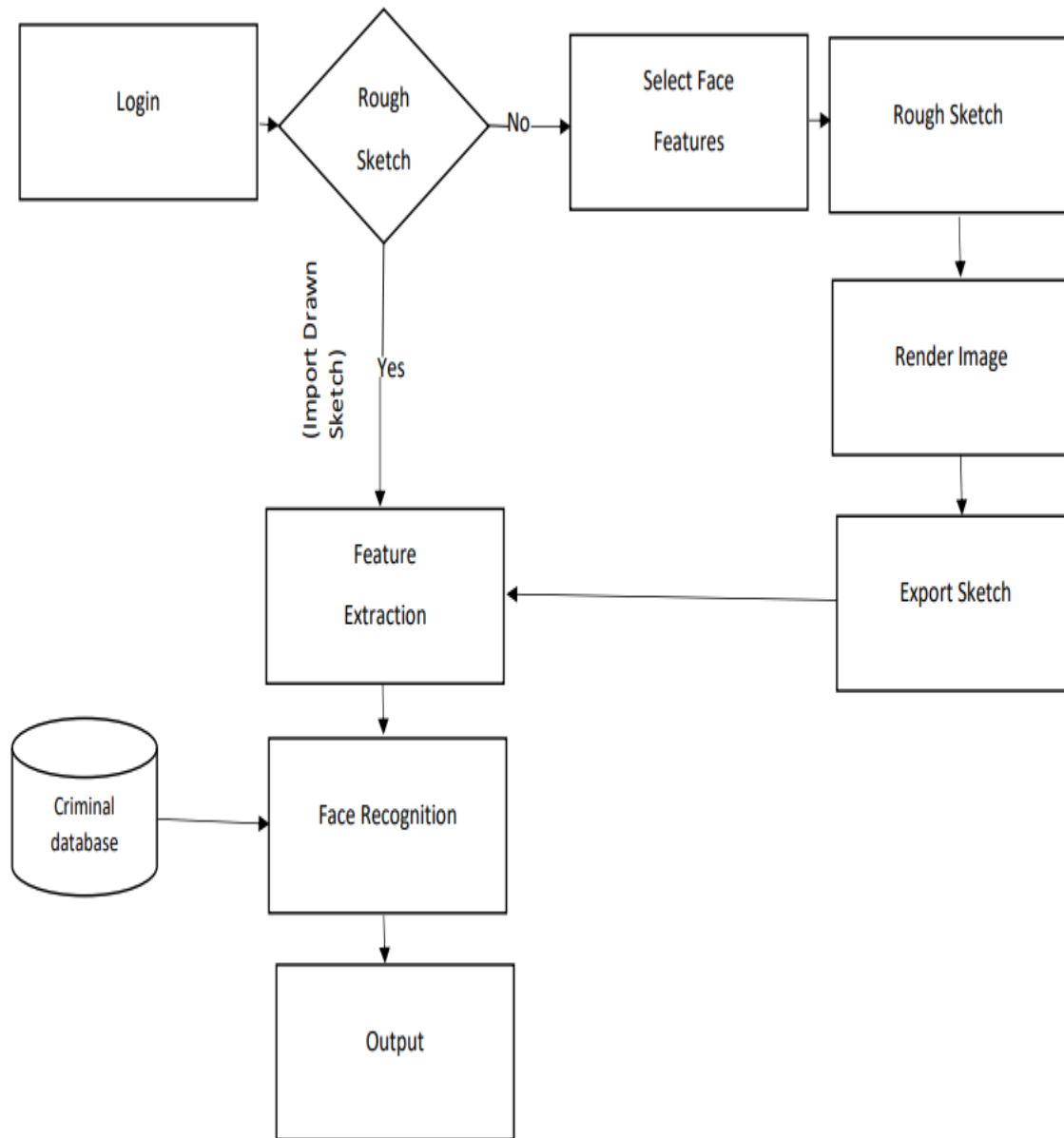


Figure 4.1(a): System Flow of Application

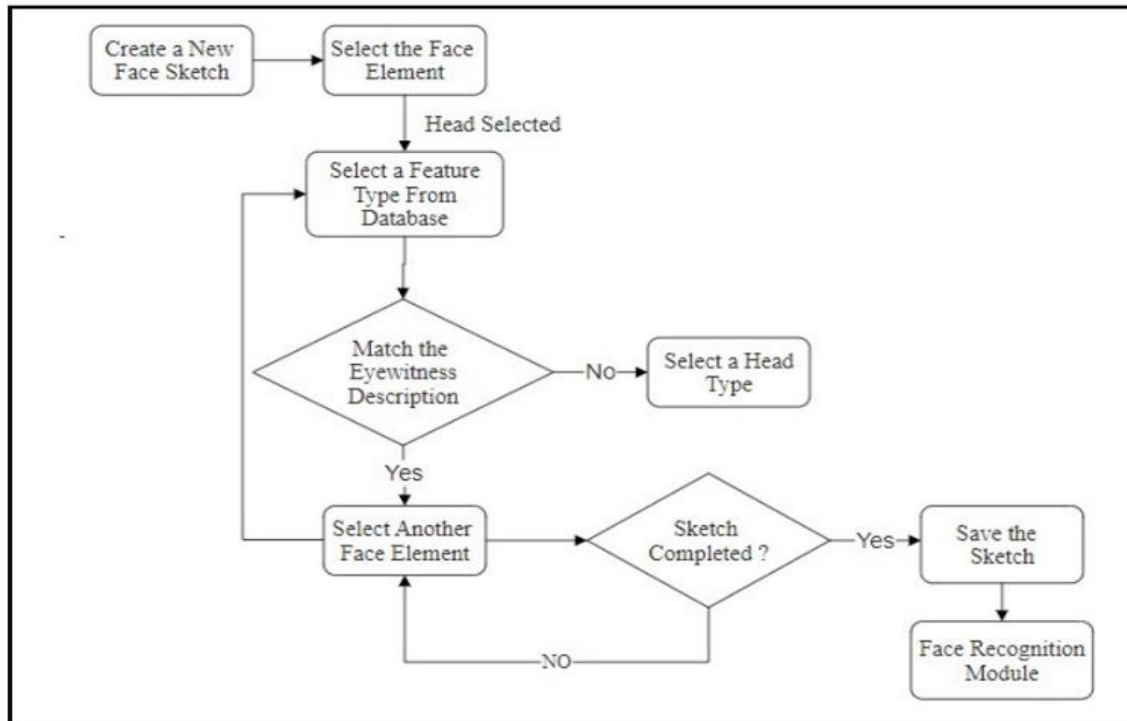


Fig 4.1(b): Flow chart for creating a sketch in the application

4.2. UML Diagrams –

4.2.1 Use case Diagram-

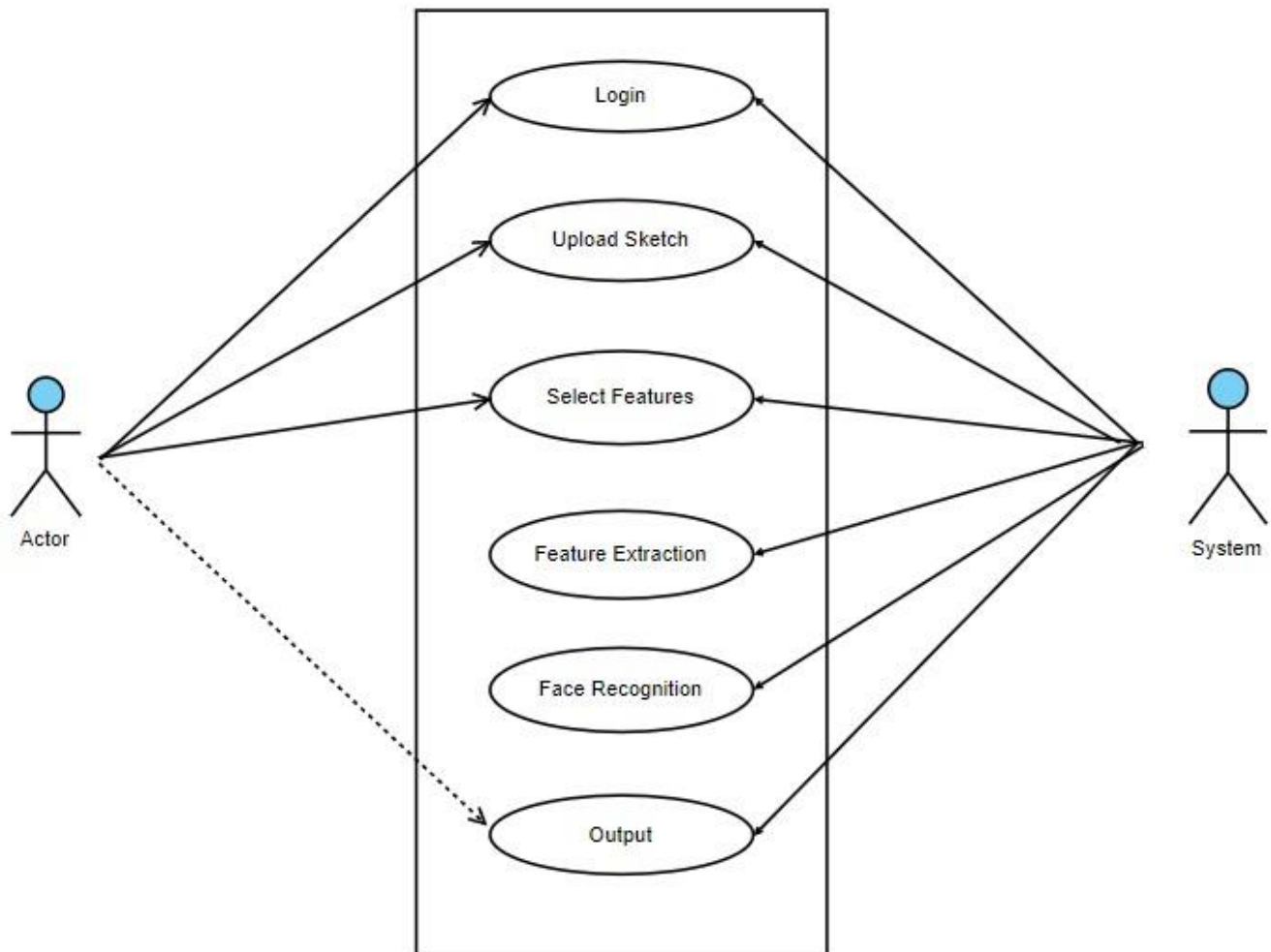


Figure 4.2.1: Use Case Diagram

4.2.2 Activity Diagram –

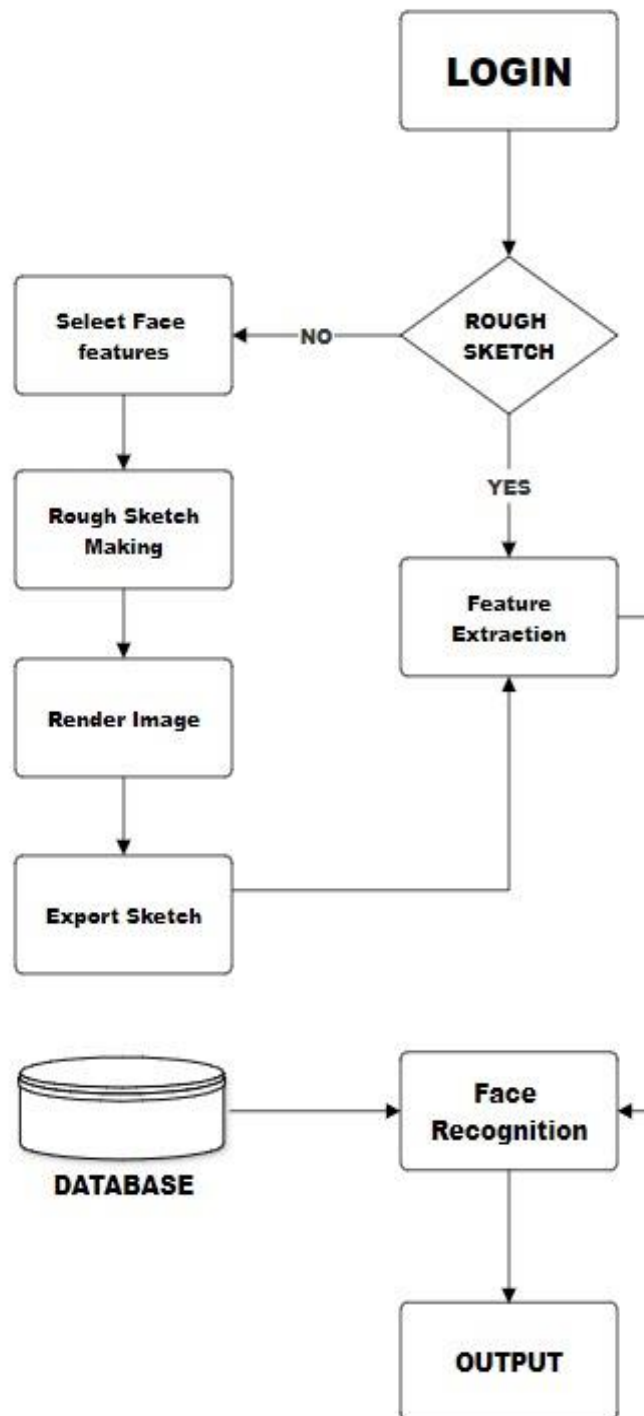


Figure 4.2.2: Activity Diagram

4.2.3 Sequence Diagram –

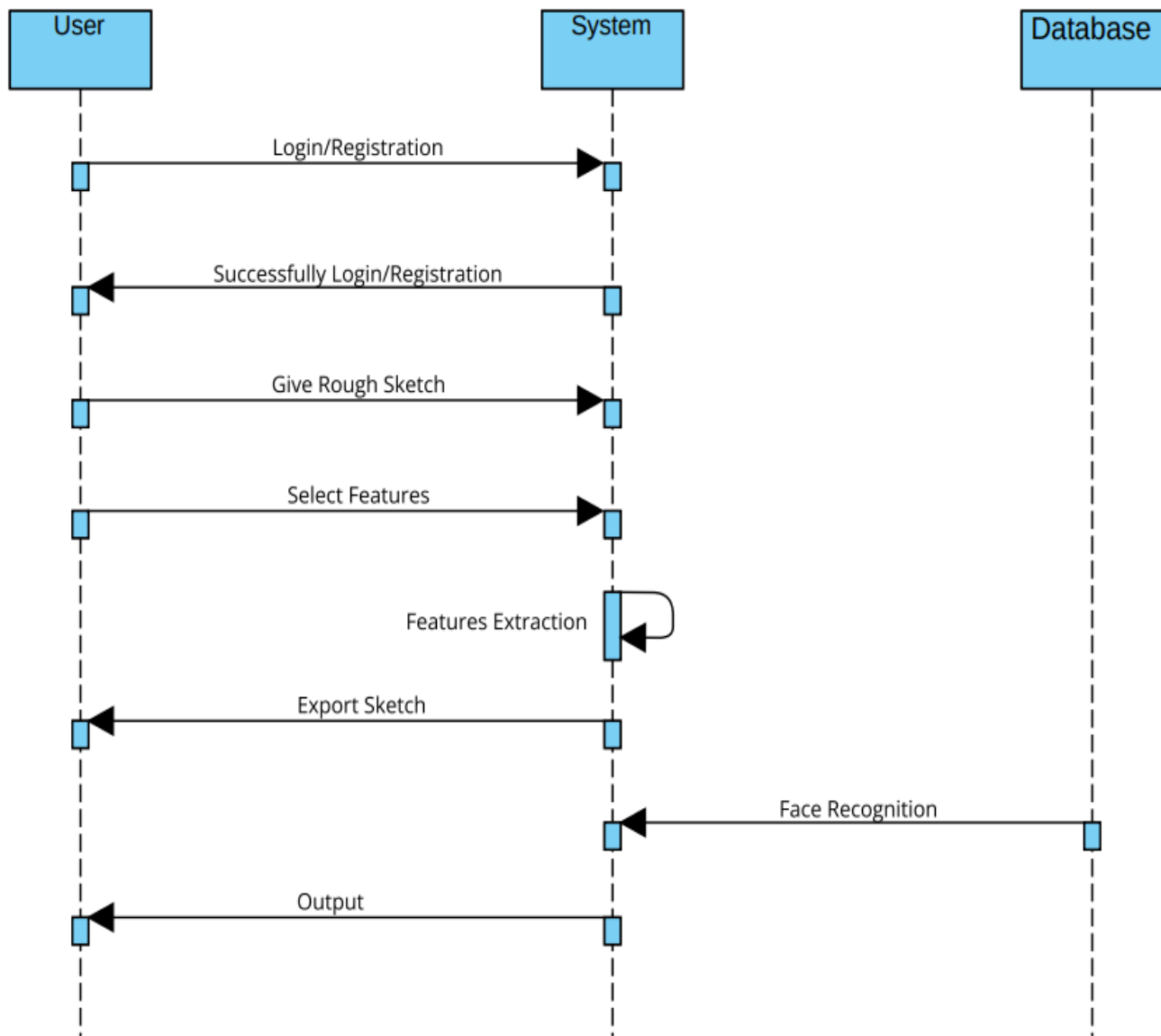


Figure 4.2.3: Sequence Diagram

4.2.4 Class Diagram-

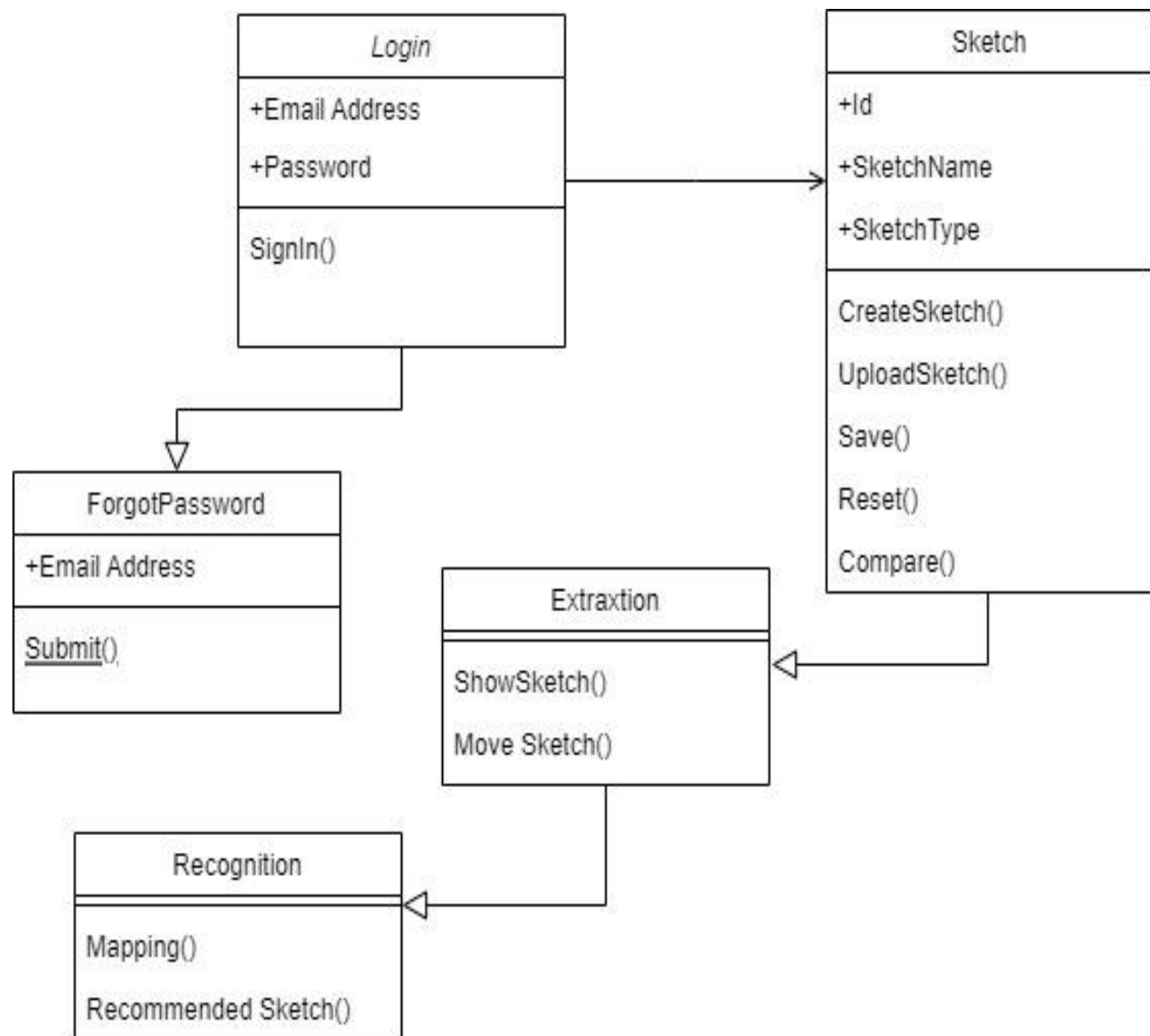


Fig 4.2.4: Class Diagram

IMPLEMENTATION DETAILS

5. IMPLEMENTATION DETAILS

5.1 Description of Tools:

Eclipse:

Eclipse is an integrated development environment used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment. Generally, Java 8 works best for most/all versions of Eclipse, including older ones. But if you want to use newer language/library features for Java development and need a newer version of Java for that purpose then newer versions should work with 2019-03 as well.

Eclipse is an open-source integrated development environment (IDE) that stands as a powerhouse in the realm of software development. It has earned its reputation as a versatile and reliable tool, and it is especially renowned as a Java IDE. Eclipse provides a comprehensive set of features for Java development, including code writing and editing, debugging, testing, and project management. It supports an array of programming languages, thanks to a vast ecosystem of plugins and extensions. Whether you're a Java developer, a C/C++ programmer, or working with languages like Python, PHP, or JavaScript, Eclipse offers tailored environments to meet your specific needs. The modularity of Eclipse ensures that you can customize your IDE by selecting and installing only the features that are relevant to your project. This flexible and modular approach keeps the IDE lightweight while allowing developers to add and configure plugins as required.

Eclipse is not only an integrated development environment (IDE) but also a platform for building and deploying applications. It supports a wide range of programming languages, including Java, C/C++, Python, PHP, and JavaScript, among others. Eclipse is highly extensible, thanks to its plugin architecture, which allows developers to enhance and customize the IDE according to their needs. The Eclipse Marketplace provides a vast repository of plugins and extensions developed by the community, covering various domains such as software development, database management, and web development. Additionally, Eclipse supports collaborative development through features like version control integration (e.g., Git), code review tools, and support for collaborative coding platforms like Code-together.

NetBeans:

NetBeans is a prominent open-source integrated development environment (IDE) renowned for its adaptability and comprehensive feature set, with a special focus on Java application development. It boasts a rich history, originating in the late 1990s, and has since become a versatile and highly configurable platform for developers. Here are some noteworthy aspects of NetBeans: Firstly, NetBeans is

open source, meaning it is freely available for developers worldwide, promoting collaboration and customization within its user community. This open approach has fostered an active and dynamic community that continually contributes to the IDE's development and improvement. NetBeans is a cross-platform IDE, designed to work seamlessly on various operating systems, ensuring that developers can choose their preferred platform without limitations. While NetBeans supports multiple programming languages, it has earned significant acclaim for its robust support for Java development. With a suite of tools and features tailored to Java developers, it is a favored choice for creating Java applications.

NetBeans offers a feature-rich environment for software development across different programming languages, with a primary focus on Java. Apart from Java development, NetBeans supports other languages such as PHP, HTML5, and C/C++. It provides a wide array of tools for building desktop, web, and mobile applications, including GUI builders, debugging tools, and project management features. NetBeans comes with built-in support for popular frameworks and technologies like Java EE, Spring, and Maven, streamlining the development process for Java developers. Its seamless integration with version control systems like Git and Subversion enables efficient collaboration among team members. Furthermore, NetBeans provides extensive documentation and tutorials to assist developers in getting started with the IDE and mastering its features.

5.2 Programming Language Description:

Java:

Java is a versatile, object-oriented, and widely used programming language known for its portability, simplicity, and reliability. Created by James Gosling and first released by Sun Microsystems in 1995, Java quickly gained popularity for its "write once, run anywhere" capability, meaning that Java programs can be executed on various platforms without modification. This platform independence is achieved through the use of the Java Virtual Machine (JVM), which translates Java bytecode into native code for a specific platform.

One of Java's key strengths is its robustness and strong memory management, which minimizes the risk of memory-related errors, such as buffer overflows. Java also features automatic garbage collection, simplifying memory management for developers. The language's rich standard library provides a broad range of pre-built classes and methods for various tasks, from basic I/O operations to complex data manipulation. Java is highly suitable for both standalone applications and web-based applications. With the introduction of Java Enterprise Edition (Java EE), it became a top choice for developing scalable, secure, and reliable server-side applications. Technologies like Servlets and JSP (Java Server Pages) have played a pivotal role in building dynamic web applications using Java.

Java is a high-level, object-oriented programming language known for its simplicity, portability, and scalability. It is widely used for developing a variety of applications, ranging from desktop software to web and mobile applications. Java's platform independence is achieved through its "write once, run anywhere" principle, where Java code can be compiled into bytecode that runs on any device with a Java Virtual Machine (JVM). This makes Java suitable for developing cross-platform applications that can run on diverse operating systems and hardware architectures. Java's extensive standard library provides a rich set of APIs for tasks such as input/output operations, networking, and database connectivity, facilitating rapid application development. Moreover, Java's strong type system and memory management features contribute to its reliability and robustness, making it a popular choice for building enterprise-grade applications.

JavaFx:

JavaFX is a versatile and powerful framework for building modern and visually appealing graphical user interfaces (GUIs) in Java applications. It has gained prominence as a successor to the aging Swing framework, offering a host of features that make it well-suited for creating interactive desktop and web applications. JavaFX's key strength lies in its ability to provide a cross-platform solution, making it possible to write GUI code once and run it on various operating systems, including Windows, macOS, and Linux. This versatility ensures that developers can reach a broad and diverse audience with their Java applications.

One of JavaFX's standout features is its capacity for rendering rich graphics and media content. It is particularly well-suited for applications that require advanced visualization, such as data dashboards, games, or simulations. JavaFX supports 2D and 3D graphics, opening up new possibilities for developers to create immersive and visually stunning user interfaces. In addition, it provides multimedia support for audio and video playback, adding another layer of interactivity to applications.

Java is a high-level, object-oriented programming language known for its simplicity, portability, and scalability. It is widely used for developing a variety of applications, ranging from desktop software to web and mobile applications. Java's platform independence is achieved through its "write once, run anywhere" principle, where Java code can be compiled into bytecode that runs on any device with a Java Virtual Machine (JVM). This makes Java suitable for developing cross-platform applications that can run on diverse operating systems and hardware architectures. Java's extensive standard library provides a rich set of APIs for tasks such as input/output operations, networking, and database connectivity, facilitating rapid application development. Moreover, Java's strong type system and memory management features contribute to its reliability and robustness, making it a popular choice for building enterprise-grade applications.

5.3 Algorithm Details:

5.3.1 Amazon Rekognition:

Amazon Rekognition is a cloud-based image and video analysis service that makes it easy to add advanced computer vision capabilities to your applications. The service is powered by proven deep learning technology and it requires no machine learning expertise to use.

Amazon Rekognition is a service offered by Amazon Web Services (AWS). It's a deep-learning-based service that makes it easy to add video and image analysis to applications. Amazon Rekognition is a machine learning product by AWS that helps in adding image and video analysis to applications. It uses deep learning to analyze images and videos.

Amazon Rekognition works with two KPI sets – Amazon Rekognition Image and Amazon Rekognition Video. These KPIs help in image and video analysis respectively. These KPIs analyze images and videos to generate insights that you can use in your apps. When your customer uploads a photo, Amazon Rekognition can detect objects or faces in that image. Your app can store this information and help develop a photo collection for your customers. They could use this to query any image search they want.

Key capabilities-

Object, Scene, and Concept Detection - Detects and classify objects, scenes, concepts, and celebrities in images.

Text Detection - Detect and recognize printed and handwritten text in images in a variety of languages.

Unsafe Content - Detect and filter explicit, inappropriate, and violent content and images. Detects granular unsafe content labels.

Celebrity recognition - Recognize tens of thousands of celebrities in your images across different categories, such as politicians, athletes, actors, and musicians.

Facial Analysis - Detect, analyze, and compare faces, along with facial attributes, such as gender, age, and emotions. Use cases may include user verification, cataloging, people counting, and public safety.

Custom Labels - Build custom classifiers to detect objects specific to your use case, such as logos, products, characters.

Image Properties - Analyze image properties like quality, color, sharpness, contrast.

Input: Access to sources of images (e.g., databases, websites).

Output: A collection of images stored in a suitable format (e.g., JPEG, PNG)

Step 1: Collect images

Step 2: Decide on classes

Step 3: Create a project

Step 4: Create training and test datasets

Step 5: Add labels to the project

Step 6: Assign image-level labels to training and test datasets

Step 7: Train your model

Step 8: Start your model

Step 9: Analyze an image with your model

Step 10: Stop your model

5.3 Output of Implementation (In the form of Screen Shots):



Fig 5.3.1: Splash Screen for our Standalone desktop application

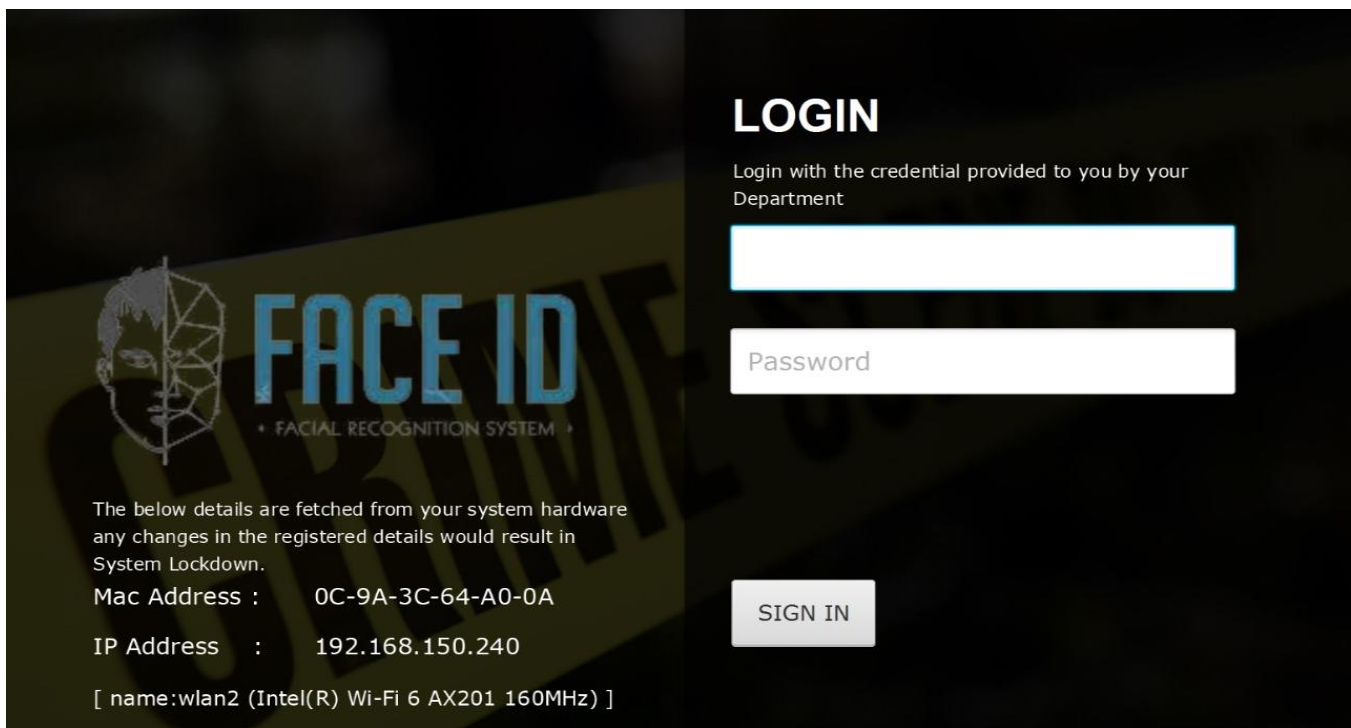


Fig 5.3.2: Login Screen for our Standalone desktop application

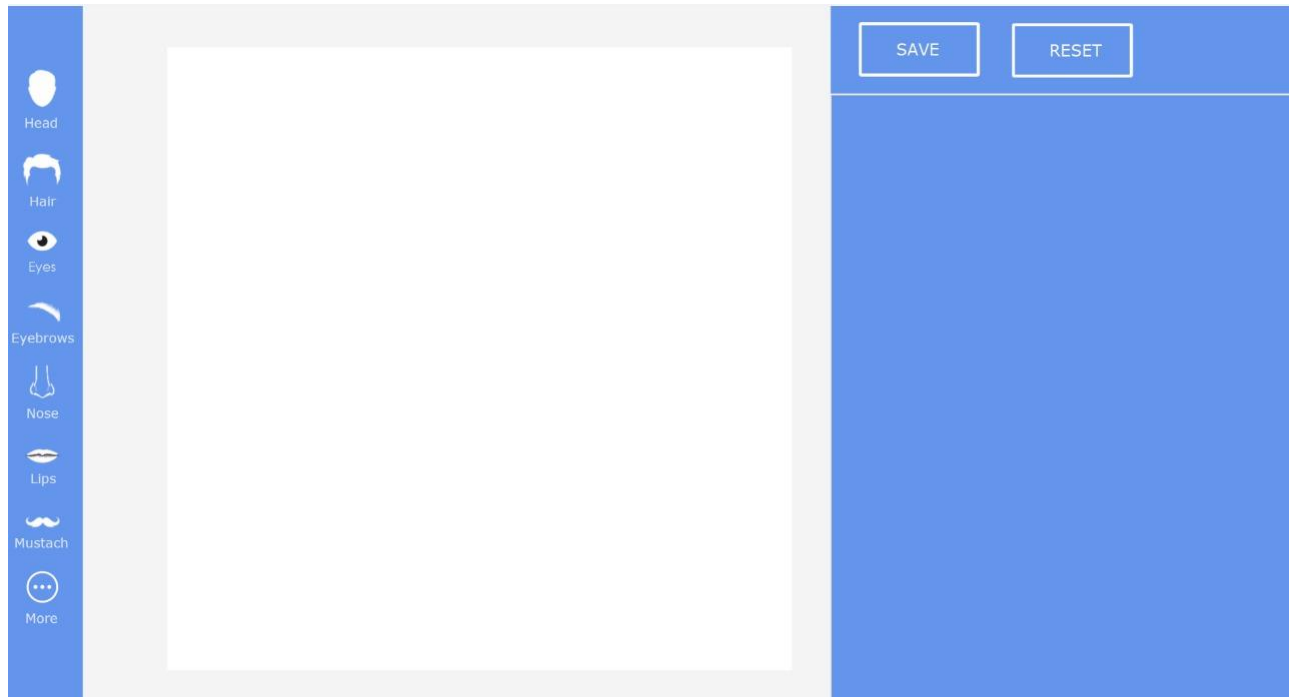


Fig 5.3.3: Dashboard to create a facial sketch

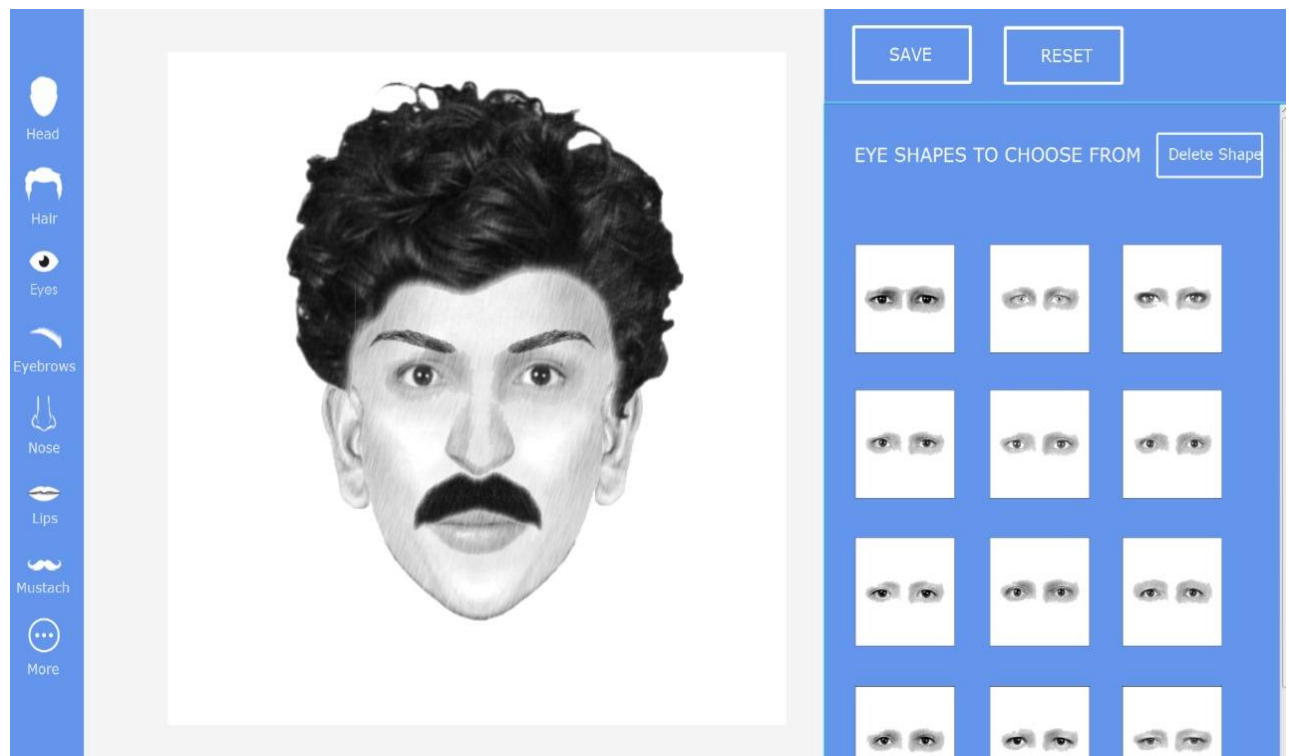


Fig 5.3.4: Complete face sketch in the Dashboard

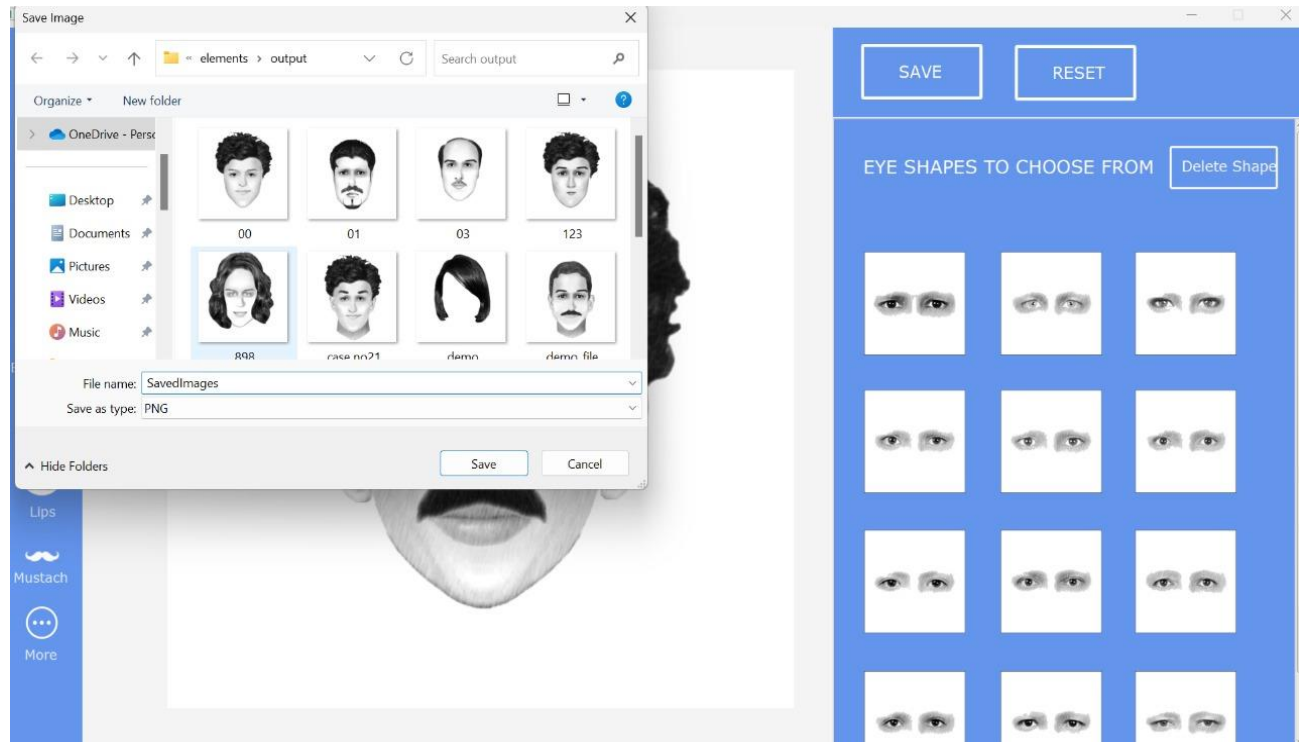


Fig 5.3.5: Saving face sketch in System

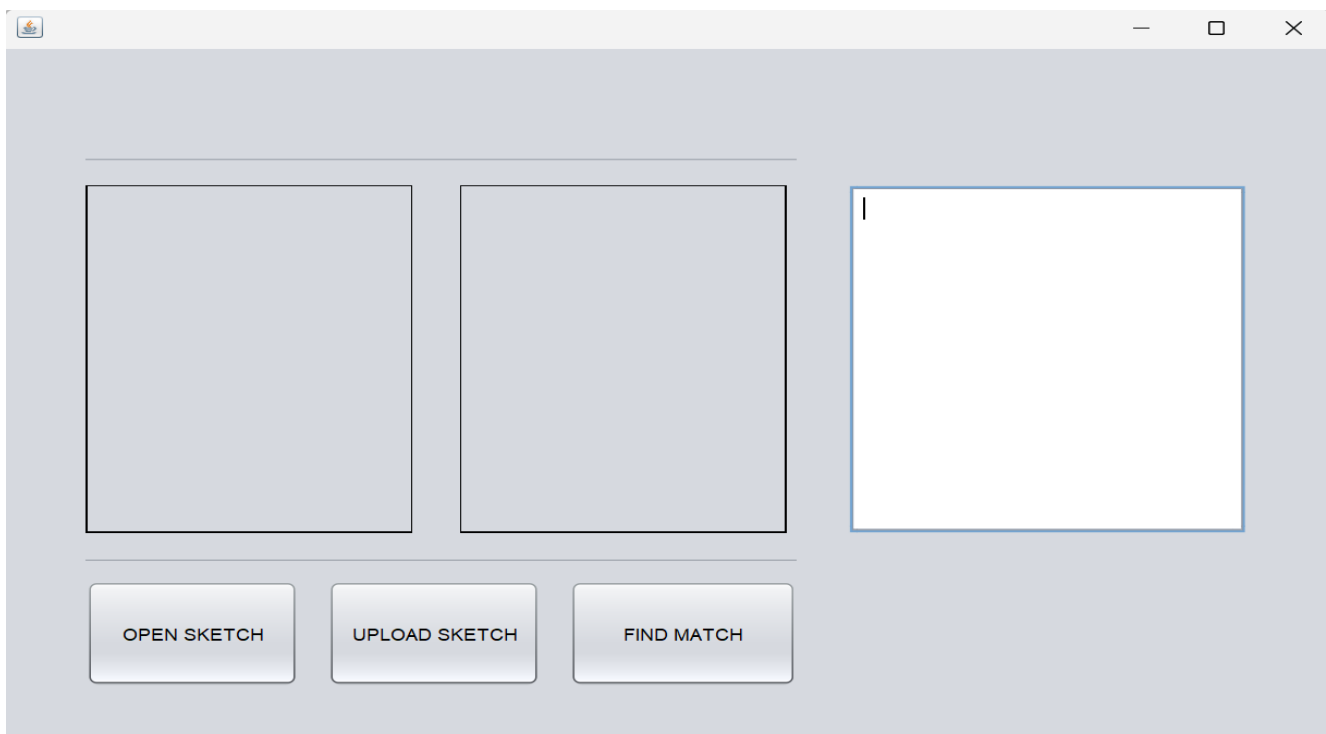


Fig 5.3.6: Dashboard for Recognition

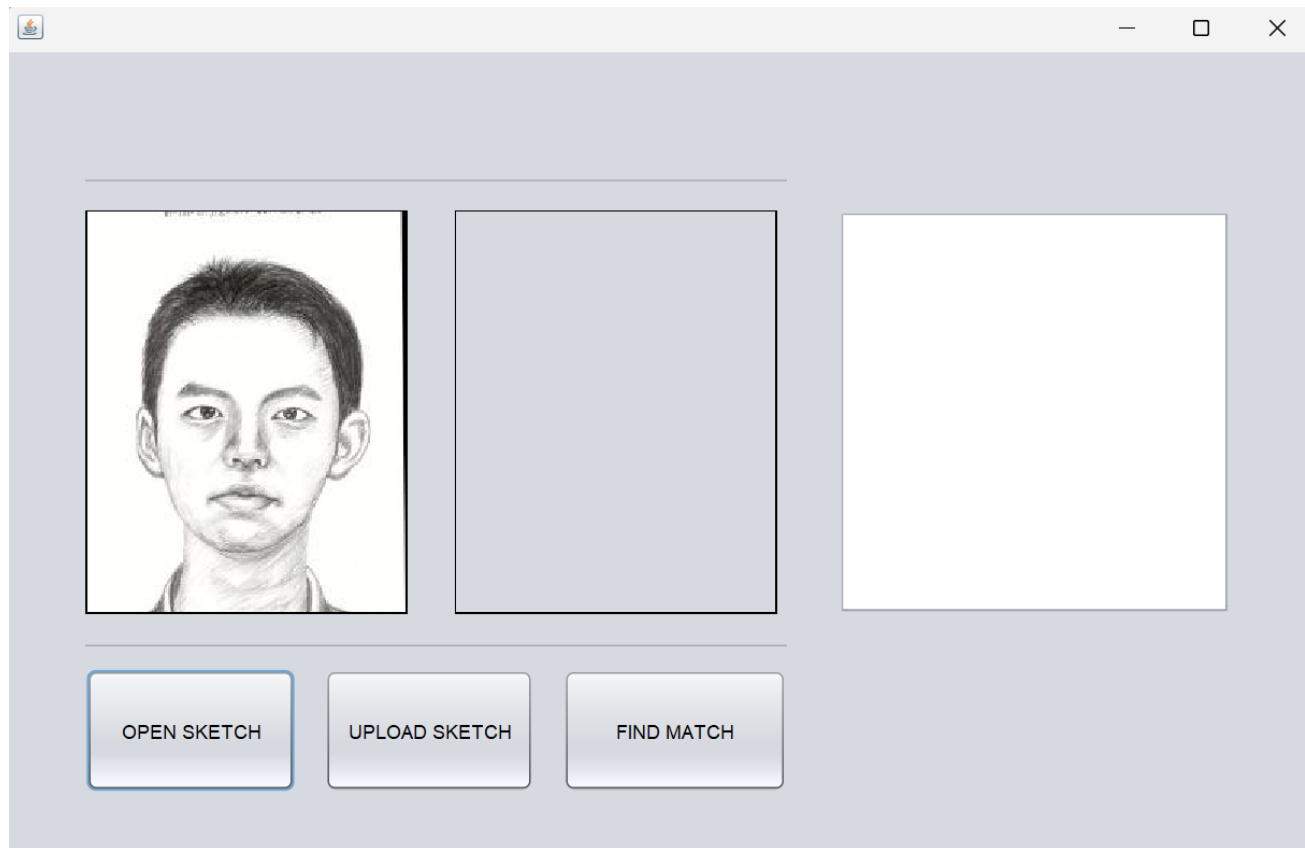


Fig 5.3.7: Opened face sketch from the system

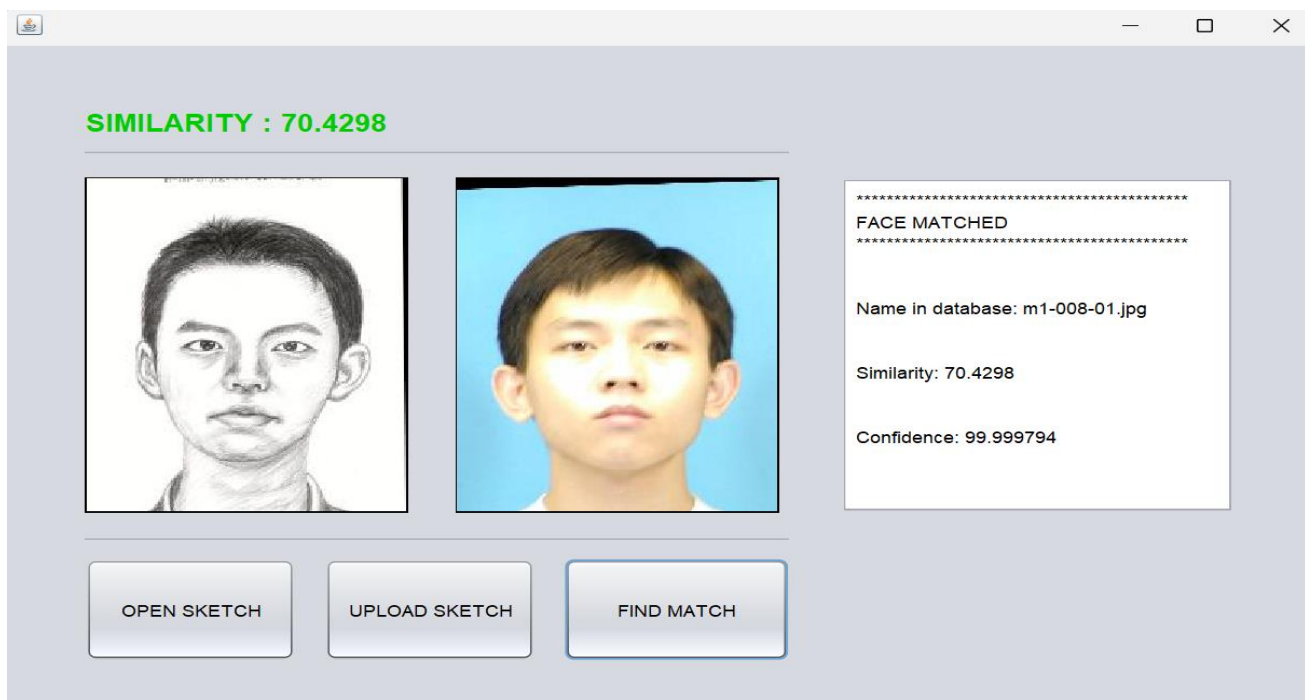


Fig 5.3.8: Face sketch matched to the database record

TESTING

6. TESTING

6.1. Types of Testing:

6.1.1. Unit Testing:

Unit testing involves testing individual components or modules of the software to ensure they function correctly in isolation, typically conducted by developers before integrating them into the larger system.

6.1.2. Integration Testing:

Integration testing verifies the interactions and interfaces between different components/modules of the software to ensure they work together as expected, identifying any integration issues or inconsistencies.

6.1.3. System Testing:

System testing evaluates the behavior of the entire software system as a whole, testing end-to-end functionality to ensure it meets specified requirements and performs as expected in a real-world environment.

6.1.4. User Acceptance Testing (UAT):

UAT involves testing the software from the end user's perspective to ensure it meets their requirements and expectations, typically conducted by stakeholders or end users before deployment.

6.1.5. Performance Testing:

Performance testing evaluates the responsiveness, scalability, and reliability of the software under various load conditions, ensuring it can handle expected levels of usage without degradation in performance.

6.1.6. Security Testing:

Security testing assesses the software's resilience against potential security threats and vulnerabilities, ensuring that sensitive data is protected, and access controls are enforced to prevent unauthorized access or malicious attacks.

6.2 Test Cases:

6.2.1. Unit Testing:

Test Case 1: Facial Feature Extraction

Input: Sample images containing faces

Expected Output: Correct identification of key facial landmarks (e.g., eyes, nose, mouth)

Assertion: Assert that the detected facial features match the ground truth coordinates within an acceptable margin of error.

Test Case 2: Database Integration

Input: Known suspect information (e.g., name, description)

Expected Output: Accurate retrieval of suspect records from the database

Assertion: Assert that the retrieved suspect information matches the expected data with high accuracy.

6.2.2. Integration Testing:

Test Case 1: User Interface Integration

Input: User interactions with the GUI (e.g., inputting suspect description, initiating facial sketch generation)

Expected Output: Smooth interaction between the GUI and backend processes

Assertion: Assert that user actions trigger the appropriate backend functionalities and that feedback is displayed correctly on the GUI.

Test Case 2: Cloud Integration

Input: Database query requests sent to the cloud server

Expected Output: Retrieval of relevant suspect information from the cloud database

Assertion: Assert that database queries are successfully executed, and responses are received within a reasonable timeframe.

6.2.3. System Testing:

Test Case 1: End-to-End Workflow

Input: Description or reference image of a suspect

Expected Output: Generation of a facial sketch and identification of potential suspects from the database

Assertion: Assert that the entire process, from sketch generation to suspect identification, executes smoothly without errors.

6.2.4. User Acceptance Testing (UAT):

Test Case 1: Sketch Accuracy

Input: User-generated facial sketches

Expected Output: Accurate depiction of facial features for identification purposes

Assertion: Assert that users can correctly identify known suspects based on the generated sketches.

Test Case 2: Ease of Use

Input: User interactions with the application interface

Expected Output: Intuitive and user-friendly experience

Assertion: Assert that users can navigate the application easily and perform tasks without encountering confusion or frustration.

6.2.5. Performance Testing:

Test Case 1: Response Time

Input: Sequential requests for facial sketch generation and suspect identification

Expected Output: Minimal delay in processing requests

Assertion: Assert that the system responds promptly to user actions and queries, maintaining low latency throughout.

6.2.6. Security Testing:

Test Case 1: Authentication

Input: Unauthorized access attempts to sensitive functionalities

Expected Output: Rejection of unauthorized access attempts

Assertion: Assert that only authenticated users with appropriate privileges can access restricted functionalities.

RESULT AND CONCLUSION

7. RESULT AND CONCLUSION

7.1 CONCLUSION

In conclusion, forensic sketching is an important tool in criminal investigations that can help identify suspects based on eyewitness accounts. The process of creating a forensic sketch involves an initial interview with the witness or victim, followed by using various techniques to create a sketch that accurately represents the suspect's appearance. However, forensic sketching can also pose challenges, such as difficulties in recalling details of the suspect's appearance and subjective interpretation of the sketch by the artist.

7.2 FUTURE SCOPE

Improved Accuracy: Future enhancements can focus on improving the accuracy of facial recognition algorithms. This could involve better handling of low-quality images, addressing bias in recognition systems, and reducing false positives and negatives.

Privacy and Ethics: As concerns around privacy and ethics grow, future research can explore ways to make facial recognition technology more transparent, secure, and respectful of individual rights.

Mobile Integration: Extending the use of face sketch creation and recognition to mobile applications can further enhance its practicality and real-world applications, such as personal security and mobile banking.

APPENDICES

APPENDIX A

PLAGIARISM REPORT OF PROJECT REPORT



Fig : Plagiarism Report (Research Paper)

APPENDIX B

1. Paper Title:

Face Sketch Creation and Recognition.

2. Name of the conference/journal where the paper was submitted:

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

3. Paper Status:

Submitted

4. Review Comments by Reviewer:

No

5. Corrective actions if any:

No







ISSN No. : 2321-9653

IJRASET

International Journal for Research in Applied Science & Engineering Technology
IJRASET is indexed with Crossref for DOI-DOI : 10.22214
Website : www.ijraset.com, E-mail : ijraset@gmail.com

Certificate

It is here by certified that the paper ID : IJRASET59671, entitled
Face Sketch Creation and Recognition
by
Mayur Shinde

after review is found suitable and has been published in
Volume 12, Issue IV, April 2024
in

International Journal for Research in Applied Science &
Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Py 
Editor in Chief, IJRASET



ISRA Journal Impact
Factor: 7.429



45.98
INDEX COPERNICUS



THOMSON REUTERS
Researcher ID: 9-8021-2016



10.22214/IJRASET
doi crossref



TOGETHER WE REACH THE GOAL
SJIF 7.429



ISSN No. : 2321-9653

IJRASET

International Journal for Research in Applied Science & Engineering Technology
IJRASET is indexed with Crossref for DOI-DOI : 10.22214
Website : www.ijraset.com, E-mail : ijraset@gmail.com

Certificate

It is here by certified that the paper ID : IJRASET59671, entitled
Face Sketch Creation and Recognition
by
Prof. Anoop Kushwaha

after review is found suitable and has been published in
Volume 12, Issue IV, April 2024
in

International Journal for Research in Applied Science &
Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Py 
Editor in Chief, IJRASET



ISRA Journal Impact
Factor: 7.429



45.98
INDEX COPERNICUS



THOMSON REUTERS
Researcher ID: 9-8021-2016



10.22214/IJRASET
doi crossref



TOGETHER WE REACH THE GOAL
SJIF 7.429

REFERENCES

REFERENCES

- [1] ARGALL, B. D., CHERNOVA, S., VELOSO, M., AND BROWNING, B. A survey of robot learning from demonstration. *Robotics and autonomous systems* 57, 5 (2009), 469–483.
- [2] S. Arya and D. M. Mount, “Algorithms for fast vector quantization” in *Proc. DCC 93: Data Compression Conf.*, J. A. Storer and M. Cohn, Eds. Piscataway, NJ: IEEE Press, pp. 381–390.
- [3] Klare, Brendan, and Anil K. Jain. "Sketch-to-images matching: a feature-based approach." *Biometric technology for human identification VII*. Vol. 7667. International Society for Optics and Photonics, 2010.
- [4] P. C. Yuen and C. H. Man, "Human Face Image Searching System Using Sketches," in *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol.37, no. 4, pp. 493-504, July 2007, doi: 10.1109/TSMCA.2007.897588.
- [5] X. Tang and X. Wang, “Face sketch recognition”, *IEEE Trans. Circuits and Systems for Video Technology*, vol. 14, no. 1, pp.50-57,2004.
- [6] Hamed Kiani Galoogahi and Terence Sim, “Face Sketch Recognition By Local Radon Binary Pattern: LRBP”, 19th IEEE International Conference on Image Processing, 2012.
- [7] Charlie Frowd, Anna Petkovic, Kamran Nawaz and Yasmeen Bashir, “Automating the Processes Involved in Facial Composite Production and Identification” *Symposium on Bio-inspired Learning and Intelligent Systems for Security*,2009.
- [8] H. Han, B. Klare, K. Bonnen, and A. Jain, “Matching composite sketches to face photos: A component-based approach,” *IEEE Trans. on Information Forensics and Security*, vol. 8, pp. 191–204, January 2013.