UNIVERSITÀ DEGLI STUDI DI PADOVA

DEI DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Department of Information Engineering

Master degree in Computer Engineering

"Multimodal biometric authentication based on voice, fingerprint and face recognition".

Supervisor: Prof. Carlo Ferrari

Master Candidate: Marco Gallo

Academic Year 2022/2023

December 14, 2023

Abstract

Biometrics is the measurement and statistical analysis of people's unique physical and behavioral characteristics. Biometric systems use personal characteristics to authenticate someone, the basic premise is that every person can be accurately identified by intrinsic physical or behavioural traits. A system first collects biometric characteristics unique to every person, these characteristics are then directly linked to verify or identify the individual. Several types of biometric systems exist that make use of a single characteristic of an individual, such as fingerprint, palm veins, iris… Multimodal biometric authentication systems instead can use several characteristics to take a final decision, the level were the information fusion happen and the type of fusion are extremely important decisions that must be taken when modelling a multimodal biometric system (i.e. sensor level fusion, feature level fusion, decision level fusion, score level fusion and hybrid fusion level). In this work after presenting the different characteristics and design choices of multimodal biometric systems, we analyse different fusion methods in literature and then finally we implement a multimodal biometric system using fingerprint, voice and face as single biometric traits and custom fusion algorithm for taking a final decision.

Contents

Chapter 1

Introduction

The term “Biometric” relates to the utilization of organic, physical, or social characteristics of an individual as a form of identification and access control, it is also used to identify individuals in groups.

Biometric identification systems which use a single biometric trait of the individual for identification and verification are called unimodal systems, some examples are fingerprint, face, iris, or vein recognition. Each biometric system has its own set of drawbacks and benefits that must be considered when designing a system.

A multimodal biometric identification system integrates two or more unimodal biometric authentication systems, a multimodal system can gather different biological characteristics such as fingerprints, faces, iris images, and so on, using independent or multiple collection methods combined into one collector, and then analyse and judge the characteristic values of multiple biometric methods to identify and authenticate.

Biometric technology is transitioning from a single-mode approach to a multimodal approach, indeed Multimodal biometric systems tend to outperform unimodal biometric systems and so they are useful in a lot of more intricate and diverse authentication scenarios, for example: defence and the intelligence, Border management, interface for criminal and civil applications, but also in domains such as Personal information and Business transactions that require fraud prevent solutions that increase security and are cost effective.

We use more than one biometric modality in multimodal biometric systems and hence we have more than one decision channels. Thus arises the need to design a mechanism which can combine the classification outcome from each biometric channel and this mechanism is known as biometric fusion. This fusion combines the measurements from different

biometric attributes to enhance the strengths and decrease the weaknesses of the individual measurements.

Fusion can be used to address several issues faced in implementation of biometric systems such as accuracy, efficiency, robustness, applicability and universality. There are various possible levels for fusing the biometric traits which can be used to increase robustness of the multimodal biometric system, they are: sensor level fusion, feature level fusion, matching score level fusion and decision level fusion.

Chapter 2

Background

in this chapter we will introduce multimodal biometric systems their characteristics and possible design choices, while implementing our system in the next chapter what shown here will be considered, also we will review some literature to investigate alternative implementation methods to our algorithm.

2.1 Background and design choices

A Biometric system is an identification system in view of the utilization of various biometric features of people by the investigation of physiological characteristics, for example, fingerprints, eye retinas and irises, voice designs, facial examples and hand estimations for authentication purposes or behavioural attributes. Authentication systems setup with one biometric modality may not be adequate for the related application as far as properties, for example, universality, distinctiveness, acceptability etc. 100% accuracy may not be accomplishable in unimodal systems by virtue of the limitations, for example, the noise in the sensor data, intra-class variations, inter-class similarities, lack of universality, spoof attacks and other vulnerabilities.

Accuracy in biometrics is measured in terms of error rates. The two mostly utilized error rates are:

False Acceptance Rate (FAR) is the percentage of identification instances in which unauthorised persons are incorrectly accepted.

False Rejection Rate (FRR) is the percentage of identification instances in which authorised persons are incorrectly rejected.

As the number of false acceptances (FAR) goes down, the number of false rejections (FRR) will go up and vice versa. The **Equal Error Rate (EER)** is the point where the percentage of false acceptances and false rejections is the same.

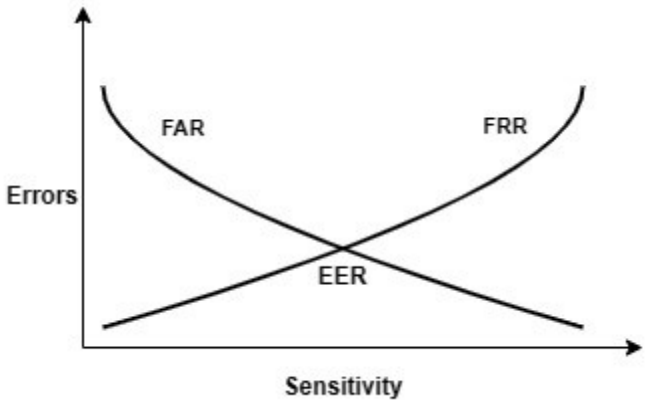

Figure 2.1: sensitivity and error relationship

Biometric traits should possess several qualities, in particular seven attributes should be considered when evaluating a biometric feature: Universality, Distinctiveness, Permanence, Collectability, Performance, Acceptability, and Circumvention.

Universality: every individual should have the biometric trait.

Distinctiveness: no two individuals should be identical in terms of the biometric traits.

Permanence: the trait should be sufficiently invariant over time.

Collectability: the trait should be easily measurable without any inconvenience to the user.

Performance: relates to accuracy, speed of the technology used.

Acceptability: stands for the user acceptance to the collection of the biometric.

Circumvention: the ease with which the biometric trait can be deceived.

Those traits are present in different qualities in different biometric technologies and is of paramount importance considering that when building a new system. In the following table we present a brief comparison of several different biometric identifier in terms of those seven features from [19]:

Table 1: Comparison of different biometric technologies

Biometric identifier / Characteristics	Finger	Facial	Iris	Hand	Retina	Signature
Universality	high	high	high	mid	high	low
Distinctiveness	high	low	high	mid	high	low
Permanence	high	mid	high	mid	mid	low
Collectability	mid	high	mid	high	low	high
Performance	high	low	high	mid	high	low
Acceptability	high	high	low	mid	low	high
Circumvention	mid	high	low	mid	low	high

Different biometric identifiers have different strength and weaknesses that can be combined to form a more powerful multimodal biometric system. In the table below from [19] we analyse those strength and weaknesses from some main biometric Identities.

Table 2: Strength and Weakness of different Biometric Identities

Biometric-Identifier	Strengths	Weakness
Finger- scan	High level of accuracy, easy to use, flexibility	Performance can deteriorate over time, unable to enroll some percentage of users
Facial- scan	Able to operate without user cooperation	Changes in physiological characteristic reduce matching accuracy
Signature- scan	Resistant to imposters	Lead to increased error rates
Hand- scan	Reliable core technology, stable physiological characteristic.	Limited accuracy
Retina- scan	Highly accurate	Difficult to use and capture
Iris-scan	Resistance to false matching	Difficult of use and capture

Unimodal Systems depending on one source of information for authentication suffer from a variety of problems such as:

-Noise in the sensed data. (e.g., due to repeated use of fingerprint sensor)

-Intra-class variation: User who is incorrectly interacting with the sensor typically causes these variations.

-Inter-class similarities: In a Biometric System where there are large numbers of users, there may be inter–class overlap in the feature space of multiple users.

-Non-Universality: The Biometric System might not be able to acquire a meaningful Biometric data from a subset of users.

-Spoof Attack: This attack occurs when signature or voice are used in Biometric System.

Some limitations of unimodal systems can be overcome by including multiple sources of information for identification using **Multimodal Biometric Systems**. Multimodal biometric systems are a refined arrangement of unimodal systems, which fuse the therapeutic measures for the downsides of the unimodal biometric system. These systems are more reliable due to the presence of multiple independent biometrics, they improve on the single biometric system in several other different ways too:

-They improve security given that it would be difficult for an imposter to spoof multiple biometric traits of a genuine user simultaneously.

-They can provide a challenge-response type of mechanism by requesting the user to present a random subset of biometric traits.

-They provide augmented accuracy because outcomes obtained from numerous cues can be fused by choosing the suitable level of fusion and applying efficient fusion scheme to achieve augmented accuracy.

-They can counter the non-universality issue of unimodal biometric framework. For example, identification of someone is still feasible by using the other cue even if due to some ailment, he is incapable to access palm-print framework.

-They can improve accessibility using any cue to access the system.

-They can counter failures more easily given that one technology alone may not influence seriously the individual identification as different technologies can be successfully employed.

Different types of multimodal biometric system can exist:

Single biometric trait, multiple sensors: Multiple sensors are used for the same biometric characteristic. The data taken from different sensors can then be combined at the feature level or matcher score level to improve the performance of the system.

Multiple biometrics: Multiple biometric traits can be combined. Different sensors are used for each biometric characteristic, the performance of the system significantly increase with this method.

Multiple units, single biometric traits: to improve system performance in an inexpensive way we can use the same sensor and get multiple different instances of a single trait, for example using different fingers.

Multiple snapshots of single biometric: multiple instances of the same biometric trait can be used, for example the same finger several times (unlike the previous method where different fingers are used).

Multiple matching algorithms for the same biometric: different methods are used in the feature extraction and matching of the biometric characteristic.

A multimodal biometric system can work in three modes:

Serial mode: The output of one biometric characteristic is used to reduce the number of possible identities, the remaining set will be the reference for the next characteristic.

Parallel mode: The information from multiple characteristics is considered together to perform recognition, this mode will be our reference when developing the project.

Hierarchical mode: Individual classifiers are combined in a tree like structure. This mode is well suited if we have many classifiers.

The four **common modules** in any biometric system are: the sensor module, the feature extraction module, the matching module, and the decision-making module.

Sensor Module: In this module the biometric sensor or scanner measure the raw data of the user which is then recorded and transferred to the next module for feature extraction. The various factors like cost and size are impacted by the design of the sensor module of the biometric system.

Feature Extraction Module In this module, the raw data transferred from the sensor module is used to generate a synoptic but indicative digital representation of the underlying traits or modalities. After extracting the features, it is given as input to the matching module for further comparison.

Matching Module The extracted features when compared with the templates in the database generate a match score. This match score may be controlled by the quality of the given biometric data. The

matching module also condensed a decision making module in which the generated match score is used to validate the claimed identity.

Decision Making Module Decision making module identifies whether the user is a genuine user or an impostor based on the match scores.

The data from the sensors can be fused at different levels to obtain the final decision of the entire multimodal:

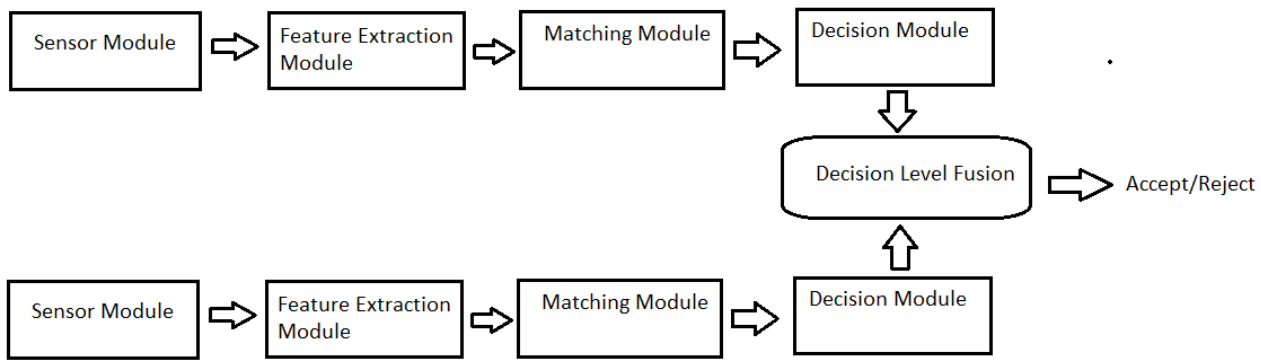

Fig 2.2: possible multi modal biometric system structure

Fusion at the sensor level: the data directly from the sensor is fused, we can either use samples of same biometric trait obtained from multiple compatible sensors or multiple instances of same biometric trait obtained using a single sensor; the data is fused at an early stage, so it has a lot of information as compared to other fusion levels.

Fusion at the feature extraction level: The data or the feature set from multiple sensors or sources are fused together. Features extracted from each sensor form a feature vector, different features vectors are then concatenated to form a single new vector, the same or different feature extraction algorithms can be used. The feature level fusion is challenging because relationship between features is not known and structurally incompatible features are common and the curse of dimensionality.

Fusion at the decision level: In decision level fusion, each sensor reaches an individual decision, the outputs are then combined using different schemes to finally reach a verdict. Decision level fusion manages very abstract information, so it's less preferred in designing multimodal biometric systems.

Fusion at score level: Each sensor provides a matching score indicating the proximity of the feature vector with the template vector. These scores are combined to assert the veracity of the claimed identity. Is necessary to normalize the scores to map the scores obtained from different matchers on to a same range.

Fusion at the score level is usually preferred because it is easier to combine the scores presented by different modalities, this type of fusion will be used in our system and several examples in literature will be presented using this system.

Some common score fusion method categories are:

Rule-based fusion methods: this method comprises of collection of basic rules such as: statistical rule-based method like MAX, MIN, linear weighted fusion (i.e. sum and product), majority voting, AND, OR. Customer defined rules are adopted for every specific solution.

Classification-based fusion methods: these methods classify the multimodal observation into one pre-defined class, different classification techniques are used. The method included in this category are Bayesian inference, dynamic Bayesian networks, maximum entropy model, support vector machine, dempster–shafer theory and neural networks.

Estimation-based fusion methods: In this category we have methods such as: the extended Kalman filter, particle filter and Kalman filter fusion, which are used primarily to deduce the state of moving object. For example, for the task of object tracking (to deduce the position of the object) different modalities like video and audio are fused.

Among the various fusion strategies, score-level fusion has become popular since it represents a good trade-off between information availability and information entropy. On one hand, most commercial biometric systems do not provide access to the raw data, nor the feature sets extracted from the data. On the other hand, while final decisions and ranks are readily accessible in most commercial systems, their entropy is rather limited compared to scores.

The scores first must be normalized and then can be combined using simple operations such as *max*, *min*, *sum* or *product*. The sum and product rules allow for weighted combinations of scores. The weighting can be differentiated in matcher specific, user specific or based on sample quality:

> In **matcher specific weighting**, weights are chosen to be proportional to the accuracy of the matcher (e.g. inversely proportional to the Equal Error Rate for the matcher).
> In **user specific weighting**, weights are assigned based on how well the matcher is known to perform for a particular individual.
> In **quality-based weighting**, the weights are assigned based on quality of the sample presented to the matcher.

Weighting techniques though offer only some performance improvements over simple sum and max fusion and these fusion methods do not require any training.

2.2 Fusion methods literature review

Several fusion methods are investigated in literature, below we present some examples seen during our research:

In [5], the authors applied face and speaker recognition algorithms on hand-held devices, equipped with lower quality audio/video capture hardware. The fusion score method applied here consist in fusing the speaker and face systems using linear weighted summation, where the weights of each classifier are learned using the minimum classification error principle on a training set, trying to optimize the equal error rate of false acceptances and false rejections under the user verification scenario. Given that this work uses two sensors only one additional parameter (the ratio of the weights of the classifiers) needs to be learned. A simple brute force sampling of the parameter space is used for this MCE training.

Authors in [6] use the speech and face modalities together to measure audio-visual synchrony for identification of in talking faces, the authors improved biometric verification performance by fusing the speech and face recognition systems using a SVM (support vector machine). Talking faces not only contain voice and image but also a third source of information: synchronization between the two.

In [9] the authors addressed the problem of score level fusion of intramodal and multimodal experts, focusing on confidence-based fusion controlled by biometric data quality. They used as features not only quality measures but also the cross terms obtained by taking the product of score and quality to generalise the fusion feature space. The study showed that the use of quality weighted scores as features in the definition of the fusion functions leads to improved performance and demonstrated that the achievable performance gain is also affected by the choice of fusion architecture.

In [10] the authors developed score-level multi-modal fusion algorithms based on predictive quality metrics and employed them for face and fingerprint biometric fusion. The causal relationships (like the fact that the match score of a gallery-probe fingerprint image pair is affected by the image qualities of the gallery and probe fingerprint images and the state of match) in the context of each fusion scenario are modelled by a probabilistic framework. The recognition/verification decision is made through probabilistic inference.

We consider the parameters:

$f_{g,a}$ A quality-related image feature vector of a gallery facial image

$f_{p,a}$ A quality-related image feature vector of a probe facial image

$q_{g,a}$ Image quality for a gallery facial image generated from $f_{g,a}$.

$q_{p,a}$ Image quality for a probe facial image generated from $f_{p,a}$.

$q_{g,i}$ Image quality for a gallery fingerprint image.

$q_{p,i}$ Image quality for a probe fingerprint image.

s_a Match score for a gallery-probe facial image pair

s_i Match score for a gallery-probe fingerprint image pair.

Once those measurement are observed, we can perform the quality-based face and fingerprint score-level fusion through probabilistic inference, the decision for match or no-match can be made by maximizing the probability of match given the values of our parameters, with $q_{g,a}$ and $q_{p,a}$ derived from the input parameters.

$$match = arg \max_{match} p(match | f_{g,a}, f_{p,a}, q_{g,i}, q_{p,i}, s_a, s_i)$$

This method has been showed to improve the verification performance over the methods based on the raw match score of a single modality.

In [11] the authors have developed a unified probabilistic framework for quality-based face recognition decisions, where the quality assessments of facial images are integrated into face recognition. The proposed algorithm significantly improves face recognition performance over a wide range of facial image quality.

Let f_g be a feature vector containing some image features such as shape coefficients of a statistical facial shape model. Let q_g be an assessment of image quality, for a gallery image. Similarly, f_p and q_p are the corresponding feature vector and the quality assessment for a probe image, respectively. Let s_{gp} be the match score for a probe/gallery image pair obtained by some face matching algorithm.

Once f_g, f_p, and s_{gp} are observed, we can perform the quality-based recognition through probabilistic inference. The match or no-match decision will be made by maximizing the joint probability of a match, q_g, and q_p given the three measurements, as follows:

$$match = arg \max_{match, q_g, q_p} p(match, q_g, q_p \,|f_g, f_p, s_{gp})$$

Which can be factorized as follows:

$$\begin{aligned} &p(match, q_g, q_p \,|f_g, f_p, s_{gp}) \\ &\quad = c * p(f_g) * p(q_g|f_g) * p(f_g) * p(f_g) * p(q_g|f_p) \\ &\quad * \mathrm{p(match)} \times \mathrm{p(s_{gp}|q_g, q_p, match)} \end{aligned}$$

Where c is a normalization factor. The proposed algorithm significantly improves face recognition performance over a wide range of facial image quality.

In [12] the authors demonstrated the benefit of fusing the voice and face modalities in scenarios where both the face and voice data suffer from extensive degradations. Several fusion rules were tested:

1) In the first case an exponential weighting factor in the product rule was introduced. This weighting factor is inversely proportional to the squared difference between face and voice scores. This mechanism tries to assign a lower weight to the fused scores in cases where the mutual confidence of the modalities is low.

2) In the second case quality values for face images and audio data along with their corresponding match scores were used, in a weighted sum rule scheme.

3) In the third case the quality values for face images and audio data along with their corresponding match scores were used, in a weighted product rule scheme.

4) In the fourth case the quality values for face images and audio data along with their corresponding match scores were introduced.

In [20] a score level-based strategy for a biometric framework with multiple cues, based on S-sums has been exhibited, S-sums were introduced by Silvert in 1979 and they are a class of binary functions that are used as a rule of combination for fuzzy sets.

An S-sum is a function S: [0,1] × [0,1] → [0,1] such that:

S (0, 0) = 0.

S (1, 1) = 1.

S is commutative.

S is increasing with respect to the two variables.

S is continuous.

S is self-dual ∀ x, y ∈ [x, y] ^ 2, S(x, y) = 1 – s(1 − x, 1 – y)

The general form is given by the formula:

$$S(x, y) = \frac{g(x, y)}{g(x, y) + g(1 - x, \quad 1 - y)},$$

here g is a continuous, positive, increasing function of [0, 1] × [0, 1] into [0, 1] , such that g(0, 0) = 0. Continuous t-norm or t-conorm can be chosen as *g*.

The two normalized matching scores will be then fused using the S-sums and the resulting score will be evaluated.

To get illicit access to a secured system submission of whipped, artificially generated, or unoriginal cue to sensor is known as spoof attack. Johnson, Tan & Schuckers in [13] suggested fusion of face, iris, and fingerprint at score level, to achieve lesser vulnerability of proposed system against spoof attacks. After the scores from each modality have been fused, giving a combined score in the range of zero to one, a threshold is implemented to make a final accept or reject decision. By varying this threshold, a performance curve known as a Detection Error Trade off (DET), can show the relationship between the false reject rate (FRR) and false accept rate (FAR). In the proposed solution the percentage of false accepts given that one or more of the modalities have been successfully spoofed is introduced as a parameter called the spoof false accept rate (SFAR). The paper proposes a method for determining, after a system assessment based on SFAR, a calculated adjustment of the operating point to ensure for a more secure system, at a cost of decreased FRR performance. It also quantitatively demonstrates how to assess the tradeoff.

Akhtar in [14] introduced serial mode of fusion to fuse fingerprint and face to achieve lesser vulnerability against spoof attacks. Comparing the results of serial mode fusion with parallel mode of fusion on two benchmark datasets have showed that serial mode of fusion is less vulnerable to spoof attacks.

Gomez Barrero in [15] suggested fusion of scores of two cues i.e., face and iris estimating the strength and speed of attack in their proposed work, validating their work on Biosecure database with EER = 0.83%.

Gupta, Walia & Sharma in [16] suggested fusion of iris, face, and fingerprint at score level to make system more robust against spoof attacks, validating everything on chimeric dataset.

Sujatha & Chilambuchelvan in [17] proposed a multimodal biometric framework that fuses the palm-print, face, iris, and signature, all was authenticated on the CASIA dataset.

S. F. Ali [18] presented a wide-ranging survey of liveness detection, spoof attacks and fingerprint. They have discussed about various algorithms and datasets for performance evaluation of multimodal framework and revealed that deep learning algorithms are superior solution to these issues.

2.3 Challenges to multimodal biometric system

When developing multimodal biometric systems several challenges present itself that must be taken into consideration and addressed:

(1) **Availability of effective sensors**: Availability of effective sensors to acquire the images irrespective of type of illumination in indoor or outdoor environments is a requirement of a multimodal biometric framework, the sensor should be fast and efficient to capture images from a distance.

(2) **Availability of appropriate database**: Numerous multimodal datasets are available either free or at a nominal cost. Selection of a well-designed dataset which was acquired while following the protocol standards may result in the expedite of the research work, but a poor dataset may result in wastage of energy, money and time while evaluating on it.

(3) **Selection of efficient fusion scheme**: The biometric traits can be fused by various types of levels of fusion: score level fusion, feature level fusion, decision level fusion, rank level fusion etc. and there are various fusion techniques for each level of fusion level. While designing a particular multimodal biometric system the selection of an appropriate level of fusion and fusion schemes plays a vital role in the performance of a multimodal biometric system. It is a challenge to decide how and when to fuse the biometric traits for designing an effective multimodal biometric system.

(4) **Privacy issues**: In biometric frameworks the personal details of individuals are stored in a database so it possible that intruders can make misuse of this data, and this will result in privacy issues, when developing a biometric system, the security of the templates must be kept in mind so that privacy issues can be mitigated.

(5) **Cost-effective system**: there is a trade-off between cost and the performance so in designing a robust, efficient, and cost-effective system this must be kept in mind.

Those challenges could be addressed in several ways:

Regarding the sensors, development of more precise and cost-effective sensors could help. Moreover, such sensors should be able to acquire samples of multiple traits in one go. In addition to that, collecting more realistic multimodal databases will always be nicer, as they can facilitate the genuine evaluation of multimodal systems.

Further, exploration and development of efficient fusion schemes are desirable, and may be achieved through learning-based fusion schemes. These schemes can learn the relationship between features and/or scores in a more representative manner, leading to superior results.

The privacy related concerns are solvable through several applications, one of which is cancellable biometrics, where the biometric templates of the subjects can be revoked in case of potential threats.

2.4 Use-cases and Future directions

In this section we cover some practical use-cases in which multimodal biometric systems are employed.

(1) **Aadhar Card**: The most prevalent example of real-time use-case of biometrics is Aadhar Card. It is a national identification card, which compromises of a typical twelve digits number for each person, allotted for example by UIDAI (Unique Identification Authority of India), It is based upon biometric traits such as iris, face, and fingerprint. To get Aadhar card an individual must provide the biographical information such as name, gender, date of birth, address, and biometric information such as fingerprint, iris, and facial image. Then to check the uniqueness of this information, it is forwarded to the Central Identities Data Repository (CIDR), where de-duplication of the information is done. After these steps a 12-digit number with lifelong identification capability is allotted to a user.

(2) **Border Control**: Another important use of multimodal biometrics is border control, which is the entirety of measures taken by a state to monitor the activities on its border and the movement of people. Modern technology has also paved way to ensure border security in the most effective way through the introduction of biometrics. Biometrics are highly effective in border management as they use biologically unique traits like face and fingerprint to identify individuals trying to enter a country. E-passport also plays an important role in border management and consist of a digital passport which has a chip embedded in it. The embedded chip holds the same data as the data page of traditional passport as well as additional biometric information. Saudi Arabia adopted this technique in border security, adopting an automated fingerprint identification system. The Saudi e-ID card is also a biometric card and is also efficient in physical and electronic identity verification. Saudi Arabia is also planning to deploy iris in addition to fingerprint to increase border security.

(3) **Law enforcement**: The use of Biometrics in law enforcement is rising, common modalities like fingerprints, face, voice & and iris are gaining attention. The use of biometrics in this sector started over 125 years ago thanks to an Argentinian criminologist. Today several countries like the United States, UK, Japan have adopted biometrics for its law enforcement. The US law enforcement agency: the FBI, has facial recognition records of more than 117 million Americans. The United Kingdom's Welsh police is planning to adopt facial recognition to detect criminals and decrease crime rate, Japan is also planning to implement facial recognition at its airports to prevent terrorists from entering in the country.

Multi modal biometric systems can develop in several various directions, among those several important fields are:

(1) Explore the possibilities of further deploying deep convolutional neural networks (CNNs) in the problem of multimodal biometrics, as CNNs have proven to be very efficient in solving various computer vision and image classification problems.

(2) Conduct real-life case-studies analysing the vulnerability of multimodal systems, as it is still not known how the real-life multimodal system behaves in response to more sophisticated spoof attacks.

(3) The feasibility of using multimodal biometrics in smartphones is an important area of investigation. Given the limited computational resources available on smartphones, devising computational-efficient multimodal systems is challenging.

(4) Discovering the optimized blend of biometric traits is still a research question to be answered. Which biometric traits will yield outperforming results when fused together, and which fusion strategy would be advantageous.

Up to now we have introduced different characteristics and design choices of biometric systems and multimodal biometric systems, all those characteristics must be considered when designing a new system:

-Type of system

-Work mode

-Choice and number of biometric indicators.

-Fusion Level

-Representation (incompatibility & unavailability of features).

-Matching score

-Decision methodology.

-Fusion methodology.

-Eventual weights of individual biometric.

-Cost versus performance trade-off.

Chapter 3

Implementation

In this chapter we will describe all phases of the process of implementation of our multi modal biometric system, and we will understand the choices made, the reasoning behind those choices and the eventual problems and solutions adopted to overcome those problems.

3.1 The general model

In our multimodal biometric system, we decided to use three individual biometric recognition methods: fingerprint recognition, face recognition and voice recognition. Those methods won't be all employed concurrently: in the first phase we ask the user for face and fingerprint recognition and consider both for recognition according to the parallel mode explained above, in the case we cannot reach a definitive solution we ask also for voice recognition and consider all three scores, this was the results of several considerations:

1) If we assume that the system will be also used in a public place then using the voice could lead to problems such as external noise or possible distortion due to wearing mask in the pandemic situation.

2) Face is a trait that everyone has while fingerprint and voice recognition aren't universally applicable.

3) Fingerprint could seem not hygienic in the pandemic situation, but many touchless fingerprint identification systems already exist, instead voice and face recognition could lead people take off their masks in places where it's not safe.

4) Face recognition systems with mask already exists and could even be used to enforce it in places where its necessary (such as covid-hospitals).

5) Fingerprint and face have higher permanence than voice.

6) Fingerprint recognition, when possible, have higher security and accuracy than the other two.

Our overall system will follow the general schema explained below, in the case the user is not authenticated after the voice recognition a supervisor intervention is required.

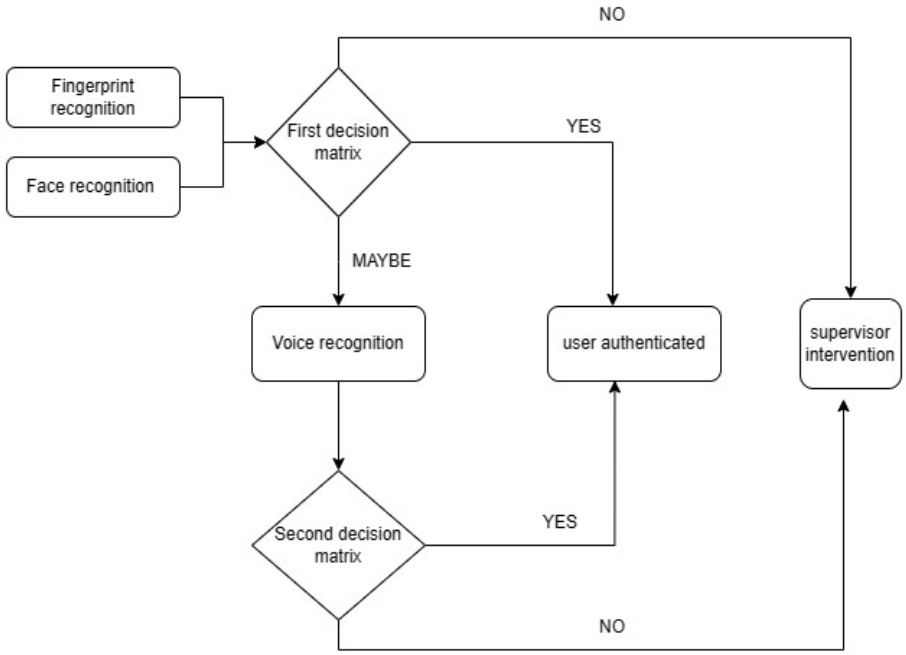

Figure 3.1: Schema of the general model.

For the fusion technique we employ score level fusion, we decided to use this method because of its simplicity and because it was well suited with the idea of using external systems for calculating the scores of single characteristics.

3.2 Decision Matrix

In our system the matching modules are responsible for the matching and normalization phases of the various single biometric systems, to evaluate the outcome we will need the scores returned by them and the values **noValue** and **yesValue** which are custom floating-point values between 0 and 1 set during the deployment of a particular instance of the system and based on the level of security required.

Each score corresponds to a confidence interval: if the score is smaller than noValue then it's in the rejection interval indicated with no, if it's bigger than yesValue than it's in the acceptance interval indicated with yes, if it's between noValue and yesValue both included than it's in the uncertainty interval indicated with maybe.

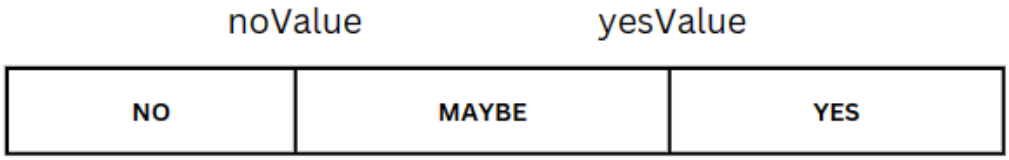

Figure 3.2: individual scores and confidence intervals.

In the first phase the scores derived from the fingerprint recognition and the face recognition decide the outcome based on the table below which link different interval values of the two individual systems with the combined corresponding outcome.

		Fingerprint recognition		
		yes	**maybe**	**no**
Face recognition	**yes**	accepted	accepted	Ask voice recognition
	maybe	accepted	Ask voice recognition	Ask voice recognition
	no	Ask voice recognition	Ask voice recognition	rejected

Figure 3.3: Face and Fingerprint recognition combined decision matrix.

In the case we ask the voice recognition all the three individual scores will be mapped in the interval values (*yes/maybe/no*) and the final decision will be based on the following matrix.

		voice recognition		
		yes	**maybe**	**no**
Face recognition	**maybe+no**	rejected	rejected	rejected
+	**maybe+maybe**	accepted	rejected	rejected
Fingerprint recognition	**no+yes**	accepted	rejected	rejected

Figure 3.3: Face and Fingerprint and voice recognition combined decision matrix.

For example, if the voice recognition score is high enough to be mapped into a yes value while the face recognition is mapped into a no value and

the fingerprint recognition into a maybe then the user is rejected. Is worth noting that for face and fingerprint recognition the combinations *yes+maybe* and *yes+yes* are not considered a because they already lead to acceptance in the first matrix, also fingerprint and face recognition have the same "weight" in the decision so only three are the possible combinations: *maybe+no, maybe+maybe, no+yes*.

3.3 Score modules outputs and decision module

In the first phase the individual matching modules of fingerprint and face recognition return an unordered array containing the relevant matches that have a similarity score bigger than the minimum value noValue to the decision module with their respective score.

Entity1	Score1
Entity2	Score2
Entity3	Score3
Entity4	Score4
Entity5	Score5

Fig 3.4: Array returned by the score module.

Then the arrays from the two biometric systems are combined in one single array, given that the individual modules could report different entities the final array score is their combination in the form of set union where every lacking score is requested.

If for example module A report entity 1 and entity 4 while module B report entity 4 and entity 2 the final score array will take into consideration entity 1,2 and 4 and so the decision module will ask the scoring module to report the missing scores for evaluation.

Entity1	Fingerprint_Score1
Entity4	Fingerprint_Score4

U

Entity2	Face_Score2
Entity4	Face_Score4

=

Entity1	Fingerprint_Score1	Face_Score1
Entity2	Fingerprint_Score2	Face_Score2
Entity4	Fingerprint_Score4	Face_Score4

Fig 3.5: Set union between the two arrays.

After that a final combined score between the two individual scores is calculated.

$$combined\_score\_entity\_n = (face\_weight * face\_score\_entity\_n + fingerprint\_weight * fingerprint\_score\_entity\_n)$$

The meaning of the weight is to be decided, a possibility could be the fact that many matching scores are found with a high value (that could even be an attack). So in this case (as an example) I could calculate the weight as function of n (number of returned entities) divided the number of scores that are bigger than the parameter ε (which represent the minimum value for an entity to be considered), Then we create an array having in each line the entity the combined score and the independent scores, ordered by the value of the combined score from highest to lowest.

Entity1	Fingerprint_Score1	Face_Score1	Combined_Score1
Entity2	Fingerprint_Score2	Face_Score2	Combined_Score2
Entity4	Fingerprint_Score4	Face_Score4	Combined_Score4

Fig 3.6: Combined array.

Then the single entities will be evaluated following the array order according to the first matrix described before, three cases can occur:

- In case the entity falls into an accepted cell then the user is accepted, and the program stop.
- In case the entity falls into a rejected cell then the next entity in the array is evaluated, if all entities are rejected then the user is rejected.
- In case the entity falls into a matrix cell which require voice recognition then the entity is put into another array of uncertain entities and the evaluation proceed.

If I arrived at the end of the array and no entity was accepted but the uncertain array is not empty, then I move to the second phase: for every entity in the uncertain array the voice recognition score is requested, and a new combined score will be created considering all tree individual systems:

$$final\_combined\_score\_entity\_n = (face\_weight * face\_score\_entity\_n + fingerprint\_weight * fingerprint\_score\_entity\_n + voice\_weight * voice\_score\_entity\_n)$$

Then a new array is created where in each line the three individual scores of the systems and the final combined scores are present and all its ordered based on this combined score from the highest value to lowest value.

Entity1	Fingerprint_Score1	Face_Score1	Voice_Score1	Final_Combined_Score1
Entity2	Fingerprint_Score2	Face_Score2	Voice_Score2	Final_Combined_Score2
Entity3	Fingerprint_Score3	Face_Score3	Voice_Score3	Final_Combined_Score3

Fig 3.7: Final combined array.

In the end each individual entity in the array is evaluated according to the second matrix following the order given by the combined score, if an entity fell into the accepted cell then the user is authenticated and the program end, if it fall into the rejected cell then I move to the next entity; if all entities are rejected then the user is rejected, in this final case manual intervention is necessary for example from a supervisor.

3.4 Individual systems used

For building the overall multi modal structure three open-source sub system have been used:

SourceAFIS is the system used for recognizing human fingerprints. Its algorithm is the result of independent design by the developers. It doesn't copy some textbook algorithm, but it does however borrow heavily from other opensource fingerprint matchers. The algorithm is capable of delivering decent accuracy and surprisingly high matching speed; it can compare two fingerprints 1:1 or search a large database 1:N for matching fingerprint. It takes fingerprint images on input and produces similarity score on output.

The algorithm is based on high-level abstractions like minutiae, ridge endings and bifurcations. Minutiae are what is saved in the template, and they consist of points on the image with associated direction angle.

Fig 3.8: Minutiae found by the algorithm in the fingerprint image [3].

After the first step, another abstraction occurs to produce edges and angles: an Edge is a line connecting two minutiae, edges are composed by a length and two angles inherited from its minutiae. Edge angles are expressed as relative to the edge. These three properties of the edge (length and two relative angles) do not change when the edge is moved or rotated so they will always be the same in different images from different angles of the same fingerprint, that's why they are so important for matching.

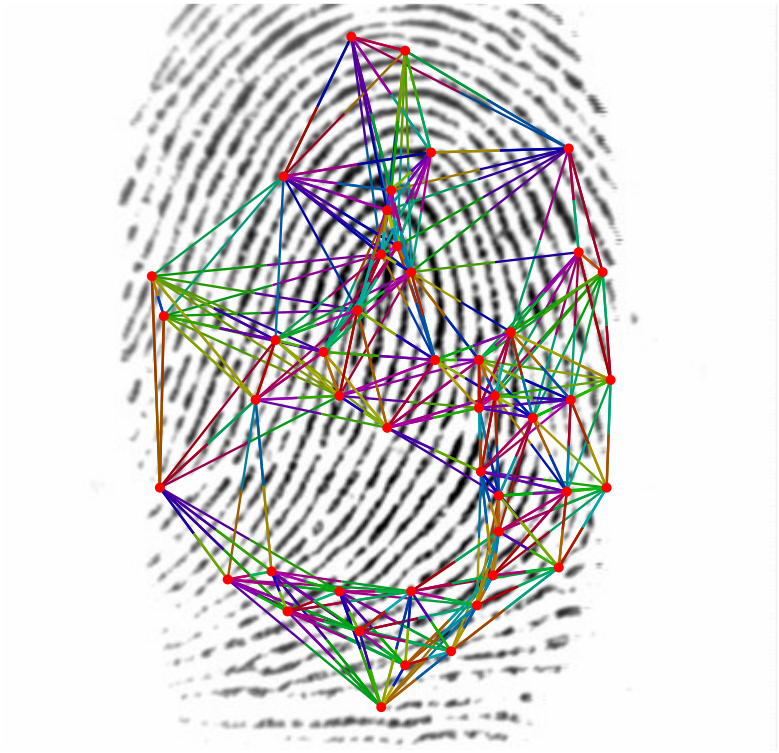

Fig 3.9: Edges and Angles found by the algorithm in the fingerprint image [3].

SourceAFIS's algorithm tries to find at least one edge shared by the two fingerprints being matched. This is done using a nearest neighbour algorithm, so a root pair is found, which is the initial pair of matched minutiae, one from each fingerprint.

Starting from the root pair, the algorithm crawls edges outwards and builds a pairing consisting of several paired minutiae and paired edges.

Fig 3.10: Matching from the root minutiae in blue with the pairing edges in green [3].

Then the scoring happens: the idea is that every paired minutia or edge is an event that is unlikely to happen randomly, starting form that assumption, the algorithm will try to decide is such pairings means it's the same person or it's just a coincidence. Generally speaking, the more of such unlikely events there are, the more likely the pairing is to represent the same finger. So, the algorithm essentially counts various matched features and scores them on how closely they match. Final sum of the partial scores is shaped to align to some scale and returned from the algorithm.

From this point on our code intervene:

We created a **getFingerprintScore** function that takes in input the float noValue explained before and the name of the image file which represent the user fingerprint. It returns an array containing the id and the normalized score of all the entities in our dataset which have a score bigger than noValue.

First the image identified by id is retrieved and a template is created, then an array containing all the names (which corresponds to the id) of the images in the dataset is created.

```
public static Results[] getFingerprintScore(float noValue, String id) {
   // array of objects results which are composed by two fields: id and probability
   Results[] resultsArray = new Results[0];

   try {

      // i use the image labeled as id as the input from the user
      var encoded = Files.readAllBytes(Paths.get(
            "C:/Users/marco/Desktop/fingerprint/testImages/" + id + ".tif"));
      var decoded = new FingerprintImage(encoded);

      // create a template
      var template = new FingerprintTemplate(decoded);

      // Creating a File object for the directory path containing
      // all the entities in the dataset
      File directoryPath = new File(
            "C:/Users/marco/Desktop/fingerprint/finger-images");
      // List of all files
      String contents[] = directoryPath.list();

      // here the file must not have . in their name
      // i split the file name and take only the part without the extension
      String[] idArray = new String[contents.length];
      for (int i = 0; i < contents.length; i++) {
         String[] res = contents[i].split("[.]", 0);
         idArray[i] = res[0];

      }
```

Finally, the images are confronted with the fingerprint in input and an array containing the pairs of id and similarity score for each image is created, the scores are then normalized and an array containing only the pairs with score bigger than noValue is returned in output.

```
        Results[] similarityArray = new Results[contents.length];
        for (int i = 0; i < contents.length; i++) {
            var candidate = new FingerprintTemplate(new FingerprintImage(Files.readAllBytes(Paths
                    .get("C:/Users/marco/Desktop/fingerprint/finger-images/" + contents[i]))));

            // match the two templates
            var matcher = new FingerprintMatcher(template);
            similarityArray[i] = new Results();
            similarityArray[i].probability = (float) matcher.match(candidate);
            similarityArray[i].id = idArray[i];
        }

        // to keep the position in the array
        int j = 0;

        for (int i = 0; i < contents.length; i++) {

            float normalizedScore = normalize(similarityArray[i].probability, min(similarityArray),
                    max(similarityArray));

            if (normalizedScore >= noValue) {

                resultsArray = resize(resultsArray);
                resultsArray[j] = new Results();
                resultsArray[j].probability = normalizedScore;
                resultsArray[j].id = similarityArray[i].id;
                j++;
            }
        }
    } catch (IOException ioe) {
        ioe.printStackTrace();
    }

    return resultsArray;

}
```

Exadel CompreFace is a free and open-source face recognition GitHub project. It's a docker-based application that can be used as a standalone server or deployed in the cloud. The system provides REST API for a variety of tasks like face recognition, face verification, face detection, landmark detection, mask detection, head pose detection, age, and gender recognition services.

CompreFace supports different models that work on CPU and GPU and is based on state-of-the-art methods and libraries like FaceNet and InsightFace.

Interactions with CompreFace will be managed with curl which is a tool for transferring data from or to a server. Curl commands will be sent to the application to interact with it, for example if we want to compare faces from the uploaded images with the face in saved image ID:

```
curl -X POST
"http://localhost:8000/api/v1/recognition/faces/<image_id>/verify?limit=<
limit>&det_prob_threshold=<det_prob_threshold>&face_plugins=<face_plugins
>&status=<status>" \
-H "Content-Type: multipart/form-data" \
-H "x-api-key: <service_api_key>" \
-F file=<local_file>
```

In output it will return a json response that must be processed:


```
{
  "result" : [ {
    "age" : {
      "probability": 0.9308982491493225,
      "high": 32,
      "low": 25
    },
    "gender" : {
      "probability": 0.9898611307144165,
      "value": "female"
    },
    "mask" : {
      "probability": 0.9999470710754395,
      "value": "without_mask"
    },
    "embedding" : [ 9.424854069948196E-4, "...", -0.011415496468544006 ],
    "box" : {
      "probability" : 1.0,
      "x_max" : 1420,
      "y_max" : 1368,
      "x_min" : 548,
      "y_min" : 295
    },
    "landmarks" : [ [ 814, 713 ], [ 1104, 829 ], [ 832, 937 ], [ 704, 1030 ], [ 1017, 1133 ] ],
    "subjects" : [ {
      "similarity" : 0.97858,
      "subject" : "subject1"
    } ],
    "execution_time" : {
      "age" : 28.0,
      "gender" : 26.0,
      "detector" : 117.0,
      "calculator" : 45.0,
      "mask": 36.0
    }
  } ],
  "plugins_versions" : {
    "age" : "agegender.AgeDetector",
    "gender" : "agegender.GenderDetector",
    "detector" : "facenet.FaceDetector",
    "calculator" : "facenet.Calculator",
    "mask": "facemask.MaskDetector"
  }
}
```

Two methods have been created for facing the issue of retrieving and processing the scores:

getFaceScore takes in input noValue and the id of the image we want to confront with the images in the database, ask using a curl command a response from the application and then process the response received giving as output an array of couples of type [id , probability], where id identify the user in the database, and probability represent a normalized similarity with the image in input.

```
public static Results[] getFaceScore(float noValue, String id) throws IOException
{
    String s1 = "curl -X POST http://localhost:8000/api/v1/recognition"
    +"/recognize?limit=1&prediction_count=999&det_prob_threshold=0.0";
    String s3 = "&status=true \\-H \"Content-Type: multipart/form-data\""
    +" \\ -H \"x-api-key: f9af2de0-c74f-4f6b-aeb9-7db1fa50be18\" \\ -F file=@C:/";
    String s4 = id;
    String s5 = ".jpg";
    String command = s1 + s3 + s4 + s5;

    Process process = Runtime.getRuntime().exec(command);
    Results[] resultsArray = null;
    int i = 0;
    BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(process.getInputStream()), 1);
    String line = null;
    String tempid = null;
```

```
    while ((line = bufferedReader.readLine()) != null) {
        StringTokenizer stringTokenizer = new StringTokenizer(line);
        // System.out.println(line);
        while (stringTokenizer.hasMoreTokens()) {
            String token = stringTokenizer.nextToken();
            if (token.equals("\"subject\"")) {
                stringTokenizer.nextToken();
                stringTokenizer.nextToken("\"");
                tempid = stringTokenizer.nextToken("\"");
            }
            if (token.equals("\"similarity\"")) {
                stringTokenizer.nextToken();
                float probability = Float.valueOf(stringTokenizer.nextToken(","));
                if (probability > noValue) {
                    resultsArray = resize(resultsArray);
                    resultsArray[i] = new Results();
                    resultsArray[i].probability = probability;
                    if (tempid == null) {
                        System.out.println("error in the curl response");
                        System.exit(1);
                    }
                    resultsArray[i].id = tempid;
                    System.out.println(resultsArray[i].id);
                    System.out.println(resultsArray[i].probability);
                    i++;
                    tempid = null;
                }
            }
        }
    }
    bufferedReader.close();
    return resultsArray;

}
```

getEntityScore takes in input the id of the image from the user (idTest) and the id of the image in the database we want to confront it with (id), ask using a curl command a response from the application and then process the input received giving as output the similarity score between the two.

```
public static float getFaceEntityScore(String id, String idTest) throws IOException {
    System.out.println(idTest);
    System.out.println("curl id required " + id);
    int i = 0;
    String[] imageID = new String[5];

    String s1 = "curl -X GET http://localhost:8000/api/v1/recognition/faces?subject=";
    String s2 = id;
    String s3 = " \\ -H \"x-api-key: f9af2de0-c74f-4f6b-aeb9-7db1fa50be18\" \\";
    String command = s1 + s2 + s3;

    Process process = Runtime.getRuntime().exec(command);

    BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(process.getInputStream()), 1);
    String line;

    while ((line = bufferedReader.readLine()) != null) {

        StringTokenizer stringTokenizer = new StringTokenizer(line);

        while (stringTokenizer.hasMoreTokens()) {
            String token = stringTokenizer.nextToken();

            if (token.equals("\"image_id\"")) {
                stringTokenizer.nextToken("\"");
                if (i + 1 == imageID.length)
                    imageID = Arrays.copyOf(imageID, i * 2);
                imageID[i] = stringTokenizer.nextToken("\"");
                System.out.println("the image id is: " + imageID[i]);
                i++;

            }

        }

    }
    bufferedReader.close();
```

```
        BufferedReader bufferedReader2 = null;
        int j = 0;

        float avarageSimilarity = 0;
        while (imageID[j] != null) {

            String a1 = "curl -X POST http://localhost:8000/api/v1/recognition/faces/";
            String a2 = imageID[j];
            String a3 = "/verify?limit=1&det_prob_threshold=0&status=false \\ ";
            String a4 = "-H \"Content-Type: multipart/form-data\" \\ ";
            String a5 = "-H \"x-api-key: f9af2de0-c74f-4f6b-aeb9-7db1fa50be18\" \\ ";
            String a6 = "-F file=@C:/";
            String a7 = idTest;
            String a8 = ".jpg";
            String command2 = a1 + a2 + a3 + a4 + a5 + a6 + a7 + a8;

            Process process2 = Runtime.getRuntime().exec(command2);

            bufferedReader2 = new BufferedReader(new InputStreamReader(process2.getInputStream()), 1);
            line = null;
            while ((line = bufferedReader2.readLine()) != null) {

                StringTokenizer stringTokenizer = new StringTokenizer(line);

                while (stringTokenizer.hasMoreTokens()) {
                    String token = stringTokenizer.nextToken();

                    if (token.equals("\"similarity\"")) {
                        stringTokenizer.nextToken();
                        float similarity = Float.valueOf(stringTokenizer.nextToken(","));
                        if (similarity < 0)
                            similarity = 0;
                        avarageSimilarity += similarity;
                    }

                }
```

```
            }
            j++;
        }
        if (bufferedReader2 != null)
            bufferedReader2.close();

        // if i found nothing
        if (j == 0) {
            System.out.println("the face database it's incomplete");
            System.exit(1);
        }
        System.out.println("the value returned by curl getFaceEntityScore is: " +
                avarageSimilarity / j);
        return avarageSimilarity / j;

    }
```

SpeechBrain is an open-source and all-in-one speech toolkit based on PyTorch which is a machine learning framework based on the Torch library, used for applications such as computer vision and natural language processing.

SpeechBrain provides different models for speaker recognition, identification, and diarization on different datasets:

- State-of-the-art performance on speaker recognition and diarization based on ECAPA-TDNN models.
- Original Xvectors implementation with PLDA.
- Spectral clustering for speaker diarization
- Libraries to extract speaker embeddings with a pre-trained model on your data.

To get the data two files are used, one in Java and one in Python:

In the Java file the method **getVoiceEntityScore** take in input the id of the two voices, then call the Python program passing as command line arguments the two ids, then process the output received and return in the similarity score between the two.

```java
package com.machinezoo.sourceafis;

import java.io.*;
import java.util.StringTokenizer;

public class SpeakerRecognition {

    public static float getVoiceEntityScore(String id1, String id) throws Exception {

        float probability = 0;
        ProcessBuilder processBuilder = new ProcessBuilder("py",
                "C:/Users/marco/Desktop/fingerprint/sourceafis-java/temp.py", id + " " + id1);
        processBuilder.redirectErrorStream(true);
        Process process = processBuilder.start();
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(process.getInputStream()), 1);
        String line;
        while ((line = bufferedReader.readLine()) != null && !line.equals("START")) {


        }
        while ((line = bufferedReader.readLine()) != null) {

            StringTokenizer stringTokenizer = new StringTokenizer(line);
            while (stringTokenizer.hasMoreTokens()) {

                stringTokenizer.nextToken("[");
                String similarity = stringTokenizer.nextToken("[");
                StringTokenizer stringTokenizerSub = new StringTokenizer(similarity);
                probability = Float.valueOf(stringTokenizerSub.nextToken("]"));

            }
        }
        bufferedReader.close();
        return probability;
    }
}
```

In the Python file first, I get the two ids from the command line, then using the method **verify_files** provided by speechbrain I get the similarity score and I print it in output so to pass the data to the Java file.

```
import os
import speechbrain as sb
from IPython.display import Audio
from speechbrain.dataio.dataio import read_audio
from speechbrain.pretrained import SpeakerRecognition
import argparse
parser = argparse.ArgumentParser()
parser.add_argument('files')
files = parser.parse_args()
namespaceString=str(files)
files=namespaceString.split("'")[1]
verification = SpeakerRecognition.from_hparams(source="speechbrain/spkrec-ecapa-voxceleb",
 savedir="pretrained_models/spkrec-ecapa-voxceleb")
file1=files.split(" ")[0]
file2=files.split(" ")[1]
print("START")
score="tensor([0])"
if file1==27 or file1== "27":
    print("tensor([0])")
else:
    score, prediction = verification.verify_files("C:/Users/marco/Desktop/fingerprint/sourceafis-java/test voices/original"
    +file2+".wav","C:/Users/marco/Desktop/fingerprint/sourceafis-java/test voices/"+file1+".wav")
    if score<0:
       score="tensor([0])"
    print(score)
```

3.5 Main algorithm

The main algorithm is divided in a main function: **decision** and several auxiliary functions:

The decision function takes in input a string which identify the user (all the user files are named with that string) and returns in output an object of class outcome, which is composed of a string representing the id of the verified subject and the probability of that subject corresponding to the user.

```
package com.machinezoo.sourceafis;

public class Outcome {
    public Outcome(String id, float subjectProbability) {
        this.verifiedId = id;
        this.probability = subjectProbability;

    }

    public String verifiedId;
    public float probability;
}
```

First the arrays of corresponding fingerprints and faces with their respective probabilities are retrieved with the use of the individual functions explained before.

```
import java.io.*;

import javax.crypto.NullCipher;

import com.machinezoo.sourceafis.Curl;

public class DecisionModule {

    final static float n = 5; // this has to be changed and chosen statistically

    public static String id;

    public Outcome decision(String identifier) throws IOException {

        id = identifier;
        // those must be custom
        float noValue = 0.3f; // represent the value under it's considered a no in
        // the evaluation matrix
        float yesValue = 0.6f; // represent the value above it's considered a yes in
        // the evaluation matrix

        // the methods populate the arrays with ALL the entities that have more than
        // noValue as score
        Results[] fingerprintArray = FingerprintOutput.getFingerprintScore(noValue,
                id);
        Results[] faceArray = Curl.getFaceScore(noValue, id);
        System.out.println("got face and id entity scores");
        if (faceArray == null || fingerprintArray == null) {
            if (faceArray == null)
                faceArray = new Results[0];

            else

                fingerprintArray = new Results[0];
        }
```

Then a new combined array is created and ordered using the uniteArray2 function (which also ask for any lacking data) and the scoreArray2 function (which order the data according to the combined score explained before). The elements in the array are then evaluated in order and a second array keeps track of the identities that falls into the uncertainty range with a 1 in the equivalent position of the combined array. If a result is found, then the algorithm stops, and an outcome object is returned.

```
// here i create the first combined array and find the missing entries
Entity2[] combinedArray = new Entity2[fingerprintArray.length +
        faceArray.length];
combinedArray = uniteArray2(fingerprintArray, faceArray);
// here i order the array
EntityScore1[] orderedArray = scoreArray2(combinedArray,
        fingerprintArray.length, faceArray.length);
// can be -1 if it's no, 0 if it's maybe and 1 if it's yes
int evaluation;
// here i keep track of all the uncertain (maybe) entities
// with a 1 in the corresponding maybe position
int[] maybeArray = new int[orderedArray.length];
boolean maybepresent = false;
for (int i = 0; i < orderedArray.length; i++) {
    maybeArray[i] = 0;
    // here i call the evaluation function on the best ranked
    evaluation = evaluate(orderedArray[i], noValue, yesValue);
    // it's a yes
    if (evaluation == 1) {
        System.out.println("identified " + orderedArray[i].id);
        System.out.println("face " + orderedArray[i].faceScore);
        System.out.println("fingerprint " + orderedArray[i].fingerprintScore);
        Outcome out = new Outcome(orderedArray[i].id, orderedArray[i].score);
        return out;
    }
    // it's a maybe then i consider also the voice recognition
    else if (evaluation == 0) {
        maybeArray[i] = 1;
        // use this variable to say that there are some maybe
        maybepresent = true;

    }
}
```

If no entities felt into the maybe range, then the user is rejected and the program stops, otherwise I create an array with only the entities in the first position in the maybe array and ask the voice score for each entity.

```
// if i arrived here it means that i didn't find any match
// so i need to proceed with the voice value
if (maybepresent == false) {
    // then there are no valid scores
    System.out.println("refusal call the supervisor");
    return null;
}
// if I'm here than there are some maybe
Entity3[] maybeEntityArray = new Entity3[orderedArray.length];
// here i create an array with only the maybe entities
int j = 0;
for (int i = 0; i < orderedArray.length; i++) {
    if (maybeArray[i] == 1) {
        maybeEntityArray[j] = new Entity3();
        maybeEntityArray[j].id = orderedArray[i].id;
        System.out.println(orderedArray[i].id + " is in maybe array in position " + j);
        maybeEntityArray[j].faceScore = orderedArray[i].faceScore;
        maybeEntityArray[j].fingerprintScore = orderedArray[i].fingerprintScore;

        try {
            maybeEntityArray[j].voiceScore = SpeakerRecognition.getVoiceEntityScore(orderedArray[i].id,
                    identifier);
            System.out.println("voice score is: " + maybeEntityArray[j].voiceScore);
            j++;
        } catch (Exception ex) {
            System.out.println("Error: exception in the voice module");
            System.exit(1);

        }
    }
}
System.out.println("resize maybe array");
maybeEntityArray = DecisionModule.resize3(maybeEntityArray);
System.out.println("resized maybe array");
int voiceNumber = 0;
```

I then score and order the obtained array and evaluate the entities, if no entity falls into the acceptance range, then the program ends, the user is rejected and a supervisor intervention is necessary, otherwise it's accepted returning the corresponding Outcome object.

```
            try {
                maybeEntityArray[j].voiceScore = SpeakerRecognition.getVoiceEntityScore(orderedArray[i].id,
                        identifier);
                System.out.println("voice score is: " + maybeEntityArray[j].voiceScore);
                j++;
            } catch (Exception ex) {
                System.out.println("Error: exception in the voice module");
                System.exit(1);

            }
        }
    }
    System.out.println("resize maybe array");
    maybeEntityArray = DecisionModule.resize3(maybeEntityArray);
    System.out.println("resized maybe array");
    int voiceNumber = 0;
    try {

        voiceNumber = (faceArray.length + fingerprintArray.length) / 2;
        System.out.println("got the voice number");
    } catch (Exception ex) {
        System.out.println("Error: exception in the voice module");
        System.exit(1);
    }
    Entity3[] finalOrderedArray = scoreArray3(maybeEntityArray, faceArray.length,
            fingerprintArray.length,
            voiceNumber);
    System.out.println("scored the final array");
    System.out.println(finalOrderedArray.length);
    for (int i = 0; i < finalOrderedArray.length; i++) {
        System.out.println(i);
        System.out.println(i);
        evaluation = evaluate2(finalOrderedArray[i], noValue, yesValue);
```

```
            if (evaluation == 1) {
                Outcome out = new Outcome(finalOrderedArray[i].id,
                        finalOrderedArray[i].score);
                System.out.println("identified " + finalOrderedArray[i].id);
                return out;
            }
        }

        // if i arrived here it means that i found no possible matching Entity
        System.out.println("Not recognised call the supervisor");
        return null;
    }
```

The two auxiliary functions **evaluate** and **evaluate2** take in input a candidate and the two values **noValue** and **yesValue** and return an integer representing in which cell of the evaluation matrix the candidate fall: 1 for yes, 0 for maybe and -1 for no. the function evaluate analyse candidates with only fingerprint and face scores while evaluate2 analyse candidates also having the face score.

```
// here i write the evaluation function of the fingerprint and the face only
static int evaluate(EntityScore1 candidate, float noValue, float yesValue) {

    if ((candidate.faceScore > yesValue && candidate.fingerprintScore > noValue)
            || (candidate.fingerprintScore > yesValue && candidate.faceScore > noValue))
        return 1;

    else if (candidate.faceScore < noValue && candidate.fingerprintScore < noValue)
        return -1;

    else
        return 0;

}

// this evaluate when i have also the voice score
static int evaluate2(Entity3 candidate, float noValue, float yesValue) {

    // conclusion not reached
    if (candidate.voiceScore < noValue)
        return -1;

    // conclusion reached
    else if (candidate.voiceScore > yesValue
            && ((candidate.faceScore > yesValue && candidate.fingerprintScore < noValue)
                    || (candidate.fingerprintScore > yesValue && candidate.faceScore < noValue)
                    || (candidate.faceScore > noValue && candidate.fingerprintScore > noValue)))
        return 1;

    // no conclusion
    else
        return 0;

}
```

The uniteArray2 function take in input the fingerprint and face arrays and unite them into a single unordered array were for each entity id correspond a fingerprint and a face score.

```
static Entity2[] uniteArray2(Results[] fingerprintArray, Results[] faceArray) {
    System.out.println(fingerprintArray.length);
    System.out.println(faceArray.length);
    Entity2[] finalArray = new Entity2[0];
    // first I insert all the elements of the fingerprint array in the final array
    for (int i = 0; i < fingerprintArray.length; i++) {
        int j = 0;
        System.out.println(i);
        // if they are the same i will find a j < faceArray.lenght
        while (j < faceArray.length && fingerprintArray[i].id != faceArray[j].id)
            j++;

        if (j == faceArray.length) { // in this case there is no correspondent
            System.out.println("call resize");
            finalArray = resizeEntity2(finalArray);

            finalArray[i].id = fingerprintArray[i].id;
            // get the face score of an entity given a particular id from the scoring
            // module
            try {
                System.out.println("call curl");
                finalArray[i].faceScore = Curl.getFaceEntityScore(finalArray[i].id, id);
                System.out.println(finalArray[i].faceScore);
            } catch (IOException ex) {
                System.out.println("IO exception");
                System.exit(1);
            }
            finalArray[i].fingerprintScore = fingerprintArray[i].probability;
        } else { // i found a correspondence
            System.out.println("found correspondence in the union");
            finalArray = resizeEntity2(finalArray);
            finalArray[i] = new Entity2();
            finalArray[i].id = fingerprintArray[i].id;
            finalArray[i].faceScore = faceArray[j].probability;
            finalArray[i].fingerprintScore = fingerprintArray[i].probability;
        }
    }
```

```
    // to keep the position in the final array
    int k = fingerprintArray.length;
    System.out.println("second phase of the union");
    for (int j = 0; j < faceArray.length; j++) {
        int i = 0;
        while (i < finalArray.length && faceArray[j].id != finalArray[i].id)
            i++;
        // there is no correspondance in the final array so i must insert it
        if (i == fingerprintArray.length) {
            finalArray = resizeEntity2(finalArray);
            finalArray[k] = new Entity2();
            finalArray[k].id = faceArray[j].id;
            System.out.println("face array id " + faceArray[j].id);
            finalArray[k].faceScore = faceArray[j].probability;
            // get the fingerprint score of an entity given a particular id
            finalArray[k].fingerprintScore = FingerprintOutput.getFingerprintEntityScore(finalArray[k].id,
                    id);

            k++;
        }
    }
    return finalArray;
    // end of the function unite array

}
```

The tree entities classes contain each an id and either one single score: fingerprint and face scores or all three possible score fields.

```
// here i write the class for the Entity object
// in this class i have the score of only one parameter and the ID of the
// Entity


class Entity {

    // represent the id of the Entity
    String id;
    // represent the score of the Entity
    float score;

    // constructor
    Entity() {
        id = null;
        score = 0;
    }

}
```

```
    public static Entity2[] resizeEntity2(Entity2[] results) {
        if (results == null) {
            Entity2[] temp = new Entity2[1];
            temp[0] = new Entity2();
            return temp;
        }
        Entity2[] temp = new Entity2[results.length + 1];
        int i = 0;
        for (i = 0; i < results.length; i++) {
            temp[i] = new Entity2();
            temp[i].id = results[i].id;
            temp[i].faceScore = results[i].faceScore;
            temp[i].fingerprintScore = results[i].fingerprintScore;

        }
        temp[i] = new Entity2();
        return temp;
    }
    public static Entity3[] resize3(Entity3[] results) {
        int j = 0;
        while (j < results.length && results[j] != null)
            j++;
        Entity3[] temp = new Entity3[j];
        int i = 0;
        for (i = 0; i < j; i++) {
            temp[i] = new Entity3();
            temp[i].id = results[i].id;
            temp[i].faceScore = results[i].faceScore;
            temp[i].fingerprintScore = results[i].fingerprintScore;
            temp[i].voiceScore = results[i].voiceScore;
        }
        return temp;
    }
}
```

3.6 Data and experimental results

The data for running the final experiments was taken from different sources: the **audio** files came mostly from the **Speaker Recognition Audio Dataset** by Vibhor Jain [21] which is composed by several speaker's audio data with length more than 1 hour for each, data is converted to wav format, 16KHz, and is split into 1min chunks. The dataset was scraped from YouTube and Librivox.

For **face** recognition **CelebFaces Attributes Dataset (CelebA)** [22] was used which is a large-scale face attributes dataset with more than **200K** celebrity images. The images in this dataset cover large pose variations and background clutter. The CelebA dataset include:

- 10,177 identities.
- 202,599 face images.
- 5 landmark locations.
- 40 binary attributes annotations per image.

For **fingerprint** instead several datasets were used: most of the images were taken from UareU [23] a dataset distributed by Neurotechnology which contains 65 fingers with 8 impressions each, in TIFF, 500dpi, 326x357px format, taken by U.are.U 4000 optical sensor by DigitalPersona, the remaining files were taken from several FCV (fingerprint verification competition) events such as FCV2000, they all contain 10 fingers with 8 impressions each, stored in TIFF format.

In the test phase 97 identities were created each composed of one triple fingerprint image, face image and one voice file. Those were then confronted with 100 subjects, each identified by a triple fingerprint image, face image and voice file. Of the 100 subjects 97 corresponded to the identities present in the dataset, which means that the triples were different, but the face images were from the same person, the fingerprint images were from the same finger, and the voice files were from the same voice; while the remaining were not present in the system dataset and should have been rejected.

The tests were done thanks to an auxiliary class, which calls the decision function for each subject, and records the number of false positives, false negatives, true positives, and true negatives. It also records in an array all the wrong identifications and to what identity they were falsely associated. It then displays the results and the content of the array.

```java
public class testfile {
    Run | Debug
    public static void main(String[] args) {
        int truePositive = 0;  int falsePositives = 0;
        int falseNegatives = 0; int trueNegatives = 0;
        DecisionModule dec = new DecisionModule();
        String[] wrongs = new String[101];
        for (int i = 1; i < 101; i++) {
            try {
                Outcome out = dec.decision(Integer.toString(i));
                if (out == null) {
                    wrongs[i] = "" + i;
                    falseNegatives++;
                } else if (out.verifiedId.equals(Integer.toString(i))) {
                    truePositive++;
                } else {
                    falsePositives++;
                    wrongs[i] = out.verifiedId;
                }
            }catch (IOException e) {
                System.out.println("test io exception conversion int to string");
                System.exit(1);
            }
        }
        System.out.println("the number of true positives is: " + truePositive);
        System.out.println("the number of false negatives is: " + falseNegatives);
        System.out.println("the number of false positives is: " + falsePositives);
        System.out.println("the number of true negatives is: " + trueNegatives);
        for (int j = 1; j < 101; j++) {
            if (wrongs[j] != null&&!wrongs[j].equals(j))
                System.out.println(j + " falsely recognised as " + wrongs[j]);
            else if (wrongs[j] != null&&wrongs[j].equals(j))
                System.out.println(j + "not recognised");
        }
    }
}
```

Several tests were conducted using the same data on just the face recognition module and the fingerprint recognition module, first a noValue was set as parameter then all the results who have a higher score than noValue were retrieved, the result with the highest score was considered the response of the system, if no results were retrieved than the system rejected the test user.

noValue	TP	FN	FP	TN
0.3	84	0	16	0
0.4	84	0	16	0
0.5	84	0	16	0
0.6	84	0	15	1
0.7	83	3	13	1
0.8	80	9	9	2
0.9	73	21	4	2

Table 3.1: Face recognition

noValue	TP	FN	FP	TN
0.3	83	0	17	0
0.4	83	0	17	0
0.5	83	0	17	0
0.6	83	0	17	0
0.7	83	0	17	0
0.8	83	0	17	0
0.9	83	0	17	0

Table 3.1: Fingerprint recognition

From these tests we can observe that: face recognition and fingerprint recognition have similar performances in the chosen dataset so a weight to give higher preference to one system over another is not useful in our situation, we so decided to disregard our idea of adding weights in the ordering formula using instead a simple non weighted equation.

On the final multimodal biometric system several tests were then conducted, setting different parameters but using the same tests set explained above, we report here the results:

YesValue	noValue	TP	FN	FP	TN
0.5	0.3	91	4	4	1
0.6	0.3	89	6	4	1
0.4	0.3	92	2	5	1

Table 3.1: Multimodal biometric system tests results

As we can observe from our tests the performances increased significantly from the individual systems, so we managed to demonstrate the effectiveness of our multimodal biometric system and how it outperforms the unimodal systems.

Chapter 4

Conclusions and future works

In this work we introduced biometric systems and their different advantages/disadvantages, we then presented the structure of multi modal biometric systems and the different strategies to build them, introducing the various modules and the fusion methods, and analysing the literature regarding fusion methods.

To take advantage of what we learned during our research we implemented an actual multi modal biometric system using three different open-source software for the single biometric features whose scores were then normalized and combined using our algorithm to get a final decision based on them and on specific parameters which are set based on the security level of the specific implementation.

During the test phase of our final project, we obtained different results based on the parameters decided but ultimately, we were able to observe how the performances of a multimodal biometric system are an increase over one single biometric system.

Future works might concern improving the algorithm for taking the final decision, creating different algorithms, and integrating different single biometric systems with better performances over the ones used.

Acknowledgements

I want to spend a few words to thank my supervisor Nicola Zingirian for the trust placed in me and in this work, but above all for having guided and helped me in this path. Finally, I would like to express my deepest appreciation to my family who believed in me since the first moment, allowing me economically and emotionally to face this journey, for the continuous encouragement and support.

Bibliography

[1] M. Ravanelli *et al.*, “SpeechBrain: A General-Purpose Speech Toolkit,” *ArXiv*, vol. abs/2106.04624, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:235377273.

[2] Exadel. 2022. CompreFace. GitHub - exadel-inc/CompreFace: Leading free and open-source face recognition system.

[3] Robert Važan.2017. SourceAFIS**.** GitHub - robertvazan/sourceafis-java: Fingerprint recognition engine for Java that takes a pair of human fingerprint images and returns their similarity score. Supports efficient 1: N search.[4] Cheniti Symmetric sum-based biometric score fusion.

[4] Garg, Suneet & Gupta, Savita. (2017). Multimodal Authentication System: An Overview. International Journal of Control Theory and Applications. 10. 111 to 119.

[5] T. Hazen, E. Weinstein, R. Kabir, A. Park, and B. Heisele. Multi-modal face and speaker identification on a handheld device. In Proceedings of the Workshop on Multimodal User Authentication, 2003.

[6] H. Bredin and G. Chollet. Audio-visual speech synchrony measure for talking-face identity verification. In ICASSP, 2007.

[7] A. Chowdhury, Y. Atoum, L. Tran, X. Liu, and A. Ross, "MSU-AVIS dataset: Fusing Face and Voice Modalities for Biometric Recognition in Indoor Surveillance Videos," *2018 24th International Conference on Pattern Recognition (ICPR)*, Beijing, China, 2018, pp. 3567-3573, doi: 10.1109/ICPR.2018.8545260.

[8] H. Vajaria, T. Islam, P. Mohanty, S. Sarkar, R. Sankar, and R. Kasturi. Evaluation and analysis of a face and voice outdoor multi-biometric system. Pattern recognition letters, 28(12):1572–1580, 2007.

[9] J. Kittler, N. Poh, O. Fatukasi, K. Messer, K. Kryszczuk, J. Richiardi, and A. Drygajlo. Quality dependent fusion of intramodal and multimodal biometric experts. In Proc. of SPIE, 2007.

[10] Y. Tong, F. W. Wheeler, and X. Liu. Improving biometric identification through quality-based face and fingerprint biometric fusion. In CVPRW, 2010.

[11] N. Ozay, Y. Tong, F. W. Wheeler, and X. Liu. Improving face recognition with a quality-based probabilistic framework. In CVPRW, 2009.

[12] A. Chowdhury, Y. Atoum, L. Tran, X. Liu and A. Ross, "MSU-AVIS dataset: Fusing Face and Voice Modalities for Biometric Recognition in Indoor Surveillance Videos," *2018 24th International Conference on Pattern Recognition (ICPR)*, Beijing, China, 2018, pp. 3567-3573, doi: 10.1109/ICPR.2018.8545260.

[13] Johnson, P. A., Tan, B., & Schuckers, S. (2010). Multimodal fusion vulnerability to non-zero effort (spoof) imposters. In 2010 IEEE International Workshop on Information Forensics and Security, WIFS 2010. https://doi.org/10. 1109/WIFS.2010.5711469.

[14] Akhtar, Z., Fumera, G., Marcialis, G. L., & Roli, F. (2012). Evaluation of serial and parallel multibiometric systems under spoofing attacks. In 2012 IEEE 5th International Conference on Biometrics: Theory, Applications and Systems, BTAS 2012 (pp. 283–288). https://doi.org/10. 1109/BTAS.2012.6374590.

[15] Gomez-Barrero, M., Galbally, J., & Fierrez, J. (2014). Efficient software attack to multimodal biometric systems and its application to face and iris fusion. Pattern Recognition Letters, 36, 243–253. https://doi.org/10.1016/j.patrec.2013. 04.029.

[16] Gupta, K., Walia, G. S., & Sharma, K. (2020). Quality based adaptive score fusion approach for multimodal biometric system. Applied Intelligence, 50(4), 1086–1099. https://doi. org/10.1007/s10489-019-01579-1.

[17] Sujatha, E., & Chilambuchelvan, A. (2018). Multimodal biometric authentication algorithm using iris, palm print, face, and signature with encoded dwt. Wireless Personal Communications, 99(1), 23–34. https://doi.org/10.1007/ s11277-017-5034-1.

[18] Ali, S. F., Khan, M. A., & Aslam, A. S. (2021). Fingerprint matching, spoof, and liveness detection: classification. 15(1).

[19] Dahea, Waleed & Fadewar, H.S. (2018). Multimodal biometric system: A review. International Journal of Engineering and Technology. 4. 25-31. 10.13140/RG.2.2.34056.65287.

[20] Cheniti, M., Boukezzoula, N.-E. and Akhtar, Z. (2018), Symmetric sum-based biometric score fusion. IET Biom., 7: 391-395 . https://doi.org/10.1049/iet-bmt.2017.0015.

[21] Vibhor Jain, Speaker Recognition Audio Dataset, version 1, retrieved 2023 from https://www.kaggle.com/datasets/vjcalling/speaker-recognition-audio-dataset.

[22] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep Learning Face Attributes in the Wild,” in *Proceedings of International Conference on Computer Vision (ICCV)*, Dec. 2015.

[23] Neurotechnology UareU retrieved from https://www.neurotechnology.com/download.html#databases.