BioGuard
Final Defense Report

Angel Van den akker and Maria Khvatova

June 4, 2024

Contents

1 Project introduction

In today's digital landscape, the need for secure and reliable authentication methods is more critical than ever. As traditional passwords become increasingly vulnerable to breaches, biometric authentication has emerged as a superior alternative, offering enhanced security through the use of unique biological characteristics. This report documents the final phase of our project, focusing on the development and integration of a comprehensive biometric authentication system.

The goal of this project is to create an advanced security solution that leverages fingeprints for user verification. Building on our previous work with fingerprint recognition, we have now created a real application. approach aims to provide a higher level of security and accuracy in user authentication.

Fingerprints, with their unique ridge and valley patterns, have been a reliable form of identification for decades. The distinctiveness of these patterns ensures a high degree of accuracy in verifying an individual's identity. Our project harnesses this reliability, aiming to deliver a biometric authentication system that is both secure and user-friendly.

In this report, we will discuss the methodologies employed in the development of our biometric system, including the design and implementation phases. We will also address the technical challenges we faced during the integration process, such as ensuring compatibility with existing hardware and maintaining user privacy.
.

This final report aims to provide a comprehensive overview of our project's achievements, highlighting the practical implications and future potential of biometric systems in enhancing digital security.

2 Introduction of the Members

Angel: With a strong passion for both sports and coding, I balance an active lifestyle with rigorous academic pursuits. This unique combination fuels a disciplined approach to studies and a dynamic engagement in various projects. Outside the classroom, Angel is an active member of sports clubs, finding time to stay physically active and fostering a healthy balance between mental and physical well-being. I generally have a collaborative spirit and always a keen enthusiasm for learning, I contribute positively to group projects and am always eager to take on new challenges.

Maria: Being at the end of the Prepa program, I was eager to take on a fascinating project. Having had several ideas about the topic of this project, e.g. developing Gesture Recognition System, a brainwave-controlled Application or a Rust-based Computational Photography Tool, I decided to propose this specific topic of Biometric Authentification. Being so captivating, vast and variable, this project seems like a perfect opportunity to gain valuable knowledge on many fields of computing.

3 Background and Related Work

3.1 Biometric Authentication

Biometric authentication is a security process that verifies an individual's identity using unique biological characteristics. Common biometric authentication methods include fingerprint recognition, facial recognition, iris scanning, voice recognition, and even behavioral patterns like typing rhythms. Unlike traditional authentication methods that rely on something the user knows (passwords, PINs) or possesses (security tokens,...), biometric authentication leverages the inherent and unique traits of individuals.

3.2 Types of Biometric Authentication

- **Fingerprint Recognition**: Uses the unique patterns of ridges and valleys on an individual's fingertips.
- **Facial Recognition**: Analyzes facial features such as the distance between the eyes, nose width, and jawline.
- **Iris and Retina Scanning**: Examines the unique patterns in the colored part of the eye (iris) or the pattern of blood vessels in the retina.
- **Voice Recognition**: Identifies individuals based on their unique vocal characteristics.
- **Behavioral Biometrics**: Includes keystroke dynamics, and other behavior-based identification methods.

3.3 Advantages of Biometric Authentication Over Traditional Methods

3.3.1 Enhanced Security

- **Uniqueness**: Biometric traits are unique to each individual, making it difficult for unauthorized users to duplicate or steal.
- **Non-replicability**: Unlike passwords or PINs, biometric data cannot be easily shared, lost, or forgotten, reducing the risk of unauthorized access.

3.3.2 Convenience and User Experience

- **Ease of Use**: Users do not need to remember complex passwords or carry additional hardware like tokens or cards.
- **Speed**: Biometric authentication can be performed quickly, often within seconds, facilitating faster access to systems and devices.

3.3.3 Reduced Administrative Overhead

- **Lower Maintenance**: Eliminates the need for password resets and related support, reducing the burden on IT departments.
- **Cost-Effective**: Over time, biometric systems can be more cost-effective by reducing the expenses associated with password management and security breaches.

3.3.4 Improved Compliance and Accountability

- **Audit Trails**: Biometric systems can provide precise records of who accessed what and when, enhancing accountability and aiding in regulatory compliance.
- **Non-Repudiation**: Because biometric data is unique and directly tied to an individual, it provides a robust method for ensuring that actions cannot be denied by the person who performed them.

3.3.5 Scalability and Flexibility

- **Integration**: Biometric systems can be integrated into various platforms, from mobile devices to large enterprise systems, providing versatile applications across different use cases.
- **Adaptability**: Biometric technologies are continually evolving, with new methods and improvements being developed to enhance security and user experience further.

3.4 Fingerprint Recognition

The use of fingerprints for identification has a long and fascinating history:

Early Use

- **Ancient Civilizations**: Fingerprints were used in ancient Babylon and China for business transactions and to authenticate documents. Clay tablets from Babylon dating back to 2000 BCE bear fingerprints.
- **14th Century Persia**: A Persian book from the 14th century notes the uniqueness of fingerprints as a means of identifying individuals.

Modern Development

- **19th Century**: The modern study of fingerprint recognition began in the 19th century. In 1892, Sir Francis Galton published "Fingerprints" establishing the individuality and permanence of fingerprints.
- **Henry Classification System**: Sir Edward Henry developed a system for classifying fingerprints, which was adopted by law enforcement agencies in the UK and other countries.
- **Early 20th Century**: Fingerprint recognition became a standard practice in forensic science for identifying criminals and verifying identities.

3.4.1 Technical Aspects of Fingerprint Recognition

Fingerprint recognition involves several technical processes to capture, store, and match fingerprints. These processes include:

Fingerprint Acquisition

- **Optical Sensors**: Use light to capture fingerprint images. When a finger is placed on a glass surface, the ridges and valleys reflect light differently, creating a visible pattern.
- **Capacitive Sensors**: Measure the electrical charge between the finger and the sensor. The ridges and valleys create different capacitance values, forming a fingerprint image.
- **Ultrasonic Sensors**: Use high-frequency sound waves to capture the detailed texture of the fingerprint, providing more accurate data even if the finger is dirty or wet.

Image Processing

- **Enhancement**: Fingerprint images are enhanced to improve clarity and contrast. Techniques such as histogram equalization and filtering are used.
- **Segmentation**: The fingerprint area is segmented from the background to focus on the relevant data.
- **Minutiae Extraction**: Key features (minutiae) such as ridge endings and bifurcations are extracted from the enhanced image. These minutiae points are critical for matching fingerprints.

Matching Algorithms

- **Pattern-Based Matching**: Compares the overall patterns and flow of ridges in the fingerprints.
- **Minutiae-Based Matching**: Compares the extracted minutiae points between the stored and input fingerprint images.
- **Hybrid Matching**: Combines both pattern-based and minutiae-based techniques to improve accuracy and reliability.

Template Storage and Security

- **Template Creation**: A fingerprint template is created from the extracted minutiae points. This template is a digital representation of the fingerprint.
- **Storage**: Templates are stored in secure databases. Encryption techniques are employed to protect the templates from unauthorized access.
- **Security Measures**: Advanced security measures, such as liveness detection, are used to prevent spoofing attacks by ensuring that the fingerprint comes from a live finger.

4 Task Distribution

Due to two members leaving our group, we redistributed the task between the two remaining members in the following way:

Task Distribution		
Task	Maria	Angel
Data Aquisition		Main
Minutiae extraction	Main	
Minutiae matching		Main
Password manager		Main
Website	Main	
UI		Main

5 System Design and Implementation

5.1 System Architecture

The general workflow of the application is as follows:

1. The application starts and establishes a connection to the SQLite database.
2. The necessary tables are created if they do not exist.
3. When a user attempts to log in, their fingerprint image is read and compared against the stored fingerprint image in the database.
4. If the user does not exist, a new user is created and their fingerprint image is stored.
5. If the user exists and the fingerprint matches, their credentials are retrieved and displayed.

5.1.1 Credential Model

```
#[derive(Debug)]
pub struct Credential {
    pub id: i64,
    pub user_id: i64,
    pub site: String,
    pub site_username: String,
    pub site_password: String,
}
```

5.2 Database Integration

The `database.rs` file manages the connection to the SQLite database and provides functions to interact with the database. The integration of the database into the system involves the following steps:

- **Establishing Connection**: The `establish_connection` function establishes a connection to the SQLite database using the SQLx library. This function uses a hardcoded database URL for simplicity but can be adapted to use environment variables.

```
pub async fn establish_connection() -> SqlitePool {
    let database_url = "sqlite://users.db";
    SqlitePool::connect(&database_url).await.unwrap()
}
```

- **Creating Tables**: The `create_tables` function creates the necessary tables in the database if they do not exist. This includes tables for users and their credentials.
- **Saving Data**: Functions such as `save_user` and `save_credentials` are used to insert data into the database. These functions accept parameters and bind them to the SQL queries to securely insert the data.
- **Retrieving Data**: Functions like `get_user` and `get_credentials` are used to fetch data from the database. These functions execute SQL queries and map the results to the respective data structures.

5.3 Models

The `models.rs` file defines the data structures used in the application.

`Credential`: Represents a user's credential with an ID, user ID, site, site username, and site password.

5.4 Fingerprint Acquisition: Final Review

As we conclude our project, it is crucial to reflect on the progress made in fingerprint acquisition, a fundamental component that has underpinned our work from the beginning. This section will provide a final review of the challenges addressed, the progress achieved, and the future prospects for this critical aspect of our project.

5.4.1 Achievements and Improvements

During the final phase of our project, significant strides have been made in refining the fingerprint acquisition process. Key achievements include:

- **Enhanced Sensor Integration:** We successfully improved the integration of the fingerprint sensor with our C++ codebase, resulting in more reliable and consistent data capture.
- **Feature Extraction Accuracy:** Enhancements in feature extraction techniques have increased the accuracy and reliability of the extracted fingerprint features, crucial for effective biometric matching.

These improvements were built on the foundational work done in the earlier phases, demonstrating our commitment to continuous enhancement.

5.4.2 Addressing Initial Challenges

The challenges identified in the initial and second phases have been systematically addressed throughout the final phase:

- **C++ Adaptation:** Our proficiency in C++ has grown, enabling us to write more efficient and robust code for fingerprint processing.

- **Windows Biometric Framework:** We have deepened our understanding and utilization of the Windows Biometric Framework, ensuring compatibility and leveraging its features for better performance.
- **Library Compatibility:** Continuous updates and testing have ensured our libraries are current and fully compatible with the latest versions, preventing potential integration issues.

By tackling these challenges head-on, we have laid a robust groundwork for the project's success.

5.4.3 Integration and Testing

In this final phase, the focus was on integrating the fingerprint acquisition module with the overall system and conducting extensive testing:

- **System Integration:** The fingerprint acquisition module has been seamlessly integrated with the database and authentication systems, ensuring smooth data flow and operational coherence.
- **Comprehensive Testing:** Rigorous testing was conducted to verify the accuracy, reliability, and performance of the fingerprint acquisition process. Various scenarios were simulated to ensure robustness.
- **User Feedback:** Preliminary user testing provided valuable feedback, leading to minor adjustments and further refinements in the acquisition process.

This phase has confirmed the operational readiness of our fingerprint acquisition system.

5.4.4 Future Prospects

Looking forward, several potential enhancements could be pursued:

- **Advanced Algorithms:** Implementing advanced machine learning algorithms for feature extraction could further enhance accuracy and reliability.
- **Sensor Technology:** Exploring and integrating newer sensor technologies may provide higher resolution and more detailed fingerprint images.

These future considerations will help in keeping the project relevant and up-to-date with technological advancements.

5.4.5 Conclusion

In conclusion, the third phase of our project has been marked by significant progress in the fingerprint acquisition process. The initial challenges were effectively addressed, leading to substantial improvements and a fully integrated system ready for deployment. The groundwork laid in the earlier phases has proven invaluable, providing a solid foundation for our advancements. Moving forward, continuous refinement and embracing new technologies will ensure the sustained success and relevance of our fingerprint acquisition system.

5.5 Minutia Extraction

Minutia extraction is a critical step in fingerprint analysis - the backbone of the entire project, involving the identification and extraction of minutiae points from fingerprint images. This process transforms the raw image data into a structured format suitable for further analysis and comparison. To better understand the intricasies of a human fingerprint, it is crucial to understand the structure of a fingerprint:

5.5.1 Types of Minutiae Points in Fingerprint Analysis

In fingerprint analysis, minutiae points are essential for identifying individuals. They come in two primary types:

1. **Ridge Endings:** Ridge endings are where fingerprint ridges abruptly stop. They're valuable because they're rare and stable features, persisting across different impressions of the same fingerprint. This stability makes them reliable for pinpointing unique characteristics.

2. **Bifurcations:** Bifurcations occur where a single ridge splits into two branches, forming a Y-shape. They're important due to their prevalence and complex geometry, aiding in detailed feature extraction and matching.

Importance of Minutiae Points:
Minutiae points are crucial because:

- They are unique to each fingerprint, allowing for accurate individual identification.
- They persist across multiple impressions of the same fingerprint, ensuring consistent identification.
- They are robust to environmental factors and partial damage, making them reliable in various conditions.
- They enable efficient processing and comparison of fingerprint data for quick identification.

In essence, minutiae points are the foundation of fingerprint analysis, enabling precise and reliable identification of individuals.

Now that we understand the importance of the fingerprint structure, we can look into how exactly its extraction is performed.

Overview of the extractor

The minutiae extraction process comprises three main stages: preprocessing, extraction, and postprocessing. Initially, in preprocessing, the input .bmp fingerprint image undergoes enhancements to improve its quality. Then, in the extraction stage, the algorithm identifies ridge structures and minutiae points using specialized techniques. This step is crucial for isolating the unique features of the fingerprint. Finally, in postprocessing, the algorithm validates and refines the extracted minutiae to ensure accuracy. The output is a binary matrix representing the fingerprint pattern, with 0s for background and 1s for ridge structures and minutiae points. This simplified process ensures a clear representation of the fingerprint for further analysis.

5.5.2 Preprocessing

Histogram Equalization and Binarization

In the preprocessing stage of fingerprint image analysis, two key techniques are commonly employed: histogram equalization and binarization. These techniques play a crucial role in enhancing the quality and preparing the fingerprint image for subsequent analysis.

Histogram Equalization

Histogram equalization is a technique used to enhance the contrast and overall brightness of an image. The main goal is to spread out the intensity values of the image such that the histogram of the output image becomes approximately uniform. This is particularly useful in fingerprint analysis because it helps to accentuate the ridge structures and minutiae points, making them more distinguishable.

The function calculates the histogram of the input grayscale image and computes the cumulative distribution function (CDF) based on the histogram. The CDF maps each pixel intensity value to a new intensity value, ensuring that the intensity values are spread out evenly across the entire dynamic range. Finally, the algorithm applies the computed CDF to each pixel in the image to perform the equalization, resulting in an output image with enhanced contrast.

Binarization

Binarization is the process of converting a grayscale image into a binary image, where each pixel is represented by either a 0 (black) or a 1 (white) based on a specified threshold value. In fingerprint analysis, binarization is used to segment the image into foreground (ridge structures and minutiae points) and background regions.

The algorithm takes the output of the histogram equalization process as input and applies a thresholding operation. Pixels with intensity values above the specified threshold are assigned a value of 1 (white), indicating foreground regions, while pixels below the threshold are assigned a value of 0 (black), representing background regions. This binary representation simplifies subsequent analysis tasks, such as minutiae extraction and feature detection.

Progress and optimization

In fact, since the second presentation, we opted out of applying another preprocessing algorithm to the fingerprints - the Fourrier transform. We noticed after a lot time of wondering why important data is missing from the resulting matrix, that the transform was actually affecting the recognition of the important ridges negatively, due to the size and the quality of out imput image. Hence, we are no longer using this function and have perfect results from the remaining two.

Summary

All in all, the preprocessing stage of fingerprint image analysis involves applying histogram equalization to enhance contrast and brightness, followed by binarization to segment the image into foreground and background regions. These techniques prepare the fingerprint image for further analysis and facilitate the accurate extraction.

5.5.3 The main extraction

Zhang-Suen Thinning Algorithm

The Zhang-Suen thinning algorithm refines ridge structures in binary images while maintaining their connectivity. The thinning process involves iteratively applying specific conditions to determine which pixels should be removed. Here are the key conditions used in the algorithm:

1. **Condition for Even Iterations:**
    - A pixel is marked for deletion if it:
        - Is a foreground pixel (i.e., its value is 1).
        - Has between 2 and 6 neighboring foreground pixels.
        - Has at least one neighboring background pixel (i.e., its value is 0).
        - Has at least one neighboring background pixel in the clockwise direction from its left neighbor.
        - Has at least one neighboring background pixel in the counterclockwise direction from its right neighbor.
2. **Condition for Odd Iterations:**
    - In addition to the conditions for even iterations, a pixel is marked for deletion during odd iterations only if at least one of its 8-connected neighbors is not background in an extended neighborhood of two pixels in each of the cardinal directions.

3. **Iteration Termination:**

   - After each sub-iteration, we check if any pixels were marked for deletion. If changes occur, we continue to the next iteration. Otherwise, if no changes are detected after an iteration, we terminate the algorithm.

By applying these specific conditions during each iteration, the Zhang-Suen thinning algorithm effectively reduces the thickness of ridge structures in binary images while preserving their essential connectivity. This iterative process refines the representation of ridge structures, resulting in a streamlined depiction of the fingerprint pattern.

Progress and optimization

Until the second presentation we have been using an algorithm that slightly differed from the one described above, but that small difference made a large impact. In critical places, such as bifurcation structures, the first algorithm performed the neighbourhood analysis in such a way that the fork of the bifurcation was unjustly loosing valuable pixels. However, having switched to the Zhang-Suen algorithm, we have successfully eliminated those faults.

5.5.4 Morphological operations

"H"-Break Removal

In the context of fingerprint image processing, "H" breaks refer to a specific pattern where ridge structures form an "H" shape due to breaks or interruptions. These breaks disrupt the continuity of the ridges and can adversely affect the accuracy of minutiae extraction.

1. **Why they are problematic:**
    - "H" breaks create discontinuities in the ridge structures, making it challenging to accurately identify minutiae points.
    - These breaks can lead to false minutiae detections or inaccuracies in the fingerprint analysis process.
2. **Removal of "H" Breaks:**
    - To remove "H" breaks, we can use morphological operations to smooth out the ridge structures and restore their continuity.
    - One approach is to apply dilation followed by erosion operations to bridge the gaps and connect the broken ridges.
    - By filling in the gaps and restoring the integrity of the ridge structures, we ensure a more accurate representation of the fingerprint pattern.

Removing Isolated Points

In the context of fingerprint image processing, isolated points refer to single foreground pixels (i.e., ridge pixels) that are not connected to any other ridge structures.

1. **Why are they problematic:**
    - Isolated points do not contribute to the meaningful structure of the fingerprint.
    - They can introduce noise, leading to false minutiae detections and inaccuracies in the analysis.
    - The presence of isolated points can complicate the process of accurately matching fingerprints.
2. **Removal of Isolated Points:**
    - To remove isolated points, we can use morphological operations that identify and eliminate these lone pixels.
    - A common approach is to apply a connectivity check to each pixel. If a foreground pixel has no neighboring foreground pixels, it is considered isolated and is removed.
    - This process involves scanning the image and setting the value of isolated pixels to background (i.e., turning them from 1 to 0).

By removing isolated points, we can reduce noise and improve the clarity and accuracy of the fingerprint pattern. This step is crucial for ensuring that the extracted minutiae points are reliable and accurate, leading to better fingerprint analysis and matching.

Removing Spikes

Spikes are small, protruding ridges that stick out from the main ridge structures. These spikes can mess up the fingerprint pattern and affect the accuracy of identifying minutia.

1. **Why they are problematic:**
    - Spikes can distort the ridge structure, leading to errors in identifying minutiae points.
    - They can create false minutiae points, making fingerprint matching more difficult.
    - The presence of spikes can make the fingerprint image look cluttered and unclear.
2. **Removal of Spikes:**
    - To remove spikes, we use morphological operations to smooth out the ridge structures.
    - We apply an erosion operation followed by a dilation operation. Erosion removes small protrusions, and dilation restores the size of the remaining ridges.
    - This process helps eliminate spikes while keeping the main ridge structures intact.

By removing spikes, we can make the fingerprint pattern more clearer and more accurate. This step ensures that the extracted minutiae points are more reliable, leading to better fingerprint analysis and matching.

5.5.5 Postprocessing

Minutiae Marking

In the introduction we already talked about the unique structures that make up a fingerprint. Minutiae marking is a crucial step in fingerprint analysis where we identify and classify key features, such as ridge endings and bifurcations, within a fingerprint image. These features are essential for accurately identifying and matching fingerprints.

Why it is important

- **Unique Identification:**
  - Each fingerprint has a unique pattern of minutiae points, which allows for precise identification of individuals.
  - By marking these points, we can create a unique signature for each fingerprint.
- **Feature Extraction:**
  - Minutiae points provide detailed information about the fingerprint's ridge structure.
  - Extracting these points helps in simplifying and accurately comparing fingerprints.
- **Data Reduction:**
  - Instead of analyzing the entire fingerprint image, we can focus on the marked minutiae points, making the process more efficient. This reduction in data size leads to faster and more efficient fingerprint matching.

The marking algorithm

The process of minutiae marking involves scanning the fingerprint image to identify and classify key features.

1. **Neighborhood Analysis:**
    - For each pixel in the fingerprint image, we examine its 8-connected neighbors (the pixels directly surrounding it).
    - This analysis helps in determining the type of minutia based on the number of neighboring ridge pixels.
2. **Identifying Ridge Endings:**
    - A ridge ending occurs when a ridge terminates abruptly.
    - In the neighborhood analysis, a ridge ending is identified when the pixel has exactly one neighboring ridge pixel.
    - These points are marked and classified as `MinutiaType::RidgeEnding`.
3. **Identifying Bifurcations:**
    - A bifurcation occurs when a single ridge splits into two branches.
    - In the neighborhood analysis, a bifurcation is identified when the pixel has exactly three neighboring ridge pixels.
    - These points are marked and classified as `MinutiaType::Bifurcation`.
4. **Storing Minutiae:**
    - The identified minutiae points are stored with their coordinates and types.
    - This structured representation of minutiae points is used for further processing and matching.

Implementation of false minutia removal using Fuzzy rules

Again, as we just mentioned, the data we acquired during the Minutia Marking has to be stored in a way that would allow us to easily access it. To achieve that, we created the following two structures: the Minutia structure and the MinutiaType enum. This significantly facilitates our work with large vectors of minutia representing a fingerprint image.

Then, for the main part of the algorithm we use fuzzy rules. Fuzzy logic deals with uncertainty by allowing for degrees of truth between 0 and 1. Fuzzy rules, expressed as "if-then" statements, define how inputs relate to outputs in a fuzzy system. By applying these rules to input data, fuzzy logic systems produce output values representing degrees of truth, enabling effective handling of complex and uncertain situations.

Fuzzy Rules and their problem

Initially some simpler and compact set of fuzzy rules are proposed for removing false minutiae as described below:

1. Rule 1: IF the distance between termination and bifurcation is less than D, THEN remove both the minutiae.
2. Rule 2: IF the distance between two bifurcations is less than D, THEN remove both the minutiae.
3. Rule 3: IF the distance between two terminations is less than D, THEN remove both the minutia.

The average inter ridge distance (D) between two neighboring ridges is computed by the formula:

$$D = \frac{\text{sum of all pixels in the row whose value is one}}{\text{row length}}$$

One significant issue with the described approach is that, despite its apparent simplicity, there's a risk of inadvertently discarding genuine minutiae alongside the erroneous ones. To illustrate, imagine a situation where there are two points marked as A and B on a fingerprint ridge. Point A represents a genuine ending of the ridge, while point B is mistakenly identified as a ridge termination. The distance between these two points is less than a certain threshold value. The concern arises when following the proposed rules, both points A and B would be removed, even though point A is a legitimate ridge termination. This highlights the problem of inadvertently eliminating genuine features along with the false ones, which is a significant issue in fingerprint analysis.

Modified Fuzzy rules

Having proposed modifications to improve the accuracy of the implementation of the fuzzy rules, the modified rules are as follows:

1. IF the distance between termination and bifurcation is less than D and both are on the same ridge THEN remove both the minutiae.
2. IF the distance between two bifurcations is less than D and both are on the same ridge THEN remove both the minutiae.
3. IF the distance between two terminations is less than D and both are on the same ridge THEN remove both the minutiae.
4. IF two terminations are within a distance D but on different ridges and their directions are synchronized with a minimal angle variation and no other termination is located between those two terminations THEN remove both the terminations.

Thus, based on these rules we can code the most important function - `remove_false_minutiae` with the following logic:

1. We extract the minutia from the enhanced preprocessed fingerprint image and determine their type using the marking algorithm;
2. We calculate the value D
3. Calculate the orientation angle θ between two serial ridges endings
4. We apply the modified fuzzy rules to the extracted minutia to remove the false data

Conclusion of the false minutia removal process:

These manipulations are extremely important for the optimization of the whole fingerprint detection process. Having run this algorithm on 20 different examples of fingerprints, I have discovered that the number of bifurcation points decreases by more than 50% and termination points - by more than 66%. Thus, during the minutia matching algorithm run we take into account only the useful data we get from the minutia, and do not waste time and computing power on processing the false minutia.

5.6 The progress and optimizations

Since the second defense, we have basically reworked the whole extraction mechanism in order for it to work with more time and memory efficiency, as well as being more concise in terms of coding style. We have implemented updated structures which allow for an easier data retrieval, rewritten auxiliary algorithms so that they obtain a wider use for the major functions, and modified the algorithms for maximum preciseness.

5.7 Minutia Matching

Minutia matching remains a central component in our project, acting as the cornerstone for the comparison of fingerprints during user authentication. Our methodology involves comparing matrices of minutiae between the enrolled user (reference) and the user attempting to authenticate. This process is critical for establishing a reliable biometric authentication system.

The comparison is carried out by employing a threshold value, which is a crucial parameter in determining the acceptance or rejection of a fingerprint match. This threshold serves as a criterion for assessing the similarity between minutiae sets. We perform a pairwise comparison of minutiae, calculating a similarity metric that encompasses spatial distance and orientation difference. Each minutia pair is evaluated against this threshold, and only those exceeding it are considered for further analysis.

The `minutiae matching` function takes references to two fingerprint images, detects minutiae points using the optimized `detect minutiae` function, and compares these points.

The refined minutiae matching algorithm significantly enhances the biometric authentication process in our project. By integrating these improvements, we aim to provide users with a more secure, reliable, and user-friendly authentication method, thereby elevating the overall user experience and security of our system.

5.8 User interface

5.8.1 User Interface (UI) Implementation

The user interface for the project is built using the Druid library, which provides a platform for building GUI applications in Rust. The interface is designed to facilitate user interactions for logging in, registering, and managing credentials. The main components of the UI include various views, widgets, and layout arrangements to provide a seamless user experience.

5.8.2 UI Structure

The application features a multi-view interface controlled by the `AppState` and `ViewSelector` structures. The primary views include:

- **Login View**
- **Register View**
- **Credentials View**

Each view is implemented as a separate layout within the main window and is conditionally rendered based on the current state of the application.

5.8.3 AppState and ViewSelector

The `AppState` struct holds the current state of the application, including user information and the selected view. It is defined as follows:

The `ViewSelector` enum defines the possible views that can be displayed

5.8.4 View Implementations

The UI is composed of several widgets and layouts. Below is a detailed description of each view.

5.8.5 Login View

The login view consists of a label, a text box for the username, and buttons for logging in, registering, and quitting the application.

5.8.6 Register View

The register view includes similar elements to the login view, with additional instructions for registering the user's fingerprint.

5.8.7 Credentials View

The credentials view allows users to add, view, and delete their credentials. It includes text boxes for site information, a button to save credentials, and a list to display saved credentials.

5.9 Website

Our website, built using HTML and CSS, acts as a central hub for all project-related information. With its simple layout and straightforward design, visitors can easily navigate through various sections to learn about our project, team members, download resources, and explore our sources.

Home Page

The home page welcomes visitors with our project's logo and a slogan, crafted with HTML and CSS for a visually pleasing presentation. It is made to fit the aesthetic of the logo of BioGuard.

Our Project

The "Our Project" section offers a introduction to our project and an invitation to explore further. Here, visitors can learn about our objectives, methodologies, and achievements. Additionally, we showcase our group efforts on Git through simple visuals, highlighting the collaborative nature of our work.

Members

In the "Members" section, visitors can read personal reflections from each team member about their experience working on the project. This section adds a personal touch to our project, allowing visitors to connect with the individuals involved and gain insights into their contributions and perspectives.

Downloads

The "Downloads" section provides access to download the project report, and the project itself.

Sources

The "Sources" section offers transparency by listing all the sources used in our project. Visitors can explore these sources to delve deeper into the background information, research, and data that informed our project's development and conclusions.

Conclusion

Our website serves as a comprehensive platform for sharing project information. With its user-friendly layout and simple design, it aims to engage visitors and provide them with insights into our project's goals, processes, and outcomes.

6 Technical Challenges

During the development and integration phases of the project, several technical challenges were encountered:

- **Database Connection Management**: Ensuring that the database connection is properly established and managed was challenging. Handling connection pooling and asynchronous database operations required careful attention to avoid issues such as connection leaks and deadlocks.
- **Asynchronous Programming**: Integrating asynchronous programming with the SQLx library and the overall Rust application was complex. Proper use of async/await syntax and error handling mechanisms was crucial to maintain the application's performance and reliability.
- **Error Handling**: Robust error handling for database operations was necessary to ensure the application could gracefully handle various failure scenarios, such as network issues or database unavailability.
- **Data Security**: Storing sensitive information, such as fingerprint images and user credentials, required implementing secure storage practices. Ensuring that data was securely stored and transmitted involved using appropriate encryption and secure coding techniques.
- **User Authentication**: Implementing fingerprint-based authentication required integrating with appropriate libraries and ensuring the accuracy and reliability of fingerprint matching. Handling edge cases, such as partial or corrupted fingerprint images, added to the complexity.

7 Visual Representation of Repository Work

On the "Our Project" section of the wensite is a visualization of all commits made to our Git repository, showcasing the collaborative efforts and detailed commit messages through a dynamic animation.

8 Teamwork Challenges

Unfortunately, we also have encountered some teamwork challenges. We struggled with communication in our group, specifically with getting progress updates from our ex-groupmates. Unfortunately for every presentation we were given little notice that the progress was not being made on their part, which made our job a bit more complicated. However, we believe that these challenges serve as an invaluable experience that can make us capable of handling situations similar to this one on our future careers as IT professionals. Us, the remaining two members, Maria and Angel, are proud of how much progress we were able to achieve working together, despite all the aforementioned difficulties.

9 Personal Feedback

Maria: Working on this project has been a very rewarding task. Despite the difficulties, we have a final working product which we are proud to present. For me personally, the best part about developing our application was the hands-on work on the treatment of the images. Having a tangible result of my efforts is invaluable to me, as it is greatly fulfilling. Besides, I got to more extensively work with front-end development, with which i had no experience prior to this. All of the experience collected during the duration of the project is contributing to a solid foundation of being a qualified IT specialist in the near future.

Angel: Overall, the project was an enjoyable and rewarding experience, despite facing some challenges with two group members. The inability to effectively communicate with them was certainly disappointing and added a layer of difficulty to the project. However, beyond these interpersonal issues, the technical aspects of the project were fascinating and highly educational.

Learning to access sensors through APIs like WinBio was an intriguing process that expanded my understanding of biometric integration. Additionally, diving into user interface development using Rust was particularly exciting. Rust's unique features and performance capabilities made the learning curve steep but ultimately rewarding.

I dedicated a significant amount of time and effort to this project, and in doing so, I gained a wealth of knowledge. The challenges and successes alike contributed to a fulfilling learning experience. Despite the hurdles, the project was enjoyable, and I derived a great deal of satisfaction from the skills and insights I acquired along the way.

10 Conclusion and Future Work

This project has been an invaluable learning experience, providing us with practical insights into database management, user authentication, and the application of biometric verification systems. Here are some of the key lessons we learned:

- **Database Management:** We gained hands-on experience in designing and managing a SQLite database, including creating tables and handling CRUD (Create, Read, Update, Delete) operations efficiently.
- **Biometric Verification:** Implementing fingerprint verification enhanced our understanding of biometric systems and their potential for secure user authentication.
- **Software Architecture:** Structuring our application using models, database handlers, and a main application loop improved our skills in organizing and maintaining complex software projects.

The project was particularly interesting because it combined several advanced concepts into a cohesive system. The integration of biometric verification with a user credential management system is a compelling application with real-world relevance. It highlighted the importance of security and efficiency in handling sensitive user data.

While we achieved significant milestones, there are several areas where future work can build upon our findings:

- **Model Refinement:** Further refining the fingerprint verification model to improve accuracy and performance, especially in handling edge cases.
- **Integration with Existing Systems:** Exploring how our model can be integrated with existing security systems and technologies to assess its practical applications and scalability.
- **Advanced Techniques:** Investigating advanced techniques, such as machine learning, to enhance the model's predictive capabilities and provide deeper insights.
- **User Experience Improvements:** Focusing on making the system more user-friendly, ensuring that it is accessible to a broader audience, including those without specialized knowledge.

Future research could explore related areas, such as the application of machine learning algorithms for improved biometric verification and collaboration with interdisciplinary teams to address more complex, multifaceted security challenges. Additionally, investigating the potential for real-time application of the system in dynamic environments offers an exciting direction for further study.

In **conclusion**, this project has provided us with valuable insights and practical experience. By building on these achievements and exploring the outlined future work, we can continue to advance our understanding and contribute meaningful solutions to the field of biometric authentication and secure data management.