**Project Title: OTP-Based Fingerprint Analysis System with PCA**

**Project Final Report:** Artificial Intelligence Lab

**Instructor:** Sohail Ahmed Malik

**Section:** BS-AI-4A

**Group Members**

Member 1: [Abdul-Rehman-Nazeer]

Member 2: [Javeria]

**Abstract**

The project involves an OTP-based fingerprint analysis system designed to verify user identity before displaying fingerprint images. The system utilizes **CLAHE** (Contrast Limited Adaptive Histogram Equalization) for image preprocessing, **Gaussian** **blurring**, and **PCA** (Principal Component Analysis) for feature extraction. It employs various tools, including **OpenCV**, **matplotlib** for visualization, and **Tkinter** for GUI-based OTP verification. The project consists of multiple components, including OTP verification, **image extraction** from ZIP files, **image preprocessing**, and **data visualization.**

**Acknowledgement**

We acknowledge the guidance and support provided by our instructor, Sohail Ahmed Malik, throughout the course. We also appreciate the assistance from our fellow students.

**Introduction**

**Background**: AI play a crucial role in developing complex systems, including biometric verification and image processing. This project explores the integration of OTP-based security with fingerprint analysis. OTP (One-Time Password) serves as an additional security measure to ensure that only authorized users can access sensitive data.

**Purpose** **and** **Objectives**: The primary objective is to create a system that combines OTP verification with fingerprint analysis, using PCA for feature extraction. The goal is to ensure that only users with the correct OTP can access the preprocessed fingerprint images, providing a secure method for data handling.

**Project Overview**

**Main Components and Functionality:** The system has several components:

1. **OTP Verification:** A GUI-based OTP verification system using Tkinter. An OTP is generated and users are prompted to enter it to access the images.
2. **Image Extraction:** Fingerprint images are extracted from a ZIP archive and loaded for processing.
3. **Image Preprocessing:** The images are processed with CLAHE and Gaussian blur to improve quality.
4. **PCA-Based Feature Extraction:** The project uses PCA to extract features from fingerprint images.
5. **Visualization:** The processed images are displayed in a grid layout with a matplotlib-based visualization.

**Block Diagram/System Overview:**

OTP Verification

OTP Input

Image Display

Image Extractor

**Design Methodology**

**Design Process:** The design process involved several steps:

1. **Requirement Analysis:** The system's security and functionality requirements were focusing on OTP integration and fingerprint analysis.
2. **Hardware Selection:** Tools such as OpenCV for image processing, matplotlib for visualization, and Tkinter for GUI were chosen no hardware was needed.
3. **Software Development**: Code was developed to handle OTP generation and verification, image extraction, preprocessing, and visualization.
4. **Testing and Validation:** The system was tested with various OTP values and fingerprint images to ensure functionality.

**Design Flow:** The design flow included:

* Defining OTP-based security requirements.
* Developing the GUI-based OTP verification system.
* Extracting and preprocessing images from a ZIP archive.
* Implementing PCA for feature extraction.
* Visualizing processed images with matplotlib.

**Hardware Description**

**Hardware Components:** No specific hardware components are used, as the project relies on software-based tools and libraries. Standard development environments and systems with sufficient processing power for image handling are assumed.

**Software Description**

**Software Tools:** The software components include:

* **OpenCV**: Used for image processing, including CLAHE and Gaussian blur.
* **Matplotlib**: Used for visualizing images in a grid layout.
* **Tkinter**: Used to create the GUI for OTP verification.
* **Scikit**-**learn**: Utilized for PCA-based feature extraction.

**Overview of Programming Languages:** The project primarily uses Python for development, with additional libraries for specific tasks.

**Results and Analysis**

**Experimental Results:** The system was tested with a variety of OTP inputs and fingerprint images. The OTP verification successfully restricted access until the correct OTP was entered. The processed fingerprint images were displayed in a grid format, showing improvements from preprocessing techniques.

**Analysis of Results:** The OTP verification system performed as expected, allowing access only with the correct OTP. Image preprocessing techniques like CLAHE and Gaussian blur improved image quality. PCA-based feature extraction allowed for further analysis and visualization.

**Discussion**

**Interpretation of Results:** The project demonstrated the effective integration of OTP-based security with fingerprint analysis. The OTP system worked as a secure access control mechanism, while preprocessing and PCA provided improved image quality and feature extraction.

**Comparison with Theoretical Expectations:** The results aligned with expectations for OTP-based security and image processing. The project successfully combined OTP verification with fingerprint analysis.

**Challenges and Solutions:** Challenges included integrating the GUI with OTP verification and handling image extraction and processing. Solutions involved using Python-based libraries for seamless integration and handling exceptions for incorrect OTP inputs.

**Conclusion**

**Summary of Key Findings:** The OTP-based fingerprint analysis system successfully combined security and image processing. The use of PCA and preprocessing techniques improved image quality and allowed for feature extraction and visualization.

**Reflection on Project Successes and Areas for Improvement:** The project succeeded in combining OTP verification with fingerprint analysis. Areas for improvement include expanding security features and enhancing image preprocessing techniques for better results.

**Future Work**

**Suggestions for Future Enhancements:** Future work could involve integrating additional security measures, improving the OTP verification process, and expanding the range of image processing techniques. Exploring machine learning-based feature extraction and analysis could also be a potential area for enhancement.

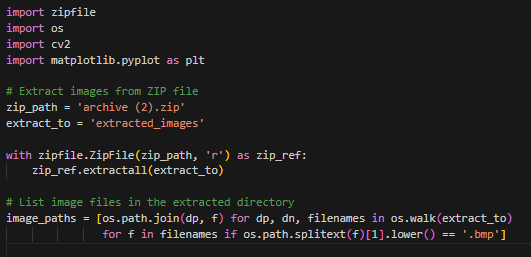
**Ideas for Expanding the Project's Functionality:** Suggestions include adding more complex biometric analysis, integrating machine learning, and exploring multi-factor authentication for increased security.

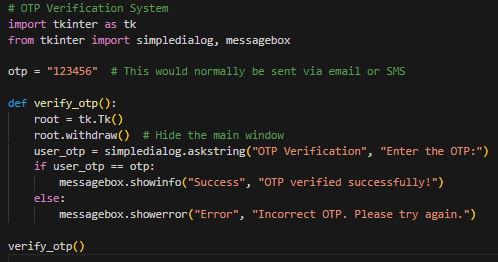
**References**

References include libraries such as OpenCV, Matplotlib, Tkinter, and Scikit-learn.

**Appendices**

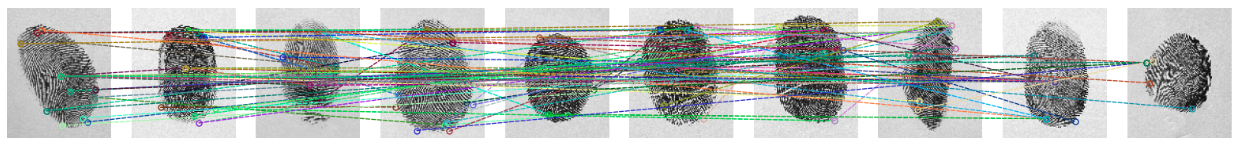
**Appendix A: Code Snippets**

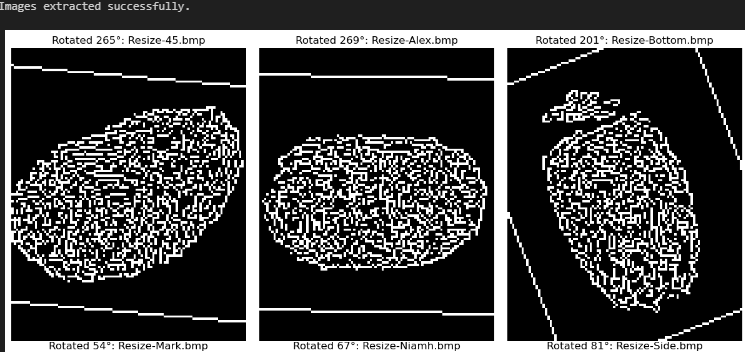




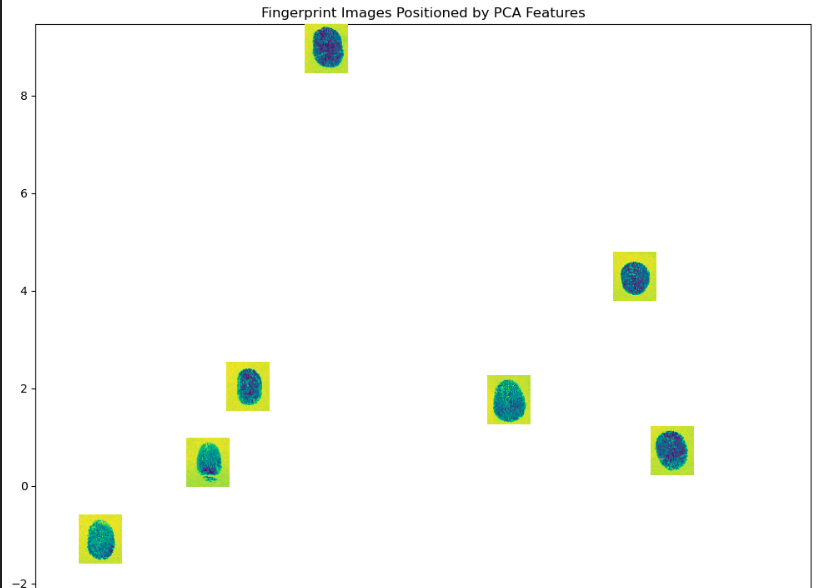
**Appendix B: Additional Figures**

**Matching Features and Storing key points.**

****

**Image Rotation**.  
  


**Fingerprint Image Positions By PCA Features.**



**References**

**Tkinter Documentation:** A guide for using Tkinter to create graphical user interfaces in Python. Available at: [Tkinter Docs](https://docs.python.org/3/library/tkinter.html)

**Scikit-learn Documentation:** Detailed information on using Scikit-learn for machine learning and PCA-based feature extraction. Available at: [Sci-kit Learn Docs](https://scikit-learn.org/0.21/documentation.html)