Biometric Matching System for Everyday Applications

José Adrián Quel Paillacho[1]

[1]*Yachay Tech University*, Urcuquí, Ecuador
jose.quel@yachaytech.edu.ec

Abstract

This project develops a program in C++ to compare biometric data, aimed at three main fields: commercial, security and identification, and medical. The program uses a structure to organize data such as height, weight, hand length, foot size, and facial ratio, and reads this information from text files. Through a function that compares these data with defined tolerances, the system determines if two biometric profiles match. During the tests, four biometric data files were compared, finding only one match between two of them. This result confirms the basic operation of the program and its ability to distinguish similar profiles. Beyond the tests, the potential of the system is discussed to improve accuracy in online purchases, reduce manufacturing waste, and contribute to environmental sustainability. In the medical field, it facilitates quick consultations and tracking of biometric histories, which could support preliminary self-diagnoses. For security, the project opens the possibility of integrating artificial intelligence for more complex and reliable biometric comparisons. Although the system is simple and has limitations, it represents a solid base for future developments that could positively impact several sectors.

Keywords: *Biometry, C++, advanced programming, structs, security, health.*

Introduction

The growth of online shopping and digital marketplaces has changed the way consumers buy products, offering unprecedented convenience and access to a wide variety of items. However, this shift has also brought significant challenges related to personalization and product fitting, especially for items where physical measurements and user characteristics play a critical role, such as clothing, footwear, furniture, and health-related equipment. One of the most common problems is the mismatch between the product specifications provided online and the actual physical attributes of the consumer. This often results in incorrect sizes, discomfort, or even health-related issues.

To address these challenges, there is growing interest in systems based on biometric data that use precise body measurements to provide personalized product recommendations. In this context, biometrics refers to the collection and analysis of unique human physical features, such as length, circumference, and body proportions. By integrating biometric data into product selection processes, it is possible to significantly reduce sizing errors, improve comfort, and enhance overall customer satisfaction.

The proposed solution uses a modular approach that collects biometric data from users, stores it in structured files, and compares it with product profiles or other biometric records. This approach aims to improve personalization in e-commerce and health contexts by using objective, measurable data instead of relying on generic sizing charts or subjective user input.

Implementing this system in C++ allows for efficient data processing and the creation of reusable components that can be extended or integrated into larger platforms in the future. Additionally, the modular design enables customization for specific applications like fashion retail, ergonomic consulting, or preliminary health diagnostics.

In summary, this project addresses an important and timely problem in product personalization and customer satisfaction by leveraging biometric data. Its successful implementation has the potential to improve user experience, reduce economic and environmental costs associated with product returns, and raise awareness about health and ergonomics. The following sections will discuss related work, system design, implementation details, and the potential impact of this approach.

Objectives

The main purpose of this project is to develop a program capable of comparing biometric data with applications in three different areas: commerce, security and identification, and the medical field. This tool aims to facilitate the personalization, validation, and tracking of biometric data in an efficient, simple, and reusable way.

The specific objectives are:

- **Create a standardized data structure** that allows clear storage and organization of relevant biometric measurements for different applications, including measurements such as height, weight, hand length, foot size, and facial proportions.

- **Implement a comparison algorithm** that can evaluate the similarity between biometric profiles considering defined tolerances for each measurement, so it can determine if two profiles match or not, adapting to the needs of each application field.

- **Ensure ease of use and maintenance** through implementation in C++, with a clear and modular design that facilitates future improvements and adaptations to different contexts.

With these objectives, the program aims not only to function correctly but also to provide practical value in various fields, demonstrating the usefulness of biometric data to solve real problems and improve quality of life.

Materials and Methods

For the development of this project, the **C++ programming language** was selected due to its power, speed, and widespread use in fields requiring efficient data handling and structured code. The idea behind choosing C++ is not only its technical capabilities but also the opportunity it offers for beginners in programming to learn about structured thinking, file manipulation, and decision-making through simple and modular logic.

The tools and resources used were:

- **Programming Language:** C++.
- **Compiler:** g++ compiler for supporting C++ programming language.
- **Code Editor:** Visual Studio Code.
- **Input Files:** .txt files containing biometric data in comma-separated format.
- **Command Line / Terminal:** Used to compile and run the program, and to view results.

Program Structure Overview

The program is organized into four main parts:

1. **Definition of the Data Structure (Structs):** A structure called BiometricData was created. This structure contains fields such as height, weight, hand length, foot size, and face ratio, along with a person identifier (ID). All the necessary biometric values are grouped together in a single object, making it easier to manage and manipulate the data. This structure simplifies both storage and access during comparisons.
2. **A function that reads the Biometric Data from a File .txt:** The program includes a function named readBiometricData, which opens a .txt file and extracts biometric values from it. Each file is expected to have just one line in the following format:

   ID,height,weight,handLength,footSize,faceRatio

   These values are read as strings and then converted into numbers using std::stod. If the file cannot be opened or does not follow the correct format, the function throws an error. This function returns a BiometricData object with all fields filled.
3. **A function that compares two .txt files:** The core comparison logic lies in the isMatch function. This function accepts two BiometricData objects and compares

each corresponding value within a predefined range of tolerance. For example, two people are considered to have the same height if the difference between their heights is no more than 5 centimeters. Similarly, weight is compared within a 5 kg margin, and other metrics have their own specific tolerance levels. If all data values fall within these ranges, the function returns true; otherwise, it returns false.

4. **The main function:** The program is executed from the terminal, taking the names of two .txt files as command-line arguments. The main function checks that the user has provided two filenames. It then reads each file using readBiometricData, calls isMatch to compare the profiles, and prints "Match" if the profiles are considered similar or "No Match" if not. If any error occurs during the process (such as file not found, wrong format, or bad data), an appropriate message is shown on screen.

Applications of methods in Real – World problems

Although the technical implementation is general, the logic behind the program can be used in several real-life scenarios depending on the context. This project was designed to be adaptable and meaningful across three key application areas:

1. **Commercial Use cases**

In the commercial sector, this system can be applied to compare physical characteristics with predefined product sizes. For instance, if a customer provides their biometric data (height, hand length, foot size), the system can match it against product specifications (such as shoe or glove sizes). The use of tolerances allows for a margin of flexibility, which is important when dealing with products that don't always fit in exact sizes.

It could also be used for ergonomic product design, helping companies ensure that products match the average dimensions of their target users. The biometric data could be used to simulate fit or comfort for tools, clothes, furniture, or wearable devices—potentially even in virtual environments.

2. **Security and Identification**

One of the most direct and impactful uses of biometric comparison is in identity verification. In this context, two biometric profiles from different sources can be compared to check if they likely belong to the same person. While this program does not use images, fingerprints, or retina scans, it applies similar logic by comparing measurable body attributes.

3. **Medical and Health monitoring**

In the medical field, consistent measurement of a person's biometric data over time can help detect changes in health. The system can be adapted to compare the current biometric profile of a patient against their historical data. If a significant change is detected (such as weight gain, height reduction due to age, or hand swelling), a flag could be raised for medical review.

Development methodology

The project followed a straightforward methodology based on iterative and modular development. It began with defining the data structure, followed by writing a function to read data from files. After ensuring that file input was handled correctly, the logic for comparing two profiles was implemented. The final step involved building the main function that ties everything together and allows the user to interact with the program via the command line.

Each step was tested with sample files to ensure correctness. Dummy data such as:

A001,1.75,70.0,18.5,27.0,1.45
A002,1.77,72.0,18.0,26.5,1.44

was used to simulate various situations where two profiles were either matching or not. This allowed for debugging both functional and logical issues early on.

Additionally, clean code practices were used: meaningful variable names, compact functions that perform only one task, comments for clarity, and a predictable structure. The idea was not just to make the program work but to ensure that it could be reused or expanded later—whether to handle more fields, multiple profiles, or a graphical interface.

Results and Discussion

Results

During the testing phase, the program was evaluated using four structured text files: data1.txt through data4.txt. Each file represented biometric data corresponding to a distinct profile, including key physical parameters such as height, weight, and facial ratios. The matching algorithm was designed to compare these metrics within a predetermined tolerance threshold.

Among the four test cases, the program identified a single relevant match between data1.txt and data4.txt, indicating that these two profiles shared biometric similarities close enough to fall within the set comparison criteria. While the dataset was minimal and controlled, this outcome confirms that the core comparison mechanism functions as expected in detecting simple biometric congruencies.

Although these results are limited in scope, they serve as a technical proof of concept for a basic yet functioning biometric comparison tool.

Discussion

Despite the simplicity of the test environment, this early prototype demonstrates the foundational potential for broader applications in various fields. One such area is the commercial sector, particularly in personalized shopping experiences. With enhanced biometric comparison tools, it would be possible to offer highly accurate virtual sizing recommendations for clothing, accessories, or ergonomic products. This would reduce return rates, improve customer satisfaction, and streamline the manufacturing process by limiting overproduction. These improvements could lead to substantial cost savings and contribute to a more sustainable production model, reducing material waste and lowering the environmental impact of mass consumer goods.

In the medical field, this system shows promise for longitudinal biometric tracking. If adapted to store and compare biometric data over time for the same individual, it could support faster diagnostics, remote consultations, and preventive care. For instance, regular comparisons

between current and historical biometric records could alert patients or doctors to concerning changes, thus facilitating early intervention. As healthcare increasingly embraces digital tools and self-monitoring, this type of biometric tracking could empower individuals to better understand and manage their health.

From a security and identification perspective, biometric comparison offers important opportunities for identity verification, fraud prevention, and even anomaly detection in sensitive environments. While the current system performs straightforward numerical matching, integrating AI-based techniques or machine learning could evolve it into a more sophisticated tool capable of analyzing patterns, learning from datasets, and adapting to a range of user profiles. Such capabilities would make the system more resilient to tampering and more accurate in complex identity scenarios.

An important limitation of the prototype is the current use of .txt files for storing biometric data. While suitable for early development and debugging, this format lacks the flexibility and structure required for real-world deployment. Future iterations of the program should move toward a customized data format, capable of storing more extensive and diverse information—such as facial imagery, timestamps, biometric metadata, and contextual health data. This would also improve interoperability with other systems, such as electronic health records, e-commerce platforms, or security infrastructure.

However, for the system to reach its full potential, it will need to overcome several current limitations. These include the lack of parameter weighting, absence of contextual differentiation (e.g., age or gender), and low scalability for large datasets. Addressing these aspects will be critical to making the system viable for real-world deployment across its three primary areas: commercial, medical, and security.

Conclusions

This project has demonstrated the feasibility of an initial biometric data comparison system developed in C++, with potential applications in key areas such as commerce, security, and medicine. Through basic tests, the comparison algorithm was validated, showing a match between two data sets, which highlights the system's potential for accuracy, even at this early stage.

Beyond the code itself, the broader impact of such tools is evident: they can optimize purchasing processes through personalized sizing, facilitate quick medical assessments, and support identity verification in security systems. Additionally, by reducing errors and unnecessary manufacturing, this type of system may also contribute positively to environmental sustainability.