Princess Nora Bint Abdulrahman University
College of Computer and Information
Sciences Information Technology

Personal Identification using Cancelable Biometrics and Deep Learning

By

Shahad Alghamdi

Sarah Aldawsari

Deema Almalki

Hatoon Abdullatif

Wed Alotaibi

Supervised By

Prof. Mohamed Batouche

A Graduation Project Report Submitted to
College of Computer Sciences and Information at PNU
in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in
Information Technology

CCIS, PNU

Riyadh, KSA

1442 - 1443H

Table of Contents

List of figures

List of tables:

List of Symbols and Abbreviations

PNU	Princess Nourah Bint Abdulrahman university
RSA	Rivest–Shamir–Adleman
DES	Data Encryption Standard
SHA1	Secure Hash Algorithm 1
DNA	Deoxyribonucleic acid
MLP	Multilayer Perceptron
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network

Acknowledgments

First and for most, "we thank God who gave us the ability, patience, and knowledge to complete this project. It has been a wonderful opportunity to gain lots of experience in projects, followed by a knowledge of how to design and analyze real projects We got to learn a lot from this project about cancelable biometrics and deep learning and how the cancelable biometrics will use the Deep learning. For that, we want to thank Princess Norah University and the Faculty of Computer and Information sciences at Princess Norah University Also, we would like to express our sincere gratitude and thanks to our supervisor Dr. Mohamed Batouche for his support, guidance, constant encouragement, and patience throughout each stage of the project. Finally, we would like to thank all the people who helped, supported, and encouraged us to successfully finish the graduation project Phase 1 whether they were in university or out of university. Moreover, that this project would not be possible to accomplish without the team members the experience of working together was beautiful and enjoyable.

Abstract

Revocable biometrics are created for the purposes of increasing the level of security and privacy of users against theft or any attack. The primary goal behind reversible biometrics is to create a new, distorted biometrics model stored in a database instead of the original templates. Besides the development of machine learning, especially deep learning, people are more and more interested in the practical uses of this technology. Image recognition is one of the most popular fields of study in recent years.

In this project, we will propose a system for personal identification which is based on cancelable biometrics and deep learning.

Keywords:

Multi-Biometrics, Cancelable Biometrics, Deep learning, Machine learning, Convolutional Neural Network, Auto-Encoder, Personal Identification.

Chapter 1: Introduction

Introduction

Personal identification can be achieved using biometrics recognition which has emerged as a reliable approach for automated human identification and is attracting significant attention from the researchers in multifaceted disciplines.
we propose the development of a reliable personal identification system which is based on cancelable biometrics and distributed deep learning

1.1 Problem Statement & Significance

The concept of the biometric is the automated recognition of human individuals based on their biological and behavioral traits. Because sometimes People forgetting their passwords even their personal identifications such as national id, license .

To Secure personal identification and biomatrices: fingerprint and face ID are safer than password.
It's not easy to hack and if it hacked, we can cancelable Biometrics and change the encryption of the biometric.

We will use Personal identification using cancelable biometrics and deep learning to solve this problem by using the face identification (Multi-biometrics) to make people lives easer so they don't have to remember every password they have and their personal identification will be more secure and safe because of cancelable biometric it has two category biometrics authentication compares data for the person's characteristics to that person's biometric "template" to determine resemblance .The data stored is then compared to the person's biometric data to be authenticated.
Biometric identification consists of determining the identity of a person. This data is then compared to the biometric data of several other persons kept in a database. Cancelable Biometric template is generated using some technique such as Hashing, Filtering, Cryptography, etc. These templates are then stored in the database.

In general, the cancelable biometrics can be used to make sure your biometrics has much higher security, because cancelable will secure and safe your biometric.

Deep learning methods learn features from data which help to generalize for other related tasks. Various correlated factors are disentangled in these learned features compared to hand-engineered features which are designed to be invariant to such factors. Face recognition can be formulated as a verification or identification problem. In verification mode, we verify whether a person is who he claims to be by comparing a person's face to his previously collected gallery images. In identification mode, a person's face is compared with the gallery images of all individuals to establish a person's identity.

The objective of our project to use the maximum advantage of this technology to make our lives easier.

1.2 Proposed Solution (System)

In this project, we will propose a system for personal identification which can be used to access to a bank remotely. The system will make use of multi-biometrics, cancelable templates, and deep learning.

1.3 Project Domain & Limitations:

- this program will apply only on Saudi Arabia, and it will be available for all age ranges

Chapter 2: Background Information and Related Work

2.Background Information & Related Work:

2.1 Background Information

2.1.1 Biometrics and Personal Identification

Biometrics: are body measurements and calculations related to human characteristics. Biometric authentication is used in computer science as a form of identification and access control.

It is also used to identify individuals in groups that are under surveillance.

Personal identification: is defined as establishing the identity of an individual and can be defined as Personal identity the unique numerical identity of a person over time. Discussions regarding personal identity typically aim to determine the necessary and sufficient conditions under which a person at one time and a person at another time can be said to be the same person, persisting through time.

There are two types of biometrics: Physical biometrics and Behavioral biometrics.

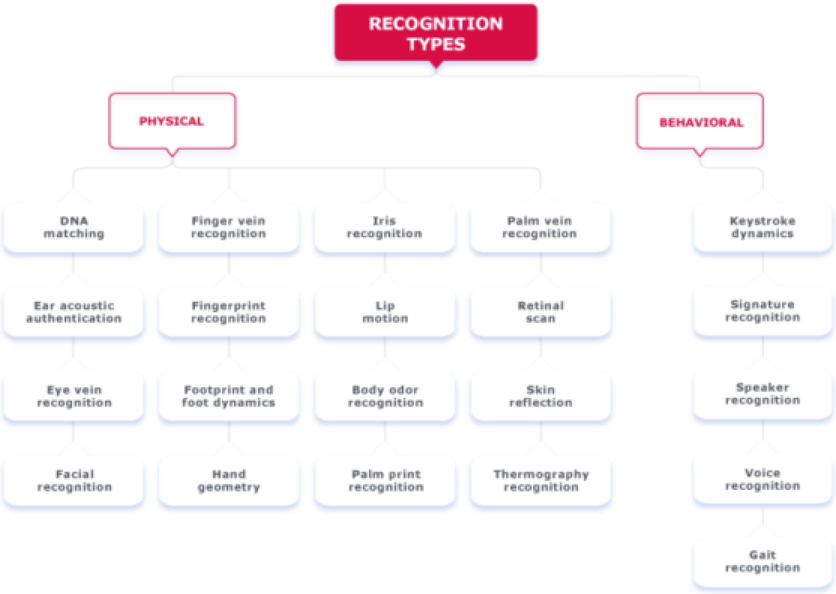


Figure 2-1: recognition types.

PHYSICAL BIOMETRICS

Physical biometrics refers to physiological features on the human body that can serve as identification, analyze data such as facial features, eye structure (retina or iris), finger parameters. Physical biometrics have become widespread, for example, access control to smartphones and laptops.

Physiological measurements

1-Face recognition face recognition technique is applications that identify or verify a person automatically from a digital image or a video frame from a video source. Facial metric technology relies on the manufacture of the specific face recognition feature, such as the position of eyes, nose and mouth, and distances between these features.

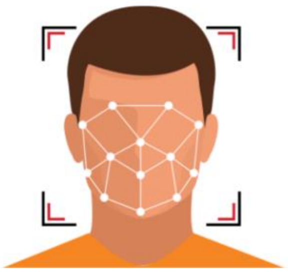


Figure 2-2: Face recognition.

2-Eye (Iris) recognition

Iris recognition or iris scanning is the process of using visible and near-infrared light to take a high-contrast photograph of a person's iris.

Figure 2-3: Iris recognition.

3-Fingerprint Recognition

Fingerprint recognition allows a person to be verified or identified through the analysis and comparison of his or her finger dermal ridges.

Figure 2-4: Fingerprint recognition

Advantages and disadvantages of Physical Biometrics:

-Advantages:

1- An identifier is inseparable from a person; it cannot be forgotten, lost, or passed on, no one can forget their face or hand or finger or eye.

2- It is quite difficult to recreate an identifier but it’s not impossible.

3- The process of biometric identification is fast and completely performed by computers.

-Disadvantages:

1- Situations can arise where biometric identifiers are damaged or unavailable for reading.
2- For many biometric identification systems, biometric scanners are quite expensive.
3- It is necessary to comply with the requirements of regulators for the protection of personal biometric data

BEHAVIOLAR BIOMETRICS

is the field of study related to the measure of uniquely identifying and measurable patterns in human activities in a person so the algorithm can identify the person by their behavioral pattern.

BEHAVIORAL MEASUREMENTS

behavioral characteristics that related to the pattern of people doing something, such as signature, typing, and mouse movement.

1-The signature recognition

is based on the dynamics of making the signature, rather than a direct comparison of the signature itself afterwards. The dynamic is measured as a mean of the pressure, direction, acceleration and the length of the strikes, and dynamic number of strokes and their duration.

Figure 2-5: Signature recognition.

2-Typing Recognition

The use of the unique characteristics of a person’s typing for establishing identity.

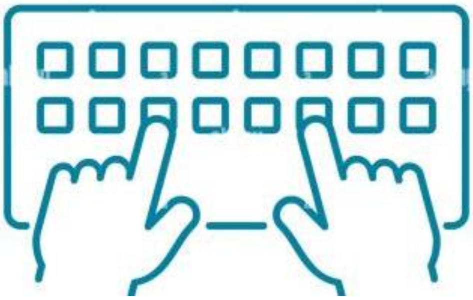


Figure 2-6: typing recognition.

3- Gait recognition.

The use of an individual’s walking style or gait to determine identity.

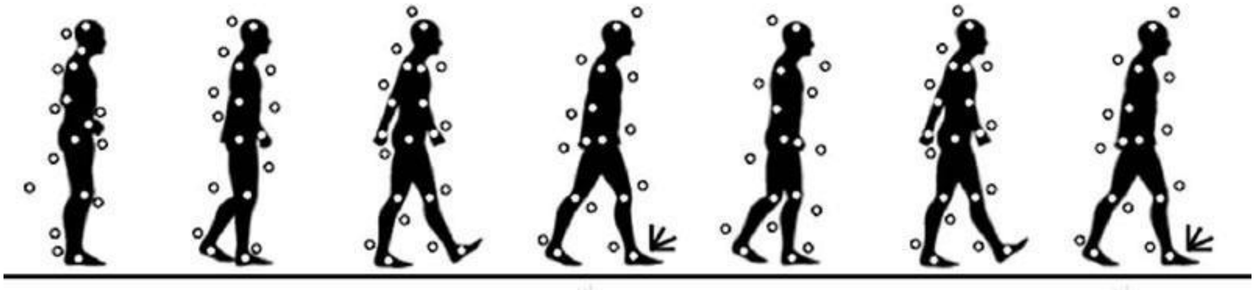


Figure 2-7: Gait recognition.

Advantages and disadvantages of Behavioral Biometrics:

-Advantages:

1- Individual user set of analyzed behavioral characteristics

2- No custom script change is required to perform identification: seamless integration method.

-Disadvantages:

1-Inaccuracies in identification may arise because the user's behavior is not always constant since they can behave differently in various situations due to fatigue, drunkenness, feeling unwell or trivial haste.

2-Requires lots of personal data to determine a user's standard behavior. Also, it’s not wildly used.

2.1.2 Cancelable Biometrics

1- Cancelable Biometrics Definition:

One advantage of passwords over biometrics is that they can be re-issued. If a token or a password is lost or stolen, it can be cancelled and replaced by a newer version. This is not naturally available in biometrics. If someone's face is compromised from a database, they cannot cancel or reissue it. If the electronic biometric identifier is stolen, it is nearly impossible to change a biometric feature.

Cancelable biometrics is a way in which to incorporate protection and the replacement features into biometrics to create a more secure system.

2- CANCELABLE BIOMETRIC TEMPLATES:

strategies for generating cancelable biometric templates. In these methods, a function that is dependent on some parameter is used to generate protected biometric templates. The parameter of the function is used as the key.

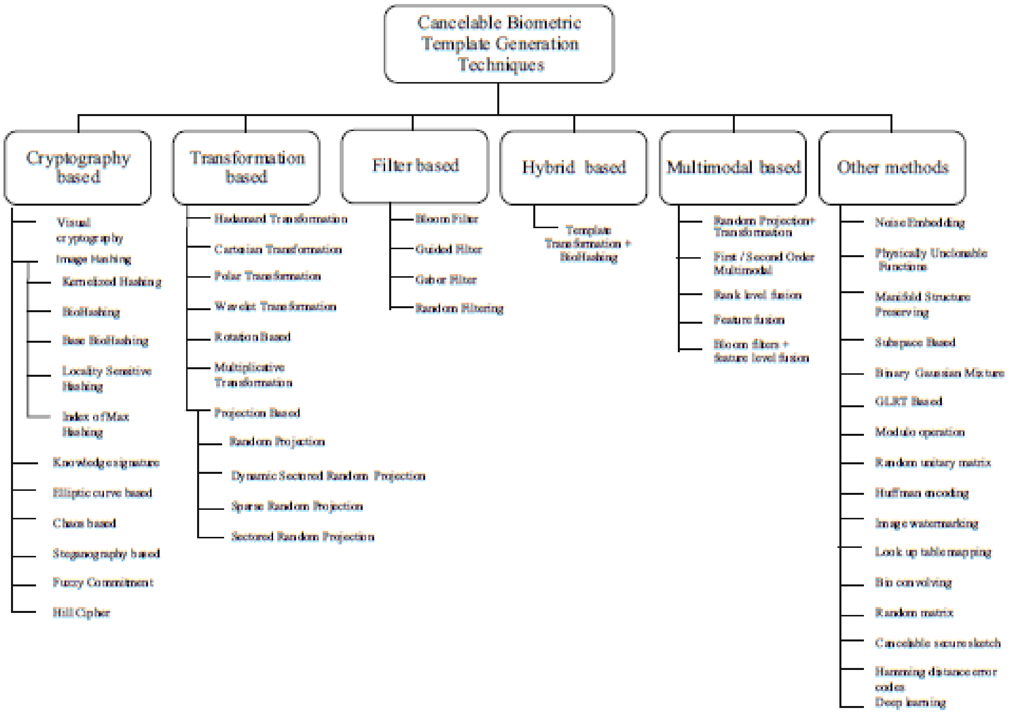


Figure 2-8: Cancelable Biometric techniques.

Cryptography based methods

Definition: Cryptography is associated with the process of converting ordinary plain text into unintelligible text and vice-versa. It is a method of storing and transmitting data in a particular form so that only those for whom it is intended can read and process it. Cryptography not only protects data from theft or alteration but can also be used for user authentication.

Description: Earlier cryptography was effectively synonymous with encryption but nowadays cryptography is mainly based on mathematical theory and computer science practice.

Modern cryptography concerns with:

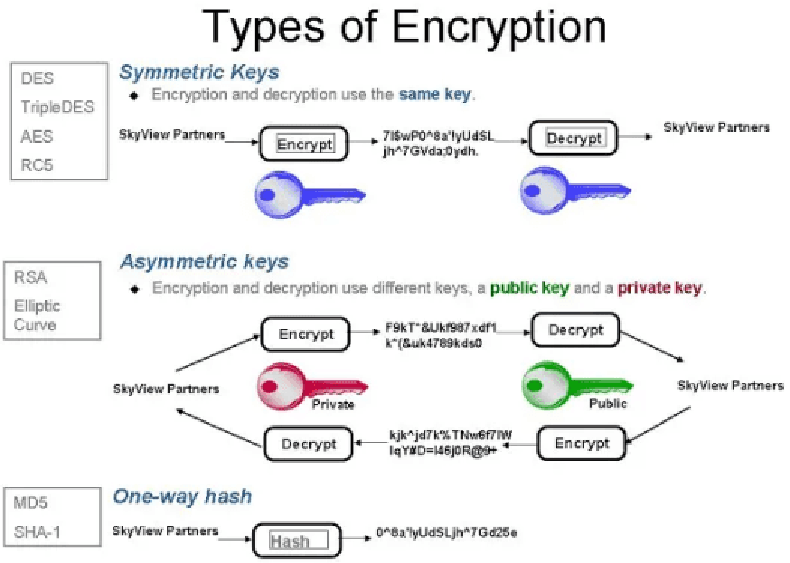
Confidentiality - Information cannot be understood by anyone

Integrity - Information cannot be altered.

Non-repudiation - Sender cannot deny his/her intentions in the transmission of the information at a later stage

Authentication - Sender and receiver can confirm each Cryptography is used in many applications like banking transactions cards, computer passwords, and e-commerce transactions.

Three types of cryptographic techniques used in general:



Types of Encryption

Symmetric Keys

- Encryption and decryption use the same key.

Asymmetric keys

- Encryption and decryption use different keys, a public key and a private key.

One-way hash

Figure 2-9: Types of encryptions.

Which Encryption Types Involve Keys?

Symmetric key and asymmetric key cryptography make use of keys for encryption and decryption.

The **hash function** doesn't require the use of any key; instead, it takes an input of arbitrary length and provides the output in fixed length.

Hashing is almost always preferable to encryption when storing passwords inside databases because in the event of a data compromise, attackers will not have access to the plaintext passwords, and the website will not ever know the user's plaintext password.

Transformation based methods

Description: In this method, the original Biometric templates are morphed by applying Different transformations e.g., Cartesian, Polar etc. In Cartesian transformation, the minutiae positions are measured in rectangular coordinates with reference to the position of the singular point by aligning x-axis with its orientation. The coordinate system is divided into cells of fixed size. The transformation causes changes in the cell positions. In Polar transformation, the minutiae positions are measured in the polar coordinate with reference to the core position.

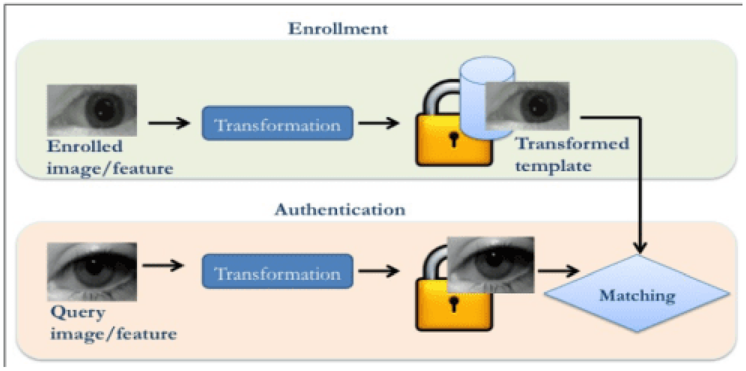


Figure 2-10: transformation-based methods.

Filter based methods

Cancelable Biometric Filter is a Convolution based method. Bloom Filters is a space efficient probabilistic data structure representing a set to support membership queries. Bloom filter-based transformation of any binary feature vector generates irreversible Cancelable Biometric templates.

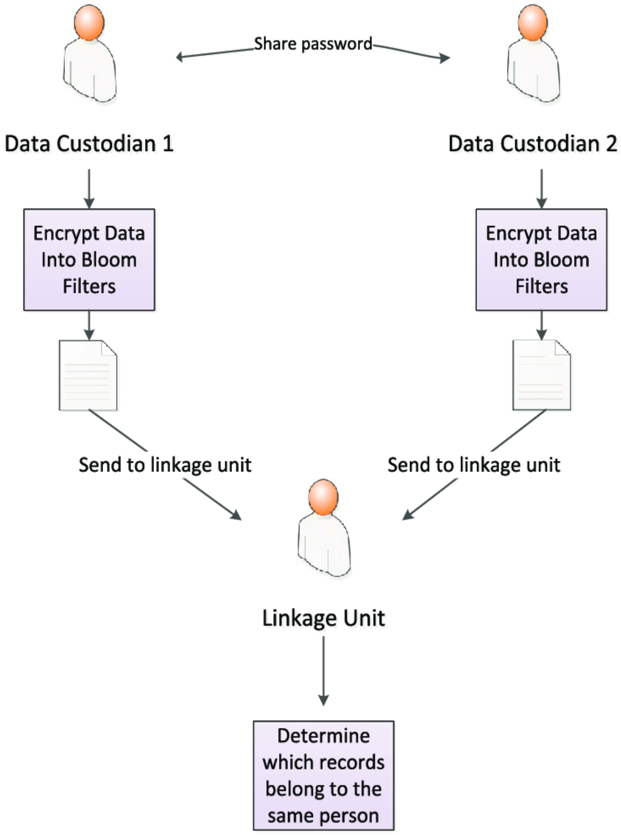

Share password
Data Custodian 1
Data Custodian 2
Encrypt Data Into Bloom Filters
Encrypt Data Into Bloom Filters
Send to linkage unit
Send to linkage unit
Linkage Unit
Determine which records belong to the same person

Figure 2-11: filter-based methods.

Hybrid methods

A hybrid encryption scheme is one that blends the convenience of an asymmetric encryption scheme with the effectiveness of a symmetric encryption scheme.

In this paper a hybrid cryptographic technique for improving data security during network transmission is proposed and their implementation and results are reported. The proposed secure cryptographic technique promises to provide the highly secure cipher generation technique using the RSA, DES and SHA1 technique.

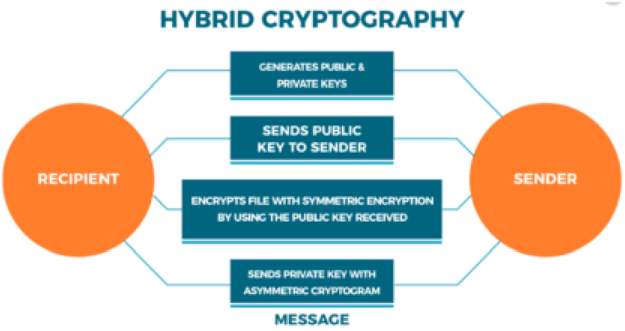

HYBRID CRYPTOGRAPHY
RECIPIENT
SENDER
GENERATES PUBLIC & PRIVATE KEYS
SENDS PUBLIC KEY TO SENDER
ENCRYPTS FILE WITH SYMMETRIC ENCRYPTION BY USING THE PUBLIC KEY RECEIVED
SENDS PRIVATE KEY WITH ASYMMETRIC CRYPTOGRAM
MESSAGE

Figure 2-12: Hybrid method.

Multimodal based methods

Security is a major issue in all recent developing technologies, for that system deals with palm-vein and Iris (Biometrics inputs) from the user and then extract the features like edge, texture using the feature extraction algorithm from both palm vein and iris captured images simultaneously and then apply cryptographic algorithm (Blow fish) to that extracted feature.

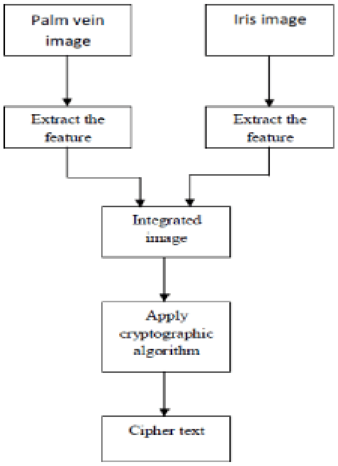


Figure 2-13: Multimodal method

2.1.3 Machine and Deep Learning

With the reinvigoration or reinvention of neural networks, deep learning has become an extremely active area of research, and one that's paving the way for modern machine learning. process of teaching a machine to think like a human being to perform a particular task, without being explicitly programmed.

Machine learning is the practice of programming computers to learn from data. we can use machine learning to solve problems that are very complex for non-machine learning software.

Deep learning is the use of neural networks with many hidden layers and raw data as inputs.

There are many deep learning architectures (figure 2-14).

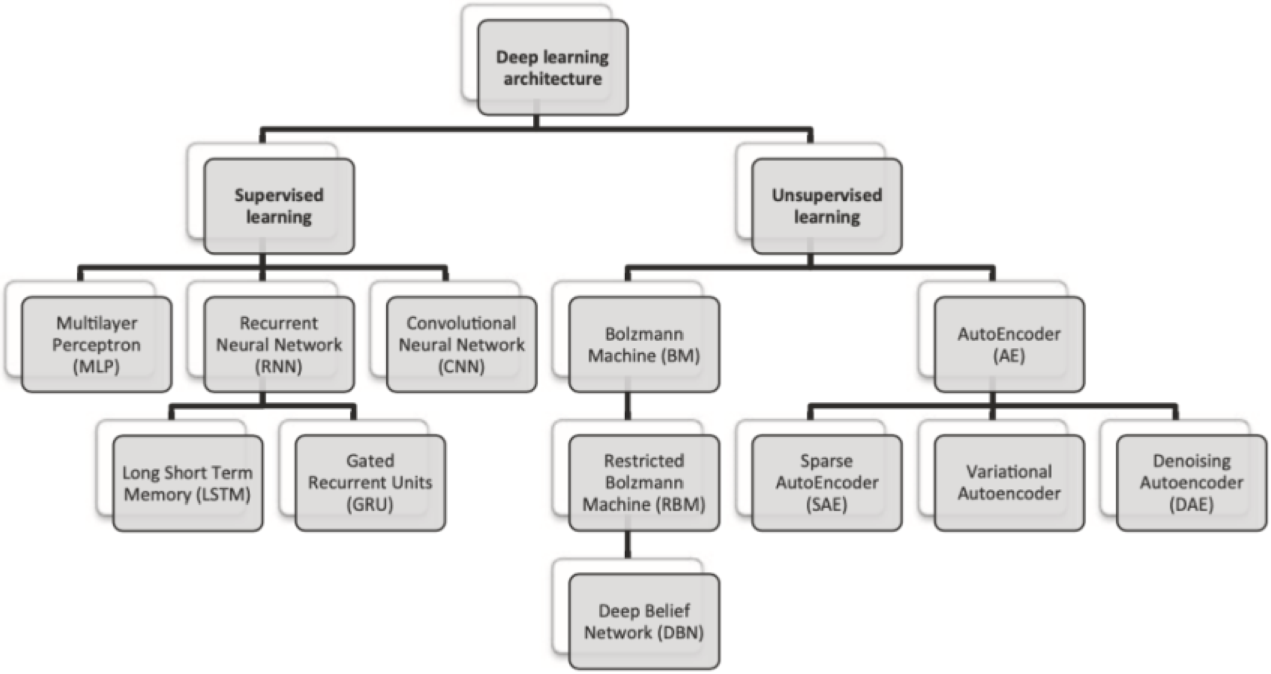

Deep learning architecture
Supervised learning
Unsupervised learning
Multilayer Perceptron (MLP)
Recurrent Neural Network (RNN)
Convolutional Neural Network (CNN)
Bolzmann Machine (BM)
AutoEncoder (AE)
Long Short Term Memory (LSTM)
Gated Recurrent Units (GRU)
Restricted Bolzmann Machine (RBM)
Sparse AutoEncoder (SAE)
Variational Autoencoder
Denoising Autoencoder (DAE)
Deep Belief Network (DBN)

Figure 2-14: Deep learning architectures.

There are different types of system of machine learning:

Supervised: in this type of the data that you feed in the algorithm with the desired solution are referred to as labels, Supervised learning groups together a task of classification.

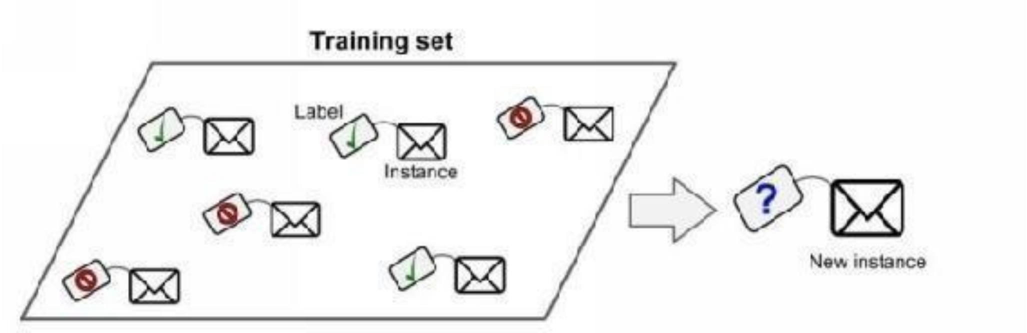


Figure 2-15: training set.

Multilayer Perceptron (MLP):

Multilayer perceptron holds many hidden layers; the neurons in the base layer *i* is completely connected to neurons in *i + 1* layer. Such type of network is restricted to have minimal hidden layers, and the data is allowed to transmit in one direction only.

Recurrent Neural Network (RNN):

RNN is a logical choice if the input data is ordered sequentially, RNNs are capable of handling long-range temporal dependencies.

Convolutional Neural Network (CNN):

The CNN has an excellent performance in machine learning problems. Specially the applications that deal with image data, such as largest image classification data set (Image Net), computer vision, and in natural language processing (NLP).

Unsupervised: In this type you can guess that the data is unlabeled. And the most important unsupervised algorithms are Clustering, Association rule learning and **Visualization** and dimensionality reduction.

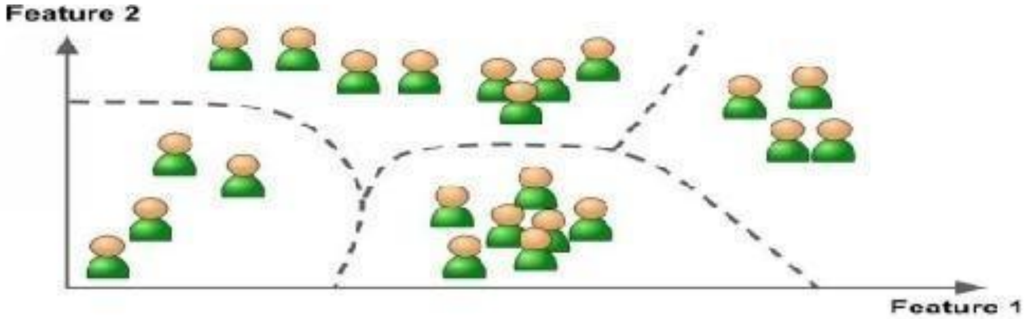


Figure 2-16: unsupervised

this is an example; suppose you've got many data on visitor Using of one of the algorithms for detecting groups with similar visitors. It may find that 65% of your visitors are males who love watching movies in the evening, while 30% watch plays in the evening; in this case, by using a clustering algorithm, it will divide every group into smaller sub-groups

Online Learning: in this type of system can learn incrementally by providing the system with all the available data as instances groups or individually, and then the system can learn quickly, we can use this type of system for problems that require the continuous flow of data, which also needs to adapt quickly to any changes. Also, you can use this type of system to work with very large data sets.

AutoEncoder: learns how to efficiently compress and encode data then learns how to reconstruct the data back from the reduced encoded representation to a representation hat is as close to the original input as possible.

Table 2-1 Compare between supervised and unsupervised learning.

Supervised Learning	**Unsupervised Learning**
it uses data that is labeled.	it uses data that is unlabeled.
it does not require excess data for accuracy.	it requires excess data for accuracy.
Computational complexity is less, i.e., it is simpler.	Computational complexity is greater, i.e., it is less simple
it does not find patterns on its own from a dataset.	it finds patterns on its own from a given dataset.

Deep learning:

Deep Learning is a new area of Machine Learning research, which has been introduced with the objective of moving Machine Learning closer to one of its original goals: Artificial Intelligence, it’s a specific method of machine learning that incur- pirates’ neural networks in successive layers to learn from data in an iterative manner. Deep learning is especially useful when you’re trying to learn patterns from unstructured data, it’s a branch of artificial intelligence and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, it is a systematic approach to leveraging advanced algorithms and models to continually train data and test with additional data to begin to apply the most appropriate machine learning algorithms to a problem

Modern deep learning libraries allow you to define and start fitting a wide range of neural network models in minutes with just a few lines of code.

Deep learning neural networks learn a mapping function from inputs to outputs. This is achieved by updating the weights of the network in response to the errors the model makes on the training dataset.

TensorFlow:

Advanced machine learning concepts utilize the manipulation and calculus of tensors, TensorFlow is an open- source end-to-end machine learning library for production and research.

Keras:

Keras is an open-source neural network library with features like fast, modular, and user friendly in Python platform that works on top of TensorFlow or Theano. Backend is a library within Keras to handle low-level computation as Keras is dedicated to advanced API wrapper.

Neural Designer:

is a deep learning tool which is used to implement analytics algorithms and make it easy to handle, it is designed with a graphical user interface that defines the flow of work and gives accurate result. It is easy to handle as there is no programming or block diagrams involved.

Deep Learning Application:

Speech Recognition:

Speech recognition uses deep learning concepts and becomes the foremost application of deep learning by cautiously using its power. the speech signal is considered as short-time stationary signal or piecewise stationary signal.

neural networks prove its efficiency in discriminative training. Neural networks provide better results for short-time signals, it is continuous speech signals.

Deep Learning in HealthCare:

Healthcare system is facing a new era by using advanced technologies and provide- Ing right treatment for right patient at right time. The deep learning architecture applicable for healthcare system mostly falls on recurrent neural networks (RNNs)

Deep Learning Applications with Python:

One of the most desirable features of a programming language used for working with the deployment of deep learning models would be the ability for quick prototyping with minimal effort.

Deep Learning for Face Recognition:

Facial recognition is one of the most prominent biometric techniques used for identity authentication and verification, it is the identification of an individual based on the photograph of their face. Deep learning methods, in the recent past, have had great success in the tasks of image recognition and classification.

Datasets:

The presence of a large-scale database is a necessary condition for the effective working of a facial recognition system. With saturation in the performance over simple databases.

Deep Learning for Fingerprint Recognition:

Fingerprint recognition refers to the identification of an individual based on the comparison of two fingerprints.

fingerprint recognition has had significant improvement over its former iterations. Many of the solutions and research upon the fingerprint recognition problem revolve around the regular minutiae-based matching.

2.2 Related Work Survey

2.2.1 Convolutional Autoencoder:

It’s a tool for extracting features from images and compressing it to a lower dimension called latent space that generated from input images, also it is used for random noise and random convolution.

The random noise is extracted features which are random noise followed by median filtering to get non-invertible templates.

The Random convolution is based transformations make use of random kernels to convolve the biometric image or features to generate cancelable templates.

So, in this Fig below it shown the processing the input biometric image, and the features that extracted from the convolutional autoencoder.

Also, Random noise is added to the obtained feature vector which is then convolved using a random kernel for generating the cancelable template.

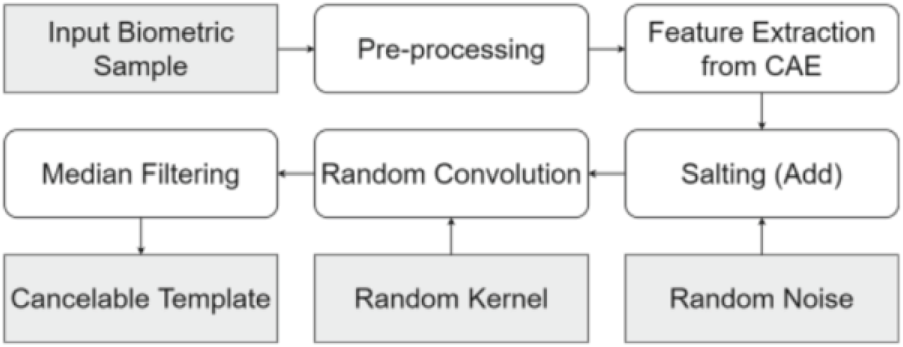


Figure 2-17: Convolutional Autoencoder

Pros:

to secure a biometric identity, is creating a **pseudo identity** that distorted in a non-invertible way by using a user specific (cancelable template).

Cons:

The identifiers directly are stored in the data base. so, if the attacker stole it, the identifiers can be used for illegitimate access to a system.

So, this is making the system under the risk and once lost, a biometric cannot be recovered.

2.2.2 Cancelable multi-biometric recognition system based on deep learning:

The figure below presents the proposed approach pipeline, which is divided into four steps: (a) detections of different facial regions; (b) extraction of deep features using multiple CNNs; (c) Using a fusion network to create a discriminative facial descriptor, and (d) using bio convolving with random kernels to protect biometric data from various threats

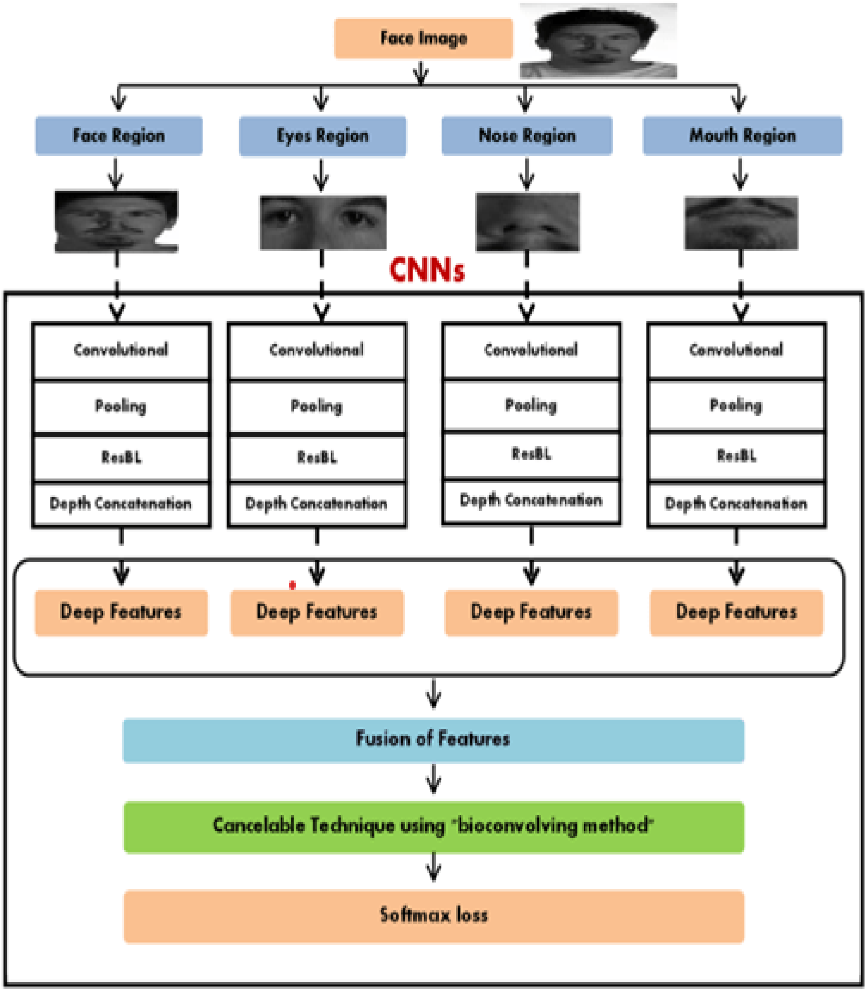


Figure 2-18: cancelable multi biometrics deep learning.

Face, nose, eyes, and mouth regions are detected from the original face images Eyes, nose, and mouth are very effective regions on which several changes are clearly noticed. These changes include laughing, closing eyes, opening mouth, or wearing glasses.

Pros:

drawback of using encryption with biometrics is the need to apply data decryption that represents an attack point

a single CNN model takes about 4 h in the training process. So, the proposed method suffers from a time consumption issue

Cons:

a bio-convolving method maintains privacy and security of biometric templates without affecting the recognition accuracy

2.2.3 Cancelable Biometrics Using Deep Learning as a Cloud Service:

Cloud computing is a technique or a tool that allows you to store your sensitive data on a logical could know as a remote database. If an electronic device has access to the web, it has access to the data and the software programs to run it. But here it had been developed by using cancelable Biometrics and Deep Learning to make your sensitive data that stored on the cloud more secure and safe.

Pros:

high accuracy, non-repudiation, and permanency. Sense the cost is so high this method provides pay as you go techneqe which allow you to pay the cost of your work only .

Cons:

So complex so it may increase costs and data storage. The cost can go into two different categories the hardware cost and the software cost so that would be expensive.

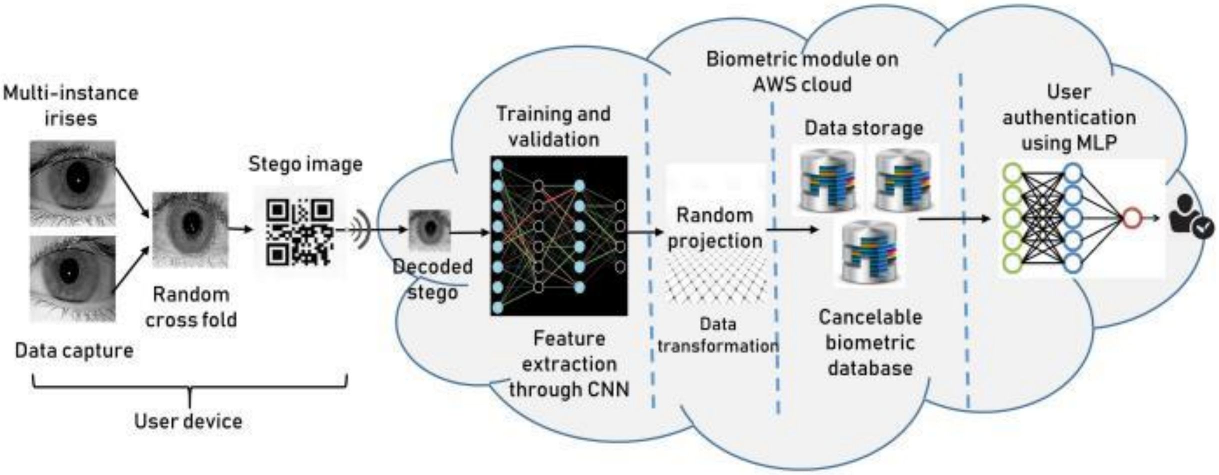


Figure 2-19: Cancelable Biometrics Using Deep Learning as a Cloud Service

2.2.4 Cancelable fusion-based face recognition:

Biometric recognition refers to the automated process of recognizing individuals using them
biometric patterns.

FR is considered as a promising option for human individuals' identification improvement and People can access their accounts by secret codes, Users could be identified by face, iris, fingerprint, blood, or DNA.

DL learns multiple levels of representations including invariance of facial expressions, pose and lighting. DL reshapes the FR research landscape with respect to datasets and evaluation protocols.

Biometric protection techniques that are used for preserving biometric authentication can be categorized to cancelable biometric techniques and biometric cryptosystems.

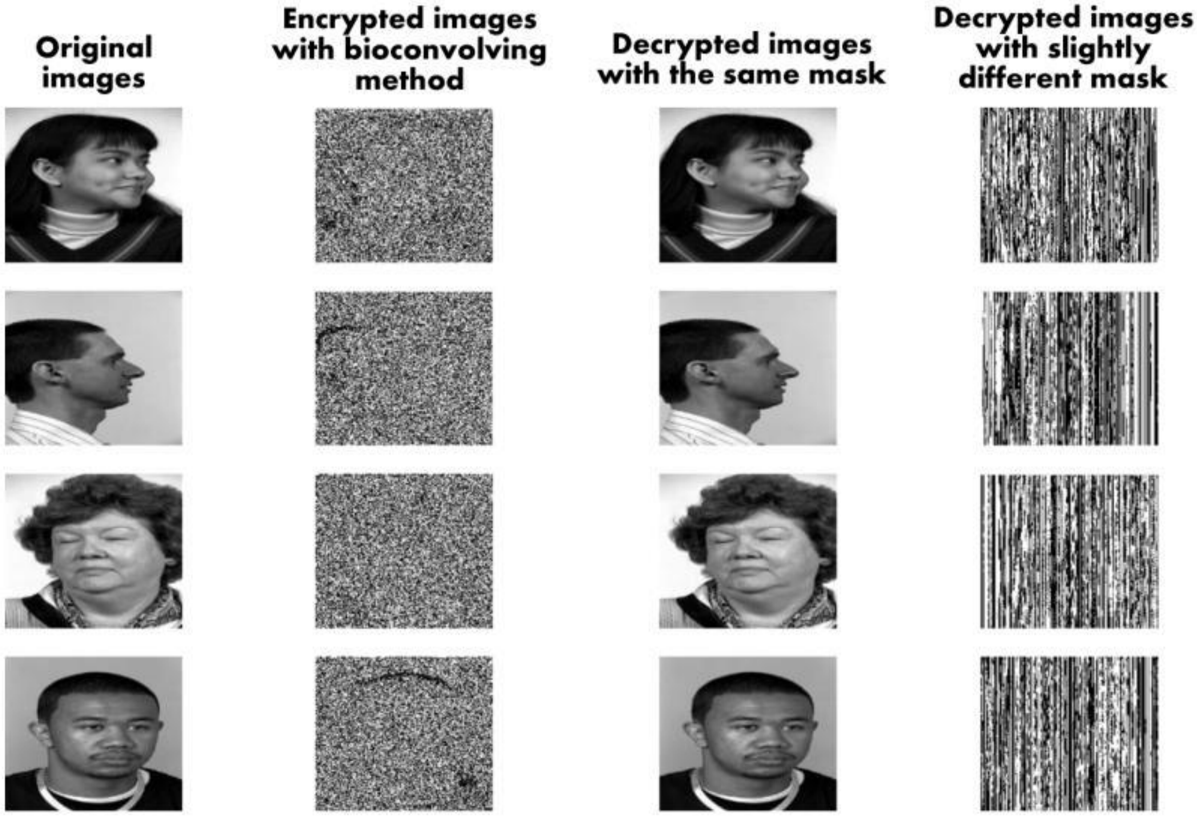

Original images
Encrypted images with bioconvolving method
Decrypted images with the same mask
Decrypted images with slightly different mask

Figure 2-20: Cancelable fusion-based face recognition.

Encryption and decryption of a class of face images with the same mask and a slightly different mask

Pros: doesn’t require multiple distributed and active co-operation of a person.

Protection can be provided without degradation in the system performance

Cons: secret codes are unique, but they could be stolen or hacked by criminals or forgotten by users.

2.2.5 Attacks and Protection in Multi-biometric System:

State of the art multi-biometric systems The biometrics framework has been divided into two categories depending on the number of attributes used, a unimodal and multi-modal system where the system uses a single biometric attribute of an individual for identification and verification There are certain limitations associated with these systems such as intra-class variation, spoofing attack , Failure to Enroll and Difference Between Classes To overcome these limitations, a multiple-biometrics system was introduced. These are systems capable of using two or more anomalies to identify an individual. The accuracy of biometric system performance can be increased by using a multi-biometric system instead of a mono-biometric system. Fusion plays a key role in multi-biometrics. It is implemented in five different levels as shown in Figure 2-21.

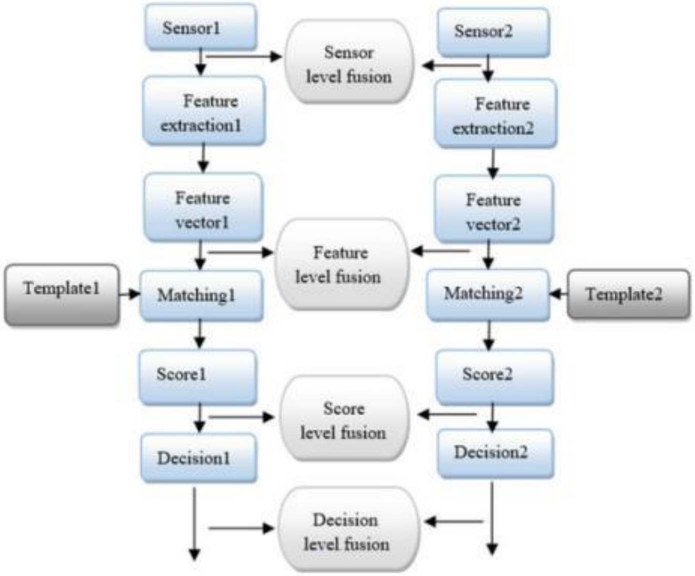


Figure 2-21: Fusion levels in biometric system.

Biometric systems may operate in two different modes, identification mode or verification mode.

In the case of identification mode, the user is identified by comparing his input to the templates already stored in the database, while in the verification mode, the identity of the user is checked against the claimed identity and checking whether the user is real or not.

In general, biometric systems are deployed in different application areas, such as commercial, for example, ATMs, distance learning and PDAs; government, for example, social security, border, and airport security; and forensics, for example, cadaver identification, paternity determination, and criminal investigation.

Pros:

Basic information for all Users registered in biometric systems are stored centrally in the form database.

Cons:

Attacks on the template database in these systems can lead to failures or deterioration in system performance.

A Survey on Biometrics and Cancelable Biometrics Systems:

The biometric traits possessed by everyone are unique and has the potential to recognize an individua, there are two phases in every conventional biometric system: enrolment phase and authentication phase.

Compared to password or token-based authentication system, biometric system using fingerprint, iris, face, voice, etc. provides better security as people cannot lose or forget their biometric trait. But the advanced technology of today’s world makes it possible to create a loophole in the biometric system.

People leave their fingerprints on whatever they touch; hence one can easily steal the fingerprint and can even make an artificial finger using the stolen fingerprint.

The person’s face can be captured by the camera even from a distance without their concern

The biometric template protection schemes are mainly divided into two categories:

- Biometric Cryptosystem

encrypts the biometric data to store it. Then, during authentication, the stored template is decrypted to do the comparison.

- Cancelable Biometrics

matches the templates in the transmuted domain itself during the authentication phase. And provide the comparison decision in terms of Match or Non-Match.

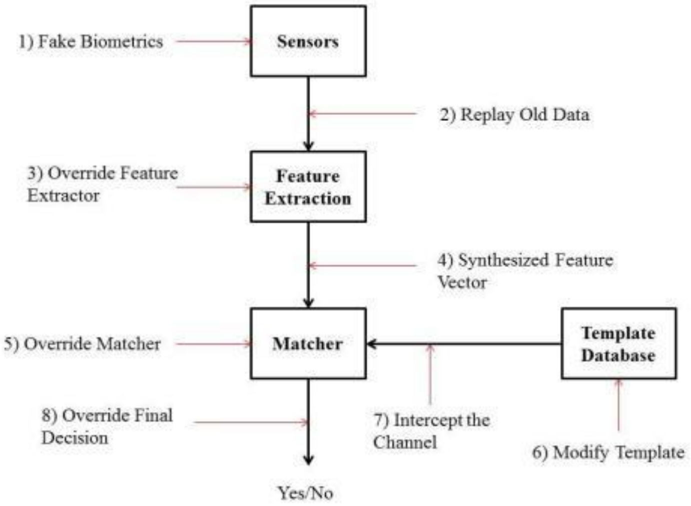


Figure 2-22: Possible Attacks on Biometric Verification System

Pros:

The advantage using biometric traits are that biometrics cannot be forgotten, cannot get lost, it is permanent and difficult to forge, and It can prevent attacks on the database against the biometric applications.

Cons:

Finding an appropriate transformation function for cancelable biometrics is a complicated task. Standard non-invertible transformation functions do not operate properly with biometric data, the user either has to memorize a password/pin or to bring the transformation parameter stored in the form of a token.

2.3 Proposed and Similar Systems comparison

Table 2-2 Compare between our system and other systems

	Our System	System based on CAE	System based on CNN
Solved Problem	Personal Identification	Authentication	Authentication
Features	- Multi-biometrics (Face + Fingerprint) - Feature Extraction using CNN/CAE/SAE/AE - Cancelable template using bioconvolution	- Multi-Biometrics - Feature Extraction using CAE - Cancelable template using random convolution	- Multi-instances - Feature Extraction using CNN - Cancelable template using random projection
Advantages	Security Feature extraction using transfer learning when using CNN	Security	Security
Limitations	Requires Training when using CAE/SAE/AE	Requires Training	Requires Training for user Authentication

Chapter 3: System Analysis

3.1 System Requirements

3.1.1 Requirements specification

In this project, we will develop a system for personal identification based on cancelable biometrics combined with deep learning. This system will be used to access remotely to a bank by using biometrics which are better than using a password. The general architecture of our system is depicted on figure 3-1.

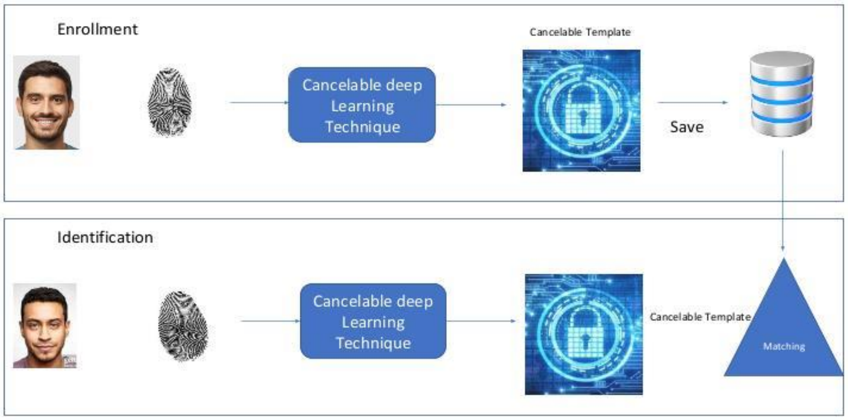


Figure 3-1: The proposed System.

3.2 Use case diagram

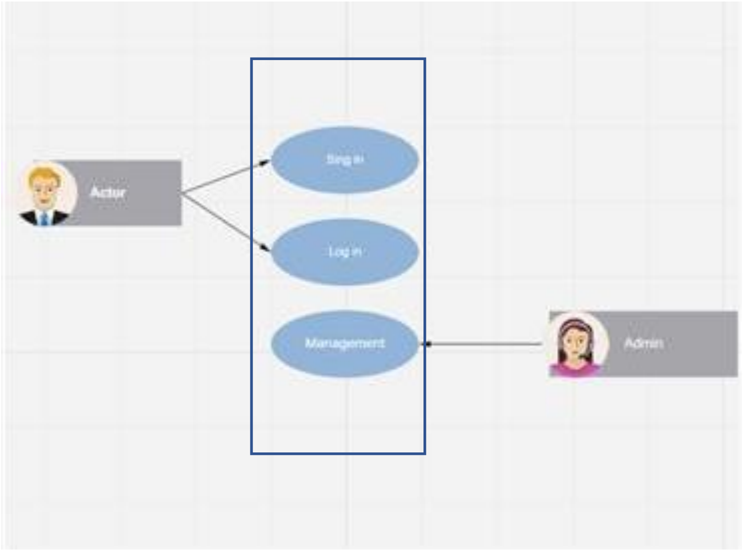


Figure 3-2: Use case diagram.

3.3 **Functional and nonfunctional Requirements**

3.3.1. Functional Requirements

The Functional Requirements of the system explain the specific functions to be performed or accomplished by the system. We have three main functions:

- Enrollment (sign up)
- System Management
- Personal Identification (Sign in)

3.3.2 **Non-functional Requirements**

- Availability
- Security
- Reliability
- Efficiency (fast response)

3.3.3 **Software requirements**

- Operating system: Windows OS.
- Program: Python with TensorFlow 2.0

3.3.4 **Hardware requirements**

- Laptop with GPU
- Smartphone

Chapter 4: System design

4.1 System Architecture

Enrollment Phase (Sign up):

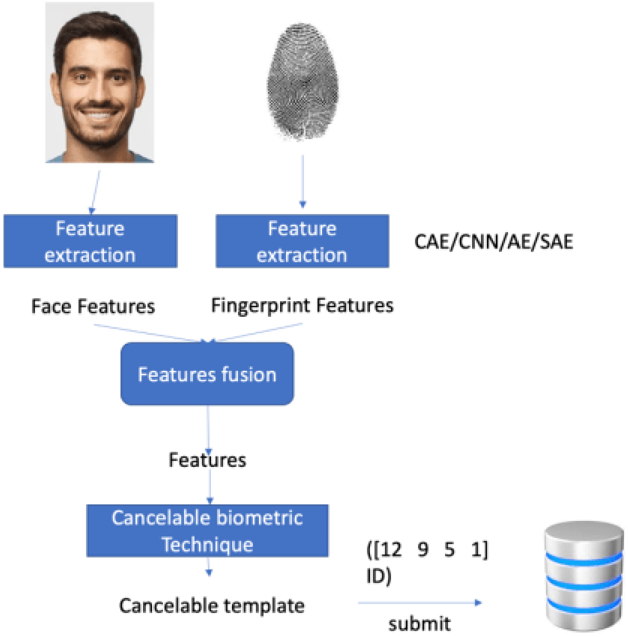


Figure 4-1: Architecture of system Enrollment Phase.

Identification Phase (Sign in):

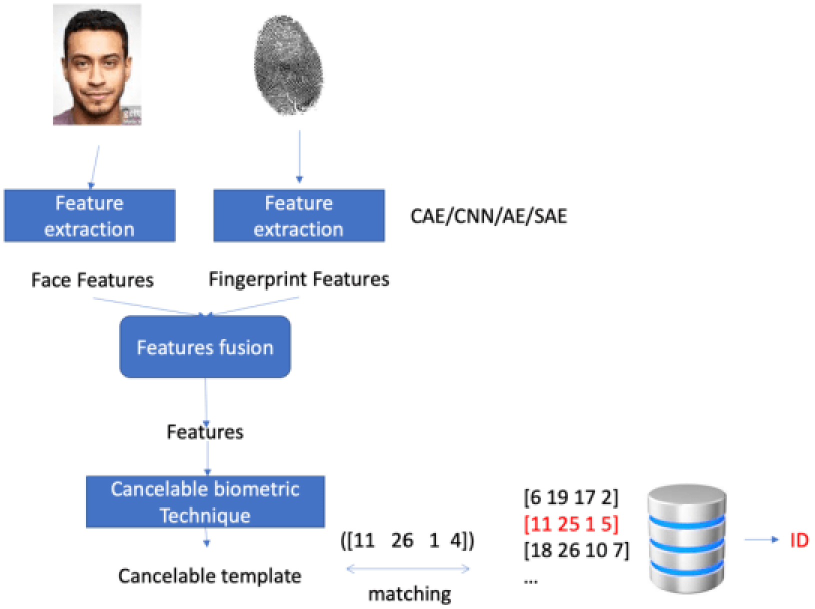


Figure 4-2: Architecture of system Identification Phase.

The proposed System :

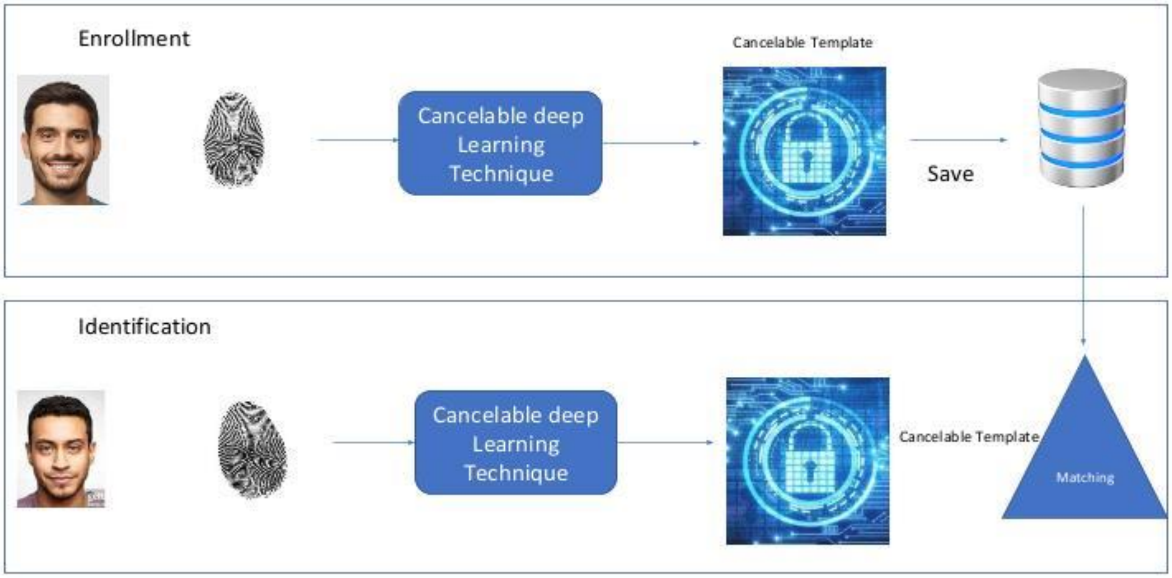


Figure 4-3: The architecture of the proposed System.

4.2 Interfaces:

Main interface:

This is the main interface , when the user click or tap the application icon ,this interface will pop-up .
the user has two choices sign in if the user has an account or sign up to create an account.

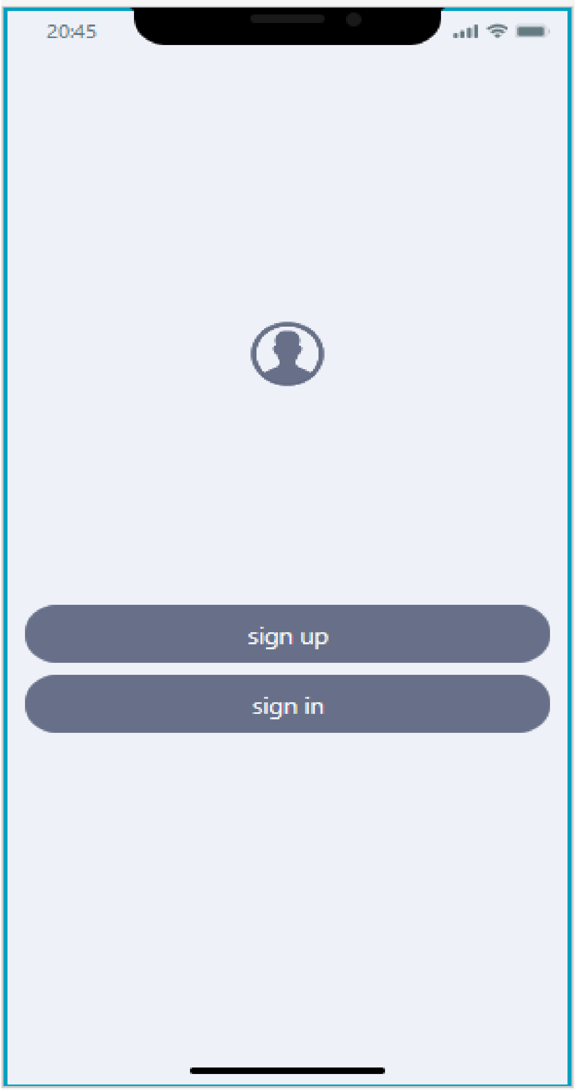


Figure 4-4: Main interface

Sign up interface:

When the user tap or click on sign up
 to create an account this interface will pop up.
And to sign up the user should enter the Email,
Name, Face ID , and Fingerprint .
After that user will click or tap sign up button
the System will save the new User information.

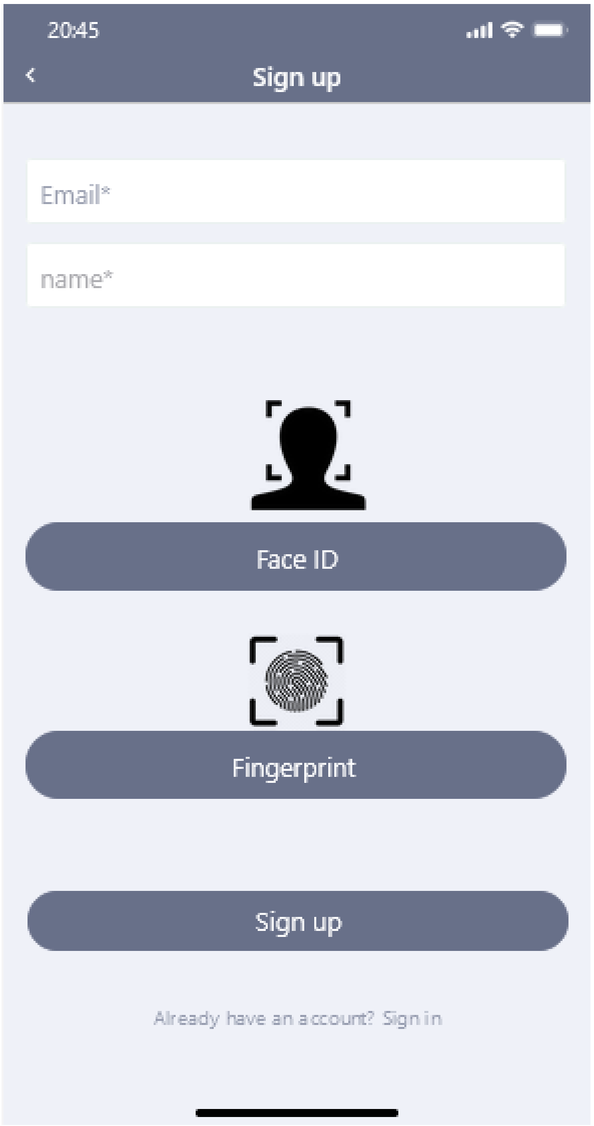
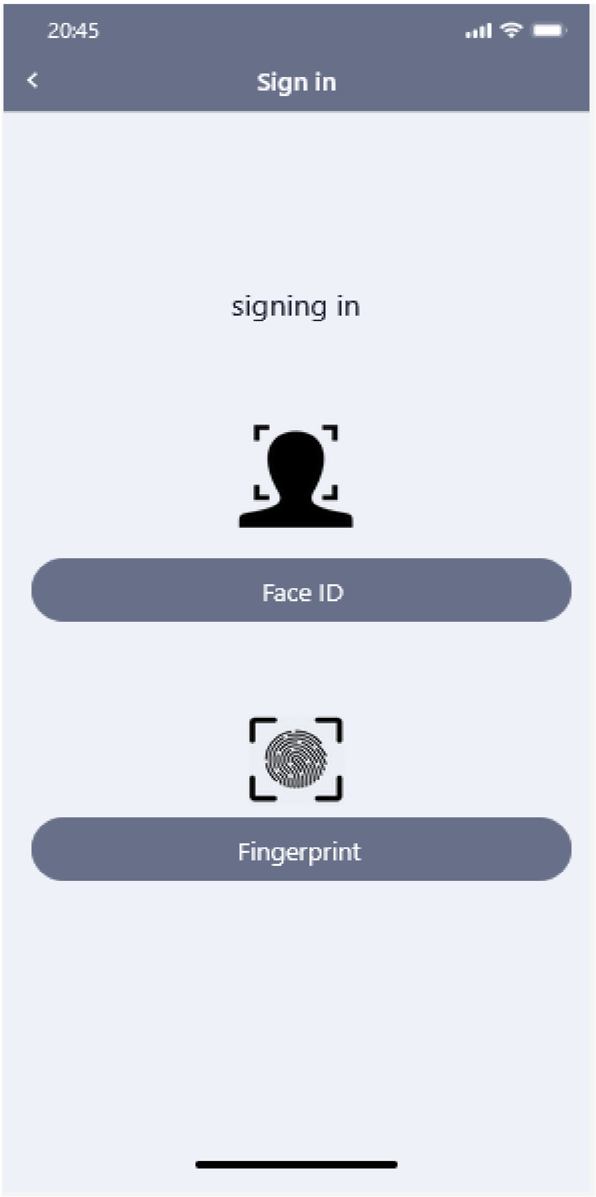


Figure 4-5: Sign up interface.

Signing in interface:

When user click or tap on sign in this interface Will pop up and in order for the user to sign in The system require Face ID and finger print of the user so the system can match them in the database and confirm the user. If the entered Biometrics matches what is saved in the data base the user shall enter his account if not then the system will deny the user



20:45

Sign in

signing in

Figure 4-6: Signing in interface.

4.3 Datasets

We will use the publicly available datasets provided by KAGGLE :

https://www.kaggle.com/vasukipatel/face-recognition-dataset?select=Dataset.csv

KAGGLE has launched new datasets, consisting of fingerprint, facial photographs, and iris scans, to help biometrics researchers to evaluate the performance of access control identity verification systems, according to an announcement by the institution.

Chapter 5:Implementation

5.1 implementation requirement

- **Software requirements**

  Python programming language (PyCharm)
  TensorFlow 2.0
  Keras

- **Hardware requirements**

  Laptop with GPU (RTX 3070 or RTX 3080 TI)

  Smartphone

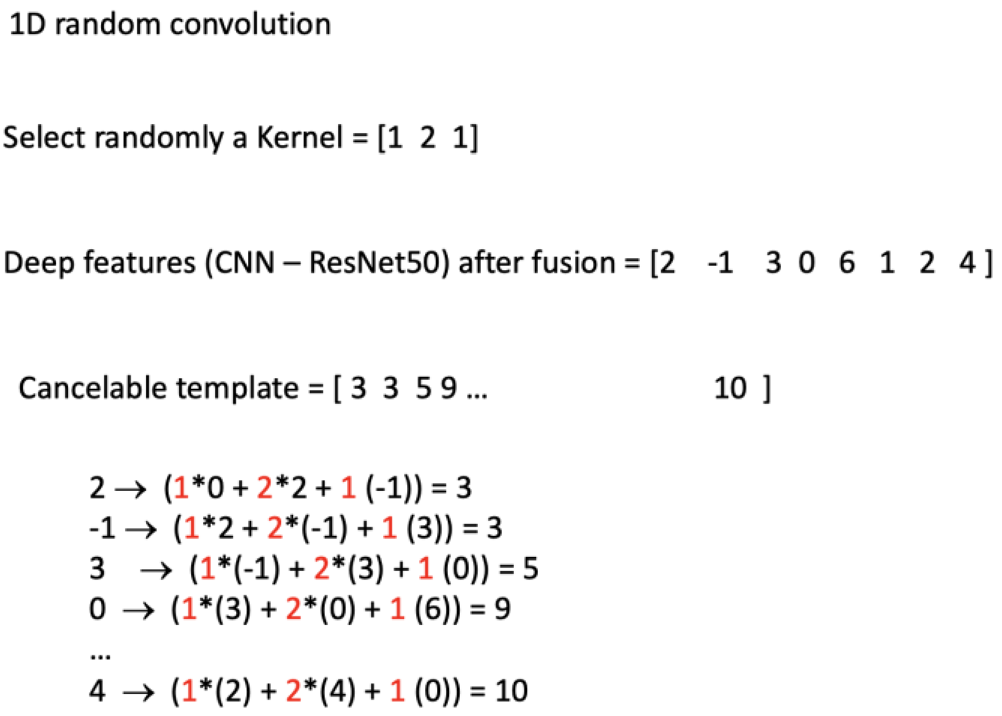
5.2 implementation details

- **data preparation**

in this phase we have the data preparation code .
Firstly, this code for each person will select randomly one face image and the fingerprint for each right index randomly.
These images will be used in the enrollment phase to build the database.

- **enrollment method**

in this phase , we will use the selected face images and fingerprint in the data preparation phase to build the database of cancelable templates.
For each person, we use the selected face image and fingerprint to generate the deep features using a pretrained CNN - ResNet50 - as feature extractor.
These deep features are then merged and modified using random convolution. The obtained features correspond to the cancelable template which will be stored in the database to be used in the identification phase.



1D random convolution

Select randomly a Kernel = [1 2 1]

Deep features (CNN – ResNet50) after fusion = [2 -1 3 0 6 1 2 4]

Cancelable template = [3 3 5 9 ... 10]

2 → (1*0 + 2*2 + 1 (-1)) = 3

-1 → (1*2 + 2*(-1) + 1 (3)) = 3

3 → (1*(-1) + 2*(3) + 1 (0)) = 5

0 → (1*(3) + 2*(0) + 1 (6)) = 9

...

4 → (1*(2) + 2*(4) + 1 (0)) = 10

Figure 5-1: Random convolution

- **Personal identification**

In this phase , we will select randomly a face image and a fingerprint for a certain person.

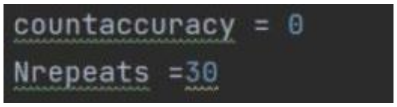
Then, the selected face image and fingerprint are used to generate the deep features based on a pretrained CNN - ResNet50 - as feature extractor.

These deep features are then merged and modified using random convolution. The obtained features correspond to the cancelable template which will be matched to the cancelable templates present in the database.

We select the person with the closest cancelable template. We use Euclidean distance to measure the closeness.

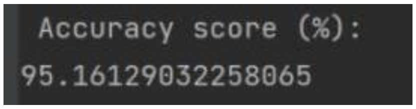
- **Evaluation**

To evaluate the performance of the system, we calculated the accuracy of the identification process with various number of face and fingerprints images for each person: 5, 10, 15, 20, 25, 30.



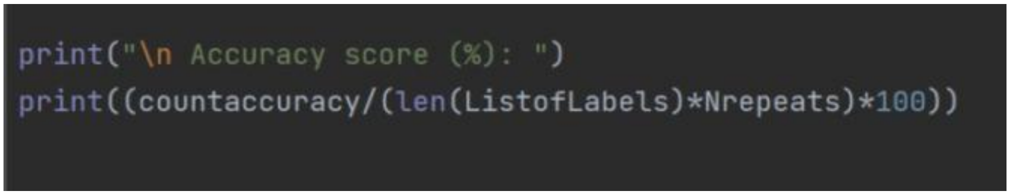
```
countaccuracy = 0
Nrepeats =30
```

Figure 5-2 Nrepeat



```
 Accuracy score (%):
95.16129032258065
```

Figure 5-3:accuracy



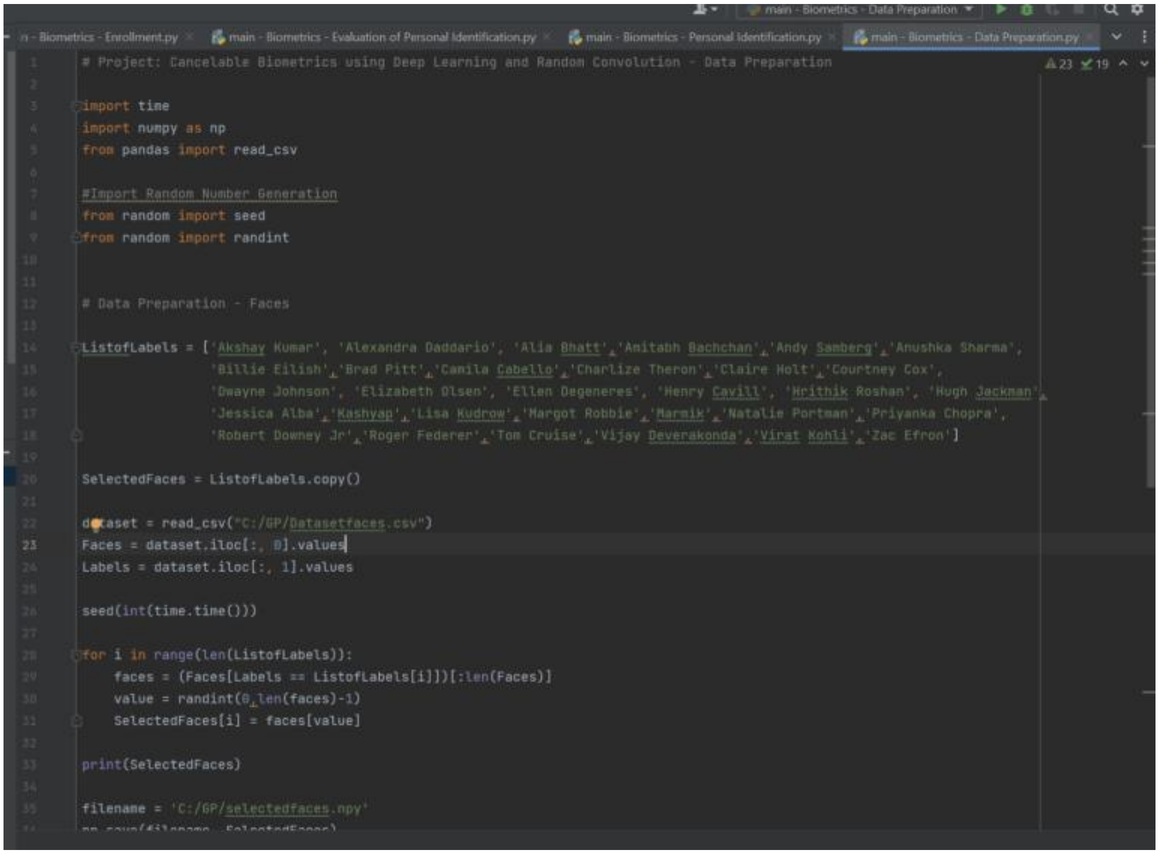
```
print("\n Accuracy score (%): ")
print((countaccuracy/(len(ListofLabels)*Nrepeats)*100))
```

Figure 5-4: Accuracy formula.

5.3 I/O screens

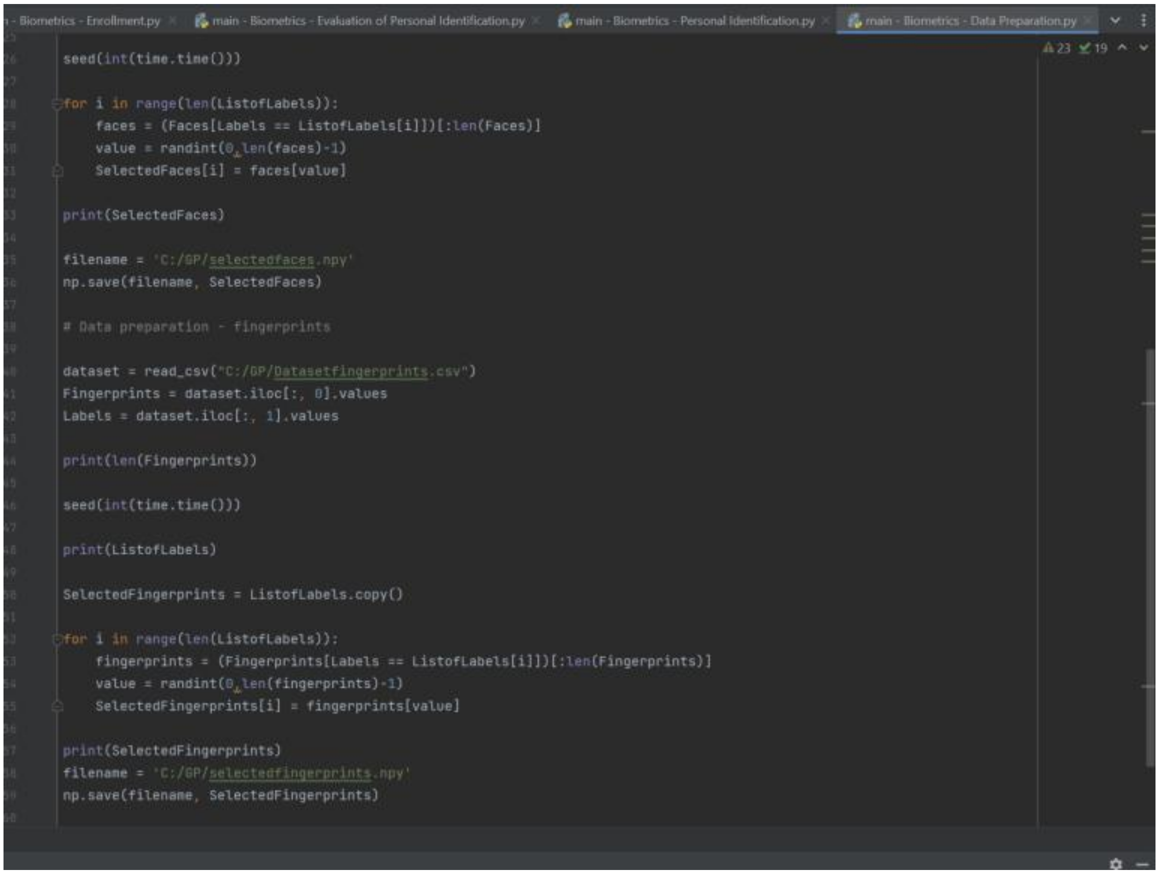
Data preparation

This is the first code we are going to run in our system Data preparation code will start by taking a random picture of the person face and their right index fingerprint form the dataset to be used in the enrollment code which is the next code.


main - Biometrics - Data Preparation
n - Biometrics - Enrollment.py
main - Biometrics - Evaluation of Personal Identification.py
main - Biometrics - Personal Identification.py
main - Biometrics - Data Preparation.py
23
19

```python
# Project: Cancelable Biometrics using Deep Learning and Random Convolution - Data Preparation

import time
import numpy as np
from pandas import read_csv

#Import Random Number Generation
from random import seed
from random import randint


# Data Preparation - Faces

ListofLabels = ['Akshay Kumar', 'Alexandra Daddario', 'Alia Bhatt','Amitabh Bachchan','Andy Samberg','Anushka Sharma',
                'Billie Eilish','Brad Pitt','Camila Cabello','Charlize Theron','Claire Holt','Courtney Cox',
                'Dwayne Johnson', 'Elizabeth Olsen', 'Ellen Degeneres', 'Henry Cavill', 'Hrithik Roshan', 'Hugh Jackman',
                'Jessica Alba','Kashyap','Lisa Kudrow','Margot Robbie','Marmik','Natalie Portman','Priyanka Chopra',
                'Robert Downey Jr','Roger Federer','Tom Cruise','Vijay Deverakonda','Virat Kohli','Zac Efron']

SelectedFaces = ListofLabels.copy()

dataset = read_csv("C:/GP/Datasetfaces.csv")
Faces = dataset.iloc[:, 0].values
Labels = dataset.iloc[:, 1].values

seed(int(time.time()))

for i in range(len(ListofLabels)):
    faces = (Faces[Labels == ListofLabels[i]])[:len(Faces)]
    value = randint(0,len(faces)-1)
    SelectedFaces[i] = faces[value]

print(SelectedFaces)

filename = 'C:/GP/selectedfaces.npy'
np.save(filename, SelectedFaces)
```

Figure 5-5:Dataperparation

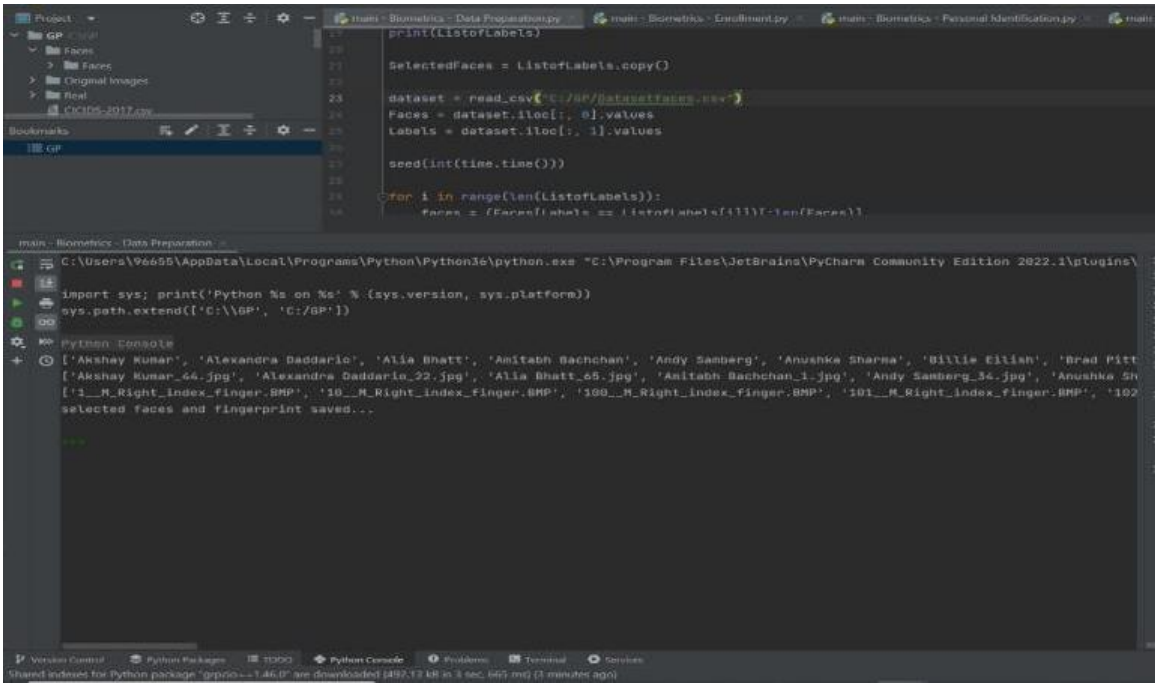

- Biometrics - Enrollment.py
main - Biometrics - Evaluation of Personal Identification.py
main - Biometrics - Personal Identification.py
main - Biometrics - Data Preparation.py
▲23 ✓19

```
seed(int(time.time()))

for i in range(len(ListofLabels)):
    faces = (Faces[Labels == ListofLabels[i]])[:len(Faces)]
    value = randint(0,len(faces)-1)
    SelectedFaces[i] = faces[value]

print(SelectedFaces)

filename = 'C:/GP/selectedfaces.npy'
np.save(filename, SelectedFaces)

# Data preparation - fingerprints

dataset = read_csv("C:/GP/Datasetfingerprints.csv")
Fingerprints = dataset.iloc[:, 0].values
Labels = dataset.iloc[:, 1].values

print(len(Fingerprints))

seed(int(time.time()))

print(ListofLabels)

SelectedFingerprints = ListofLabels.copy()

for i in range(len(ListofLabels)):
    fingerprints = (Fingerprints[Labels == ListofLabels[i]])[:len(Fingerprints)]
    value = randint(0,len(fingerprints)-1)
    SelectedFingerprints[i] = fingerprints[value]

print(SelectedFingerprints)
filename = 'C:/GP/selectedfingerprints.npy'
np.save(filename, SelectedFingerprints)
```

Figure 5-6:Dataperparation

Output

The first line of the output is the list of labels and the second line is faces pictures and the third line is the fingerprint of the right index for each person and it is organized the first fingerprint and the first face picture belong to the first person in the list of labels.



Project
GP
Faces
Faces
Original Images
Real
CICIDS-2017.csv
Bookmarks
GP
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Personal Identification.py
main - Biometrics - Data Preparation
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services

```
print(ListofLabels)

SelectedFaces = ListofLabels.copy()

dataset = read_csv("C:/GP/DatasetFaces.csv")
Faces = dataset.iloc[:, 0].values
Labels = dataset.iloc[:, 1].values

seed(int(time.time()))

for i in range(len(ListofLabels)):
    faces = (Faces[Labels == ListofLabels[i]])[:len(Faces)]
```

```
C:\Users\96655\AppData\Local\Programs\Python\Python36\python.exe "C:\Program Files\JetBrains\PyCharm Community Edition 2022.1\plugins\

import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['C:\\GP', 'C:/GP'])

Python Console
['Akshay Kumar', 'Alexandra Daddario', 'Alia Bhatt', 'Amitabh Bachchan', 'Andy Samberg', 'Anushka Sharma', 'Billie Eilish', 'Brad Pitt
['Akshay Kumar_44.jpg', 'Alexandra Daddario_22.jpg', 'Alia Bhatt_65.jpg', 'Amitabh Bachchan_1.jpg', 'Andy Samberg_34.jpg', 'Anushka Sh
['1__M_Right_index_finger.BMP', '10__M_Right_index_finger.BMP', '100__M_Right_index_finger.BMP', '101__M_Right_index_finger.BMP', '102
selected faces and fingerprint saved...

>>>
```

Shared indexes for Python package "grpcio==1.46.0" are downloaded (497.13 kB in 3 sec, 665 ms) (3 minutes ago)

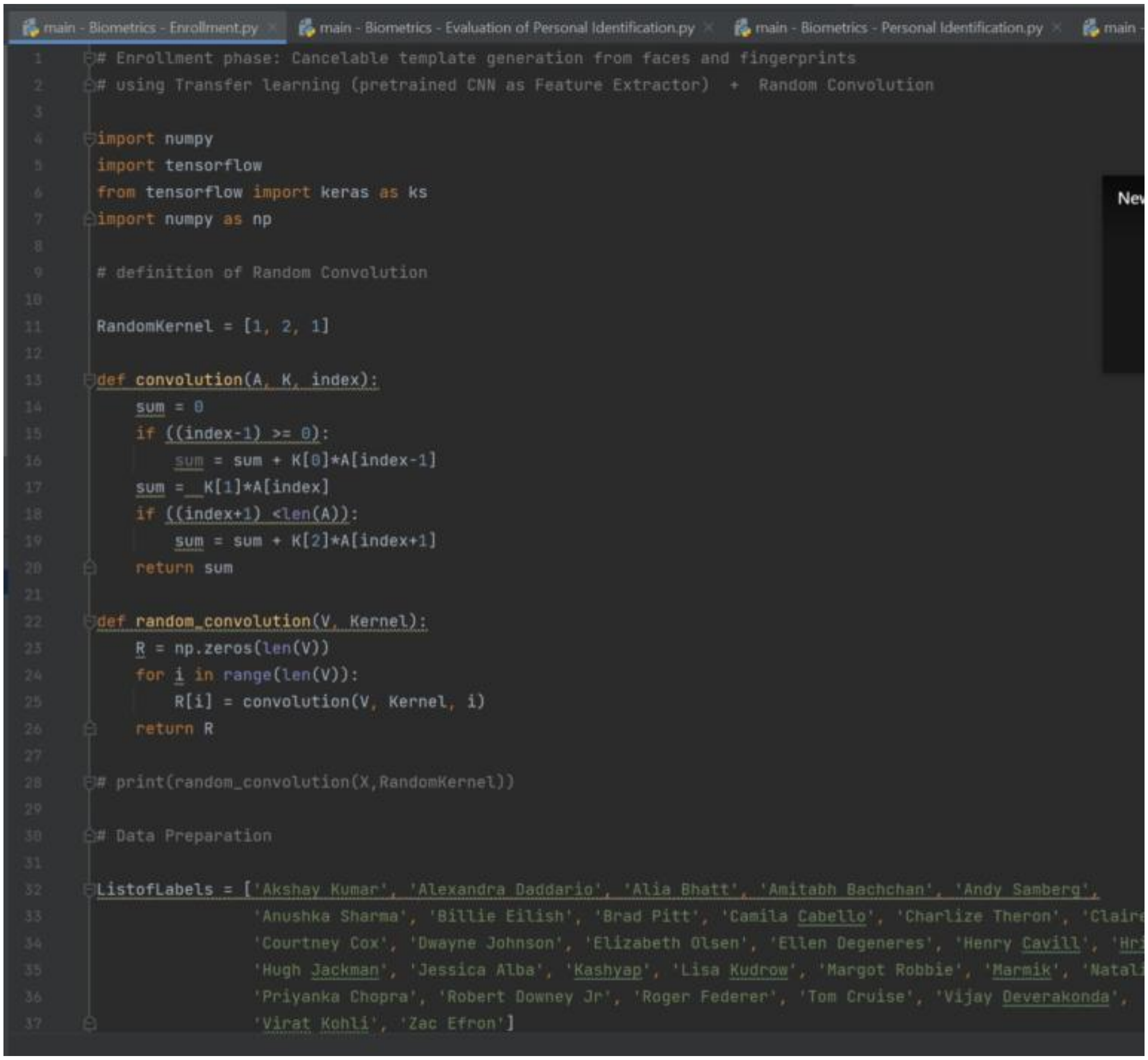
Figure 5-7:Dataperparation Output

Enrollment

In this code the Enrollment code it will start by reading two files the “SelectedFaces” file and the “SelectedFingerprints” file those files that have been used in the Data preparation code, it will extract the deep features of each picture then resize it by using CNN - Resnet50 so it will be suitable to be used, then we will use random convolution on each picture and merge them together to create a single cancelable template for each person.

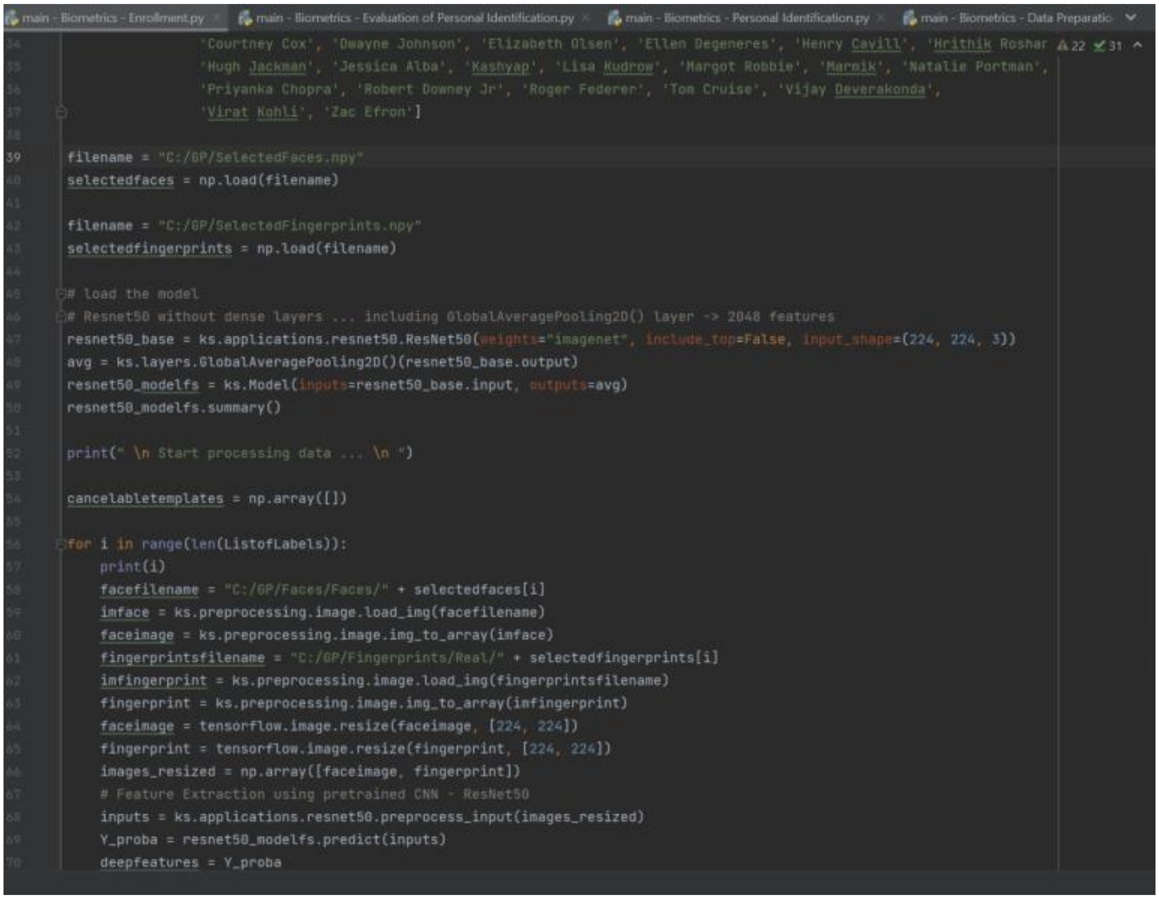
(1 face) + (1 fingerprint) = 1 cancelable template

The cancelable template will be saved in the “DB” file and the ID in the “IDs” file.


main - Biometrics - Enrollment.py
main - Biometrics - Evaluation of Personal Identification.py
main - Biometrics - Personal Identification.py
main -
Nev

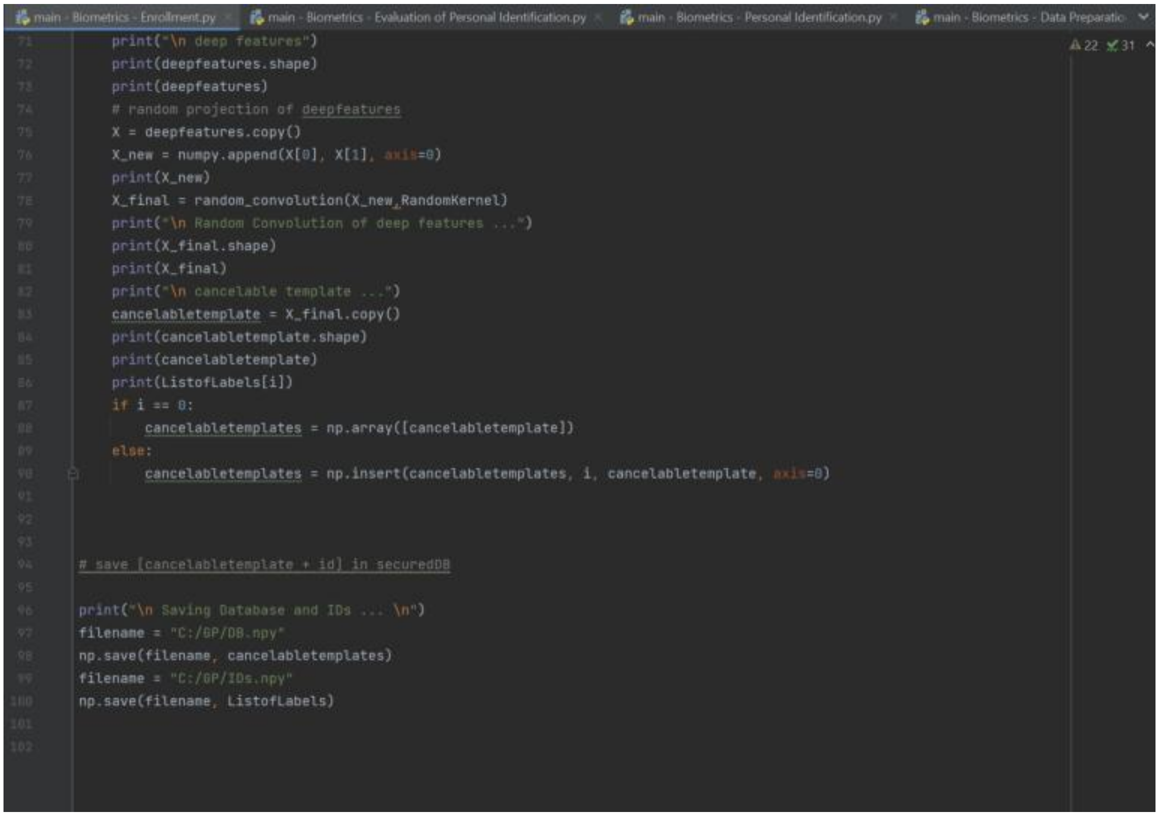
```
# Enrollment phase: Cancelable template generation from faces and fingerprints
# using Transfer learning (pretrained CNN as Feature Extractor)  +  Random Convolution

import numpy
import tensorflow
from tensorflow import keras as ks
import numpy as np

# definition of Random Convolution

RandomKernel = [1, 2, 1]

def convolution(A, K, index):
    sum = 0
    if ((index-1) >= 0):
        sum = sum + K[0]*A[index-1]
    sum = _K[1]*A[index]
    if ((index+1) <len(A)):
        sum = sum + K[2]*A[index+1]
    return sum

def random_convolution(V, Kernel):
    R = np.zeros(len(V))
    for i in range(len(V)):
        R[i] = convolution(V, Kernel, i)
    return R

# print(random_convolution(X,RandomKernel))

# Data Preparation

ListofLabels = ['Akshay Kumar', 'Alexandra Daddario', 'Alia Bhatt', 'Amitabh Bachchan', 'Andy Samberg',
                'Anushka Sharma', 'Billie Eilish', 'Brad Pitt', 'Camila Cabello', 'Charlize Theron', 'Claire
                'Courtney Cox', 'Dwayne Johnson', 'Elizabeth Olsen', 'Ellen Degeneres', 'Henry Cavill', 'Hri
                'Hugh Jackman', 'Jessica Alba', 'Kashyap', 'Lisa Kudrow', 'Margot Robbie', 'Marmik', 'Natali
                'Priyanka Chopra', 'Robert Downey Jr', 'Roger Federer', 'Tom Cruise', 'Vijay Deverakonda',
                'Virat Kohli', 'Zac Efron']
```

Figure 5-8: Enrollment


main - Biometrics - Enrollment.py
main - Biometrics - Evaluation of Personal Identification.py
main - Biometrics - Personal Identification.py
main - Biometrics - Data Preparatio
22
31

```
                 'Courtney Cox', 'Dwayne Johnson', 'Elizabeth Olsen', 'Ellen Degeneres', 'Henry Cavill', 'Hrithik Roshar
                 'Hugh Jackman', 'Jessica Alba', 'Kashyap', 'Lisa Kudrow', 'Margot Robbie', 'Marmik', 'Natalie Portman',
                 'Priyanka Chopra', 'Robert Downey Jr', 'Roger Federer', 'Tom Cruise', 'Vijay Deverakonda',
                 'Virat Kohli', 'Zac Efron']

filename = "C:/GP/SelectedFaces.npy"
selectedfaces = np.load(filename)

filename = "C:/GP/SelectedFingerprints.npy"
selectedfingerprints = np.load(filename)

# load the model
# Resnet50 without dense layers ... including GlobalAveragePooling2D() layer -> 2048 features
resnet50_base = ks.applications.resnet50.ResNet50(weights="imagenet", include_top=False, input_shape=(224, 224, 3))
avg = ks.layers.GlobalAveragePooling2D()(resnet50_base.output)
resnet50_modelfs = ks.Model(inputs=resnet50_base.input, outputs=avg)
resnet50_modelfs.summary()

print(" \n Start processing data ... \n ")

cancelabletemplates = np.array([])

for i in range(len(ListofLabels)):
    print(i)
    facefilename = "C:/GP/Faces/Faces/" + selectedfaces[i]
    imface = ks.preprocessing.image.load_img(facefilename)
    faceimage = ks.preprocessing.image.img_to_array(imface)
    fingerprintsfilename = "C:/GP/Fingerprints/Real/" + selectedfingerprints[i]
    imfingerprint = ks.preprocessing.image.load_img(fingerprintsfilename)
    fingerprint = ks.preprocessing.image.img_to_array(imfingerprint)
    faceimage = tensorflow.image.resize(faceimage, [224, 224])
    fingerprint = tensorflow.image.resize(fingerprint, [224, 224])
    images_resized = np.array([faceimage, fingerprint])
    # Feature Extraction using pretrained CNN - ResNet50
    inputs = ks.applications.resnet50.preprocess_input(images_resized)
    Y_proba = resnet50_modelfs.predict(inputs)
    deepfeatures = Y_proba
```

Figure 5-9: Enrollment



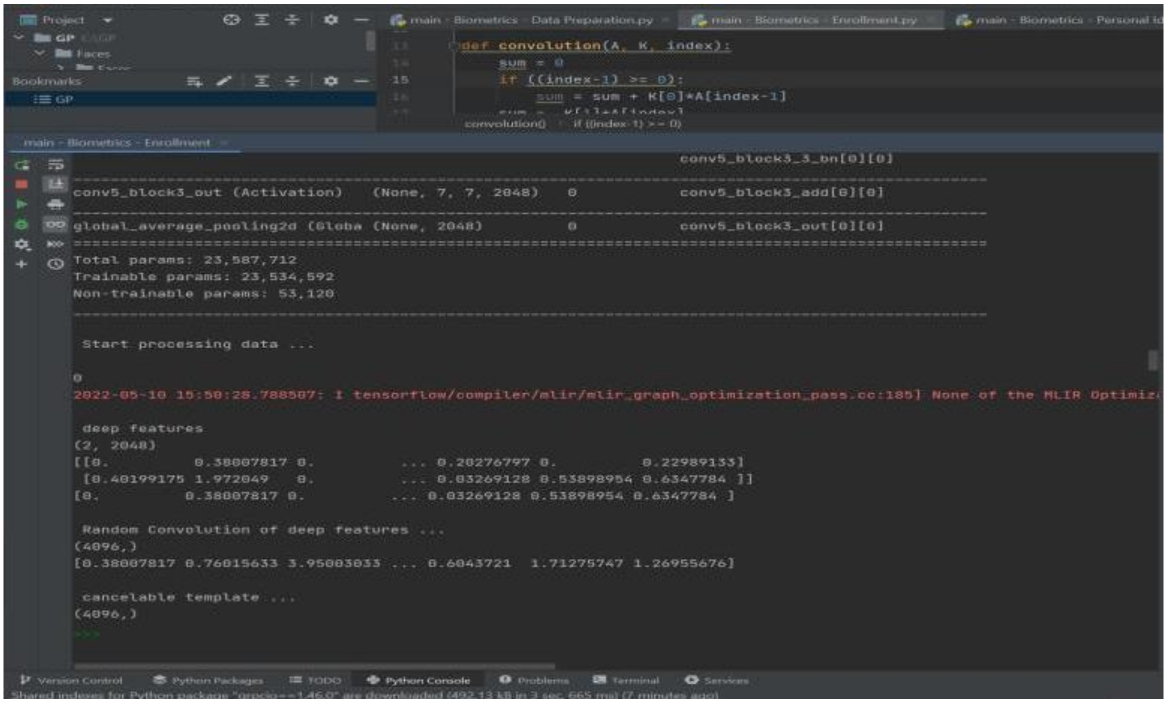
main - Biometrics - Enrollment.py
main - Biometrics - Evaluation of Personal Identification.py
main - Biometrics - Personal Identification.py
main - Biometrics - Data Preparatio
22
31

```
    print("\n deep features")
    print(deepfeatures.shape)
    print(deepfeatures)
    # random projection of deepfeatures
    X = deepfeatures.copy()
    X_new = numpy.append(X[0], X[1], axis=0)
    print(X_new)
    X_final = random_convolution(X_new,RandomKernel)
    print("\n Random Convolution of deep features ...")
    print(X_final.shape)
    print(X_final)
    print("\n cancelable template ...")
    cancelabletemplate = X_final.copy()
    print(cancelabletemplate.shape)
    print(cancelabletemplate)
    print(ListofLabels[i])
    if i == 0:
        cancelabletemplates = np.array([cancelabletemplate])
    else:
        cancelabletemplates = np.insert(cancelabletemplates, i, cancelabletemplate, axis=0)



# save [cancelabletemplate + id] in securedDB

print("\n Saving Database and IDs ... \n")
filename = "C:/GP/DB.npy"
np.save(filename, cancelabletemplates)
filename = "C:/GP/IDs.npy"
np.save(filename, ListofLabels)
```

Figure 5-10: Enrollment

Output

First we have the deep features extraction then the random convolution of deep features then we have the cancelable template and the name of each person and their ID at the end it will be Saved in “DB” file and “IDs” file.

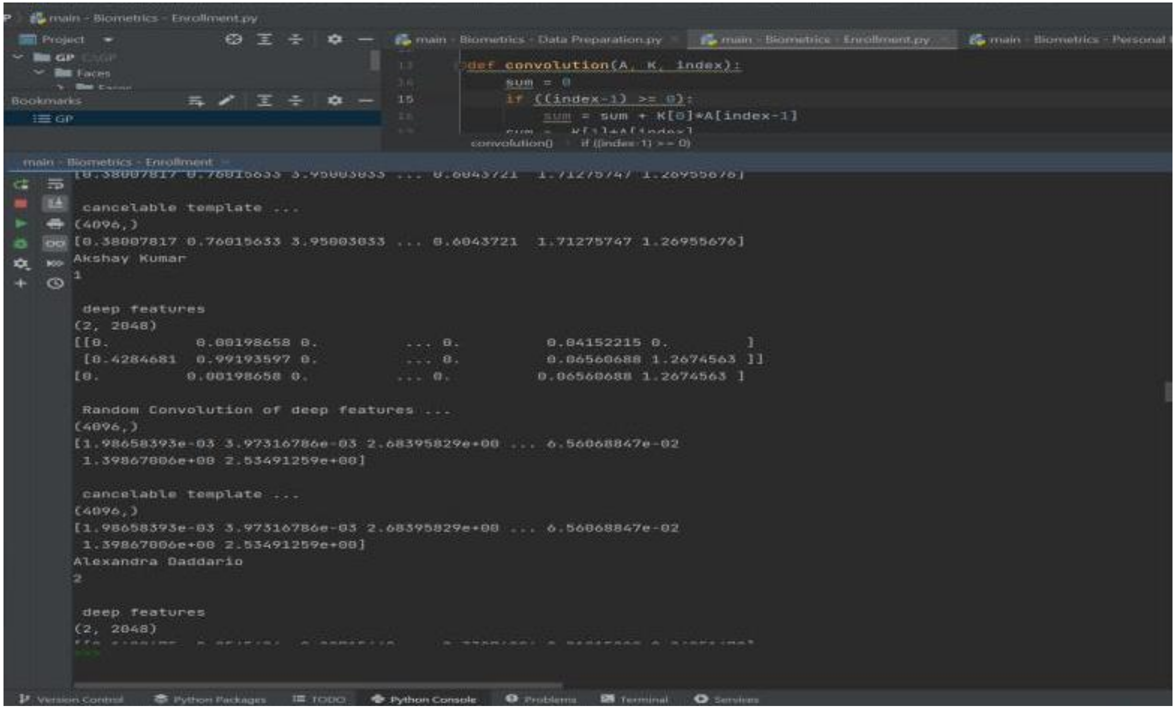

Project
GP
Faces
Bookmarks
GP
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Personal Id
convolution() if ((index-1) >= 0)
main - Biometrics - Enrollment
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services

```
def convolution(A, K, index):
    sum = 0
    if ((index-1) >= 0):
        sum = sum + K[0]*A[index-1]
```

```
                                                               conv5_block3_3_bn[0][0]
______________________________________________________________________________________________
conv5_block3_out (Activation)   (None, 7, 7, 2048)   0           conv5_block3_add[0][0]
______________________________________________________________________________________________
global_average_pooling2d (Globa (None, 2048)         0           conv5_block3_out[0][0]
==============================================================================================
Total params: 23,587,712
Trainable params: 23,534,592
Non-trainable params: 53,120
______________________________________________________________________________________________

 Start processing data ...

0
2022-05-10 15:50:28.788507: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR Optimiz

 deep features
(2, 2048)
[[0.         0.38007817 0.         ... 0.20276797 0.         0.22989133]
 [0.40199175 1.972049   0.         ... 0.03269128 0.53898954 0.6347784 ]]
[0.         0.38007817 0.         ... 0.03269128 0.53898954 0.6347784 ]

 Random Convolution of deep features ...
(4096,)
[0.38007817 0.76015633 3.95003033 ... 0.6043721  1.71275747 1.26955676]

 cancelable template ...
(4096,)
>>>
```

Shared indexes for Python package "grpcio==1.46.0" are downloaded (492.13 kB in 3 sec, 665 ms) (7 minutes ago)

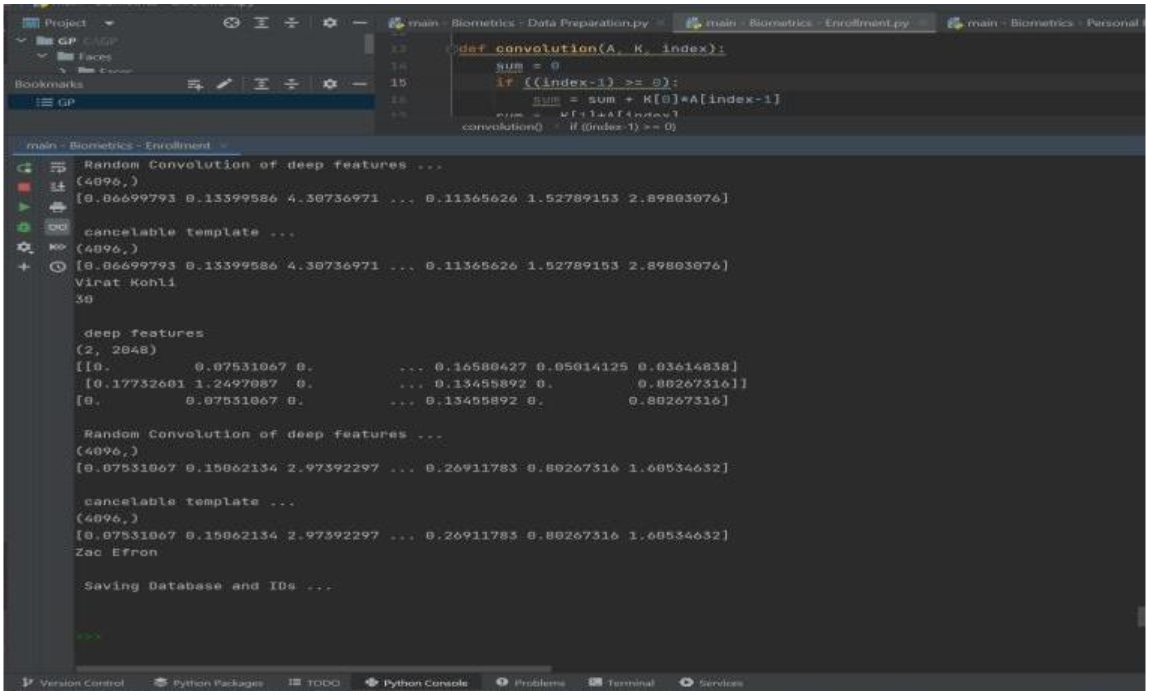
Figure 5-11: Enrollment output


main - Biometrics - Enrollment.py
Project
GP
Faces
Bookmarks
GP
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Personal
convolution() if ((index-1) >= 0)
main - Biometrics - Enrollment
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services

```
def convolution(A, K, index):
    sum = 0
    if ((index-1) >= 0):
        sum = sum + K[0]*A[index-1]
```

```
[0.38007817 0.76015633 3.95003033 ... 0.6043721  1.71275747 1.26955676]

 cancelable template ...
(4096,)
[0.38007817 0.76015633 3.95003033 ... 0.6043721  1.71275747 1.26955676]
Akshay Kumar
1

 deep features
(2, 2048)
[[0.         0.00198658 0.         ... 0.         0.04152215 0.        ]
 [0.4284681  0.99193597 0.         ... 0.         0.06560688 1.2674563 ]]
[0.         0.00198658 0.         ... 0.         0.06560688 1.2674563 ]

 Random Convolution of deep features ...
(4096,)
[1.98658393e-03 3.97316786e-03 2.68395829e+00 ... 6.56068847e-02
 1.39867806e+00 2.53491259e+00]

 cancelable template ...
(4096,)
[1.98658393e-03 3.97316786e-03 2.68395829e+00 ... 6.56068847e-02
 1.39867806e+00 2.53491259e+00]
Alexandra Daddario
2

 deep features
(2, 2048)
>>>
```

Figure 5-12: Enrollment output


Project
GP
Faces
Bookmarks
GP
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Personal
convolution() › if ((index-1) >= 0)
main - Biometrics - Enrollment
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services

```
def convolution(A, K, index):
    sum = 0
    if ((index-1) >= 0):
        sum = sum + K[0]*A[index-1]
```

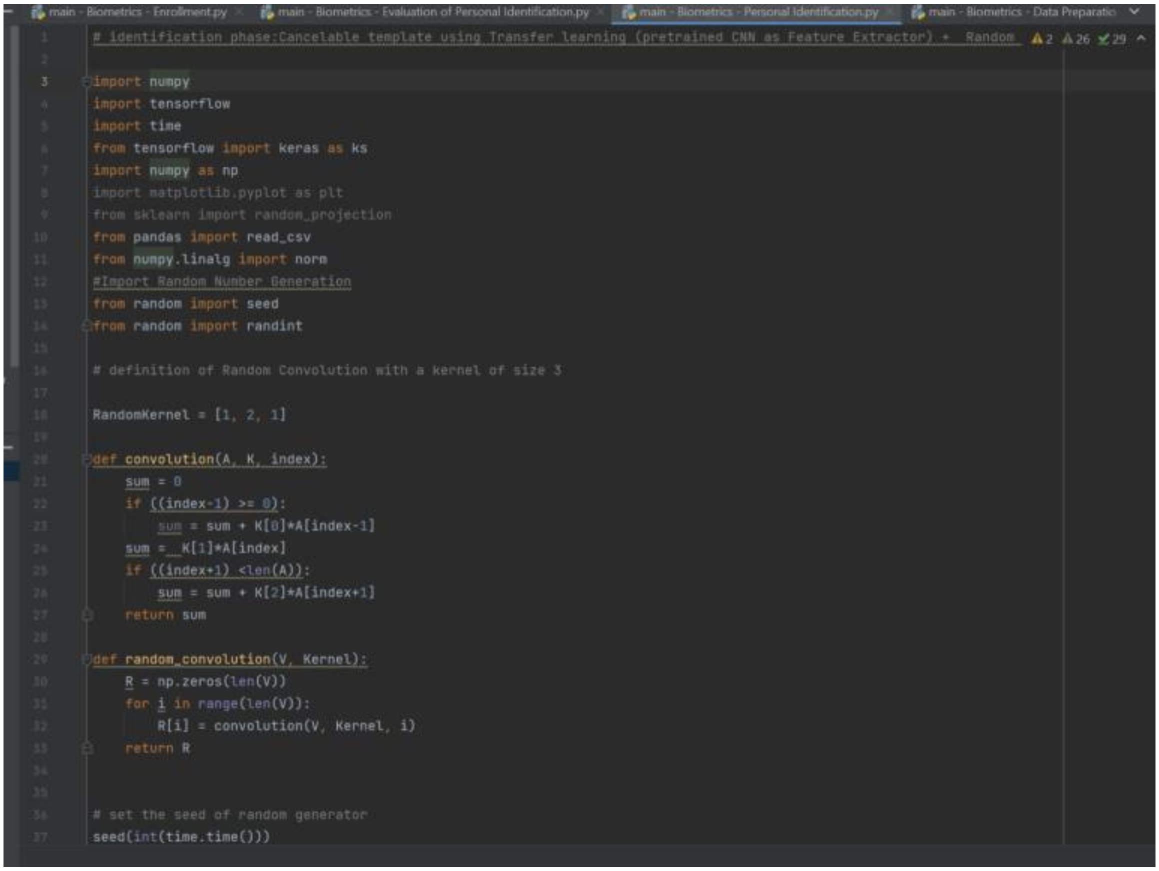
```
 Random Convolution of deep features ...
(4096,)
[0.86699793 0.13399586 4.30736971 ... 0.11365626 1.52789153 2.89803076]

 cancelable template ...
(4096,)
[0.86699793 0.13399586 4.30736971 ... 0.11365626 1.52789153 2.89803076]
Virat Kohli
30

 deep features
(2, 2048)
[[0.         0.07531067 0.         ... 0.16580427 0.05014125 0.03614838]
 [0.17732601 1.2497087  0.         ... 0.13455892 0.         0.80267316]]
[0.         0.07531067 0.         ... 0.13455892 0.         0.80267316]

 Random Convolution of deep features ...
(4096,)
[0.07531067 0.15062134 2.97392297 ... 0.26911783 0.80267316 1.60534632]

 cancelable template ...
(4096,)
[0.07531067 0.15062134 2.97392297 ... 0.26911783 0.80267316 1.60534632]
Zac Efron

 Saving Database and IDs ...


>>>
```

Figure 5-13: Enrollment output

Personal identification

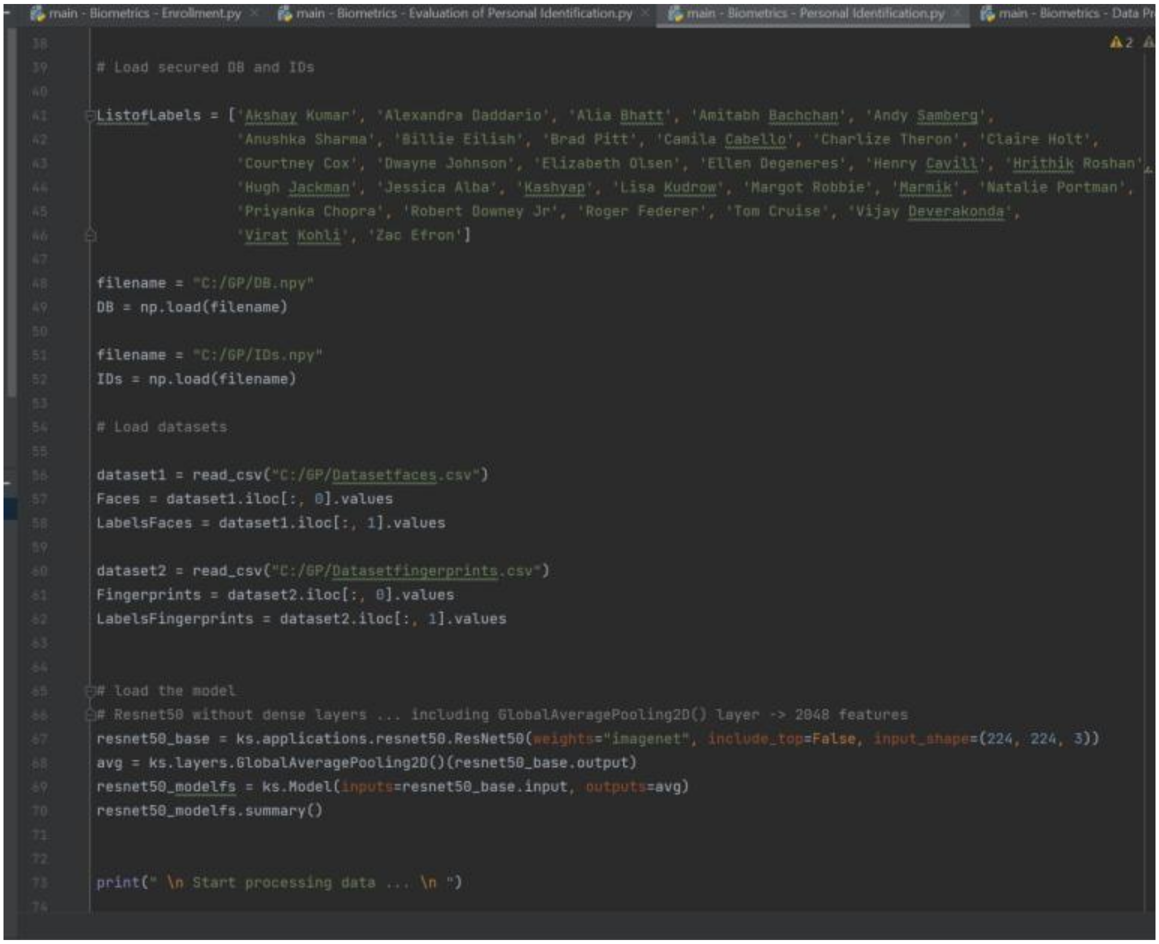
Our third code to be run is the personal identification code it will start by loading the "DB" file and the "IDs" file for each person.

Then it will start to select one random person form the "ListofLabels" every time we run the code and start the matching process for this person which is comparing between the dataset of the face picture and the fingerprint that have been selected in the Data preparation code and a the dataset of the persons and it will match with the nearest distance and the closest matches numbers that the system can find.


main - Biometrics - Enrollment.py
main - Biometrics - Evaluation of Personal Identification.py
main - Biometrics - Personal Identification.py
main - Biometrics - Data Preparatio
▲2 ▲26 ✓29

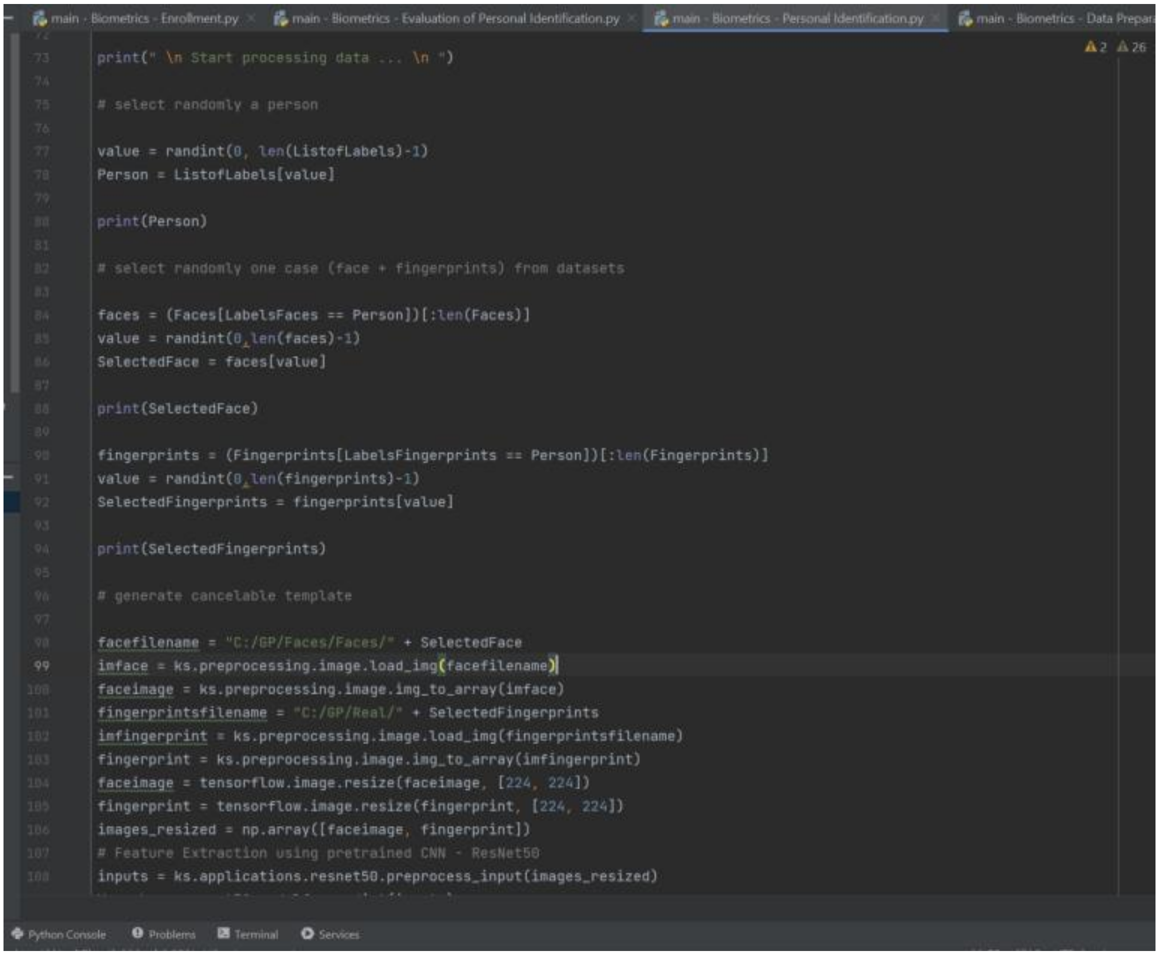
```
# identification phase:Cancelable template using Transfer learning (pretrained CNN as Feature Extractor) +  Random

import numpy
import tensorflow
import time
from tensorflow import keras as ks
import numpy as np
import matplotlib.pyplot as plt
from sklearn import random_projection
from pandas import read_csv
from numpy.linalg import norm
#Import Random Number Generation
from random import seed
from random import randint

# definition of Random Convolution with a kernel of size 3

RandomKernel = [1, 2, 1]

def convolution(A, K, index):
    sum = 0
    if ((index-1) >= 0):
        sum = sum + K[0]*A[index-1]
    sum = _K[1]*A[index]
    if ((index+1) <len(A)):
        sum = sum + K[2]*A[index+1]
    return sum

def random_convolution(V, Kernel):
    R = np.zeros(len(V))
    for i in range(len(V)):
        R[i] = convolution(V, Kernel, i)
    return R


# set the seed of random generator
seed(int(time.time()))
```

Figure 5-14: Personal identification.


main - Biometrics - Enrollment.py
main - Biometrics - Evaluation of Personal Identification.py
main - Biometrics - Personal Identification.py
main - Biometrics - Data Pr
2

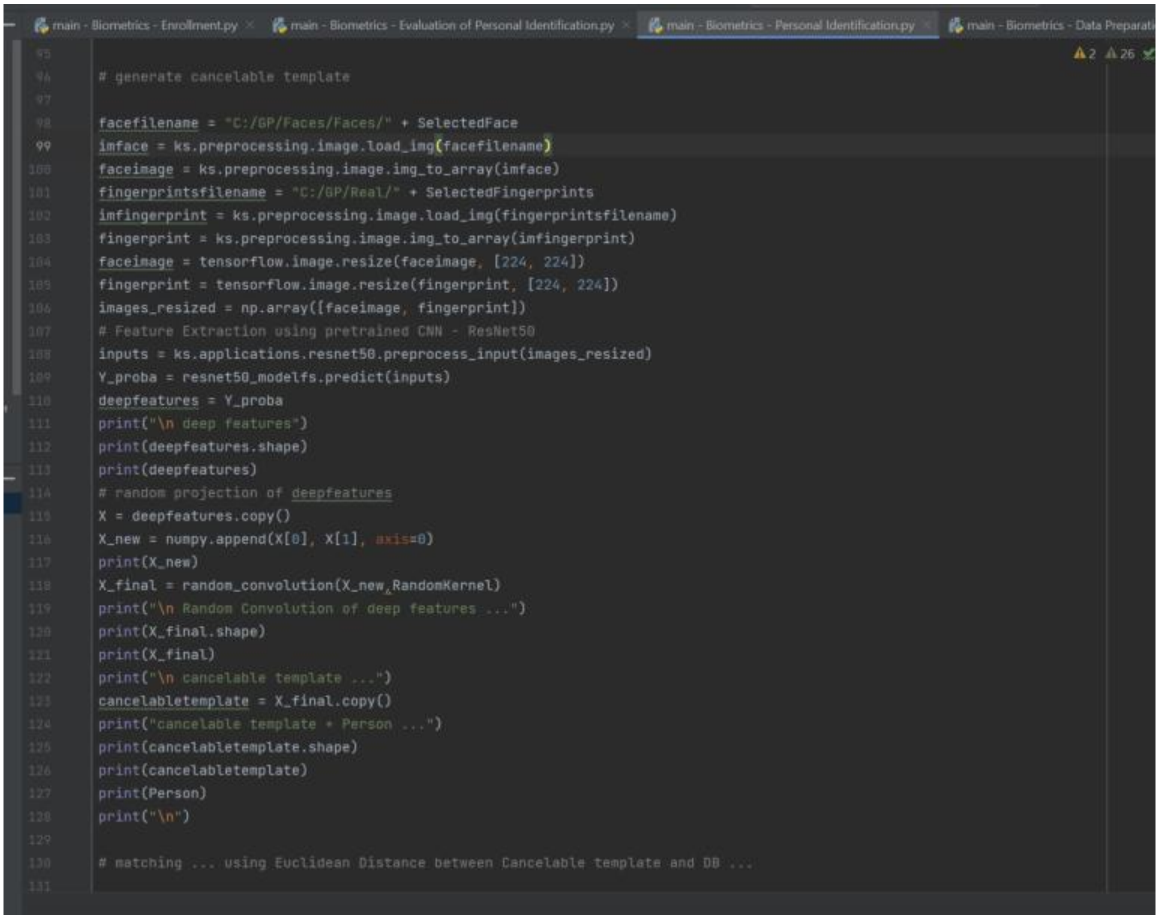
```python

# Load secured DB and IDs

ListofLabels = ['Akshay Kumar', 'Alexandra Daddario', 'Alia Bhatt', 'Amitabh Bachchan', 'Andy Samberg',
                'Anushka Sharma', 'Billie Eilish', 'Brad Pitt', 'Camila Cabello', 'Charlize Theron', 'Claire Holt',
                'Courtney Cox', 'Dwayne Johnson', 'Elizabeth Olsen', 'Ellen Degeneres', 'Henry Cavill', 'Hrithik Roshan',
                'Hugh Jackman', 'Jessica Alba', 'Kashyap', 'Lisa Kudrow', 'Margot Robbie', 'Marmik', 'Natalie Portman',
                'Priyanka Chopra', 'Robert Downey Jr', 'Roger Federer', 'Tom Cruise', 'Vijay Deverakonda',
                'Virat Kohli', 'Zac Efron']

filename = "C:/GP/DB.npy"
DB = np.load(filename)

filename = "C:/GP/IDs.npy"
IDs = np.load(filename)

# Load datasets

dataset1 = read_csv("C:/GP/Datasetfaces.csv")
Faces = dataset1.iloc[:, 0].values
LabelsFaces = dataset1.iloc[:, 1].values

dataset2 = read_csv("C:/GP/Datasetfingerprints.csv")
Fingerprints = dataset2.iloc[:, 0].values
LabelsFingerprints = dataset2.iloc[:, 1].values


# load the model
# Resnet50 without dense layers ... including GlobalAveragePooling2D() layer -> 2048 features
resnet50_base = ks.applications.resnet50.ResNet50(weights="imagenet", include_top=False, input_shape=(224, 224, 3))
avg = ks.layers.GlobalAveragePooling2D()(resnet50_base.output)
resnet50_modelfs = ks.Model(inputs=resnet50_base.input, outputs=avg)
resnet50_modelfs.summary()


print(" \n Start processing data ... \n ")

```

Figure 5-15: Personal identification.


main - Biometrics - Enrollment.py
main - Biometrics - Evaluation of Personal Identification.py
main - Biometrics - Personal Identification.py
main - Biometrics - Data Prepara
▲2 ▲26
Python Console
Problems
Terminal
Services

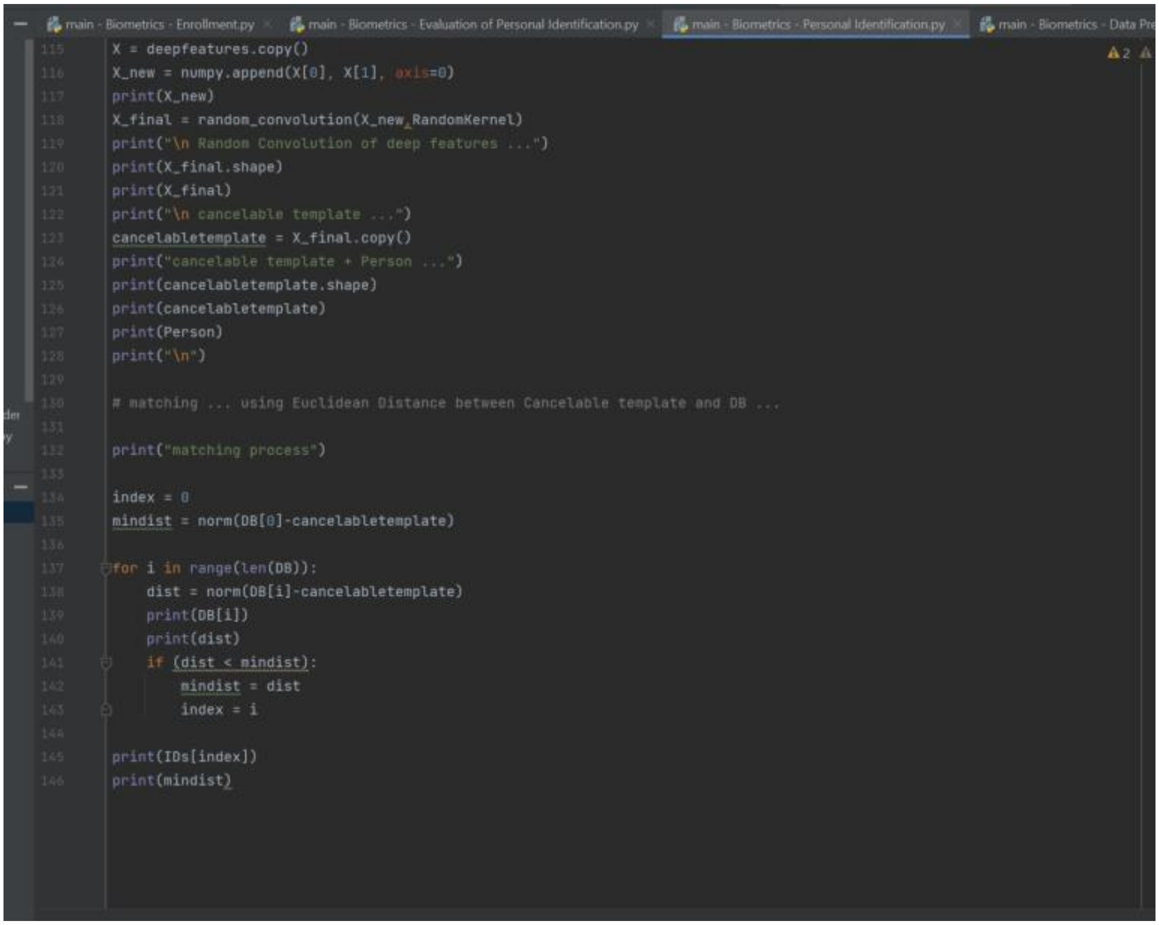
```
print(" \n Start processing data ... \n ")

# select randomly a person

value = randint(0, len(ListofLabels)-1)
Person = ListofLabels[value]

print(Person)

# select randomly one case (face + fingerprints) from datasets

faces = (Faces[LabelsFaces == Person])[:len(Faces)]
value = randint(0,len(faces)-1)
SelectedFace = faces[value]

print(SelectedFace)

fingerprints = (Fingerprints[LabelsFingerprints == Person])[:len(Fingerprints)]
value = randint(0,len(fingerprints)-1)
SelectedFingerprints = fingerprints[value]

print(SelectedFingerprints)

# generate cancelable template

facefilename = "C:/GP/Faces/Faces/" + SelectedFace
imface = ks.preprocessing.image.load_img(facefilename)
faceimage = ks.preprocessing.image.img_to_array(imface)
fingerprintsfilename = "C:/GP/Real/" + SelectedFingerprints
imfingerprint = ks.preprocessing.image.load_img(fingerprintsfilename)
fingerprint = ks.preprocessing.image.img_to_array(imfingerprint)
faceimage = tensorflow.image.resize(faceimage, [224, 224])
fingerprint = tensorflow.image.resize(fingerprint, [224, 224])
images_resized = np.array([faceimage, fingerprint])
# Feature Extraction using pretrained CNN - ResNet50
inputs = ks.applications.resnet50.preprocess_input(images_resized)
```

Figure 5-16: Personal identification.


main - Biometrics - Enrollment.py
main - Biometrics - Evaluation of Personal Identification.py
main - Biometrics - Personal Identification.py
main - Biometrics - Data Preparati
2
26

```

# generate cancelable template

facefilename = "C:/GP/Faces/Faces/" + SelectedFace
imface = ks.preprocessing.image.load_img(facefilename)
faceimage = ks.preprocessing.image.img_to_array(imface)
fingerprintsfilename = "C:/GP/Real/" + SelectedFingerprints
imfingerprint = ks.preprocessing.image.load_img(fingerprintsfilename)
fingerprint = ks.preprocessing.image.img_to_array(imfingerprint)
faceimage = tensorflow.image.resize(faceimage, [224, 224])
fingerprint = tensorflow.image.resize(fingerprint, [224, 224])
images_resized = np.array([faceimage, fingerprint])
# Feature Extraction using pretrained CNN - ResNet50
inputs = ks.applications.resnet50.preprocess_input(images_resized)
Y_proba = resnet50_modelfs.predict(inputs)
deepfeatures = Y_proba
print("\n deep features")
print(deepfeatures.shape)
print(deepfeatures)
# random projection of deepfeatures
X = deepfeatures.copy()
X_new = numpy.append(X[0], X[1], axis=0)
print(X_new)
X_final = random_convolution(X_new,RandomKernel)
print("\n Random Convolution of deep features ...")
print(X_final.shape)
print(X_final)
print("\n cancelable template ...")
cancelabletemplate = X_final.copy()
print("cancelable template + Person ...")
print(cancelabletemplate.shape)
print(cancelabletemplate)
print(Person)
print("\n")

# matching ... using Euclidean Distance between Cancelable template and DB ...

```

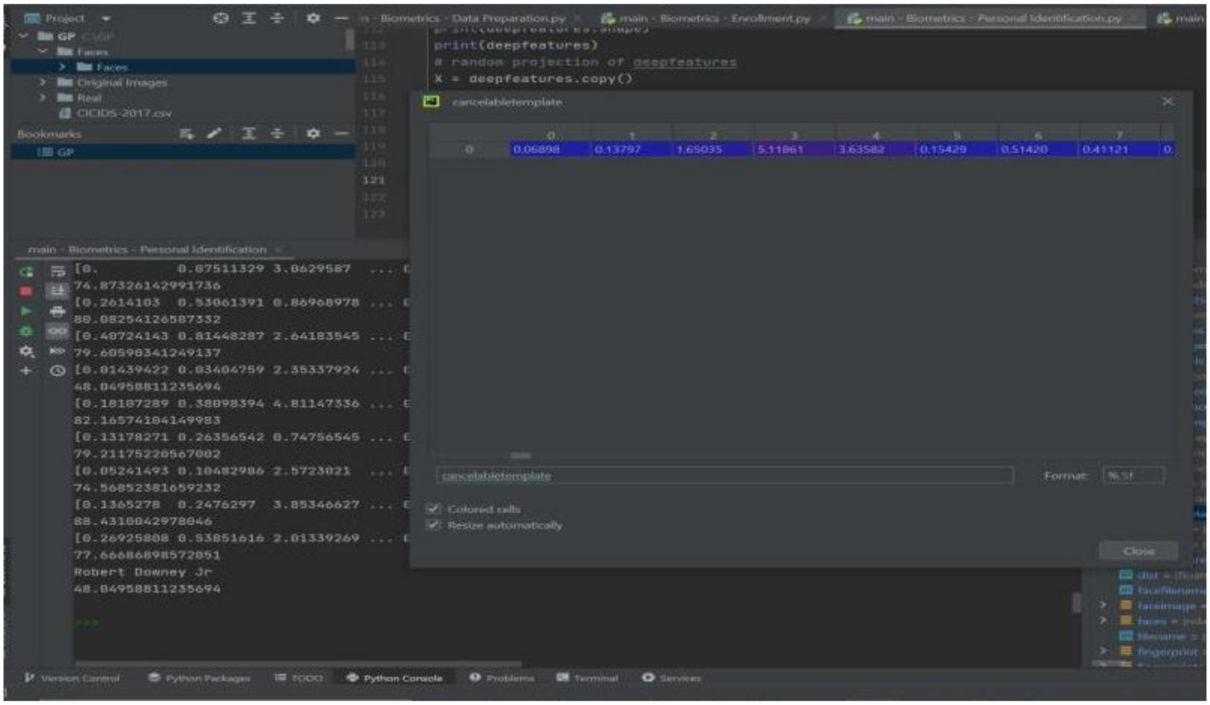
Figure 5-17: Personal identification.



main - Biometrics - Enrollment.py
main - Biometrics - Evaluation of Personal Identification.py
main - Biometrics - Personal Identification.py
main - Biometrics - Data Pre
⚠2

```python
X = deepfeatures.copy()
X_new = numpy.append(X[0], X[1], axis=0)
print(X_new)
X_final = random_convolution(X_new,RandomKernel)
print("\n Random Convolution of deep features ...")
print(X_final.shape)
print(X_final)
print("\n cancelable template ...")
cancelabletemplate = X_final.copy()
print("cancelable template + Person ...")
print(cancelabletemplate.shape)
print(cancelabletemplate)
print(Person)
print("\n")

# matching ... using Euclidean Distance between Cancelable template and DB ...

print("matching process")

index = 0
mindist = norm(DB[0]-cancelabletemplate)

for i in range(len(DB)):
    dist = norm(DB[i]-cancelabletemplate)
    print(DB[i])
    print(dist)
    if (dist < mindist):
        mindist = dist
        index = i

print(IDs[index])
print(mindist)
```

Figure 5-18: Personal identification.



```
print(deepfeatures)
# random projection of deepfeatures
X = deepfeatures.copy()
```

cancelabletemplate

	0	1	2	3	4	5	6	7
0	0.06898	0.13797	1.65035	5.11861	3.63582	0.15429	0.51420	0.41121

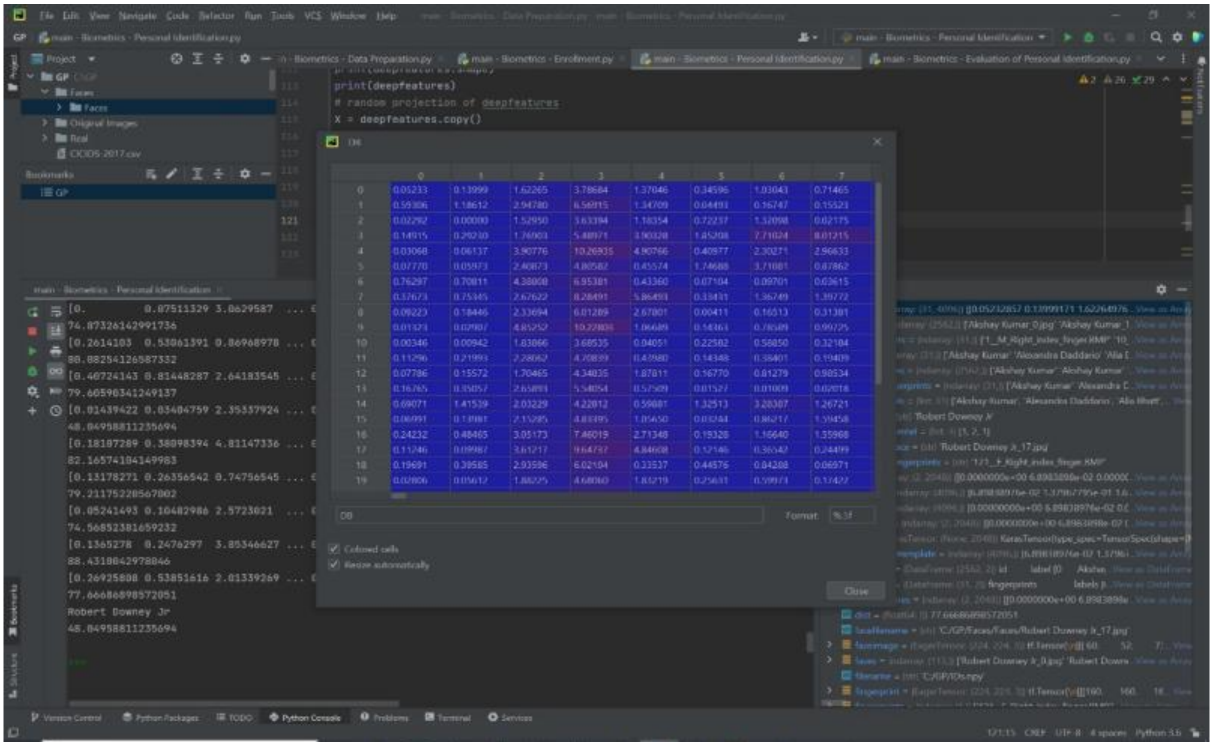
cancelabletemplate Format: %.5f

☑ Colored cells
☑ Resize automatically

```
[0.         0.07511329 3.0629587  ... 
74.87326142991736
[0.2614103  0.53061391 0.86968978 ... 
80.08254126587332
[0.40724143 0.81448287 2.64183545 ... 
79.60598341249137
[0.01439422 0.03404759 2.35337924 ... 
48.04958811235694
[0.18187289 0.38098394 4.81147336 ... 
82.16574104149983
[0.13178271 0.26356542 0.74756545 ... 
79.21175220567002
[0.05241493 0.10482986 2.5723021  ... 
74.56852381659232
[0.1365278  0.2476297  3.85346627 ... 
88.43100429780046
[0.26925808 0.53851616 2.01339269 ... 
77.66686898572051
Robert Downey Jr
48.04958811235694
```

Project
GP
Faces
Faces
Original Images
Real
CICIDS-2017.csv
Bookmarks
GP
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Personal Identification.py
main - Biometrics - Personal Identification
Close
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services

Figure 5-19: Personal identification.



```
print(deepfeatures)
# random projection of deepfeatures
X = deepfeatures.copy()
```

DB

	0	1	2	3	4	5	6	7
0	0.05233	0.13999	1.62265	3.78684	1.37046	0.34596	1.03043	0.71465
1	0.59306	1.18612	2.94780	6.56915	1.34709	0.04493	0.16747	0.15523
2	0.02292	0.00000	1.52950	3.63394	1.18354	0.72237	1.32098	0.02175
3	0.14915	0.29230	1.76903	5.48971	3.90328	1.85208	7.71024	8.01215
4	0.03068	0.06137	3.90776	10.26935	4.90766	0.40977	2.30271	2.96633
5	0.07770	0.05973	2.40873	4.80582	0.45574	1.74688	3.71081	0.87862
6	0.76297	0.70811	4.38008	6.95381	0.43360	0.07104	0.09701	0.03615
7	0.37673	0.75345	2.67622	8.28491	5.86493	0.33431	1.36749	1.39772
8	0.09223	0.18446	2.33694	6.01289	2.67001	0.00411	0.16513	0.31381
9	0.01323	0.02907	4.85252	10.22806	1.06689	0.14363	0.78589	0.99725
10	0.00346	0.00942	1.83866	3.68535	0.04051	0.22582	0.58850	0.32104
11	0.11296	0.21993	2.28062	4.70839	0.47980	0.14348	0.38401	0.19409
12	0.07786	0.15572	1.70465	4.34035	1.87811	0.16770	0.81279	0.98534
13	0.16765	0.35057	2.65893	5.54054	0.57509	0.01527	0.01009	0.02018
14	0.69071	1.41539	2.03229	4.22812	0.59881	1.32513	3.28387	1.26721
15	0.06991	0.13981	2.15285	4.83395	1.05450	0.03244	0.86217	1.59458
16	0.24232	0.48465	3.05173	7.46019	2.71348	0.19328	1.16640	1.55968
17	0.11246	0.09987	3.61217	9.64737	4.84608	0.12146	0.36542	0.24419
18	0.19691	0.39585	2.93596	6.02104	0.33537	0.44576	0.04208	0.06971
19	0.02806	0.05612	1.88225	4.68060	1.83219	0.25631	0.59973	0.17427

DB Format: %.5f

☑ Colored cells
☑ Resize automatically

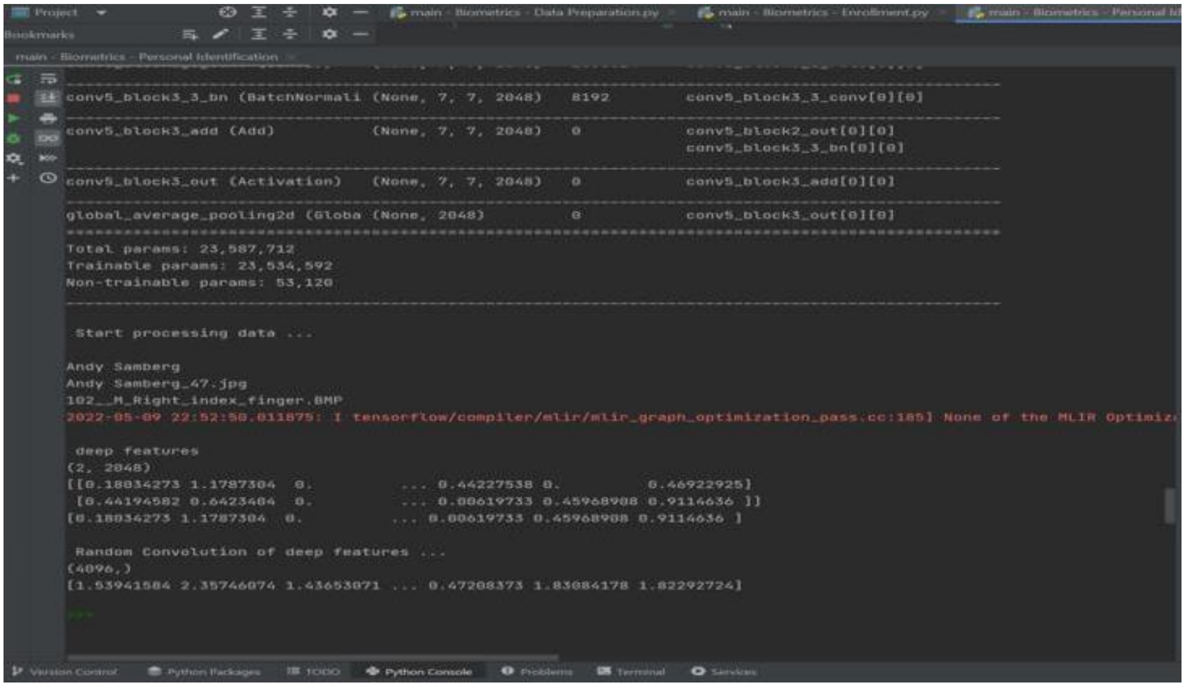
```
[0.         0.07511329 3.0629587  ... 
74.87326142991736
[0.2614103  0.53061391 0.86968978 ... 
80.08254126587332
[0.40724143 0.81448287 2.64183545 ... 
79.60598341249137
[0.01439422 0.03404759 2.35337924 ... 
48.04958811235694
[0.18187289 0.38098394 4.81147336 ... 
82.16574104149983
[0.13178271 0.26356542 0.74756545 ... 
79.21175220567002
[0.05241493 0.10482986 2.5723021  ... 
74.56852381659232
[0.1365278  0.2476297  3.85346627 ... 
88.43100429780046
[0.26925808 0.53851616 2.01339269 ... 
77.66686898572051
Robert Downey Jr
48.04958811235694
```

File Edit View Navigate Code Refactor Run Tools VCS Window Help
GP
main - Biometrics - Personal Identification.py
Project
Faces
Original Images
Real
CICIDS-2017.csv
Bookmarks
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Evaluation of Personal Identification.py
main - Biometrics - Personal Identification
Close
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services

Figure 5-20: Personal identification.

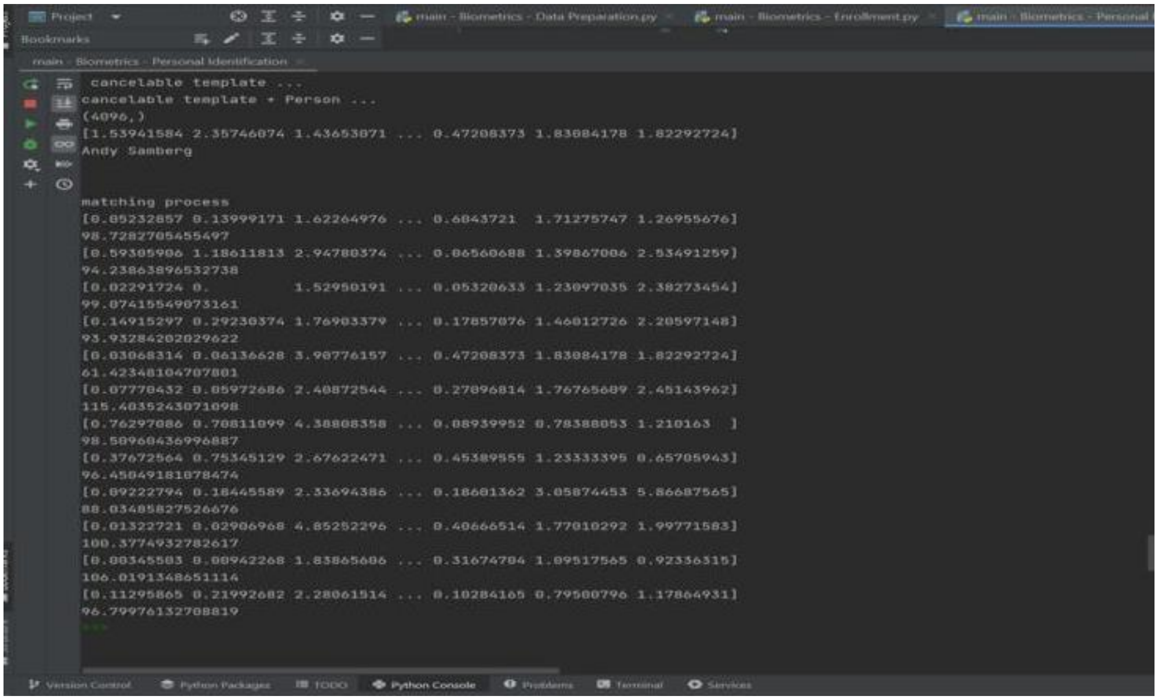
Output

The system will start processing data by choosing a random person and their face picture and fingerprint then it will start the deep features extraction then the random convolution of deep features then we have the cancelable template and the cancelable template plus person, finally the matching process will begin and at the end it will print the name of the person that have been chosen.


Project
Bookmarks
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Personal Id
main - Biometrics - Personal Identification
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services

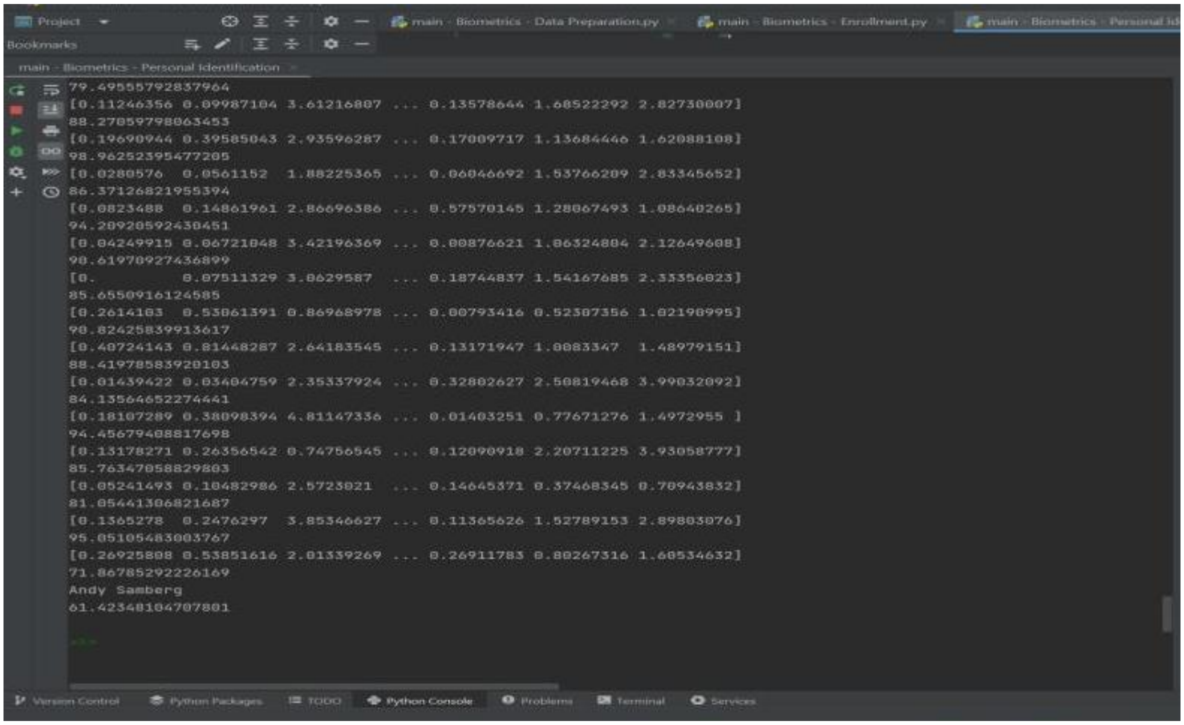
```
-----------------------------------------------------------------------------------------------
conv5_block3_3_bn (BatchNormali (None, 7, 7, 2048)   8192        conv5_block3_3_conv[0][0]
-----------------------------------------------------------------------------------------------
conv5_block3_add (Add)          (None, 7, 7, 2048)   0           conv5_block2_out[0][0]
                                                                 conv5_block3_3_bn[0][0]
-----------------------------------------------------------------------------------------------
conv5_block3_out (Activation)   (None, 7, 7, 2048)   0           conv5_block3_add[0][0]
-----------------------------------------------------------------------------------------------
global_average_pooling2d (Globa (None, 2048)         0           conv5_block3_out[0][0]
===============================================================================================
Total params: 23,587,712
Trainable params: 23,534,592
Non-trainable params: 53,120
-----------------------------------------------------------------------------------------------

 Start processing data ...

Andy Samberg
Andy Samberg_47.jpg
102__M_Right_index_finger.BMP
2022-05-09 22:52:50.011875: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR Optimiz

 deep features
(2, 2048)
[[0.18034273 1.1787304  0.         ... 0.44227538 0.         0.46922925]
 [0.44194582 0.6423404  0.         ... 0.00619733 0.45968908 0.9114636 ]]
[0.18034273 1.1787304  0.         ... 0.00619733 0.45968908 0.9114636 ]

 Random Convolution of deep features ...
(4096,)
[1.53941584 2.35746074 1.43653071 ... 0.47208373 1.83084178 1.82292724]
```

Figure 5-21: Personal identification output.


Project
Bookmarks
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Personal
main - Biometrics - Personal Identification
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services

```
 cancelable template ...
cancelable template + Person ...
(4096,)
[1.53941584 2.35746074 1.43653071 ... 0.47208373 1.83084178 1.82292724]
Andy Samberg


matching process
[0.05232857 0.13999171 1.62264976 ... 0.6043721  1.71275747 1.26955676]
98.7282705455497
[0.59305906 1.18611813 2.94780374 ... 0.06560688 1.39867006 2.53491259]
94.23863896532738
[0.02291724 0.         1.52950191 ... 0.05320633 1.23097035 2.38273454]
99.07415549073161
[0.14915297 0.29230374 1.76903379 ... 0.17857076 1.46012726 2.20597148]
93.93284202029622
[0.03068314 0.06136628 3.90776157 ... 0.47208373 1.83084178 1.82292724]
61.42348104707801
[0.07778432 0.05972686 2.40872544 ... 0.27896814 1.76765689 2.45143962]
115.4035243071098
[0.76297086 0.70811099 4.38808358 ... 0.08939952 0.78388053 1.210163  ]
98.58968436996887
[0.37672564 0.75345129 2.67622471 ... 0.45389555 1.23333395 0.65705943]
96.45049181078474
[0.09222794 0.18445589 2.33694386 ... 0.18601362 3.05874453 5.86687565]
88.03485827526676
[0.01322721 0.02906968 4.85252296 ... 0.40666514 1.77010292 1.99771583]
100.3774932782617
[0.00345503 0.00942268 1.83865606 ... 0.31674704 1.09517565 0.92336315]
106.0191348651114
[0.11295865 0.21992682 2.28061514 ... 0.10284165 0.79500796 1.17864931]
96.79976132708819
```

Figure 5-22: Personal identification output.


Project
Bookmarks
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Personal Id
main - Biometrics - Personal Identification
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services

```
79.49555792837964
[0.11246356 0.09987104 3.61216807 ... 0.13578644 1.68522292 2.82730007]
88.27059798063453
[0.19690944 0.39585043 2.93596287 ... 0.17009717 1.13684446 1.62088108]
98.96252395477205
[0.0280576  0.0561152  1.88225365 ... 0.06046692 1.53766209 2.83345652]
86.37126821955394
[0.0823488  0.14861961 2.86696386 ... 0.57570145 1.28067493 1.08640265]
94.20920592430451
[0.04249915 0.06721048 3.42196369 ... 0.00876621 1.86324804 2.12649608]
90.61970927436899
[0.         0.07511329 3.8629587  ... 0.18744837 1.54167685 2.33356023]
85.6550916124585
[0.2614103  0.53061391 0.86968978 ... 0.00793416 0.52307356 1.02190995]
90.82425839913617
[0.40724143 0.81448287 2.64183545 ... 0.13171947 1.0083347  1.48979151]
88.41978583920103
[0.01439422 0.03404759 2.35337924 ... 0.32802627 2.50819468 3.99032092]
84.13564652274441
[0.18107289 0.38098394 4.81147336 ... 0.01403251 0.77671276 1.4972955 ]
94.45679408817698
[0.13178271 0.26356542 0.74756545 ... 0.12090918 2.20711225 3.93058777]
85.76347058829803
[0.05241493 0.10482986 2.5723021  ... 0.14645371 0.37468345 0.70943832]
81.05441306821687
[0.1365278  0.2476297  3.85346627 ... 0.11365626 1.52789153 2.89803076]
95.05105483003767
[0.26925808 0.53851616 2.01339269 ... 0.26911783 0.80267316 1.60534632]
71.86785292226169
Andy Samberg
61.42348104707801

>>>
```

Figure 5-23: Personal identification output.

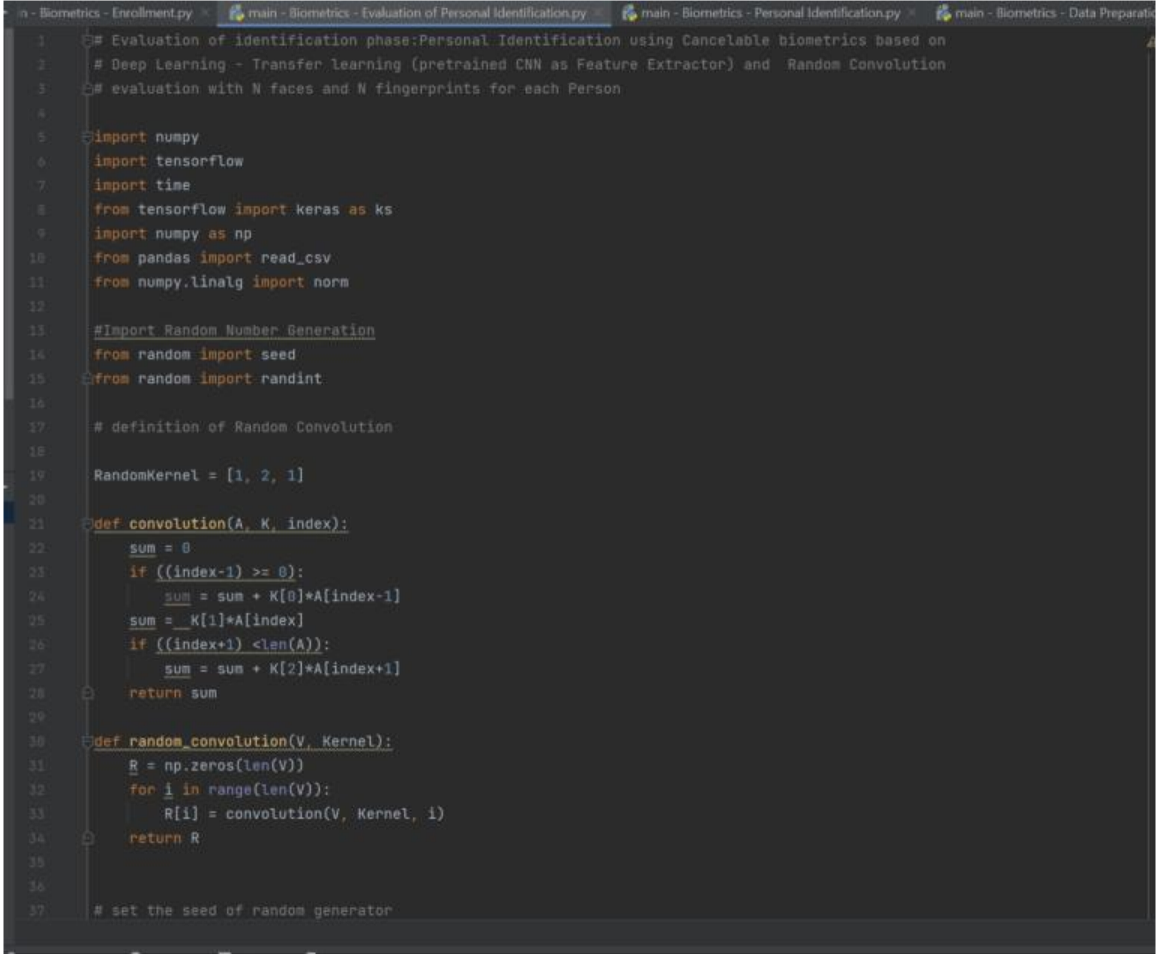
Evaluation

in this code we wanted to evaluate the accuracy of our system so we can make sure that it is efficient, it will start by loading from “DB” and "IDs” files then reading “Datasetsetfaces” and “Datasetfingerprints”.

The coutnaccuracy is the counter and it will start from 0 to the repeats which is 30 it will choose one random person (face + fingerprint) from the dataset and generate the cancelable template and Feature Extraction using pretrained CNN - ResNet50 and random convolution of deepfeatures.

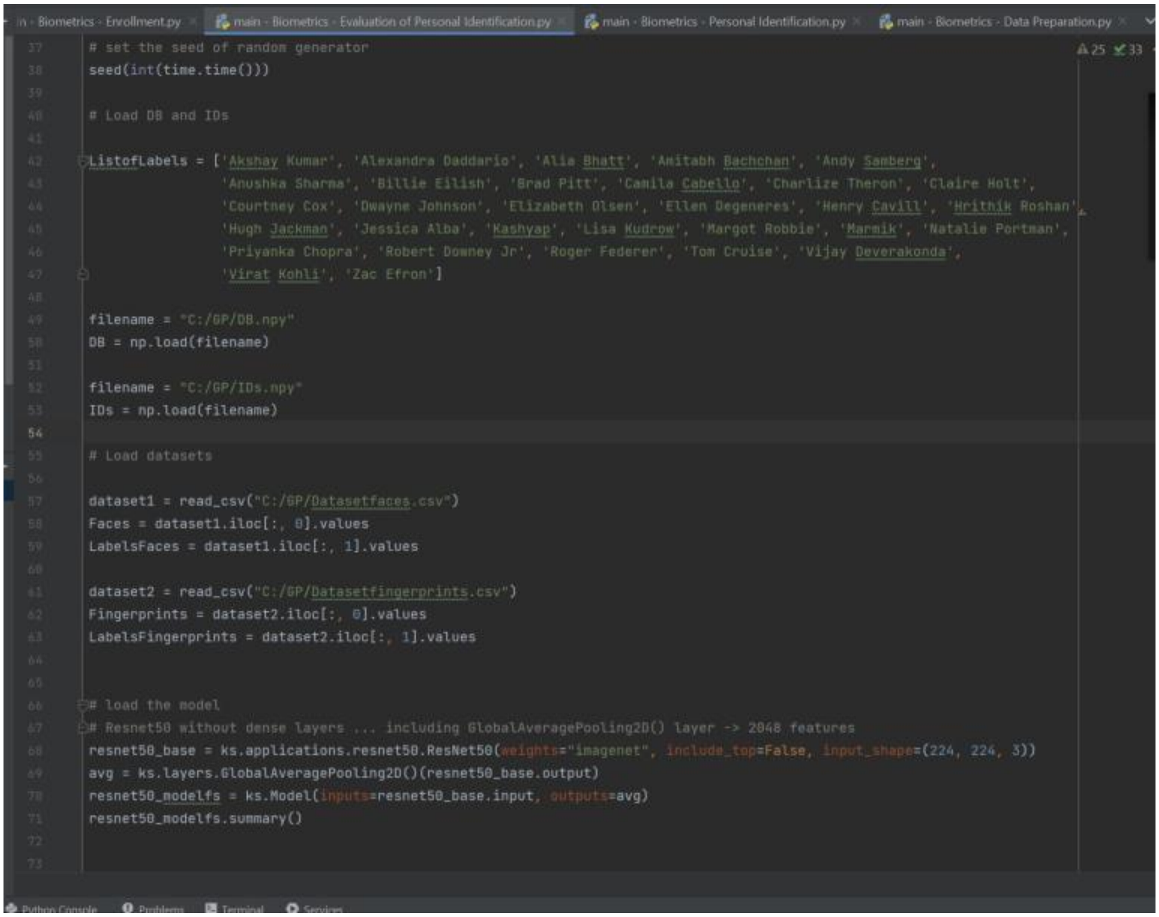
If the system found in the dataset a closer numbers to the choosen person it would be updated untill it find the closest one and as the distance get closer as much as better.

At the end we will evaluate the accuracy by using this formula ((countaccuracy/(len(Listoflabels)*Nrepeats)*100))


n - Biometrics - Enrollment.py
main - Biometrics - Evaluation of Personal Identification.py
main - Biometrics - Personal Identification.py
main - Biometrics - Data Preparatio

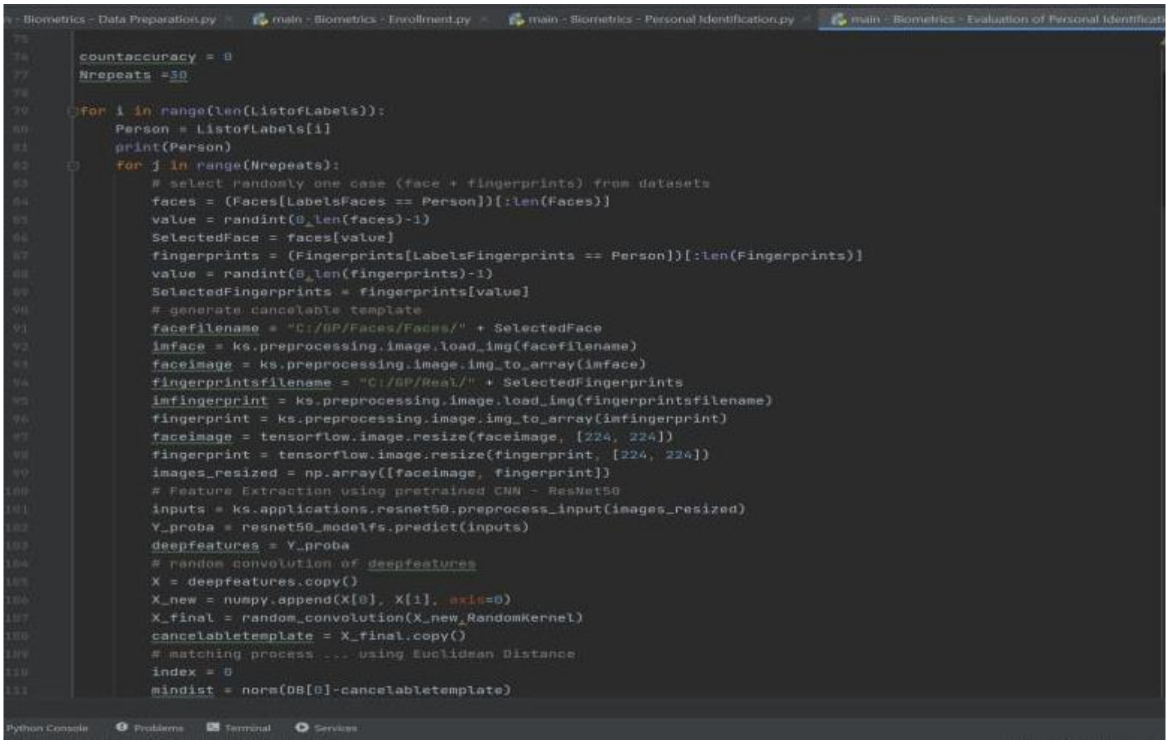
```python
# Evaluation of identification phase:Personal Identification using Cancelable biometrics based on
# Deep Learning - Transfer learning (pretrained CNN as Feature Extractor) and  Random Convolution
# evaluation with N faces and N fingerprints for each Person

import numpy
import tensorflow
import time
from tensorflow import keras as ks
import numpy as np
from pandas import read_csv
from numpy.linalg import norm

#Import Random Number Generation
from random import seed
from random import randint

# definition of Random Convolution

RandomKernel = [1, 2, 1]

def convolution(A, K, index):
    sum = 0
    if ((index-1) >= 0):
        sum = sum + K[0]*A[index-1]
    sum =  K[1]*A[index]
    if ((index+1) <len(A)):
        sum = sum + K[2]*A[index+1]
    return sum

def random_convolution(V, Kernel):
    R = np.zeros(len(V))
    for i in range(len(V)):
        R[i] = convolution(V, Kernel, i)
    return R


# set the seed of random generator
```

Figure 5-24: Evaluation..


n - Biometrics - Enrollment.py
main - Biometrics - Evaluation of Personal Identification.py
main - Biometrics - Personal Identification.py
main - Biometrics - Data Preparation.py
▲25 ✓33
Python Console
Problems
Terminal
Services

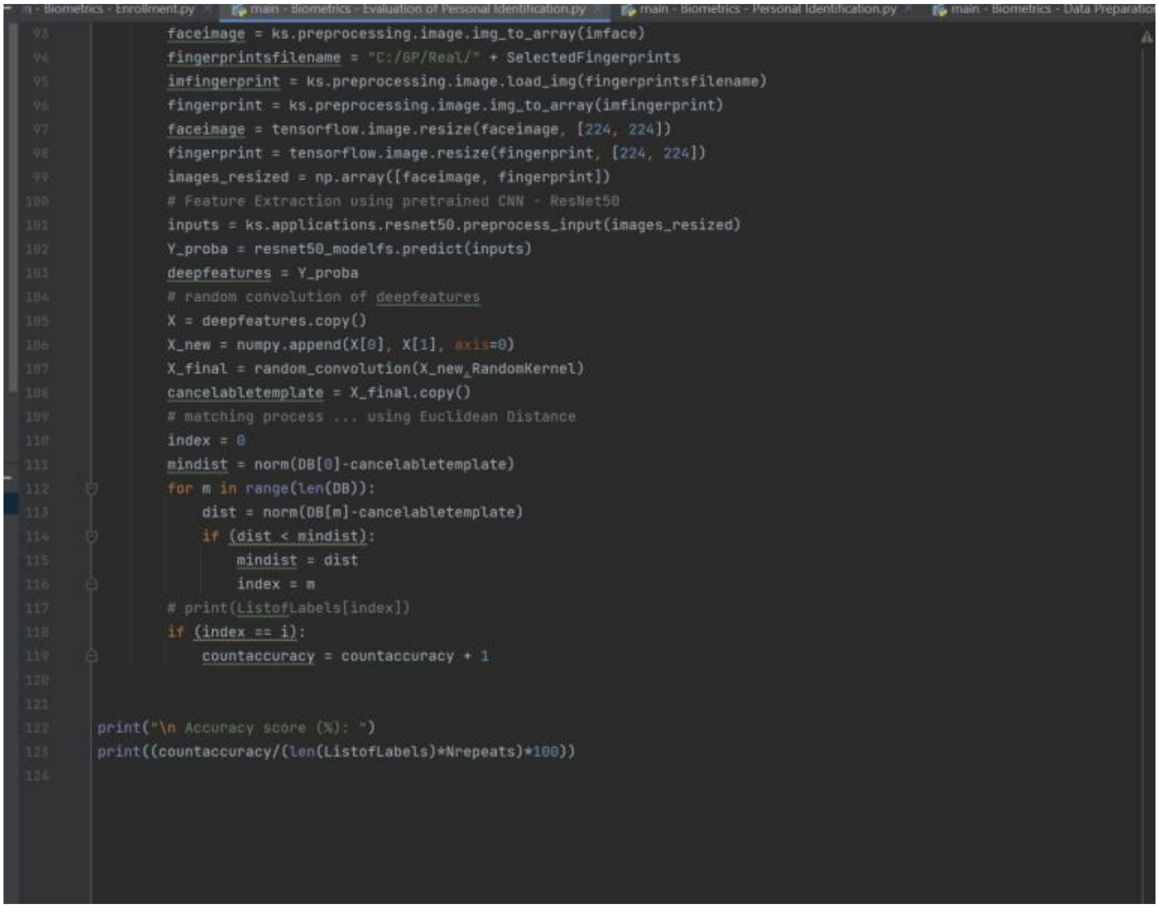
```python
# set the seed of random generator
seed(int(time.time()))

# Load DB and IDs

ListofLabels = ['Akshay Kumar', 'Alexandra Daddario', 'Alia Bhatt', 'Amitabh Bachchan', 'Andy Samberg',
                'Anushka Sharma', 'Billie Eilish', 'Brad Pitt', 'Camila Cabello', 'Charlize Theron', 'Claire Holt',
                'Courtney Cox', 'Dwayne Johnson', 'Elizabeth Olsen', 'Ellen Degeneres', 'Henry Cavill', 'Hrithik Roshan',
                'Hugh Jackman', 'Jessica Alba', 'Kashyap', 'Lisa Kudrow', 'Margot Robbie', 'Marmik', 'Natalie Portman',
                'Priyanka Chopra', 'Robert Downey Jr', 'Roger Federer', 'Tom Cruise', 'Vijay Deverakonda',
                'Virat Kohli', 'Zac Efron']

filename = "C:/GP/DB.npy"
DB = np.load(filename)

filename = "C:/GP/IDs.npy"
IDs = np.load(filename)

# Load datasets

dataset1 = read_csv("C:/GP/Datasetfaces.csv")
Faces = dataset1.iloc[:, 0].values
LabelsFaces = dataset1.iloc[:, 1].values

dataset2 = read_csv("C:/GP/Datasetfingerprints.csv")
Fingerprints = dataset2.iloc[:, 0].values
LabelsFingerprints = dataset2.iloc[:, 1].values


# load the model
# Resnet50 without dense layers ... including GlobalAveragePooling2D() layer -> 2048 features
resnet50_base = ks.applications.resnet50.ResNet50(weights="imagenet", include_top=False, input_shape=(224, 224, 3))
avg = ks.layers.GlobalAveragePooling2D()(resnet50_base.output)
resnet50_modelfs = ks.Model(inputs=resnet50_base.input, outputs=avg)
resnet50_modelfs.summary()
```

Figure 5-25: Evaluation..


- Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Personal Identification.py
main - Biometrics - Evaluation of Personal Identificati
Python Console
Problems
Terminal
Services

```python
countaccuracy = 0
Nrepeats =30

for i in range(len(ListofLabels)):
    Person = ListofLabels[i]
    print(Person)
    for j in range(Nrepeats):
        # select randomly one case (face + fingerprints) from datasets
        faces = (Faces[LabelsFaces == Person])[:len(Faces)]
        value = randint(0,len(faces)-1)
        SelectedFace = faces[value]
        fingerprints = (Fingerprints[LabelsFingerprints == Person])[:len(Fingerprints)]
        value = randint(0,len(fingerprints)-1)
        SelectedFingerprints = fingerprints[value]
        # generate cancelable template
        facefilename = "C:/GP/Faces/Faces/" + SelectedFace
        imface = ks.preprocessing.image.load_img(facefilename)
        faceimage = ks.preprocessing.image.img_to_array(imface)
        fingerprintsfilename = "C:/GP/Real/" + SelectedFingerprints
        imfingerprint = ks.preprocessing.image.load_img(fingerprintsfilename)
        fingerprint = ks.preprocessing.image.img_to_array(imfingerprint)
        faceimage = tensorflow.image.resize(faceimage, [224, 224])
        fingerprint = tensorflow.image.resize(fingerprint, [224, 224])
        images_resized = np.array([faceimage, fingerprint])
        # Feature Extraction using pretrained CNN - ResNet50
        inputs = ks.applications.resnet50.preprocess_input(images_resized)
        Y_proba = resnet50_modelfs.predict(inputs)
        deepfeatures = Y_proba
        # random convolution of deepfeatures
        X = deepfeatures.copy()
        X_new = numpy.append(X[0], X[1], axis=0)
        X_final = random_convolution(X_new,RandomKernel)
        cancelabletemplate = X_final.copy()
        # matching process ... using Euclidean Distance
        index = 0
        mindist = norm(DB[0]-cancelabletemplate)
```

Figure 5-26: Evaluation..



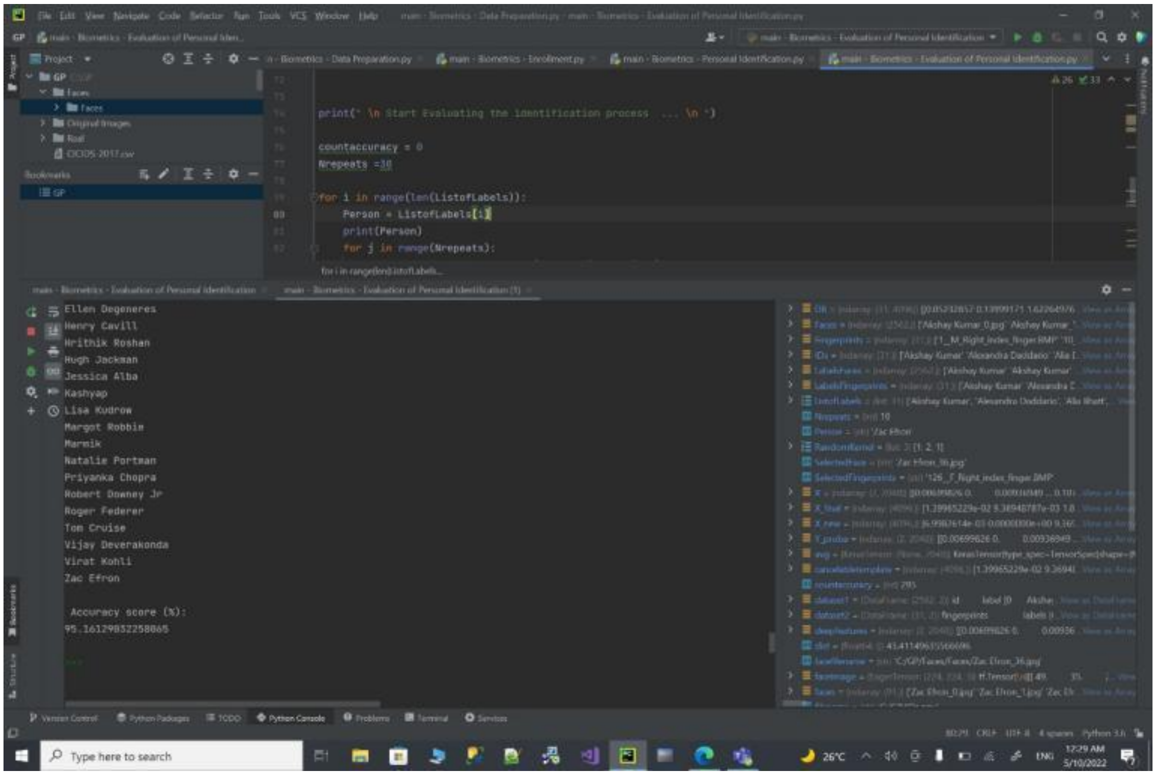
n - Biometrics - Enrollment.py
main - Biometrics - Evaluation of Personal Identification.py
main - Biometrics - Personal Identification.py
main - Biometrics - Data Preparatio

```python
        faceimage = ks.preprocessing.image.img_to_array(imface)
        fingerprintsfilename = "C:/GP/Real/" + SelectedFingerprints
        imfingerprint = ks.preprocessing.image.load_img(fingerprintsfilename)
        fingerprint = ks.preprocessing.image.img_to_array(imfingerprint)
        faceimage = tensorflow.image.resize(faceimage, [224, 224])
        fingerprint = tensorflow.image.resize(fingerprint, [224, 224])
        images_resized = np.array([faceimage, fingerprint])
        # Feature Extraction using pretrained CNN - ResNet50
        inputs = ks.applications.resnet50.preprocess_input(images_resized)
        Y_proba = resnet50_modelfs.predict(inputs)
        deepfeatures = Y_proba
        # random convolution of deepfeatures
        X = deepfeatures.copy()
        X_new = numpy.append(X[0], X[1], axis=0)
        X_final = random_convolution(X_new,RandomKernel)
        cancelabletemplate = X_final.copy()
        # matching process ... using Euclidean Distance
        index = 0
        mindist = norm(DB[0]-cancelabletemplate)
        for m in range(len(DB)):
            dist = norm(DB[m]-cancelabletemplate)
            if (dist < mindist):
                mindist = dist
                index = m
        # print(ListofLabels[index])
        if (index == i):
            countaccuracy = countaccuracy + 1


print("\n Accuracy score (%): ")
print((countaccuracy/(len(ListofLabels)*Nrepeats)*100))
```

Figure 5-27: Evaluation..

Output

High accuracy when choosing 30 repeats



```
print(" \n Start Evaluating the identification process  ... \n ")

countaccuracy = 0
Nrepeats =30

for i in range(len(ListofLabels)):
    Person = ListofLabels[i]
    print(Person)
    for j in range(Nrepeats):
```

```
Ellen Degeneres
Henry Cavill
Hrithik Roshan
Hugh Jackman
Jessica Alba
Kashyap
Lisa Kudrow
Margot Robbie
Marmik
Natalie Portman
Priyanka Chopra
Robert Downey Jr
Roger Federer
Tom Cruise
Vijay Deverakonda
Virat Kohli
Zac Efron

 Accuracy score (%):
95.16129032258065
```

Figure 5-28: Evaluation output

Chapter 6:Testing

6.1 Test plan

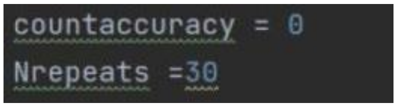
For our testing plan we chose the accuracy test because it is popular among supervised systems.

And also with higher accuracy we will get great and excellent results**.**

we will create six cases to test the level of accuracy and evaluate the performance of the system.

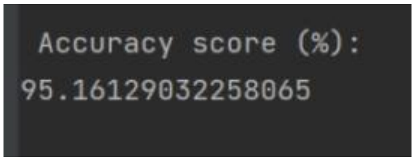
We have tried (5,10,15,20,25,30) repeats and the best final accuracy was excellent which is 30 repeats.

That was our test plan in general**.**



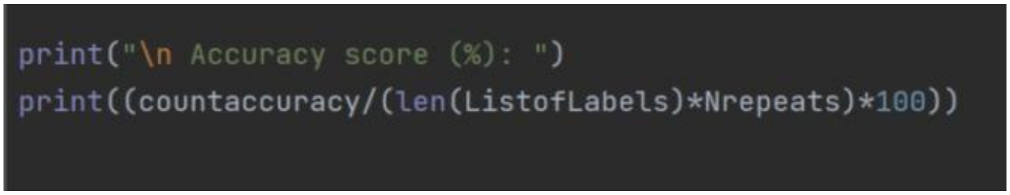
```
countaccuracy = 0
Nrepeats =30
```

Figure 6-1:Nrepeats.



```
 Accuracy score (%):
95.16129032258065
```

Figure 6-2:Accuracy.



```
print("\n Accuracy score (%): ")
print((countaccuracy/(len(ListofLabels)*Nrepeats)*100))
```

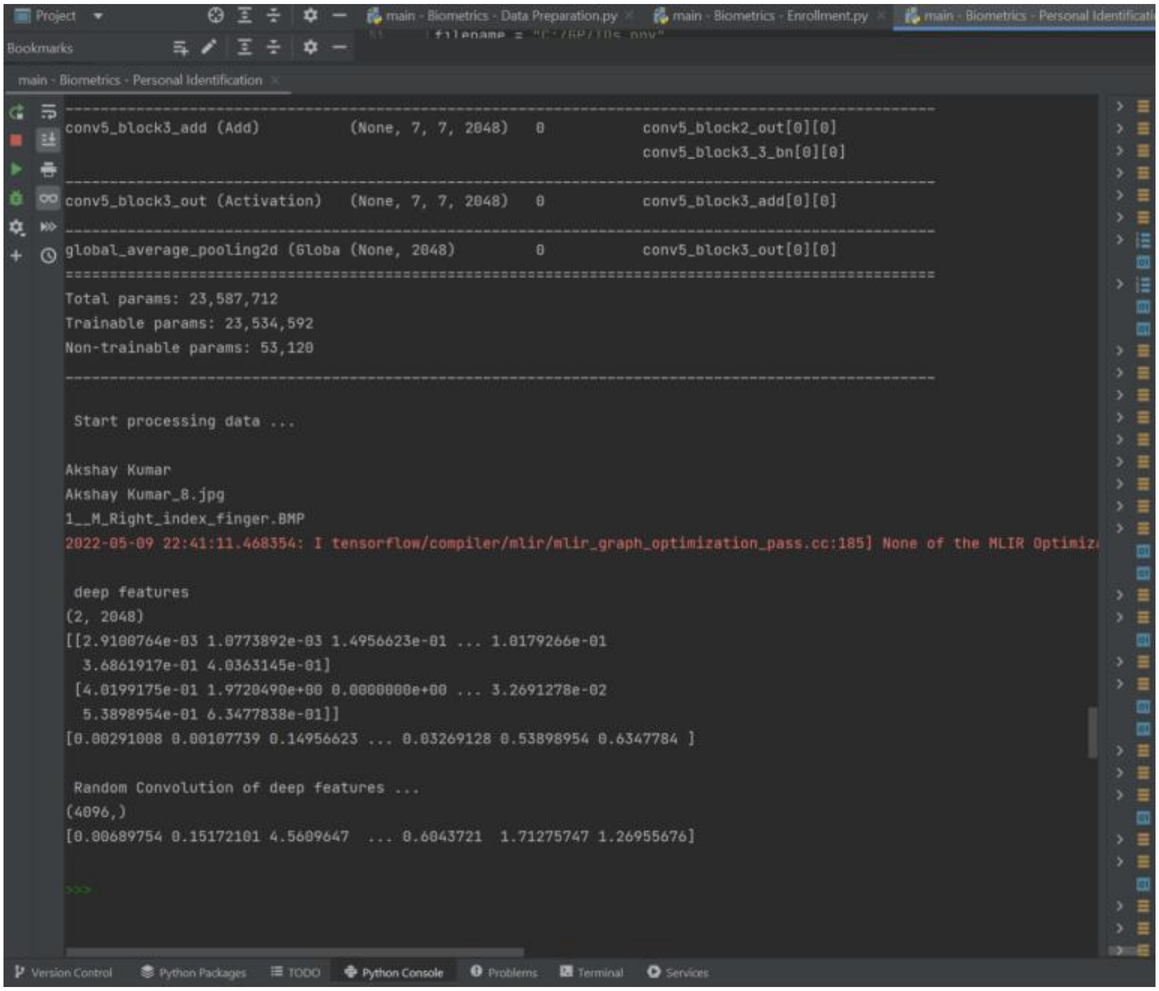
Figure 6-3: Accuracy Formula.

6.2 Test cases

Here we decided to test the system by making five test cases ,and these test cases are chosen randomly by the system that we have programmed. we mean by that we have run the code five times and every run we will have new result , because the system is choosing them randomly and see the resulting output.

by that move we making sure that the program is giving us the right and correct results.

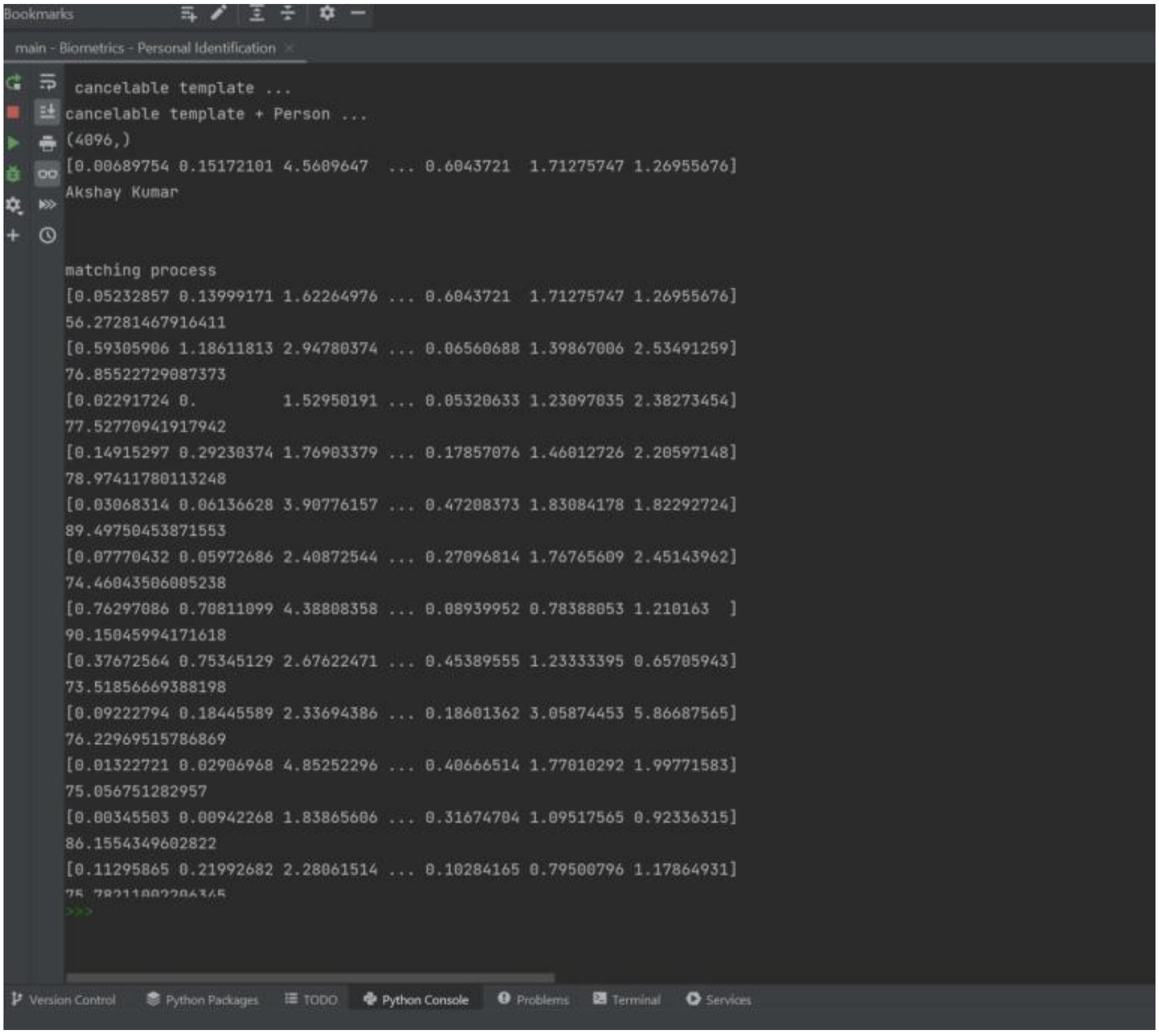
Case 1



Project
Bookmarks
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Personal Identificati
main - Biometrics - Personal Identification
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services

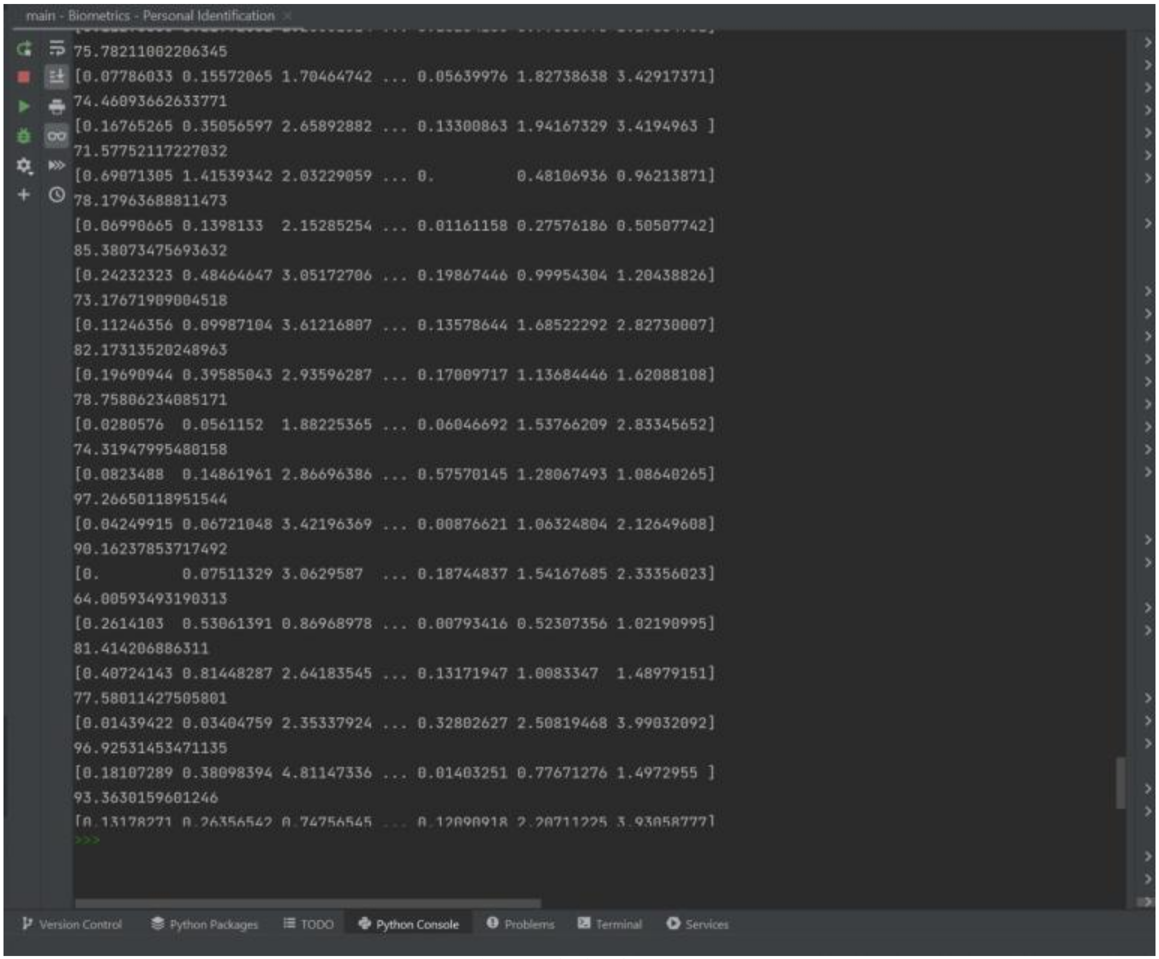
```
----------------------------------------------------------------------------------------------------
conv5_block3_add (Add)          (None, 7, 7, 2048)   0           conv5_block2_out[0][0]
                                                                 conv5_block3_3_bn[0][0]
----------------------------------------------------------------------------------------------------
conv5_block3_out (Activation)   (None, 7, 7, 2048)   0           conv5_block3_add[0][0]
----------------------------------------------------------------------------------------------------
global_average_pooling2d (Globa (None, 2048)         0           conv5_block3_out[0][0]
====================================================================================================
Total params: 23,587,712
Trainable params: 23,534,592
Non-trainable params: 53,120
----------------------------------------------------------------------------------------------------

 Start processing data ...

Akshay Kumar
Akshay Kumar_8.jpg
1__M_Right_index_finger.BMP
2022-05-09 22:41:11.468354: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR Optimiz

 deep features
(2, 2048)
[[2.9100764e-03 1.0773892e-03 1.4956623e-01 ... 1.0179266e-01
  3.6861917e-01 4.0363145e-01]
 [4.0199175e-01 1.9720490e+00 0.0000000e+00 ... 3.2691278e-02
  5.3898954e-01 6.3477838e-01]]
[0.00291008 0.00107739 0.14956623 ... 0.03269128 0.53898954 0.6347784 ]

 Random Convolution of deep features ...
(4096,)
[0.00689754 0.15172101 4.5609647  ... 0.6043721  1.71275747 1.26955676]


>>>
```

Figure 6-4: Case1.



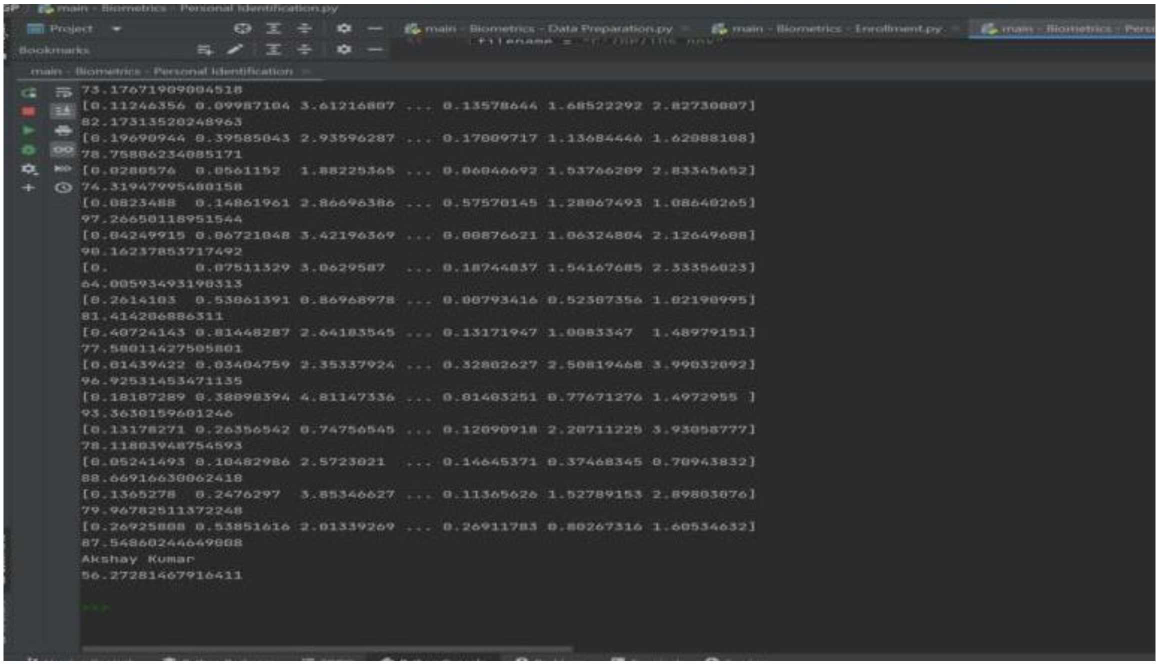
```
 cancelable template ...
cancelable template + Person ...
(4096,)
[0.00689754 0.15172101 4.5609647  ... 0.6043721  1.71275747 1.26955676]
Akshay Kumar


matching process
[0.05232857 0.13999171 1.62264976 ... 0.6043721  1.71275747 1.26955676]
56.27281467916411
[0.59305906 1.18611813 2.94780374 ... 0.06560688 1.39867006 2.53491259]
76.85522729087373
[0.02291724 0.         1.52950191 ... 0.05320633 1.23097035 2.38273454]
77.52770941917942
[0.14915297 0.29230374 1.76903379 ... 0.17857076 1.46012726 2.20597148]
78.97411780113248
[0.03068314 0.06136628 3.90776157 ... 0.47208373 1.83084178 1.82292724]
89.49750453871553
[0.07770432 0.05972686 2.40872544 ... 0.27096814 1.76765609 2.45143962]
74.46043506005238
[0.76297086 0.70811099 4.38808358 ... 0.08939952 0.78388053 1.210163  ]
90.15045994171618
[0.37672564 0.75345129 2.67622471 ... 0.45389555 1.23333395 0.65705943]
73.51856669388198
[0.09222794 0.18445589 2.33694386 ... 0.18601362 3.05874453 5.86687565]
76.22969515786869
[0.01322721 0.02906968 4.85252296 ... 0.40666514 1.77010292 1.99771583]
75.056751282957
[0.00345503 0.00942268 1.83865606 ... 0.31674704 1.09517565 0.92336315]
86.1554349602822
[0.11295865 0.21992682 2.28061514 ... 0.10284165 0.79500796 1.17864931]
75.78211002206345
>>>
```

Figure 6-5: Case1.



```
75.78211002206345
[0.07786033 0.15572065 1.70464742 ... 0.05639976 1.82738638 3.42917371]
74.46093662633771
[0.16765265 0.35056597 2.65892882 ... 0.13300863 1.94167329 3.4194963 ]
71.57752117227032
[0.69071305 1.41539342 2.03229059 ... 0.         0.48106936 0.96213871]
78.17963688811473
[0.06990665 0.1398133  2.15285254 ... 0.01161158 0.27576186 0.50507742]
85.38073475693632
[0.24232323 0.48464647 3.05172706 ... 0.19867446 0.99954304 1.20438826]
73.17671909004518
[0.11246356 0.09987104 3.61216807 ... 0.13578644 1.68522292 2.82730007]
82.17313520248963
[0.19690944 0.39585043 2.93596287 ... 0.17009717 1.13684446 1.62088108]
78.75806234085171
[0.0280576  0.0561152  1.88225365 ... 0.06046692 1.53766209 2.83345652]
74.31947995480158
[0.0823488  0.14861961 2.86696386 ... 0.57570145 1.28067493 1.08640265]
97.26650118951544
[0.04249915 0.06721048 3.42196369 ... 0.00876621 1.06324804 2.12649608]
90.16237853717492
[0.         0.07511329 3.0629587  ... 0.18744837 1.54167685 2.33356023]
64.00593493190313
[0.2614103  0.53061391 0.86968978 ... 0.00793416 0.52307356 1.02190995]
81.414206886311
[0.40724143 0.81448287 2.64183545 ... 0.13171947 1.0083347  1.48979151]
77.58011427505801
[0.01439422 0.03404759 2.35337924 ... 0.32802627 2.50819468 3.99032092]
96.92531453471135
[0.18107289 0.38098394 4.81147336 ... 0.01403251 0.77671276 1.4972955 ]
93.3630159601246
[0.13178271 0.26356542 0.74756545 ... 0.12090918 2.20711225 3.93058777]
>>>
```

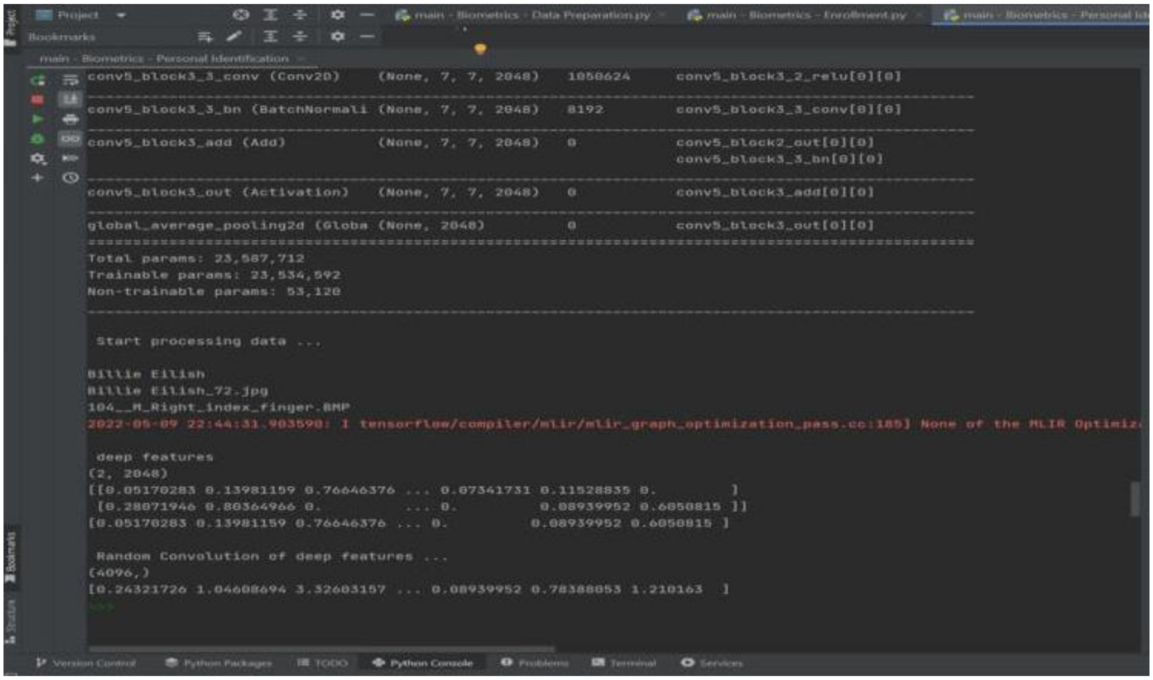
Figure 6-6: Case1.


main - Biometrics - Personal Identification.py
Project
Bookmarks
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Pers
main - Biometrics - Personal Identification

```
73.17671909004518
[0.11246356 0.09987104 3.61216807 ... 0.13578644 1.68522292 2.82730007]
82.17313520248963
[0.19690944 0.39585043 2.93596287 ... 0.17009717 1.13684446 1.62088108]
78.75806234085171
[0.0280576  0.0561152  1.88225365 ... 0.06046692 1.53766209 2.83345652]
74.31947995480158
[0.0823488  0.14861961 2.86696386 ... 0.57570145 1.28067493 1.08640265]
97.26650118951544
[0.04249915 0.06721048 3.42196369 ... 0.00876621 1.06324804 2.12649608]
90.16237853717492
[0.         0.07511329 3.0629587  ... 0.18744837 1.54167685 2.33356023]
64.00593493190313
[0.2614103  0.53061391 0.86968978 ... 0.00793416 0.52307356 1.02190995]
81.414206886311
[0.40724143 0.81448287 2.64183545 ... 0.13171947 1.0083347  1.48979151]
77.58011427505801
[0.01439422 0.03404759 2.35337924 ... 0.32802627 2.50819468 3.99032092]
96.92531453471135
[0.18107289 0.38098394 4.81147336 ... 0.01403251 0.77671276 1.4972955 ]
93.3630159601246
[0.13178271 0.26356542 0.74756545 ... 0.12090918 2.20711225 3.93058777]
78.11803948754593
[0.05241493 0.10482986 2.5723021  ... 0.14645371 0.37468345 0.70943832]
88.66916630062418
[0.1365278  0.2476297  3.85346627 ... 0.11365626 1.52789153 2.89803076]
79.96782511372248
[0.26925808 0.53851616 2.01339269 ... 0.26911783 0.80267316 1.60534632]
87.54860244649008
Akshay Kumar
56.27281467916411

>>>
```

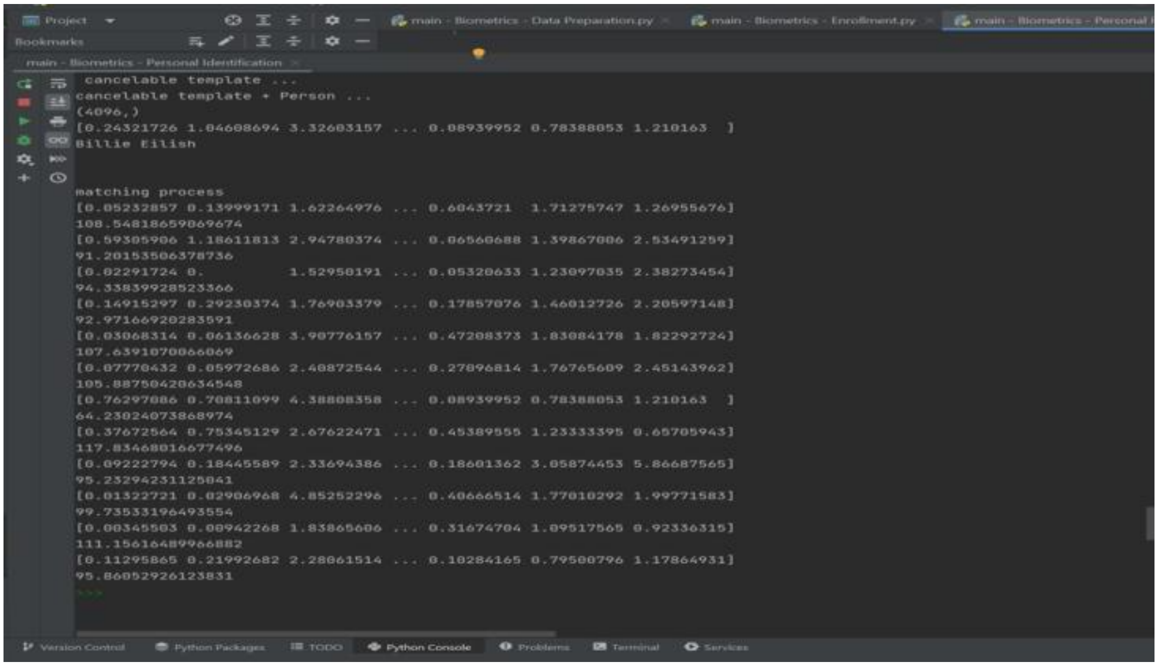
Figure 6-7: Case1.

Case 2


Project
Bookmarks
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Personal Id
main - Biometrics - Personal Identification
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services

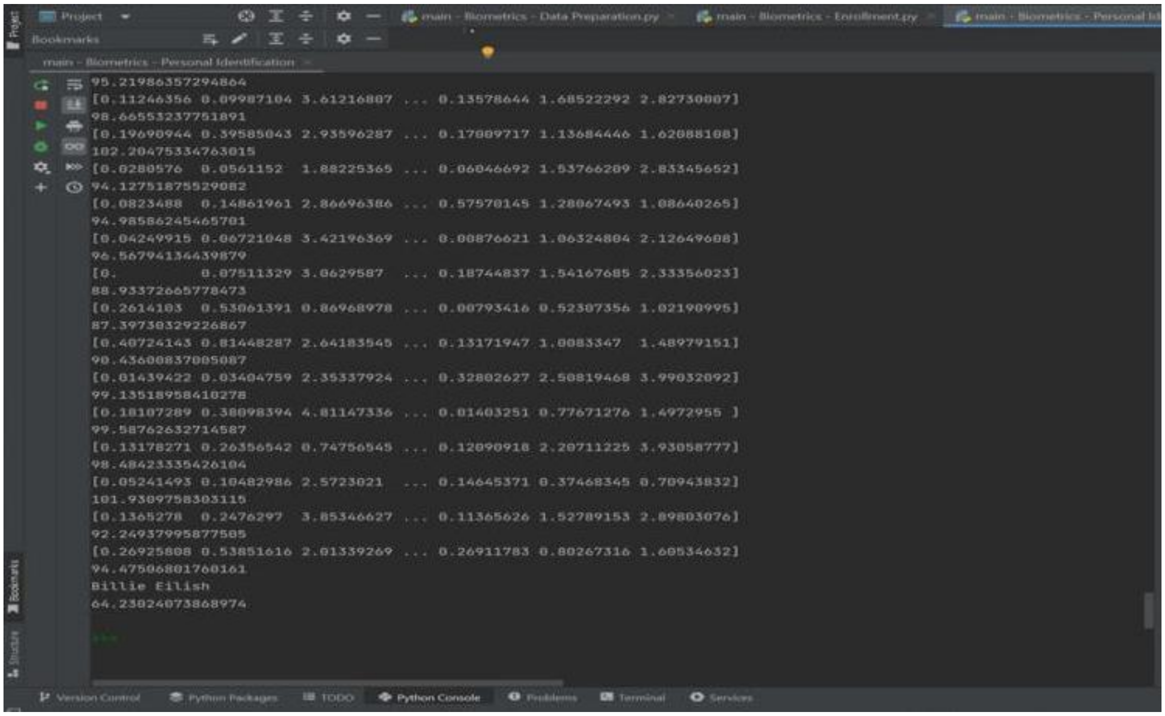
```
conv5_block3_3_conv (Conv2D)    (None, 7, 7, 2048)   1050624     conv5_block3_2_relu[0][0]
------------------------------------------------------------------------------------------------
conv5_block3_3_bn (BatchNormali (None, 7, 7, 2048)   8192        conv5_block3_3_conv[0][0]
------------------------------------------------------------------------------------------------
conv5_block3_add (Add)          (None, 7, 7, 2048)   0           conv5_block2_out[0][0]
                                                                 conv5_block3_3_bn[0][0]
------------------------------------------------------------------------------------------------
conv5_block3_out (Activation)   (None, 7, 7, 2048)   0           conv5_block3_add[0][0]
------------------------------------------------------------------------------------------------
global_average_pooling2d (Globa (None, 2048)         0           conv5_block3_out[0][0]
================================================================================================
Total params: 23,587,712
Trainable params: 23,534,592
Non-trainable params: 53,120
------------------------------------------------------------------------------------------------

 Start processing data ...

Billie Eilish
Billie Eilish_72.jpg
104__M_Right_index_finger.BMP
2022-05-09 22:44:31.903590: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR Optimiz

 deep features
(2, 2048)
[[0.05170283 0.13981159 0.76646376 ... 0.07341731 0.11528835 0.        ]
 [0.28071946 0.80364966 0.         ... 0.         0.08939952 0.6050815 ]]
[0.05170283 0.13981159 0.76646376 ... 0.         0.08939952 0.6050815 ]

 Random Convolution of deep features ...
(4096,)
[0.24321726 1.04608694 3.32603157 ... 0.08939952 0.78388053 1.210163  ]
>>>
```

Figure 6-8: Case2.


Project
Bookmarks
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Personal I
main - Biometrics - Personal Identification
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services

```
 cancelable template ...
cancelable template + Person ...
(4096,)
[0.24321726 1.04608694 3.32603157 ... 0.08939952 0.78388053 1.210163  ]
Billie Eilish


matching process
[0.05232857 0.13999171 1.62264976 ... 0.6043721  1.71275747 1.26955676]
108.54818659069674
[0.59305906 1.18611813 2.94780374 ... 0.06560688 1.39867006 2.53491259]
91.20153506378736
[0.02291724 0.         1.52950191 ... 0.05320633 1.23097035 2.38273454]
94.33839928523366
[0.14915297 0.29230374 1.76903379 ... 0.17857076 1.46012726 2.20597148]
92.97166920283591
[0.03068314 0.06136628 3.90776157 ... 0.47208373 1.83084178 1.82292724]
107.6391070066069
[0.07770432 0.05972686 2.40872544 ... 0.27896814 1.76765609 2.45143962]
105.88750420634548
[0.76297086 0.70811099 4.38808358 ... 0.08939952 0.78388053 1.210163  ]
64.23024073868974
[0.37672564 0.75345129 2.67622471 ... 0.45389555 1.23333395 0.65705943]
117.83468016677496
[0.09222794 0.18445589 2.33694386 ... 0.18601362 3.05874453 5.86687565]
95.23294231125041
[0.01322721 0.02906968 4.85252296 ... 0.40666514 1.77010292 1.99771583]
99.73533196493554
[0.00345503 0.00942268 1.83865606 ... 0.31674704 1.09517565 0.92336315]
111.15616489966882
[0.11295865 0.21992682 2.28061514 ... 0.10284165 0.79500796 1.17864931]
95.86052926123831
>>>
```

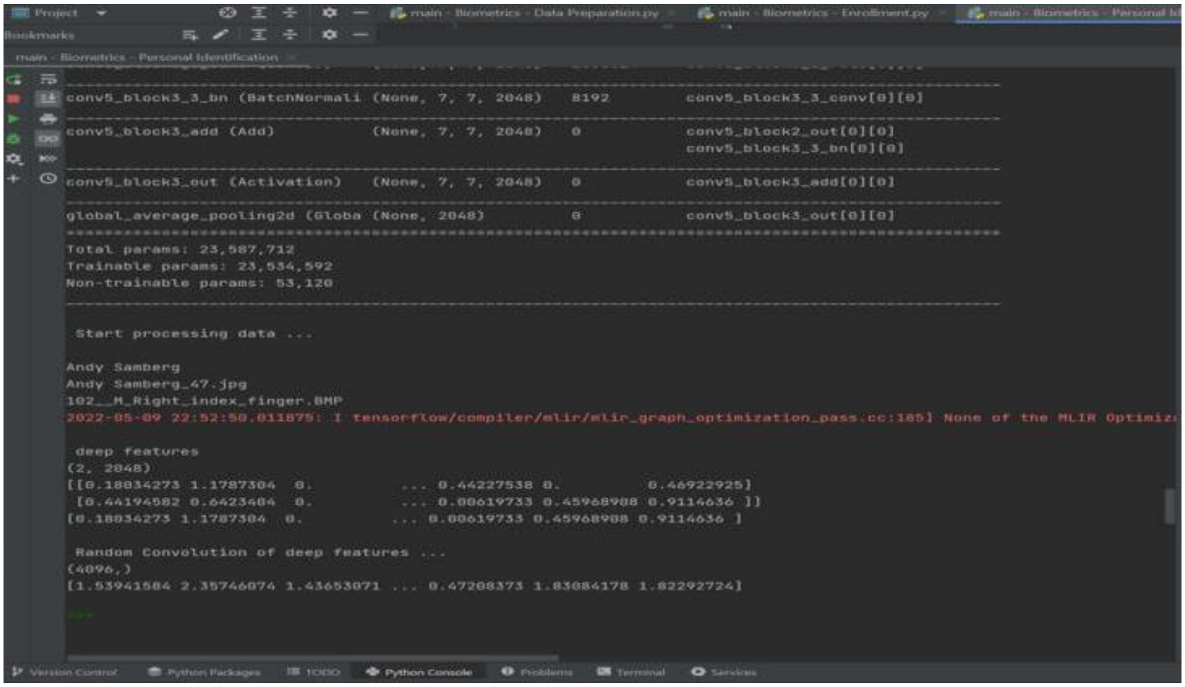
Figure 6-9: Case2.


Project
Bookmarks
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Personal Id
main - Biometrics - Personal Identification
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services
Project
Bookmarks
Structure

```
95.21986357294864
[0.11246356 0.09987104 3.61216807 ... 0.13578644 1.68522292 2.82730007]
98.66553237751891
[0.19690944 0.39585043 2.93596287 ... 0.17009717 1.13684446 1.62088108]
102.20475334763015
[0.0280576  0.0561152  1.88225365 ... 0.06046692 1.53766209 2.83345652]
94.12751875529082
[0.0823488  0.14861961 2.86696386 ... 0.57570145 1.28067493 1.08640265]
94.98586245465701
[0.04249915 0.06721048 3.42196369 ... 0.00876621 1.06324804 2.12649608]
96.56794134439879
[0.         0.07511329 3.0629587  ... 0.18744837 1.54167685 2.33356023]
88.93372665778473
[0.2614103  0.53061391 0.86968978 ... 0.00793416 0.52307356 1.02190995]
87.39730329226867
[0.40724143 0.81448287 2.64183545 ... 0.13171947 1.0083347  1.48979151]
90.43600837005087
[0.01439422 0.03404759 2.35337924 ... 0.32802627 2.50819468 3.99032092]
99.13518958410278
[0.18107289 0.38098394 4.81147336 ... 0.01403251 0.77671276 1.4972955 ]
99.58762632714587
[0.13178271 0.26356542 0.74756545 ... 0.12090918 2.20711225 3.93058777]
98.48423335426104
[0.05241493 0.10482986 2.5723021  ... 0.14645371 0.37468345 0.70943832]
101.9309758303115
[0.1365278  0.2476297  3.85346627 ... 0.11365626 1.52789153 2.89803076]
92.24937995877505
[0.26925808 0.53851616 2.01339269 ... 0.26911783 0.80267316 1.60534632]
94.47506801760161
Billie Eilish
64.23024073868974

>>>
```

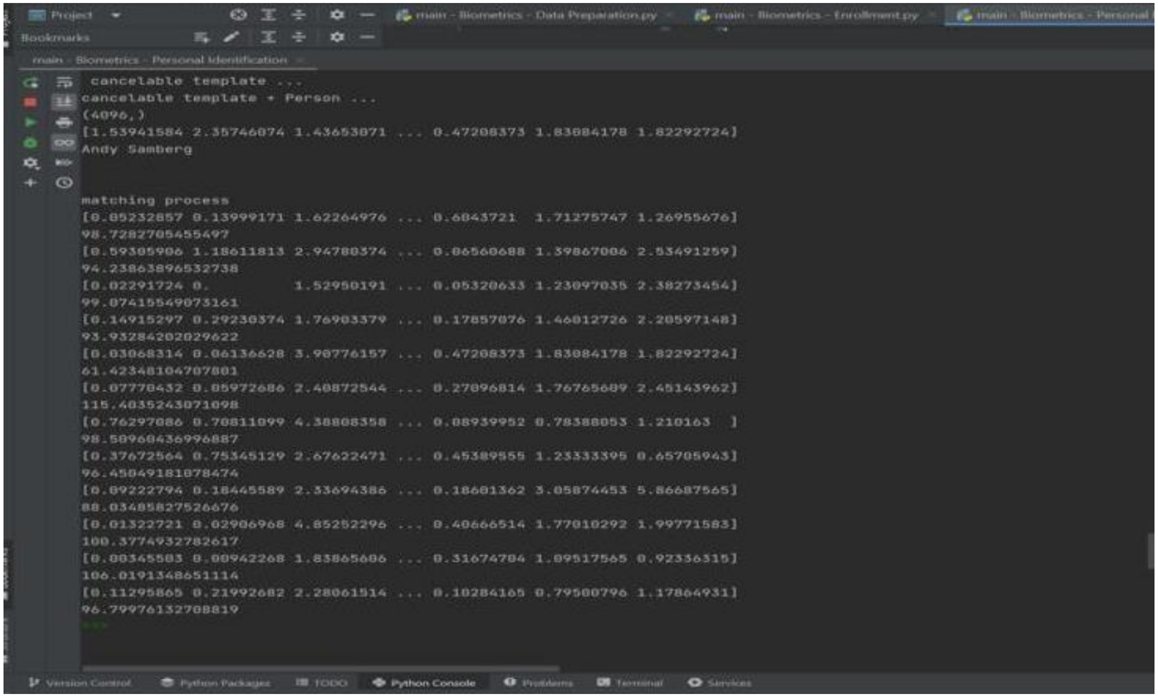
Figure 6-10: Case2.

Case 3


Project
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Personal Id
Bookmarks
main - Biometrics - Personal Identification
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services

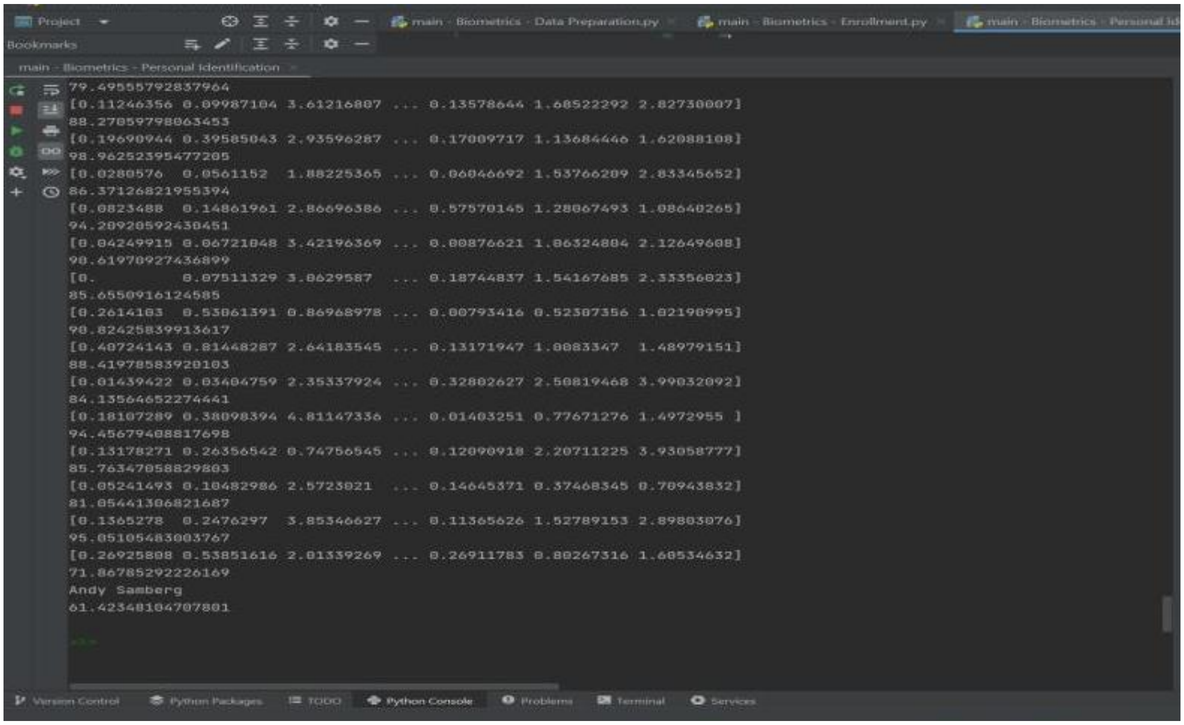
```
--------------------------------------------------------------------------------------------------
conv5_block3_3_bn (BatchNormali (None, 7, 7, 2048)   8192        conv5_block3_3_conv[0][0]
--------------------------------------------------------------------------------------------------
conv5_block3_add (Add)          (None, 7, 7, 2048)   0           conv5_block2_out[0][0]
                                                                 conv5_block3_3_bn[0][0]
--------------------------------------------------------------------------------------------------
conv5_block3_out (Activation)   (None, 7, 7, 2048)   0           conv5_block3_add[0][0]
--------------------------------------------------------------------------------------------------
global_average_pooling2d (Globa (None, 2048)         0           conv5_block3_out[0][0]
==================================================================================================
Total params: 23,587,712
Trainable params: 23,534,592
Non-trainable params: 53,120
--------------------------------------------------------------------------------------------------

 Start processing data ...

Andy Samberg
Andy Samberg_47.jpg
102__M_Right_index_finger.BMP
2022-05-09 22:52:50.011875: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR Optimiz

 deep features
(2, 2048)
[[0.18034273 1.1787304  0.         ... 0.44227538 0.         0.46922925]
 [0.44194582 0.6423484  0.         ... 0.00619733 0.45968908 0.9114636 ]]
[0.18034273 1.1787304  0.         ... 0.00619733 0.45968908 0.9114636 ]

 Random Convolution of deep features ...
(4096,)
[1.53941584 2.35746074 1.43653071 ... 0.47208373 1.83084178 1.82292724]
>>>
```

Figure 6-11: Case3.


Project
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Personal I
Bookmarks
main - Biometrics - Personal Identification
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services

```
 cancelable template ...
cancelable template + Person ...
(4096,)
[1.53941584 2.35746074 1.43653071 ... 0.47208373 1.83084178 1.82292724]
Andy Samberg


matching process
[0.05232857 0.13999171 1.62264976 ... 0.6043721  1.71275747 1.26955676]
98.7282705455497
[0.59305906 1.18611813 2.94780374 ... 0.06560688 1.39867006 2.53491259]
94.23863896532738
[0.02291724 0.         1.52950191 ... 0.05320633 1.23097035 2.38273454]
99.07415549073161
[0.14915297 0.29230374 1.76903379 ... 0.17857076 1.46012726 2.20597148]
93.93284202029622
[0.03068314 0.06136628 3.90776157 ... 0.47208373 1.83084178 1.82292724]
61.42348104707801
[0.07770432 0.05972686 2.40872544 ... 0.27096814 1.76765689 2.45143962]
115.4035243071098
[0.76297086 0.70811099 4.38808358 ... 0.08939952 0.78388053 1.210163  ]
98.58960436996887
[0.37672564 0.75345129 2.67622471 ... 0.45389555 1.23333395 0.65705943]
96.45049181078474
[0.09222794 0.18445589 2.33694386 ... 0.18601362 3.05874453 5.86687565]
88.03485827526676
[0.01322721 0.02906968 4.85252296 ... 0.40666514 1.77010292 1.99771583]
100.3774932782617
[0.00345503 0.00942268 1.83865606 ... 0.31674704 1.09517565 0.92336315]
106.0191348651114
[0.11295865 0.21992682 2.28061514 ... 0.10284165 0.79500796 1.17864931]
96.79976132708819
>>>
```

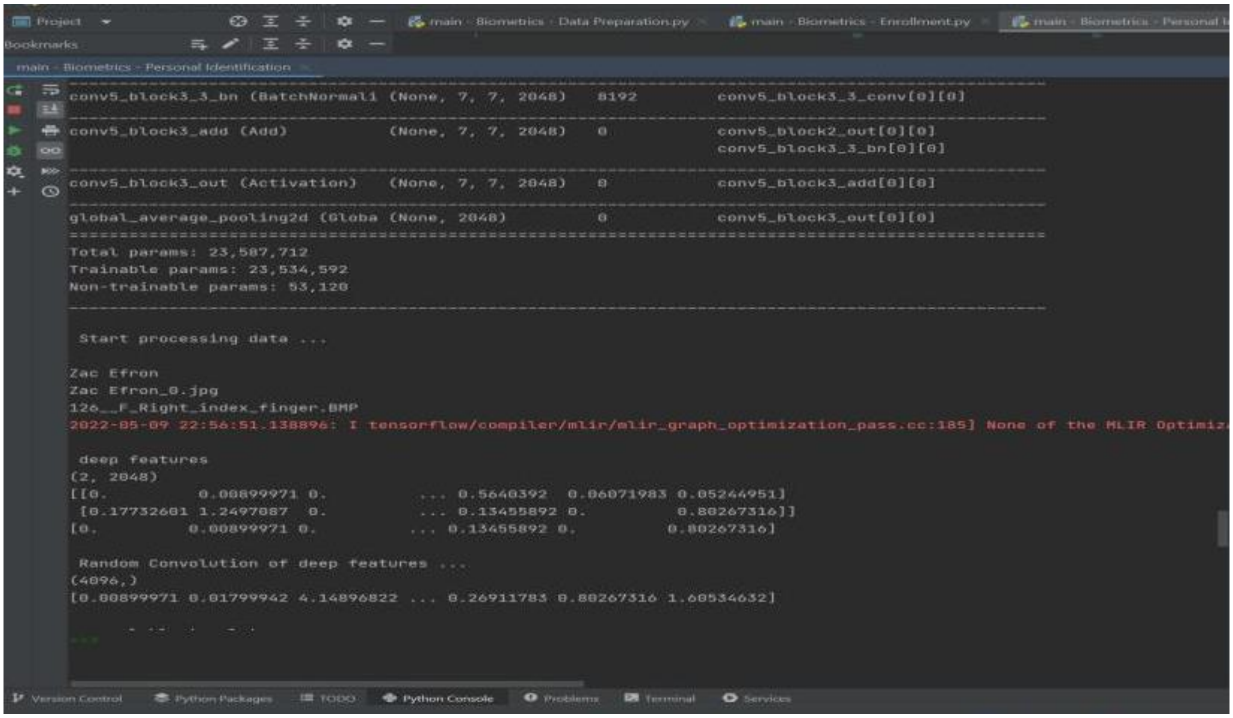
Figure 6-12: Case3.


Project
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Personal Id
Bookmarks
main - Biometrics - Personal Identification
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services

```
79.49555792837964
[0.11246356 0.09987104 3.61216807 ... 0.13578644 1.68522292 2.82730007]
88.27059798063453
[0.19690944 0.39585043 2.93596287 ... 0.17009717 1.13684446 1.62088108]
98.96252395477205
[0.0280576  0.0561152  1.88225365 ... 0.06046692 1.53766209 2.83345652]
86.37126821955394
[0.0823488  0.14861961 2.86696386 ... 0.57570145 1.28067493 1.08640265]
94.28920592430451
[0.04249915 0.06721048 3.42196369 ... 0.00876621 1.86324804 2.12649608]
90.61970927436899
[0.         0.07511329 3.0629587  ... 0.18744837 1.54167685 2.33356023]
85.6550916124585
[0.2614103  0.53061391 0.86968978 ... 0.00793416 0.52307356 1.02190995]
90.82425839913617
[0.40724143 0.81448287 2.64183545 ... 0.13171947 1.0083347  1.48979151]
88.41978583920103
[0.01439422 0.03404759 2.35337924 ... 0.32802627 2.50819468 3.99032092]
84.13564652274441
[0.18107289 0.38098394 4.81147336 ... 0.01403251 0.77671276 1.4972955 ]
94.45679408817698
[0.13178271 0.26356542 0.74756545 ... 0.12090918 2.20711225 3.93058777]
85.76347058829803
[0.05241493 0.10482986 2.5723021  ... 0.14645371 0.37468345 0.70943832]
81.05441306821687
[0.1365278  0.2476297  3.85346627 ... 0.11365626 1.52789153 2.89803076]
95.05105483003767
[0.26925808 0.53851616 2.01339269 ... 0.26911783 0.80267316 1.60534632]
71.86785292226169
Andy Samberg
61.42348104707801

>>>
```

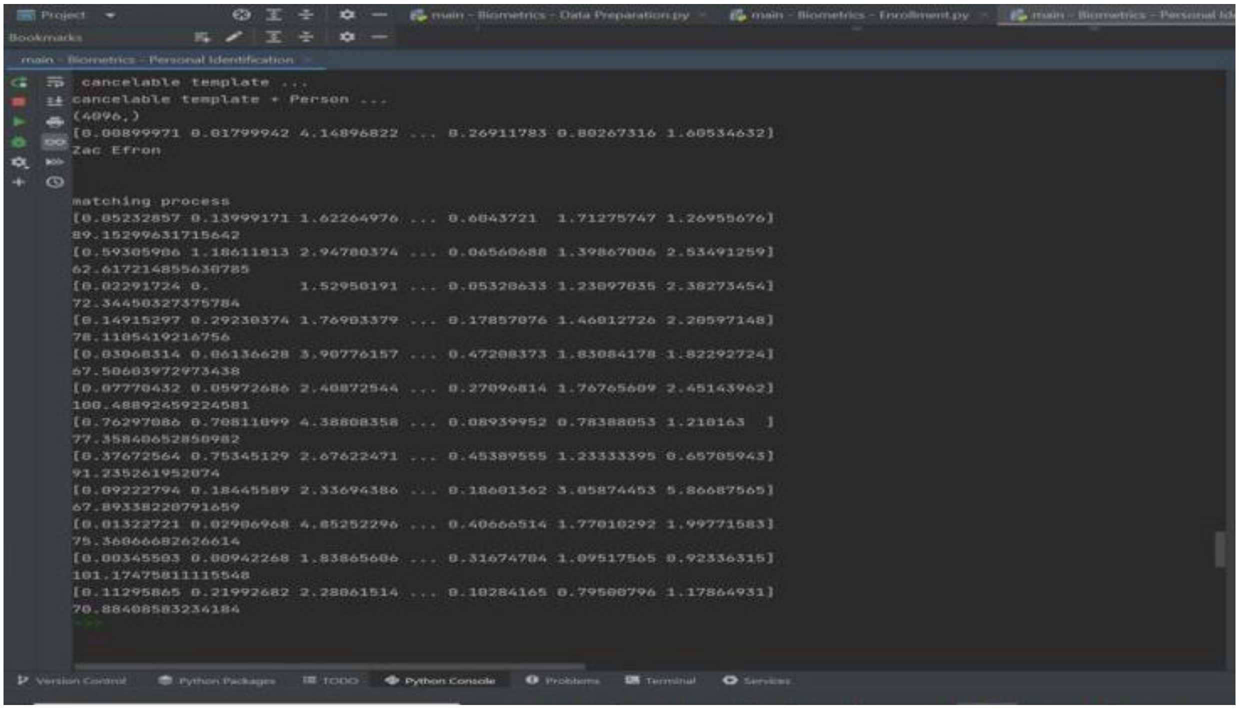
Figure 6-13: Case3.

Case 4


Project
Bookmarks
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Personal Id
main - Biometrics - Personal Identification
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services

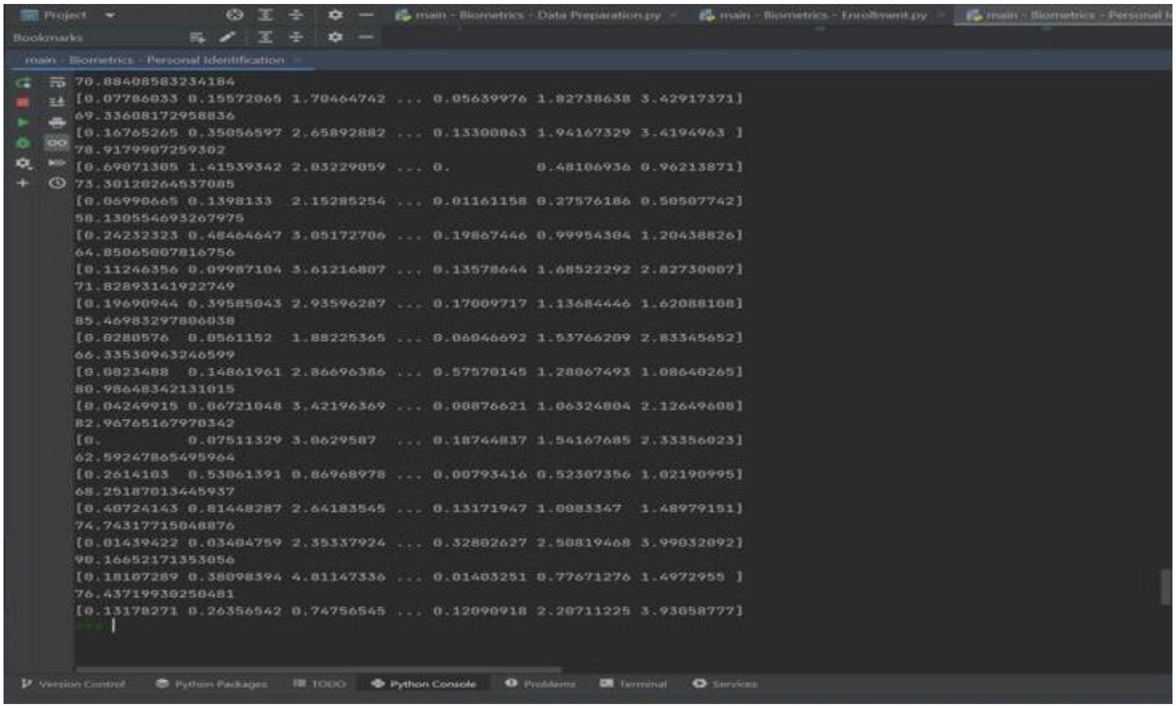
```
--------------------------------------------------------------------------------------------------
conv5_block3_3_bn (BatchNormali (None, 7, 7, 2048)   8192        conv5_block3_3_conv[0][0]
--------------------------------------------------------------------------------------------------
conv5_block3_add (Add)          (None, 7, 7, 2048)   0           conv5_block2_out[0][0]
                                                                 conv5_block3_3_bn[0][0]
--------------------------------------------------------------------------------------------------
conv5_block3_out (Activation)   (None, 7, 7, 2048)   0           conv5_block3_add[0][0]
--------------------------------------------------------------------------------------------------
global_average_pooling2d (Globa (None, 2048)         0           conv5_block3_out[0][0]
==================================================================================================
Total params: 23,587,712
Trainable params: 23,534,592
Non-trainable params: 53,120
--------------------------------------------------------------------------------------------------

 Start processing data ...

Zac Efron
Zac Efron_0.jpg
126__F_Right_index_finger.BMP
2022-05-09 22:56:51.138896: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR Optimiz

 deep features
(2, 2048)
[[0.         0.00899971 0.         ... 0.5640392  0.06071983 0.05244951]
 [0.17732601 1.2497087  0.         ... 0.13455892 0.         0.80267316]]
[0.         0.00899971 0.         ... 0.13455892 0.         0.80267316]

 Random Convolution of deep features ...
(4096,)
[0.00899971 0.01799942 4.14896822 ... 0.26911783 0.80267316 1.60534632]

>>>
```

Figure 6-14: Case4


Project
Bookmarks
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Personal Id
main - Biometrics - Personal Identification
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services

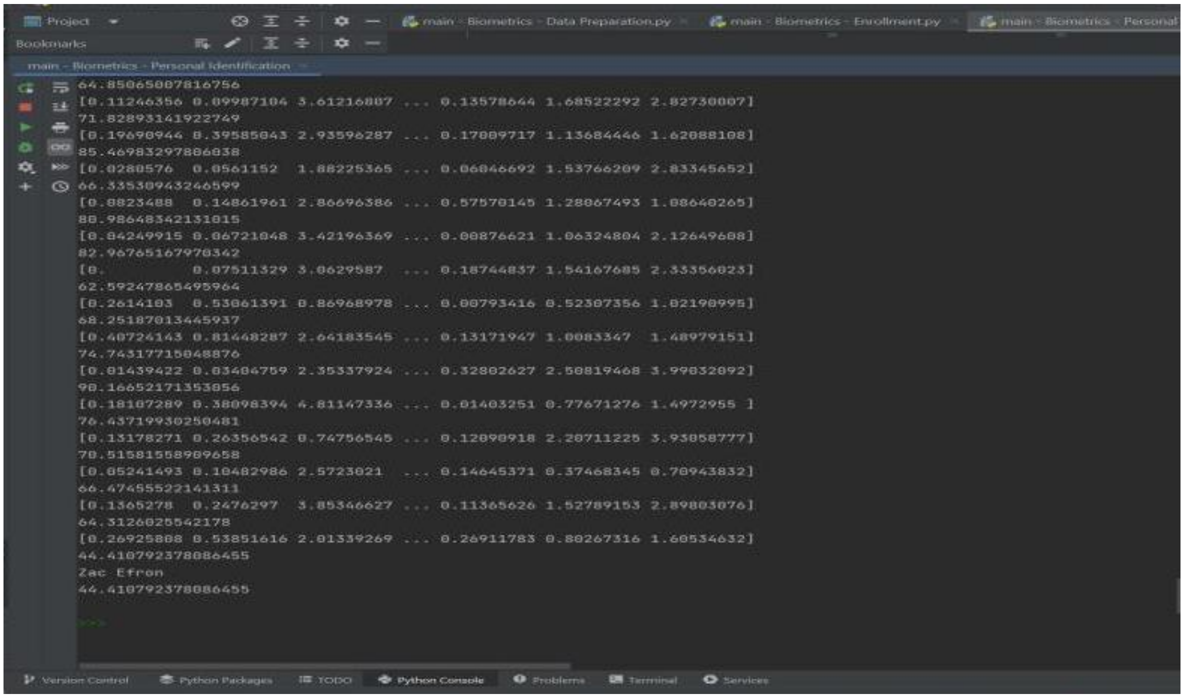
```
 cancelable template ...
cancelable template + Person ...
(4096,)
[0.00899971 0.01799942 4.14896822 ... 0.26911783 0.80267316 1.60534632]
Zac Efron


matching process
[0.05232857 0.13999171 1.62264976 ... 0.6043721  1.71275747 1.26955676]
89.15299631715642
[0.59305986 1.18611813 2.94780374 ... 0.06560688 1.39867006 2.53491259]
62.617214855630785
[0.02291724 0.         1.52950191 ... 0.05320633 1.23097035 2.38273454]
72.34450327375784
[0.14915297 0.29230374 1.76903379 ... 0.17857076 1.46012726 2.20597148]
78.1105419216756
[0.03068314 0.06136628 3.90776157 ... 0.47208373 1.83084178 1.82292724]
67.50603972973438
[0.07770432 0.05972686 2.40872544 ... 0.27096814 1.76765609 2.45143962]
100.48892459224581
[0.76297086 0.70811099 4.38808358 ... 0.08939952 0.78388053 1.210163  ]
77.35840652850982
[0.37672564 0.75345129 2.67622471 ... 0.45389555 1.23333395 0.65705943]
91.235261952074
[0.09222794 0.18445589 2.33694386 ... 0.18601362 3.05874453 5.86687565]
67.89338220791659
[0.01322721 0.02906968 4.05252296 ... 0.40666514 1.77010292 1.99771583]
75.36066682626614
[0.00345503 0.00942268 1.83865606 ... 0.31674704 1.09517565 0.92336315]
101.17475811115548
[0.11295865 0.21992682 2.28061514 ... 0.10284165 0.79500796 1.17864931]
70.88408583234184
>>>
```

Figure 6-15: Case4


Project
Bookmarks
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Personal I
main - Biometrics - Personal Identification
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services

```
70.88408583234184
[0.07786033 0.15572065 1.70464742 ... 0.05639976 1.82738638 3.42917371]
69.33608172958836
[0.16765265 0.35056597 2.65892882 ... 0.13300863 1.94167329 3.4194963 ]
78.9179907259302
[0.69071305 1.41539342 2.03229059 ... 0.         0.48106936 0.96213871]
73.30120264537085
[0.06990665 0.1398133  2.15285254 ... 0.01161158 0.27576186 0.50507742]
58.130554693267975
[0.24232323 0.48464647 3.05172706 ... 0.19867446 0.99954304 1.20438826]
64.85065007816756
[0.11246356 0.09987104 3.61216807 ... 0.13578644 1.68522292 2.82730007]
71.82893141922749
[0.19690944 0.39585043 2.93596287 ... 0.17009717 1.13684446 1.62088108]
85.46983297806038
[0.0280576  0.0561152  1.88225365 ... 0.06046692 1.53766209 2.83345652]
66.33530943246599
[0.0823488  0.14861961 2.86696386 ... 0.57570145 1.28067493 1.08640265]
80.98648342131015
[0.04249915 0.06721048 3.42196369 ... 0.00876621 1.06324804 2.12649608]
82.96765167970342
[0.         0.07511329 3.0629587  ... 0.18744837 1.54167685 2.33356023]
62.59247865495964
[0.2614103  0.53061391 0.86968978 ... 0.00793416 0.52307356 1.02190995]
68.25187013445937
[0.40724143 0.81448287 2.64183545 ... 0.13171947 1.0083347  1.48979151]
74.74317715048876
[0.01439422 0.03404759 2.35337924 ... 0.32802627 2.50819468 3.99032092]
90.16652171353056
[0.18107289 0.38098394 4.81147336 ... 0.01403251 0.77671276 1.4972955 ]
76.43719930250481
[0.13178271 0.26356542 0.74756545 ... 0.12090918 2.20711225 3.93058777]
>>>
```

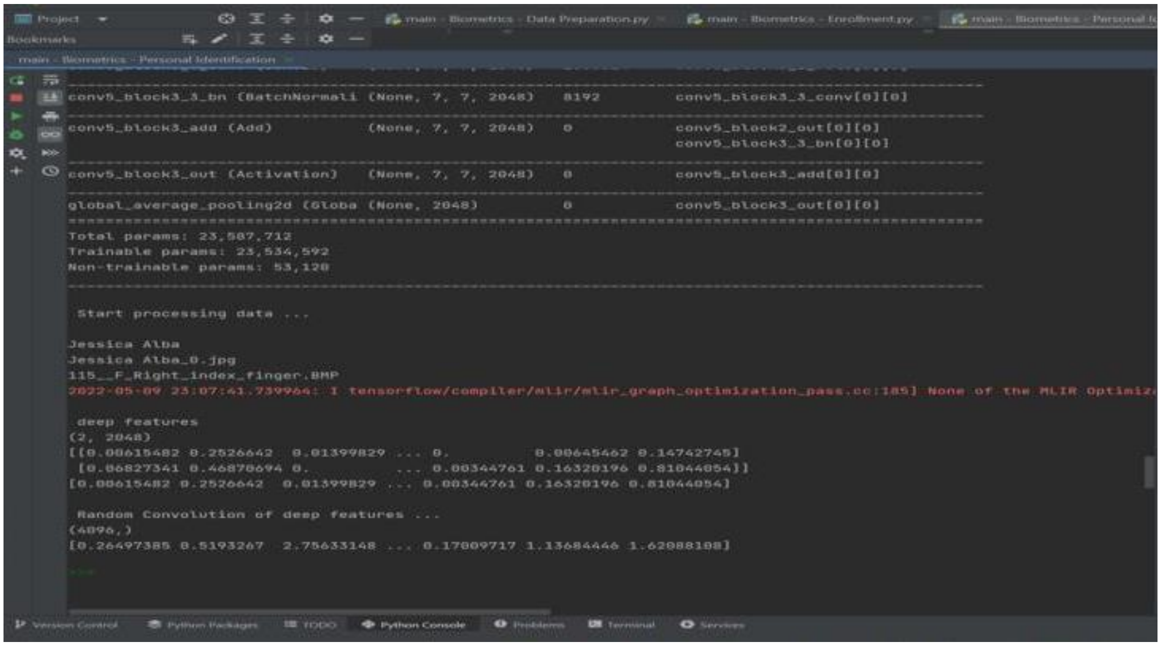
Figure 6-16: Case4


Project
Bookmarks
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Personal
main - Biometrics - Personal Identification
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services

```
64.85065007816756
[0.11246356 0.09987104 3.61216807 ... 0.13578644 1.68522292 2.82730007]
71.82893141922749
[0.19690944 0.39585043 2.93596287 ... 0.17009717 1.13684446 1.62088108]
85.46983297806038
[0.0280576  0.0561152  1.88225365 ... 0.06046692 1.53766209 2.83345652]
66.33530943246599
[0.0823488  0.14861961 2.86696386 ... 0.57570145 1.28067493 1.08640265]
80.98648342131015
[0.04249915 0.06721048 3.42196369 ... 0.00876621 1.06324804 2.12649608]
82.96765167970342
[0.         0.07511329 3.0629587  ... 0.18744837 1.54167685 2.33356023]
62.59247865495964
[0.2614103  0.53061391 0.86968978 ... 0.00793416 0.52307356 1.02190995]
68.25187013445937
[0.40724143 0.81448287 2.64183545 ... 0.13171947 1.0083347  1.48979151]
74.74317715048876
[0.01439422 0.03404759 2.35337924 ... 0.32802627 2.50819468 3.99032092]
90.16652171353056
[0.18107289 0.38098394 4.81147336 ... 0.01403251 0.77671276 1.4972955 ]
76.43719930250481
[0.13178271 0.26356542 0.74756545 ... 0.12090918 2.20711225 3.93058777]
70.51581558909658
[0.05241493 0.10482986 2.5723021  ... 0.14645371 0.37468345 0.70943832]
66.47455522141311
[0.1365278  0.2476297  3.85346627 ... 0.11365626 1.52789153 2.89803076]
64.3126025542178
[0.26925808 0.53851616 2.01339269 ... 0.26911783 0.80267316 1.60534632]
44.410792378086455
Zac Efron
44.410792378086455

>>>
```

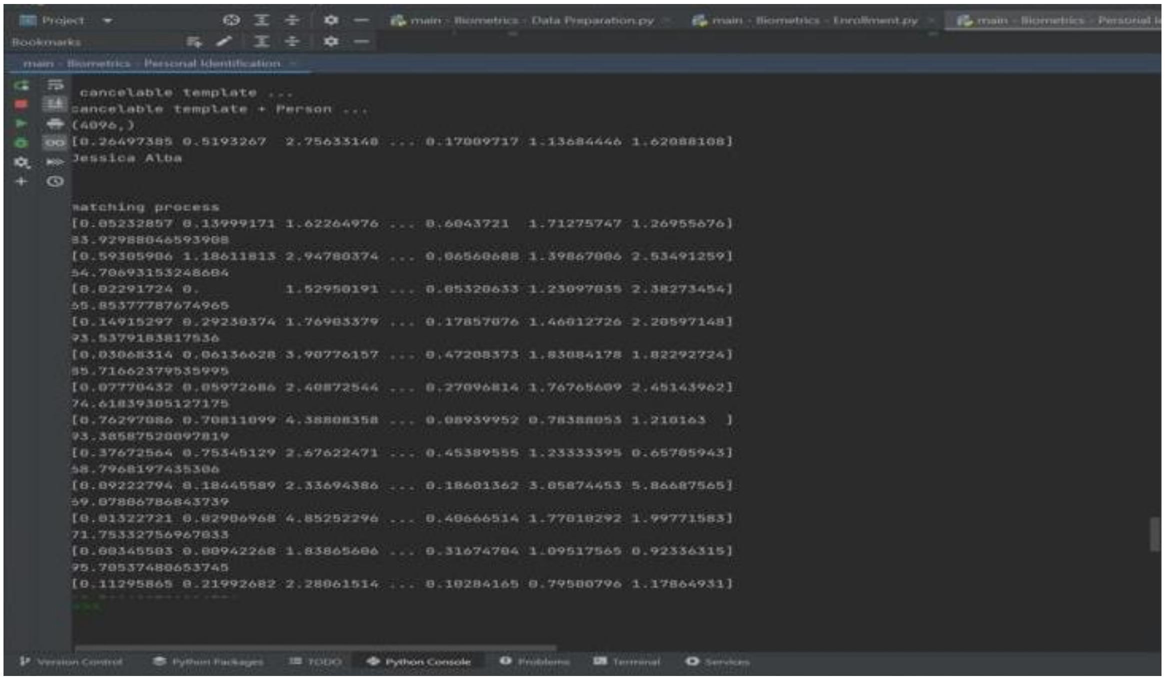
Figure 6-17: Case4

Case 5


Project
Bookmarks
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Personal Id
main - Biometrics - Personal Identification
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services

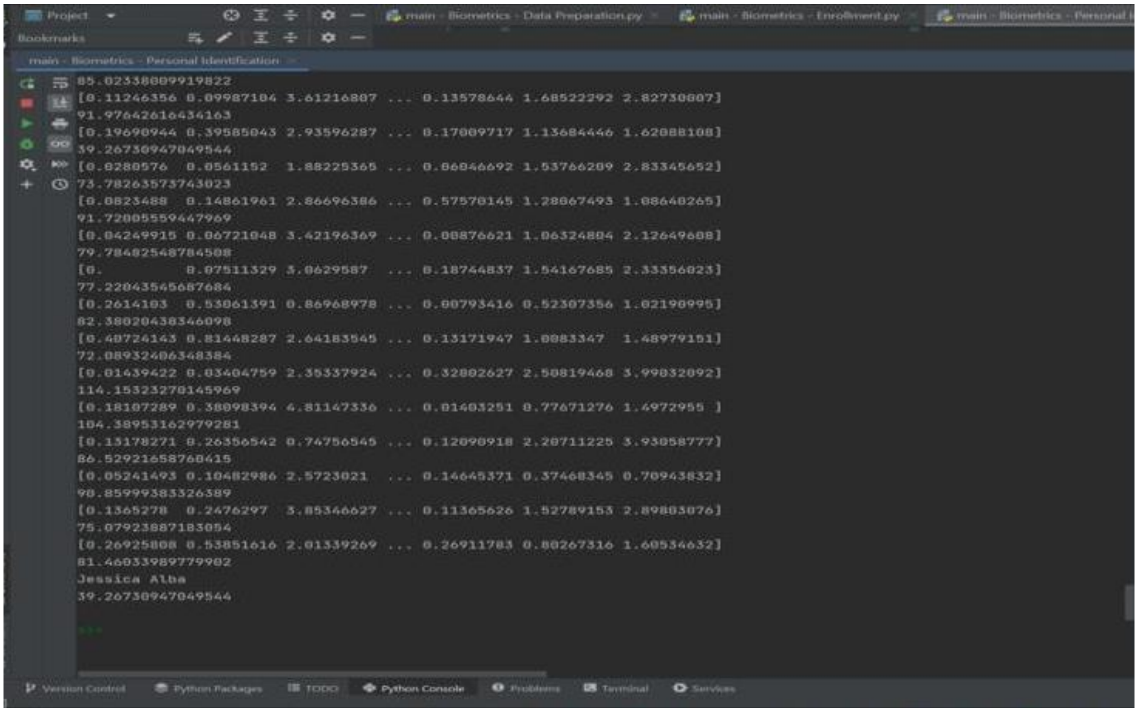
```
conv5_block3_3_bn (BatchNormali (None, 7, 7, 2048)   8192        conv5_block3_3_conv[0][0]
__________________________________________________________________________________________________
conv5_block3_add (Add)          (None, 7, 7, 2048)   0           conv5_block2_out[0][0]
                                                                 conv5_block3_3_bn[0][0]
__________________________________________________________________________________________________
conv5_block3_out (Activation)   (None, 7, 7, 2048)   0           conv5_block3_add[0][0]
__________________________________________________________________________________________________
global_average_pooling2d (Globa (None, 2048)         0           conv5_block3_out[0][0]
==================================================================================================
Total params: 23,587,712
Trainable params: 23,534,592
Non-trainable params: 53,120
__________________________________________________________________________________________________

 Start processing data ...

Jessica Alba
Jessica Alba_0.jpg
115__F_Right_index_finger.BMP
2022-05-09 23:07:41.739964: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR Optimiz

 deep features
(2, 2048)
[[0.00615482 0.2526642  0.01399829 ... 0.         0.00645462 0.14742745]
 [0.06827341 0.46870694 0.         ... 0.00344761 0.16320196 0.81044054]]
[0.00615482 0.2526642  0.01399829 ... 0.00344761 0.16320196 0.81044054]

 Random Convolution of deep features ...
(4096,)
[0.26497385 0.5193267  2.75633148 ... 0.17009717 1.13684446 1.62088108]
```

Figure 6-18: Case5


Project
Bookmarks
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Personal I
main - Biometrics - Personal Identification
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services

```
 cancelable template ...
cancelable template + Person ...
(4096,)
[0.26497385 0.5193267  2.75633148 ... 0.17009717 1.13684446 1.62088108]
Jessica Alba


matching process
[0.05232857 0.13999171 1.62264976 ... 0.6043721  1.71275747 1.26955676]
83.92988046593908
[0.59305906 1.18611813 2.94780374 ... 0.06560688 1.39867006 2.53491259]
54.70693153248684
[0.02291724 0.         1.52950191 ... 0.05320633 1.23097035 2.38273454]
55.85377787674965
[0.14915297 0.29230374 1.76983379 ... 0.17857076 1.46012726 2.20597148]
93.5379183817536
[0.03068314 0.06136628 3.90776157 ... 0.47208373 1.83084178 1.82292724]
85.71662379535995
[0.07770432 0.05972686 2.40872544 ... 0.27096814 1.76765609 2.45143962]
74.61839305127175
[0.76297086 0.70811099 4.38808358 ... 0.08939952 0.78388053 1.210163  ]
93.38587520097819
[0.37672564 0.75345129 2.67622471 ... 0.45389555 1.23333395 0.65705943]
58.7968197435306
[0.09222794 0.18445589 2.33694386 ... 0.18601362 3.05874453 5.86687565]
59.07886786843739
[0.01322721 0.02906968 4.85252296 ... 0.40666514 1.77010292 1.99771583]
71.75332756967833
[0.00345583 0.00942268 1.83865606 ... 0.31674704 1.09517565 0.92336315]
95.70537480653745
[0.11295865 0.21992682 2.28061514 ... 0.10284165 0.79500796 1.17864931]
```

Figure 6-19: Case5



Project
Bookmarks
main - Biometrics - Data Preparation.py
main - Biometrics - Enrollment.py
main - Biometrics - Personal I
main - Biometrics - Personal Identification
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services

```
85.02338009919822
[0.11246356 0.09987104 3.61216807 ... 0.13578644 1.68522292 2.82730007]
91.97642616434163
[0.19690944 0.39585043 2.93596287 ... 0.17009717 1.13684446 1.62088108]
39.26730947049544
[0.0280576  0.0561152  1.88225365 ... 0.06046692 1.53766209 2.83345652]
73.78263573743023
[0.0823488  0.14861961 2.86696386 ... 0.57570145 1.28067493 1.08640265]
91.72005559447969
[0.04249915 0.06721048 3.42196369 ... 0.00876621 1.06324804 2.12649608]
79.78482548784508
[0.         0.07511329 3.0629587  ... 0.18744837 1.54167685 2.33356023]
77.22043545687684
[0.2614103  0.53061391 0.86968978 ... 0.00793416 0.52307356 1.02190995]
82.38020438346098
[0.40724143 0.81448287 2.64183545 ... 0.13171947 1.0083347  1.48979151]
72.08932406348384
[0.01439422 0.03404759 2.35337924 ... 0.32802627 2.50819468 3.99032092]
114.15323270145969
[0.18107289 0.38098394 4.81147336 ... 0.01403251 0.77671276 1.4972955 ]
104.38953162979281
[0.13178271 0.26356542 0.74756545 ... 0.12090918 2.20711225 3.93058777]
86.52921658760415
[0.05241493 0.10482986 2.5723021  ... 0.14645371 0.37468345 0.70943832]
90.85999383326389
[0.1365278  0.2476297  3.85346627 ... 0.11365626 1.52789153 2.89803076]
75.07923887183054
[0.26925808 0.53851616 2.01339269 ... 0.26911783 0.80267316 1.60534632]
81.46033989779902
Jessica Alba
39.26730947049544

>>>
```

Figure 6-20: Case5

6.3 Test results

Nrepeats	accuracy
5	90.9677%
10	91.3548%
15	91.9929%
20	92.3271%
25	94.2580%
30	95.1612%

Table :Tests Results.

Chapter 7:Conclusion

7.1 Evaluation

In this project, we have reviewed some of the latest research, which related to cancelable biometrics using deep learning .

Our system is composite of two biometrics face, fingerprint. These biometrics, in our system will be used cancelable templets, and it will help us to overcome the standard security and privacy problems of traditional biological systems and increase the security levels.

This project presented a cancelable-biometrics security system that depends on merging two biometrics for the same person based on the random convolution, our system will merge two encrypted and cancelabled biomatrices which are the face and fingerprint per person. The encrypted biometrics will be saved in cancelable template. Therefore an original template is distorted and encrypted to be stored in the database.

It can be seen that the project has achieved the main aims which increase the security level of authentication, and protection technology to ensure reliable identification of people.

7.2 Future work

In future work, we plan to explore the potential of deep learning to extract high-level representations from data. The system should be tested on a larger database with noisy samples to validate the robustness of the model.

References

References

[1] RecFaces. 2020. *Two Main Types of Biometrics: Physical vs. Behavioral Biometrics | RecFaces*. [online] Available at: <https://recfaces.com/articles/types-of-biometrics> [Accessed 1 December 2021].

[2] Verizon. 2019 Data Breach Investigations Report. Accessed: Dec. 2019. [Online]. Available: https://enterprise.verizon.com/resources/reports/dbir/

[3] S. Barra, K.-K. R. Choo, M. Nappi, A. Castiglione, F. Narducci, and R. Ranjan, ‘‘Biometrics-as-a-service: Cloud-based technology, systems, and applications,’’ IEEE Cloud Comput., vol. 5, no. 4, pp. 33–37, Jul./Aug. 2018.

[4] K. Bailey and K. Curran, ‘‘An evaluation of image based steganography methods,’’ Multimedia Tools Appl., vol. 30, no. 1, pp. 55–88, Jul. 2006.

[5] A. Kumar and K. Pooja, ‘‘Steganography—A data hiding technique,’’ Int. J. Comput. Appl., vol. 9, no. 7, pp. 19–23, Nov. 2010.

[6] E. Abdellatef, N. Ismail, S. Abd Elrahman, K. Ismail, M. Rihan and F. Abd El-Samie, "Cancelable multi-biometric recognition system based on deep learning", 2021. .

[7] S. K. Singh, P. Roy, B. Raman, and P. Nagabhushan, Copmuter Vision and image processing 5th International Conference CVIP 2020. Prayagrai, India, December 4-6, 2020, revised selected papers, part I, Singapore: Spriager Singapore. 2021.

[8] V. Patel, N. Ratha and R. Chellappa, "Cancelable Biometrics: A review", *IEEE Signal Processing Magazine*, vol. 32, no. 5, pp. 54-65, 2015. Available: 10.1109/msp.2015.2434151.

[9] M. Ekman, *Learning Deep Learning : Theory and Practice of Neural Networks, Computer Vision, Natural Language Processing, and Transformers Using TensorFlow*. Addison Wesley Professional, 2021

[10] E. Abdellatef, N. Ismail, S. Abd Elrahman, K. Ismail, M. Rihan and F. Abd El-Samie, "Cancelable fusion-based face recognition", *Multimedia Tools and Applications*, vol. 78, no. 22, pp. 31557-31580, 2019. Available: 10.1007/s11042-019-07848-y.

[11] S. Raschka and V. Mirjalili, *Python machine learning*. .

[12] Arxiv.org, 2021. [Online]. Available: https://arxiv.org/pdf/1910.01389.pdf. [Accessed: 02- Dec- 2021].

[13] Engineering.jhu.edu, 2021. [Online]. Available: https://engineering.jhu.edu/vpatel36/wp-content/uploads/2018/08/SPM_CB_v6.pdf. [Accessed: 02- Dec- 2021].

[14] A. Jin and L. Hui, "Cancelable biometrics", 2021. .

[15] 2021. [Online]. Available: https://www.hilarispublisher.com/open-access/a-review-of-cancelable-biometric-authentication-methods-2155-6180-1000398.pdf. [Accessed: 02-Dec- 2021].

[16] Iaeng.org, 2021. [Online]. Available: http://www.iaeng.org/publication/WCECS2014/WCECS2014_pp199-204.pdf. [Accessed: 02- Dec- 2021].

[17] "Cryptography, Fields of study, Abstract, Principal terms", Science.jrank.org, 2021. [Online]. Available: https://science.jrank.org/programming/Cryptography.html. [Accessed: 02- Dec- 2021].

[18] F. Zheng, J. Shi, Y. Yang, W. Zheng and L. Cui, "A transformation-based method for auditing the IS-A hierarchy of biomedical terminologies in the Unified Medical Language System", 2021. .

[19] S. Ghouzali, O. Nafea, A. Wadood and M. Hussain, "Cancelable Multimodal Biometrics Based on Chaotic Maps", 2021. .

[20] "How Convolution Neural Networks interpret images", Medium, 2021. [Online]. Available: https://towardsdatascience.com/how-convolution-neural-networks-interpret-images-1f99913070b2. [Accessed: 02- Dec- 2021].

[21] Iopscience.iop.org, 2021. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1087/6/062032/pdf. [Accessed: 02- Dec- 2021].

[22] "Convolutional Neural Network (CNN) and its Application- All u need to know", Medium, 2021. [Online]. Available: https://medium.com/analytics-vidhya/convolutional-neural-network-cnn-and-its-application-all-u-need-to-know-f29c1d51b3e5. [Accessed: 02- Dec- 2021].

[23]"Face Recognition Dataset", *Kaggle.com*, 2022. [Online]. Available: https://www.kaggle.com/vasukipatel/face-recognition-dataset?select=Dataset.csv. [Accessed: 07- May- 2022].