Personal Identification using Cancelable Biometrics and Deep Learning

Presented By

Shahad Alghamdi
Sarah Aldawsari
Deema Almalki
Hatoon Abdullatif
Wed Alotaibi

Supervised By

Prof. Mohamed Batouche

TABLE OF CONTENTS

01

Overview

02

Implementation

03

Testing

04

Conclusion

01

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.

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 all the time even their personal identifications.

If they lost it and it takes space and sometimes you will have to pay for you belongs to get it back or to get new ones.

To Secure personal identification and biomatrices: fingerprint and face ID are safer than password.

It’s not easy to hack and if it is hacked, we can cancel Biometrics and change the encryption of the biometric.

Problem Statement & Significance

We will use Personal identification using cancelable biometrics and deep learning to solve this problem by using the biometrics identification (Multi-biometrics) to make people lives easier so they don’t have to remember every password.

Their personal identification will be more secure and safe because of cancelable biometrics. The input biometrics are transformed into cancelable templates which are stored in database.

During authentication or identification, the transformed input biometrics is compared to the templates in the database

Proposed Solution & Objective

A program that make use of multi-biometrics for personal identification rather than using the personal information such as (id, license, bank account pin password). So, users don’t care about their password as they will open the program using their biometrics (faces, fingerprints).

Proposed and Similar Systems comparison

	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

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)

Non-functional Requirements:

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

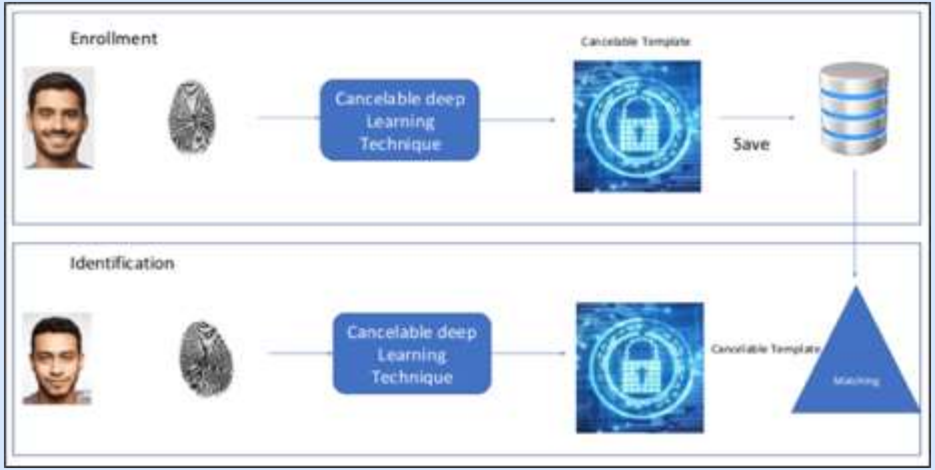
Hardware Requirements

Laptop with GPU (RTX 3070 or RTX 3080 TI)

Software Requirements:

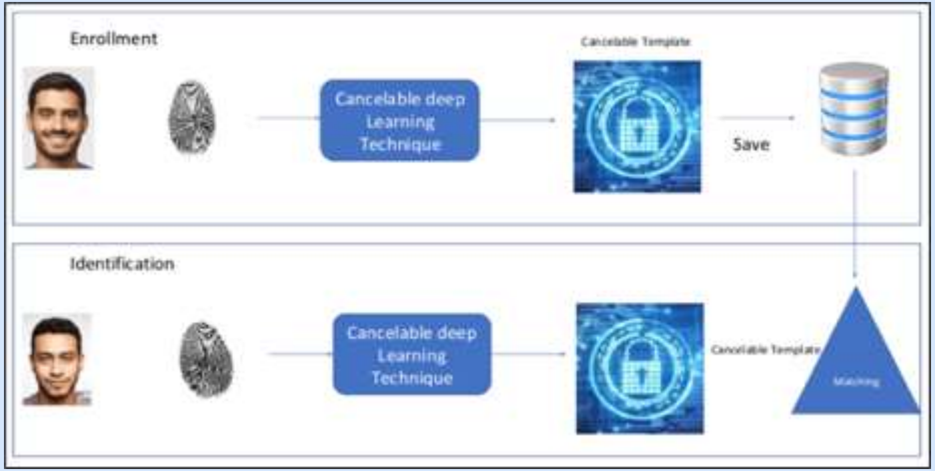
- Python
- TensorFlow 2.0
- Keras

Proposed System

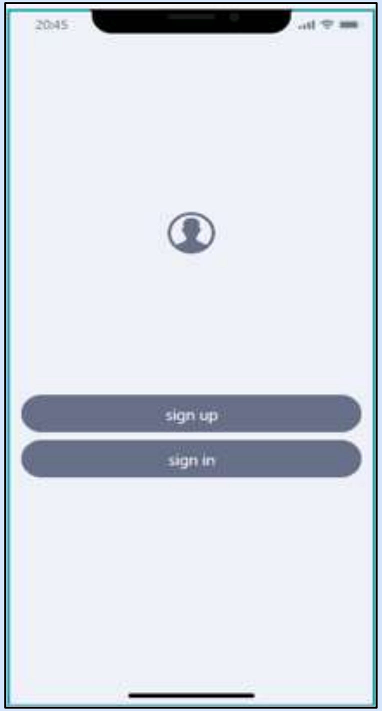


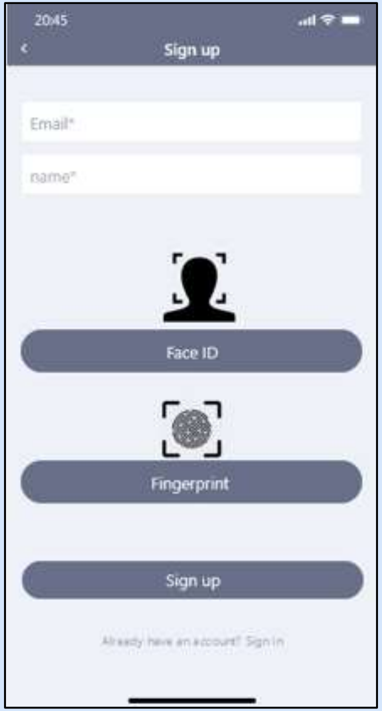
In this project, we will develop a system for personal identification based on cancelable Multi-biometrics combined with deep learning. This system will be used to access remotely to a bank by using biometrics instead of using a password.

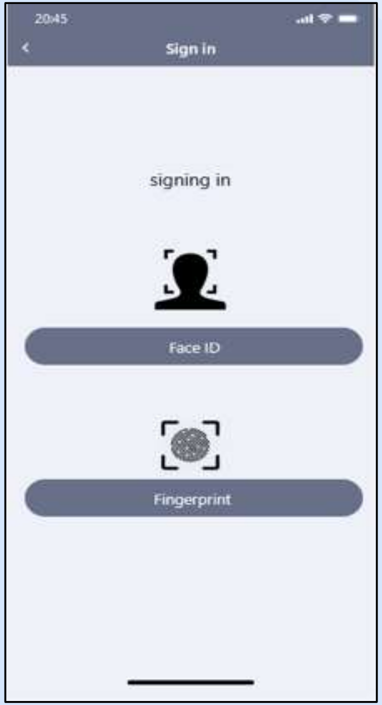
System design



System interface

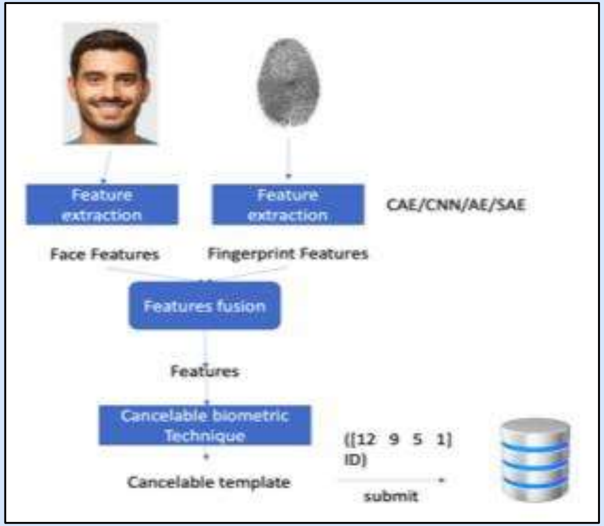




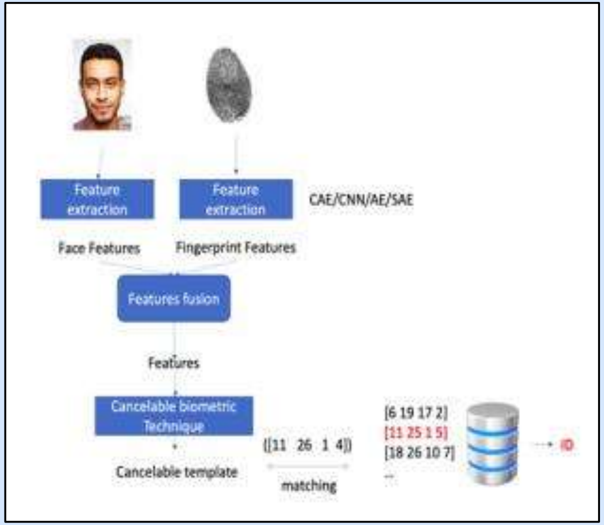


System Architecture

Enrollment Phase (Sign up):



Identification Phase (Sign in):



Cancelable Technique

4 5 2 9 → 12 9 5 1

02

Implementation

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.

Implementation details

Python code: (Data preparation)

```
“ dataset = read_csv("D:/Data/Biometrics/Faces/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]

filename = 'D:/Data/Biometrics/Faces/selectedfaces.npy'
np.save(filename,  SelectedFaces)
”
```

Implementation details

Python code: (Data preparation)

```
dataset = read_csv("D:/Data/Biometrics/Fingerprints/Datasetfingerprints.csv")
Fingerprints = dataset.iloc[:, 0].values
Labels = dataset.iloc[:, 1].values

seed(int(time.time()))

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]

filename = 'D:/Data/Biometrics/Fingerprints/selectedfingerprints.npy'
np.save(filename, SelectedFingerprints)
```

Implementation details

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.

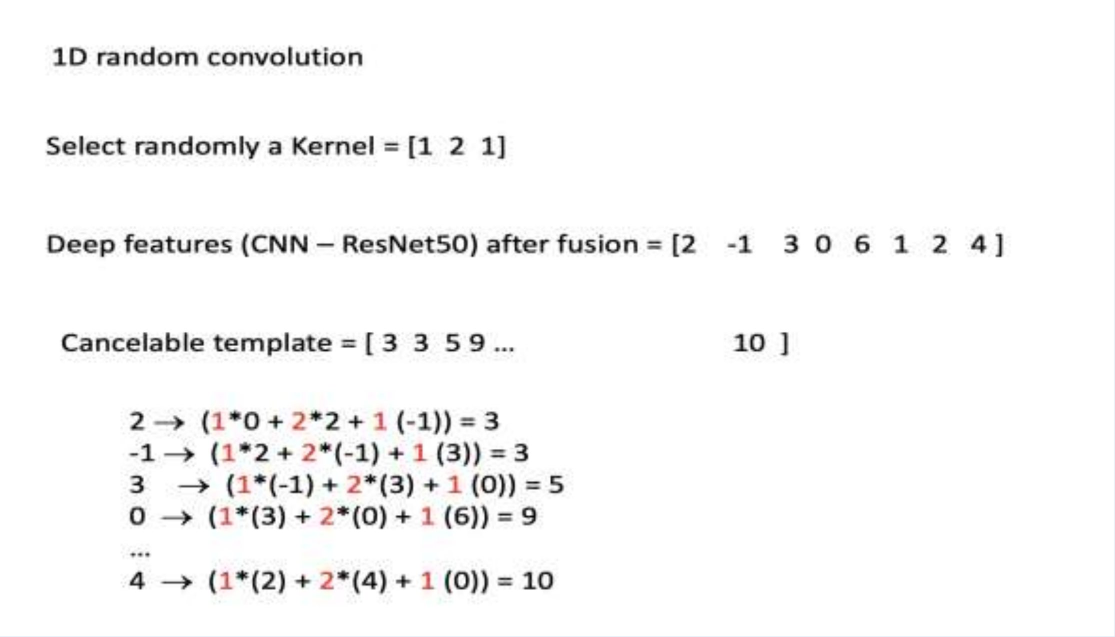
These deep features are then merged and modified using random convolution. The obtained features correspond to the cancelable template which will be stored in the database to be used in the identification phase.

Implementation details

enrollment method :

```
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(deepfeatures)

# random convolution of deepfeatures
X = deepfeatures.copy()
X_new = numpy.append(X[0], X[1], axis=0)
X_final = random_convolution(X_new,RandomKernel)
# Store X_final in the database
```

Implementation details



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

Implementation details

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.

Implementation details

Personal identification :

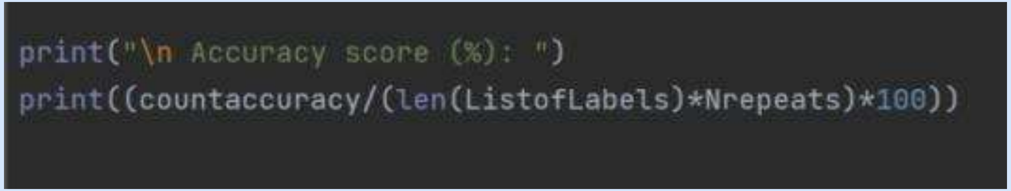
```
# matching ... using Euclidean Distance between Cancelable template and DB ...

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)
```

Implementation details

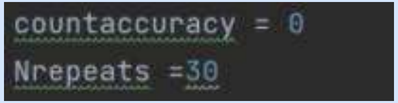
Evaluation :

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.

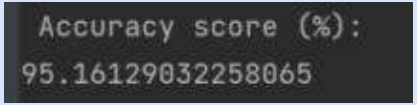
Implementation details



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



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



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

03

Testing

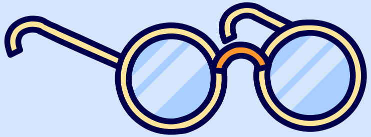
Test plan

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

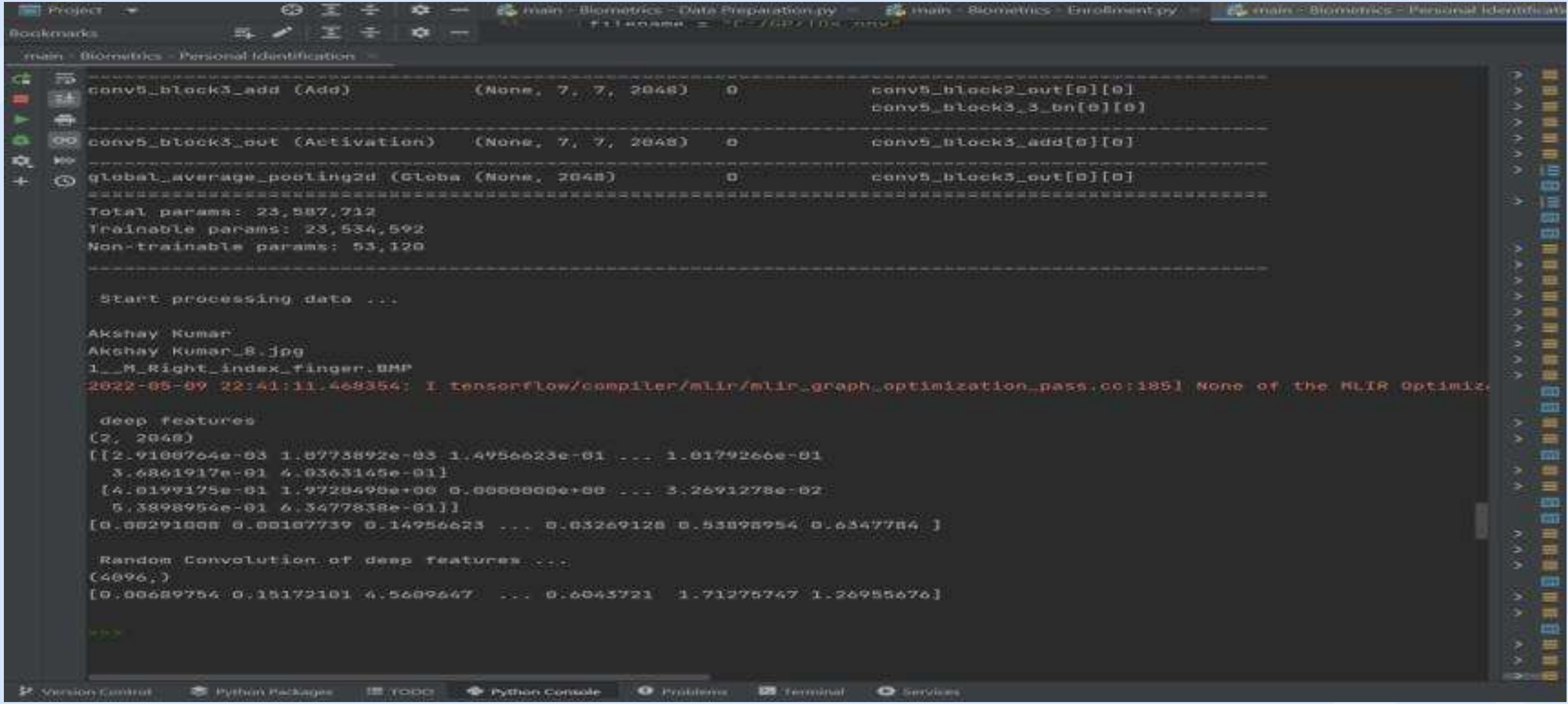
And also with higher accuracy we will get great and excellent results.
we will create six cases to test the level of accuracy and evaluate the performance of the system.
We have tried (5,10,15,20,25,30) repeats and the best final accuracy was excellent which is 30 repeats.

That was our test plan in general.

Test cases

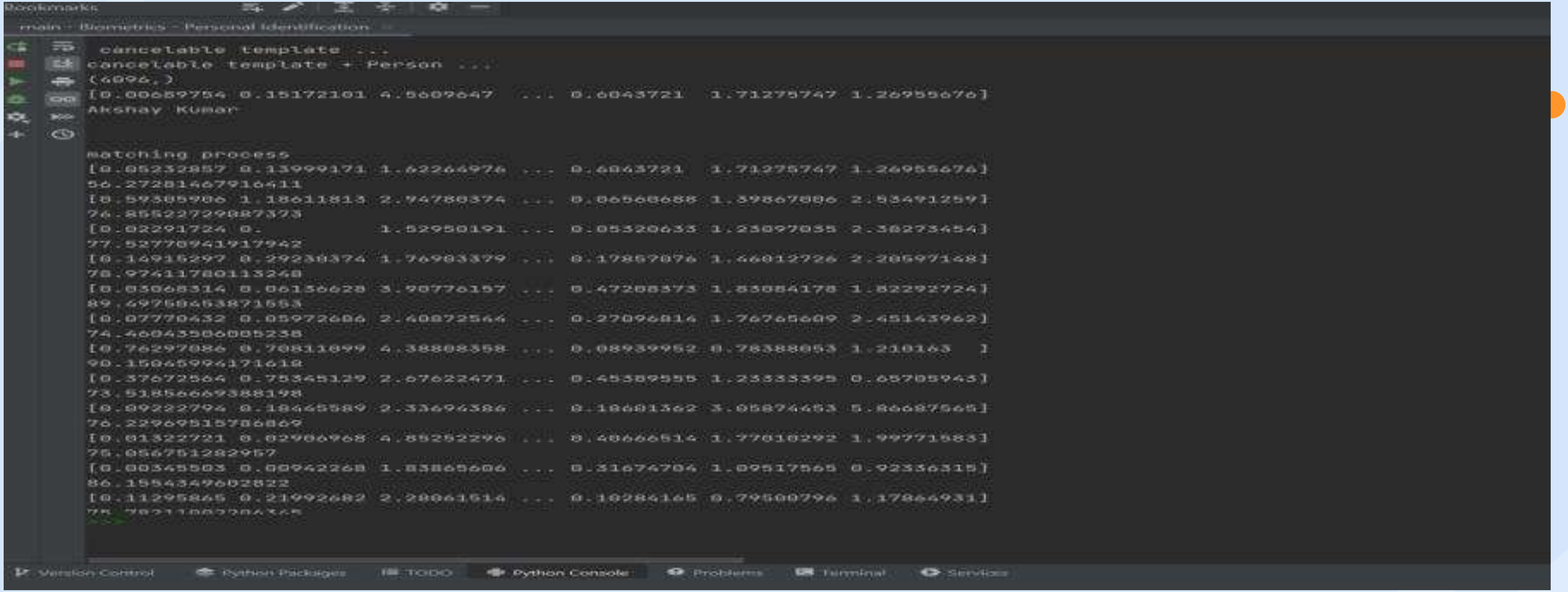


Case 1-1


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

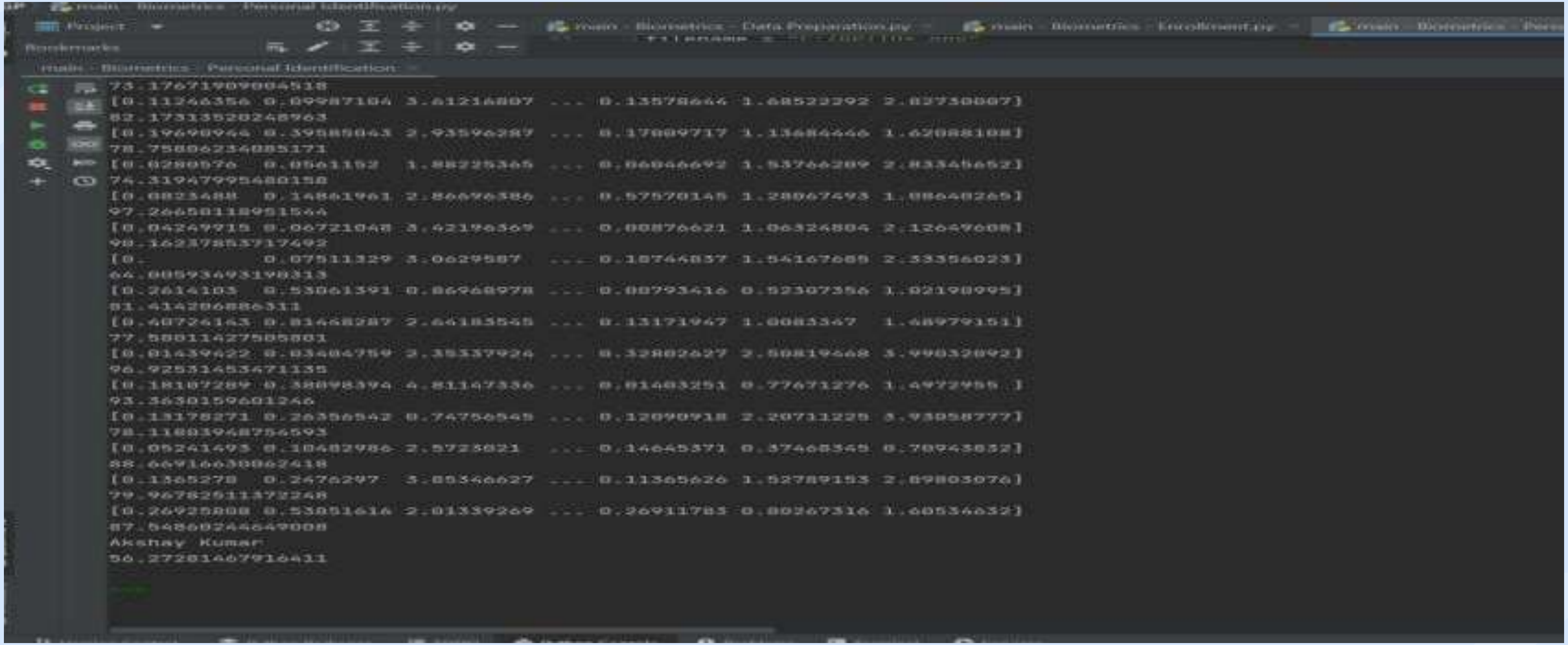
```
________________________________________________________________________________________________________
conv5_block3_add (Add)          (None, 7, 7, 2048)   0              conv5_block2_out[0][0]
                                                                    conv5_block3_3_bn[0][0]
________________________________________________________________________________________________________
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)
[[2.9108764e-03 1.07738892e-03 1.49566623e-01 ... 1.01792660e-01
  3.6861917e-01 4.0363145e-01]
 [4.0199175e-01 1.9728490e+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]

>>>
```

Case 1-2



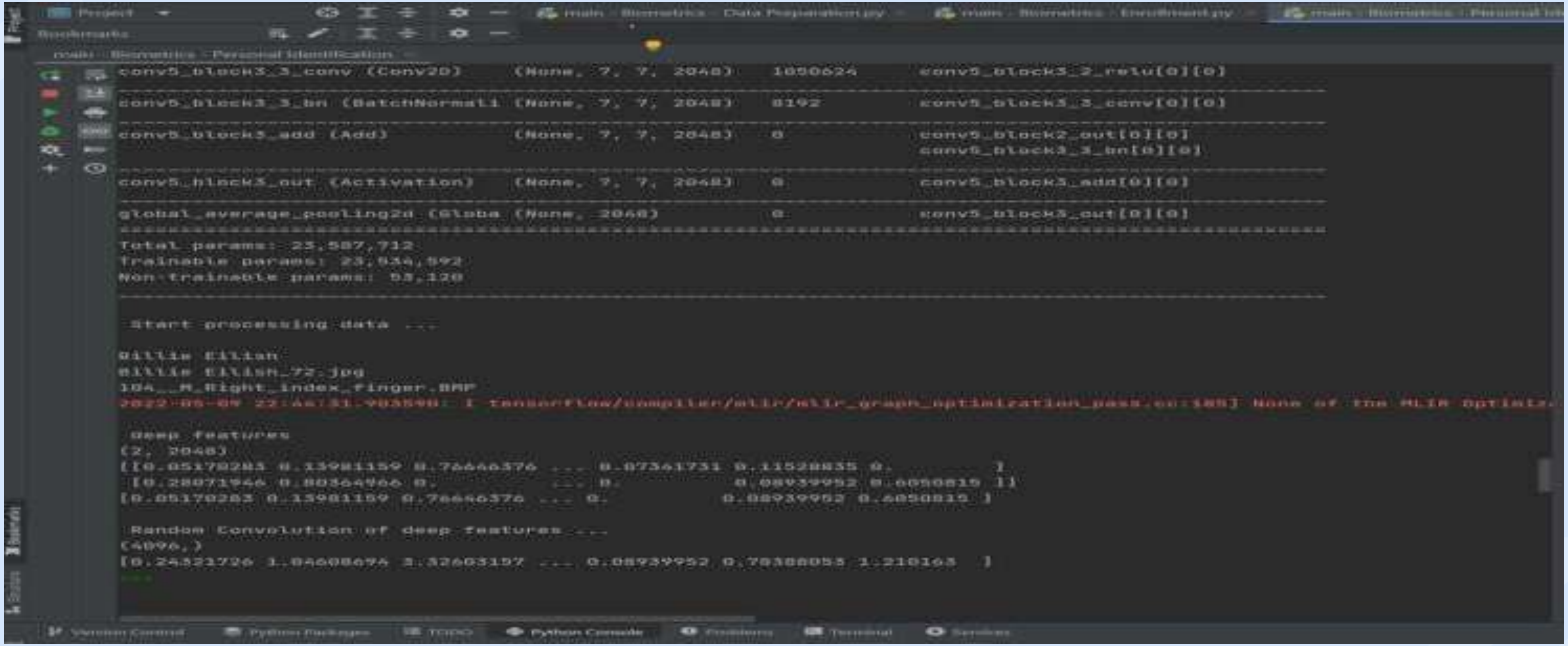
```
 cancelable template ...
cancelable template + Person ...
(4096,)
[0.00659754 0.15172101 4.5609647  ... 0.6043721  1.71275747 1.26955676]
Akshay Kumar


matching process
[0.05232857 0.13999171 1.62264976 ... 0.6043721  1.71275747 1.26955676]
56.27281467916411
[0.59305906 1.18611813 2.94780374 ... 0.06568688 1.39867806 2.53491259]
76.85522729887373
[0.02291724 0.         1.52950191 ... 0.05320633 1.23097035 2.38273454]
77.52770941917942
[0.14915297 0.29230374 1.76983379 ... 0.17857876 1.46012726 2.28597168]
78.97411780113248
[0.03068314 0.06136628 3.90776157 ... 0.47208373 1.83084178 1.82292724]
89.49758453871553
[0.07770432 0.05972686 2.40872544 ... 0.27096816 1.76765609 2.45143962]
74.46043586085238
[0.76297086 0.70811099 4.38808358 ... 0.08939952 0.78388853 1.210163   ]
90.15045994171618
[0.37672564 0.75345129 2.67622471 ... 0.45389555 1.23333395 0.65705943]
73.51856669388198
[0.09222794 0.18465589 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]
78.78211002204345
>>>
```

Case 1-3


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

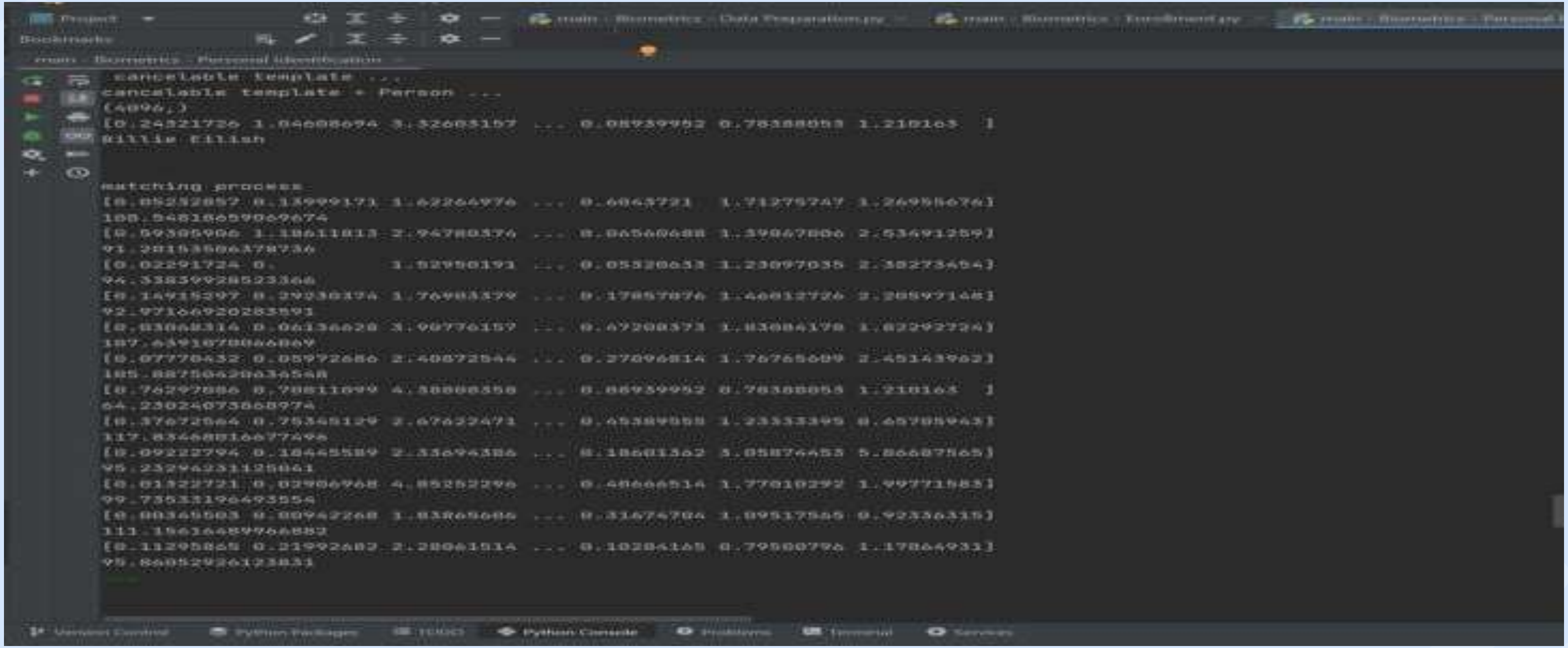
```
73.17671909004518
[0.11246356 0.09987104 3.61216807 ... 0.13578644 1.68522292 2.82730007]
82.17313520248963
[0.19690944 0.39585043 2.93596287 ... 0.17809717 1.13684446 1.62088108]
78.75880623408517 1
[0.0280076  0.0561152  1.88225365 ... 0.06046692 1.53766289 2.83345652]
74.31947995480158
[0.0823488  0.14861961 2.86696386 ... 0.57570145 1.28067493 1.08648265]
97.26658118951564
[0.04249715 0.06721048 3.42196369 ... 0.00876621 1.06324804 2.12647608]
98.16237853717492
[0.         0.07511329 3.0629587  ... 0.18744837 1.54167685 2.33356023]
64.08593493198313
[0.2614103  0.53061391 0.86968978 ... 0.08793416 0.52307356 1.02198995]
81.41420688631 1
[0.40726143 0.81468287 2.64183545 ... 0.13171947 1.0083347  1.48979151]
77.58011427585801
[0.01439622 0.03484759 2.35337924 ... 0.32882627 2.50819648 3.99832892]
96.92531453471135
[0.18107289 0.38098394 4.81147336 ... 0.01483251 0.77671276 1.4972955 ]
93.36301596012 46
[0.13178271 0.26356542 0.74756545 ... 0.12090918 2.20711225 3.93858777]
78.11803948756593
[0.05241493 0.10482986 2.5723821  ... 0.14645371 0.37468345 0.78943832]
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
```

Case 2-1


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 22:46:31.983598: 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 1.84608694 3.32603197 ... 0.08939952 0.78388053 1.210163  ]
>>>
```

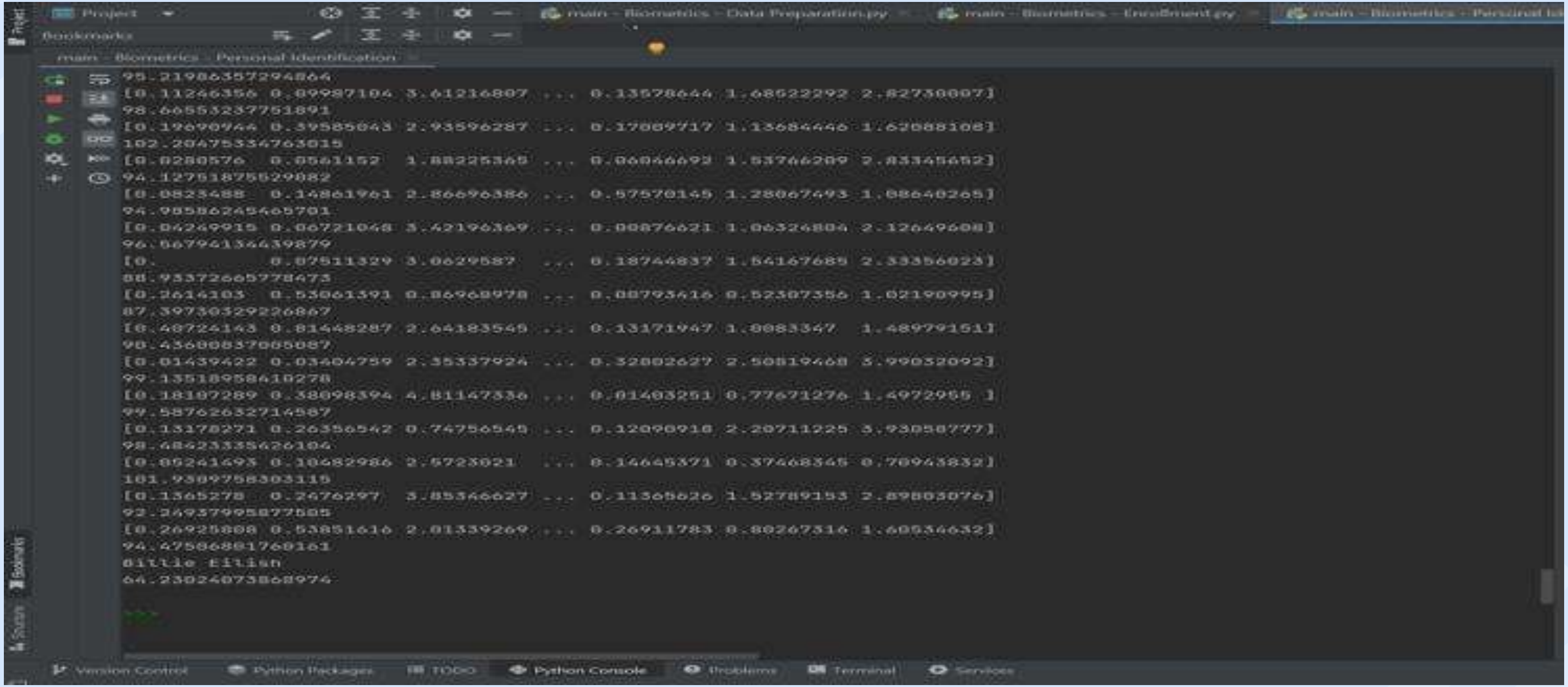
Case 2-2



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

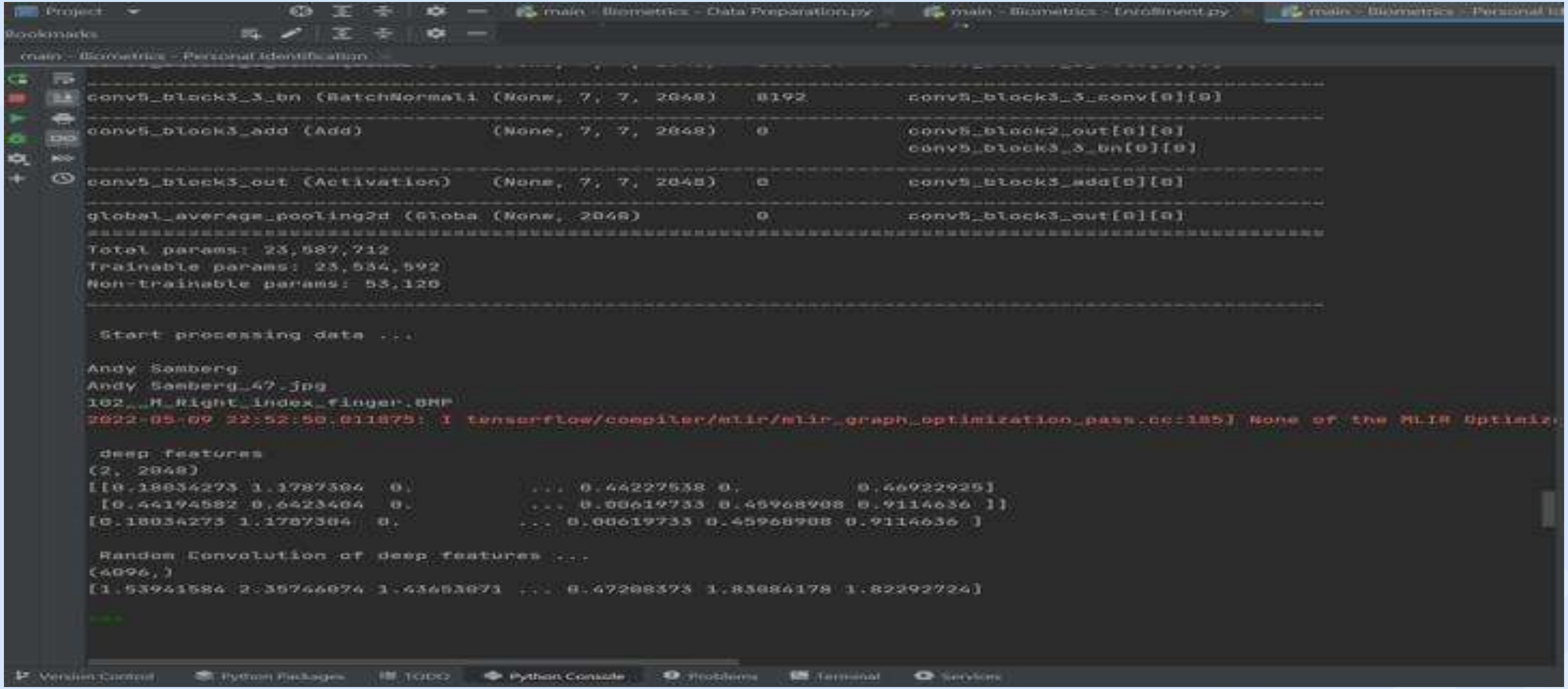
```
 cancelable template ...
cancelable template + Person ...
(4096,)
[0.24321726 1.84608694 3.32683157 ... 0.08939952 0.78388853 1.218163  ]
Billie Eilish


matching process
[0.05232857 0.13999171 1.62264976 ... 0.6843721  1.71275747 1.26955676]
108.54818659069674
[0.59385906 1.18611813 2.94780376 ... 0.06560688 1.39867806 2.53491259]
91.28153586378736
[0.02291724 0.         1.52950191 ... 0.05520633 1.23097035 2.38273454]
94.33835992852336
[0.14915297 0.29230174 1.76983379 ... 0.17857876 1.46812726 2.28597148]
92.97166920283591
[0.83868314 0.06136628 3.98776157 ... 0.47208373 1.83684178 1.82292724]
187.65918788668697
[0.07770432 0.05972686 2.40872544 ... 0.27896814 1.76765689 2.45143962]
185.88750642863548
[0.76297886 0.78811699 4.38888358 ... 0.86939952 0.78388855 1.218163  ]
64.23024073868974
[0.37672564 0.75345129 2.67622471 ... 0.65389855 1.23333395 0.65785943]
117.83468816677498
[0.09222794 0.18445589 2.33694386 ... 0.18681362 3.85874453 5.86687565]
95.23294231125861
[0.01322721 0.02986968 4.85252296 ... 0.48666514 1.77818292 1.99771583]
99.73533196493554
[0.88345583 0.80942268 1.83869686 ... 0.31674784 1.89517585 0.92336315]
111.15616489766882
[0.11295865 0.21992682 2.28861514 ... 0.10284165 0.79588796 1.17864931]
95.86852926123831
```

Case 2-3


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

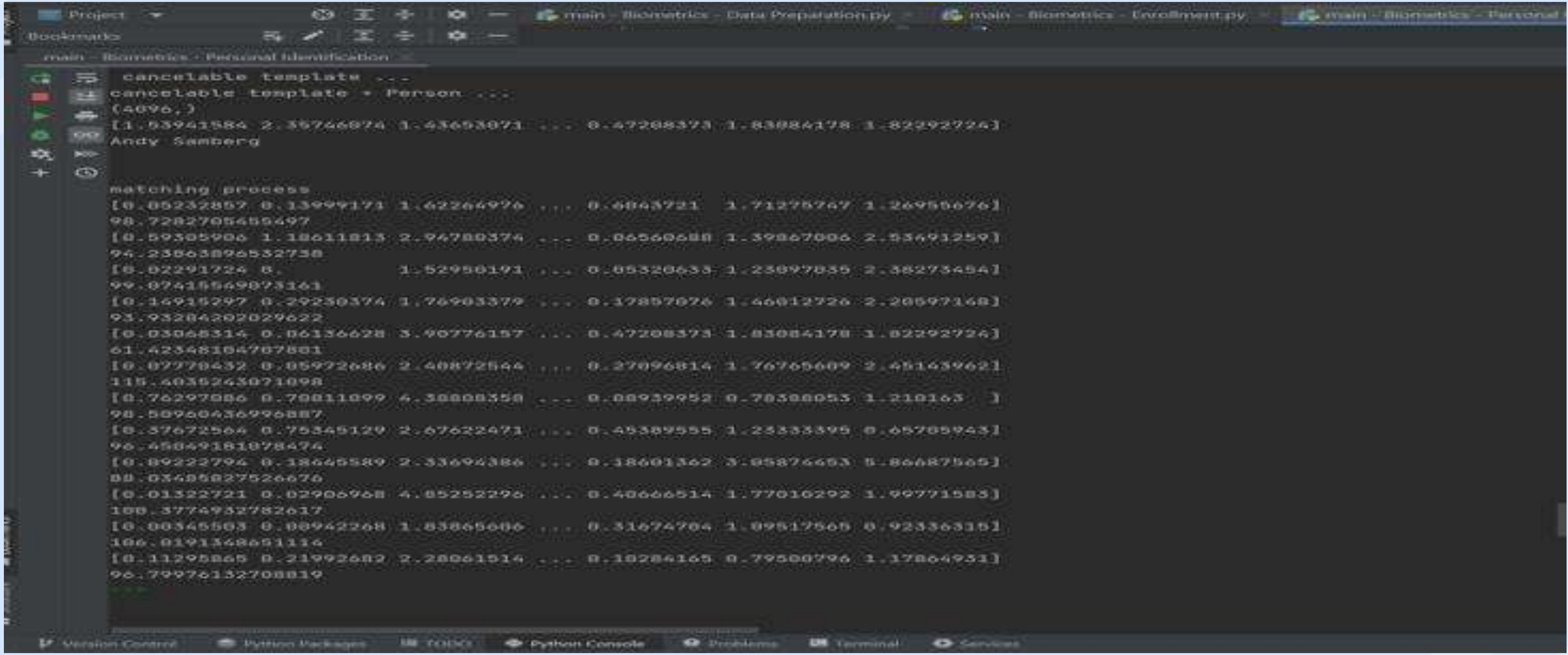
```
95.21986357294864
[0.11246356 0.89987104 3.61216807 ... 0.13578644 1.68522292 2.82730007]
98.66553237751891
[0.19690944 0.39585043 2.93596287 ... 0.17089717 1.13684446 1.62088108]
102.20475334763015
[0.0280576  0.0561152  1.88225365 ... 0.06046692 1.53766289 2.83345652]
94.12751875529082
[0.0823488  0.14861961 2.86696386 ... 0.57570145 1.28067493 1.08640265]
94.98586245465701
[0.04249915 0.06721048 3.42196369 ... 0.00876621 1.06326804 2.12649608]
96.56794134439879
[0.         0.87511329 3.0629587  ... 0.18744837 1.54167685 2.33356023]
88.93372665778473
[0.2614103  0.53061391 0.86968978 ... 0.08793416 0.52307356 1.02190995]
87.39730329226867
[0.40724143 0.81448287 2.64183545 ... 0.13171947 1.0083347  1.48979151]
98.43600837005087
[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]
98.48423335626104
[0.05241493 0.10482986 2.5723021  ... 0.14645371 0.37468345 0.78943832]
101.9389758303115
[0.1365278  0.2476297  3.85346627 ... 0.11365626 1.52789153 2.89803076]
92.24937995877585
[0.26925808 0.53851616 2.81339269 ... 0.26911783 0.80267316 1.60534632]
94.47586881760161
Billie Eilish
64.23024073868974

>>>
```

Case 3-1


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 0.6423404  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 1.43653073 ... 0.47208373 1.83884178 1.82292724]

>>>
```

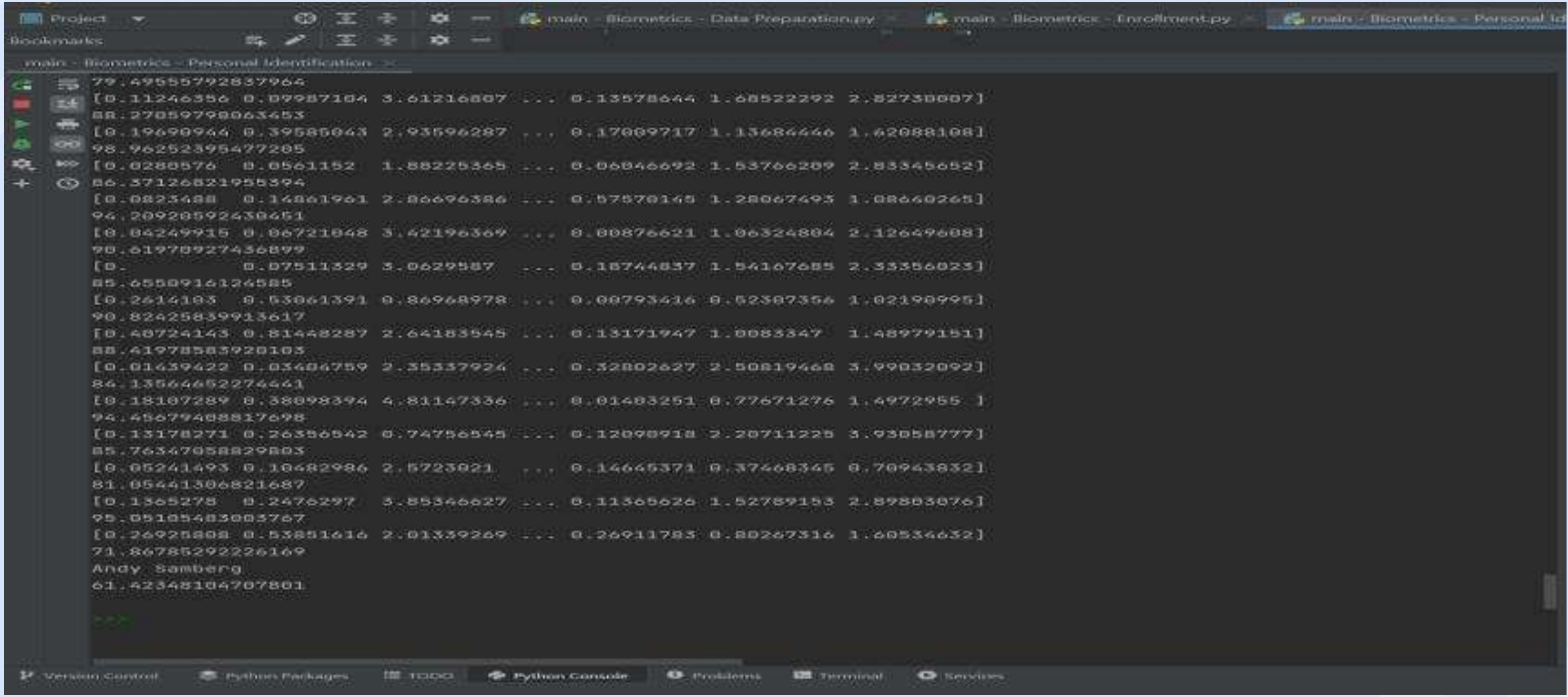
Case 3-2



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

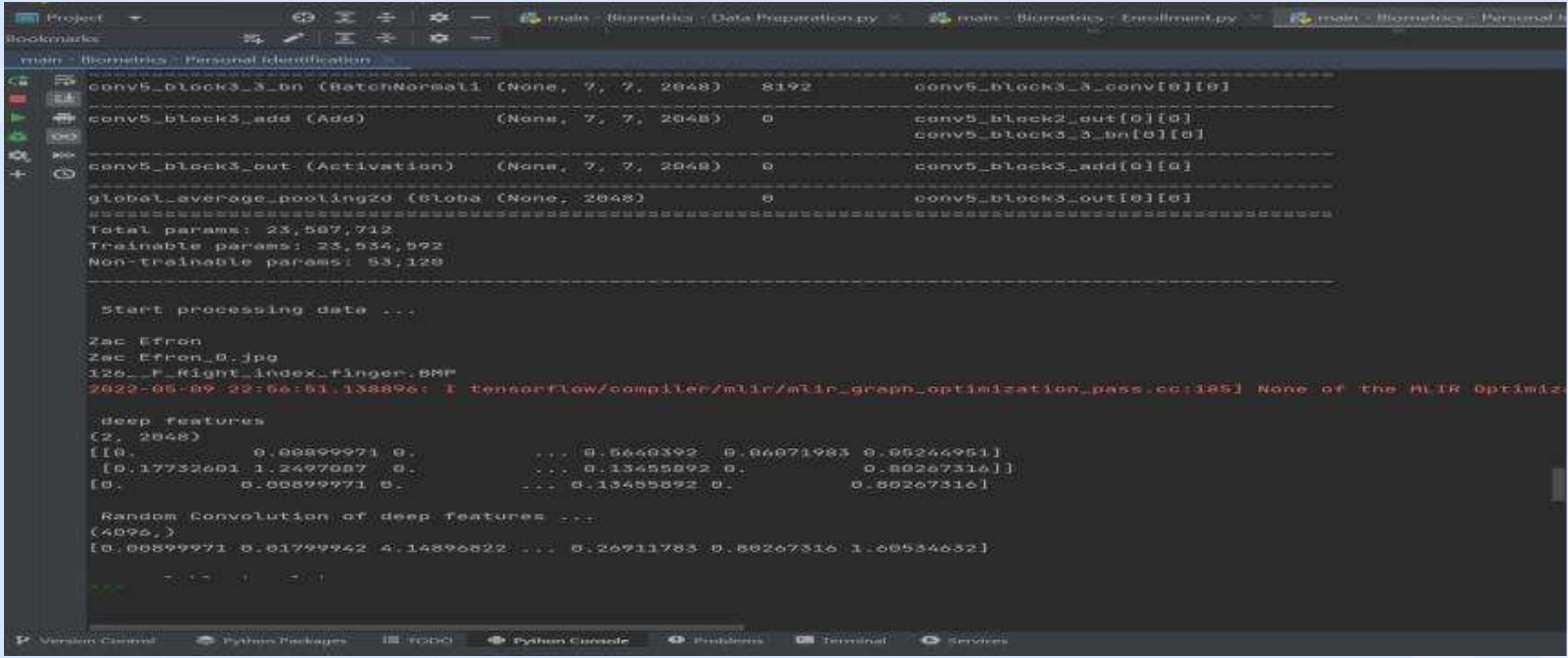
```
 cancelable template ...
cancelable template + Person ...
(4096,)
[1.53941584 2.35746874 1.43653071 ... 0.47208373 1.83084178 1.82292724]
Andy Samberg


matching process
[0.05232857 0.13999171 1.62264976 ... 0.6843721  1.71275767 1.26955676]
98.72827055655497
[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]
99.07415549873161
[0.16915297 0.29230374 1.76903379 ... 0.17857876 1.46012726 2.20597168]
93.93284202029622
[0.03068314 0.86136628 3.90776157 ... 0.47208373 1.83084178 1.82292724]
61.42348184787881
[0.07778432 0.85972686 2.40872544 ... 0.27096814 1.76765609 2.45143962]
115.4035243071098
[0.76297086 0.70811099 4.38808358 ... 0.08939952 0.78388053 1.218163  ]
98.50960436996887
[0.37672564 0.75345129 2.67622471 ... 0.45389555 1.23333395 0.65785943]
96.45849181878474
[0.89222794 0.18645589 2.33694386 ... 0.18601362 3.85876653 5.86687565]
88.03485827526676
[0.01322721 0.02906968 4.85252296 ... 0.40666514 1.77010292 1.99771583]
100.37749327782617
[0.00345503 0.00942268 1.83865686 ... 0.31674784 1.09517565 0.92336315]
106.01913486851116
[0.11295865 0.21992682 2.28061514 ... 0.18284165 0.79500796 1.17864931]
96.79976132708819
>>>
```

Case 3-3


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

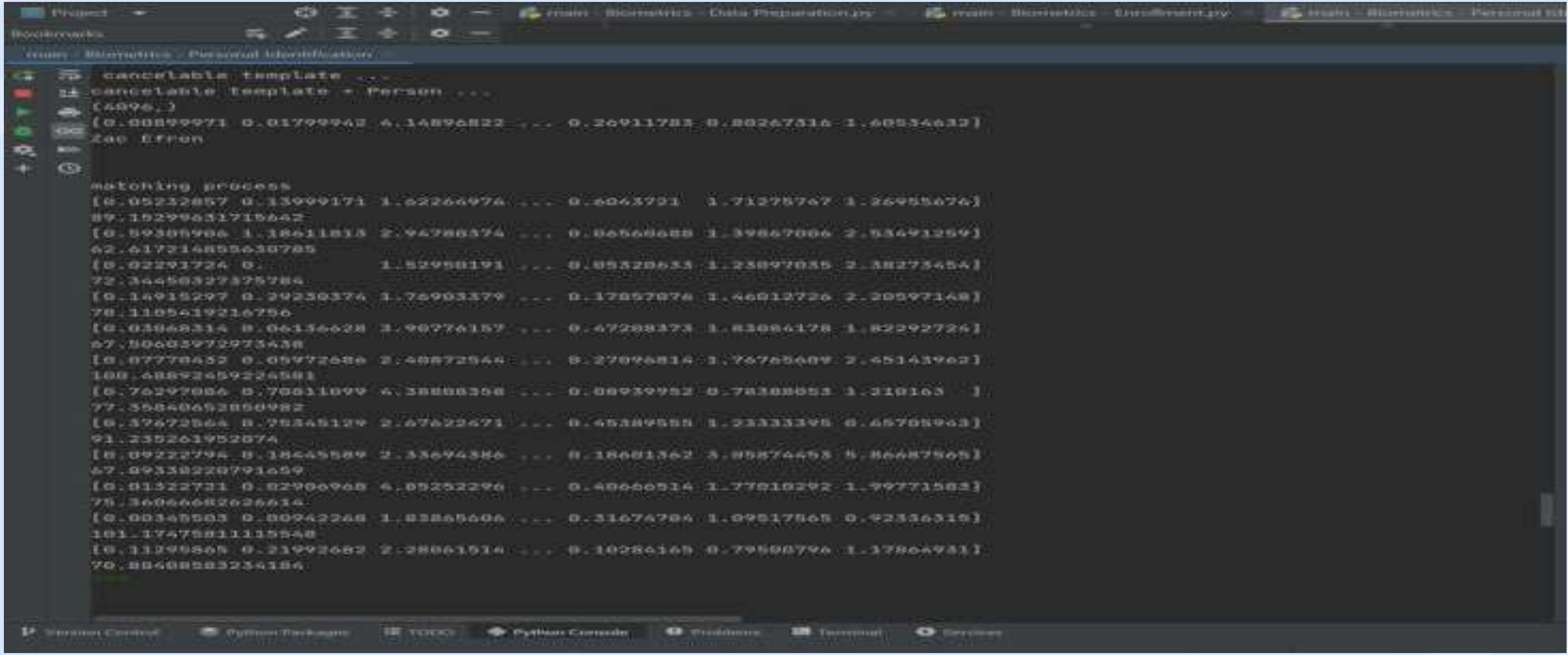
```
79.49555792837964
[0.11246356 0.09987104 3.61216807 ... 0.13578644 1.68522292 2.82730007]
88.27059798063453
[0.19690944 0.39585043 2.93596287 ... 0.17009717 1.13684446 1.62088108]
98.96252395477205
[0.0280576  0.0561152  1.88225365 ... 0.06046692 1.53766209 2.83345652]
86.37126821955394
[0.0823488  0.14861961 2.86696386 ... 0.57570145 1.28067493 1.08660265]
94.20920592438451
[0.04249915 0.06721048 3.42196369 ... 0.00876621 1.06324804 2.12649608]
90.61970927436899
[0.         0.07511329 3.0629587  ... 0.18744837 1.54167685 2.33356023]
85.6558916124585
[0.2614103  0.53061391 0.86968978 ... 0.00793416 0.52387356 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]
84.13564652274461
[0.18107289 0.38098394 4.81147336 ... 0.01483251 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

>>>
```

Case 4-1


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

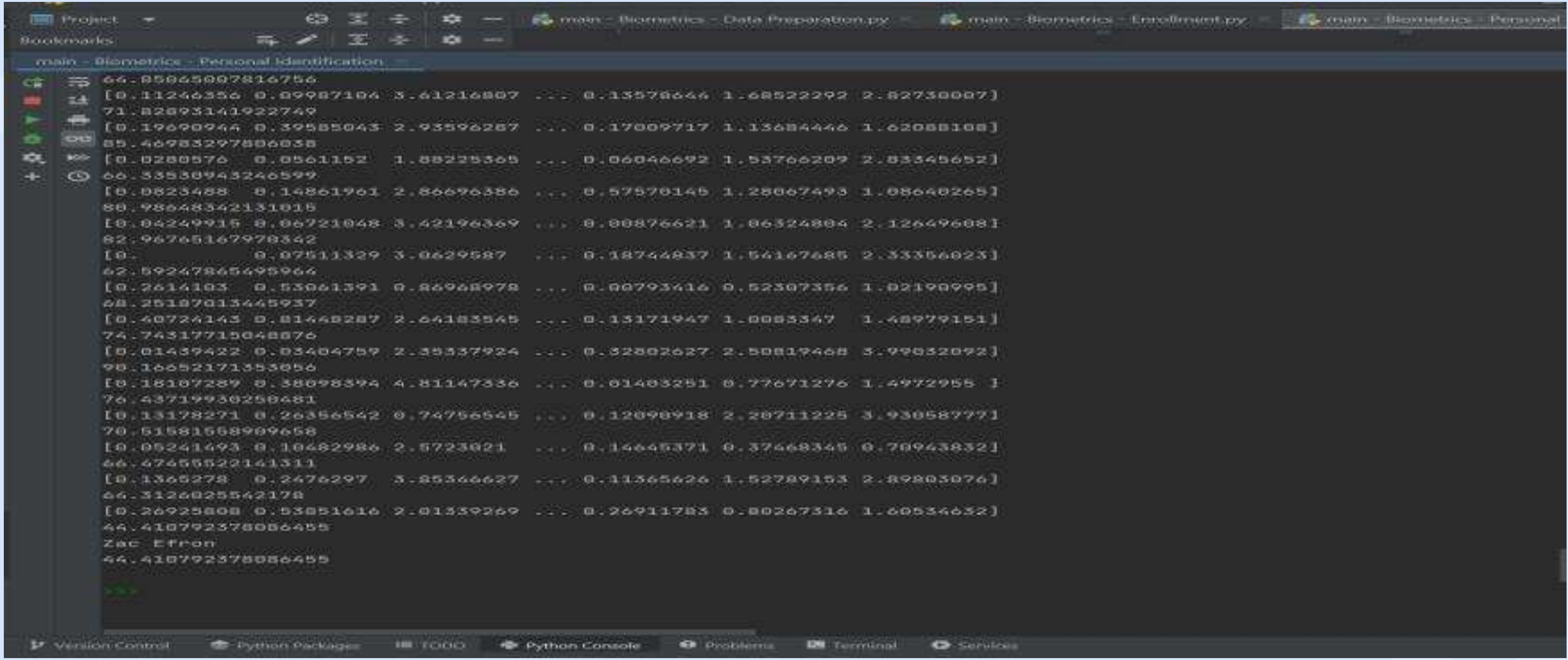
```
==================================================================================================
conv5_block3_3_bn (BatchNormali (None, 7, 7, 2048)   8192        conv5_block3_3_conv[0][0]
__________________________________________________________________________________________________
conv5_block3_add (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]
```

Case 4-2


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

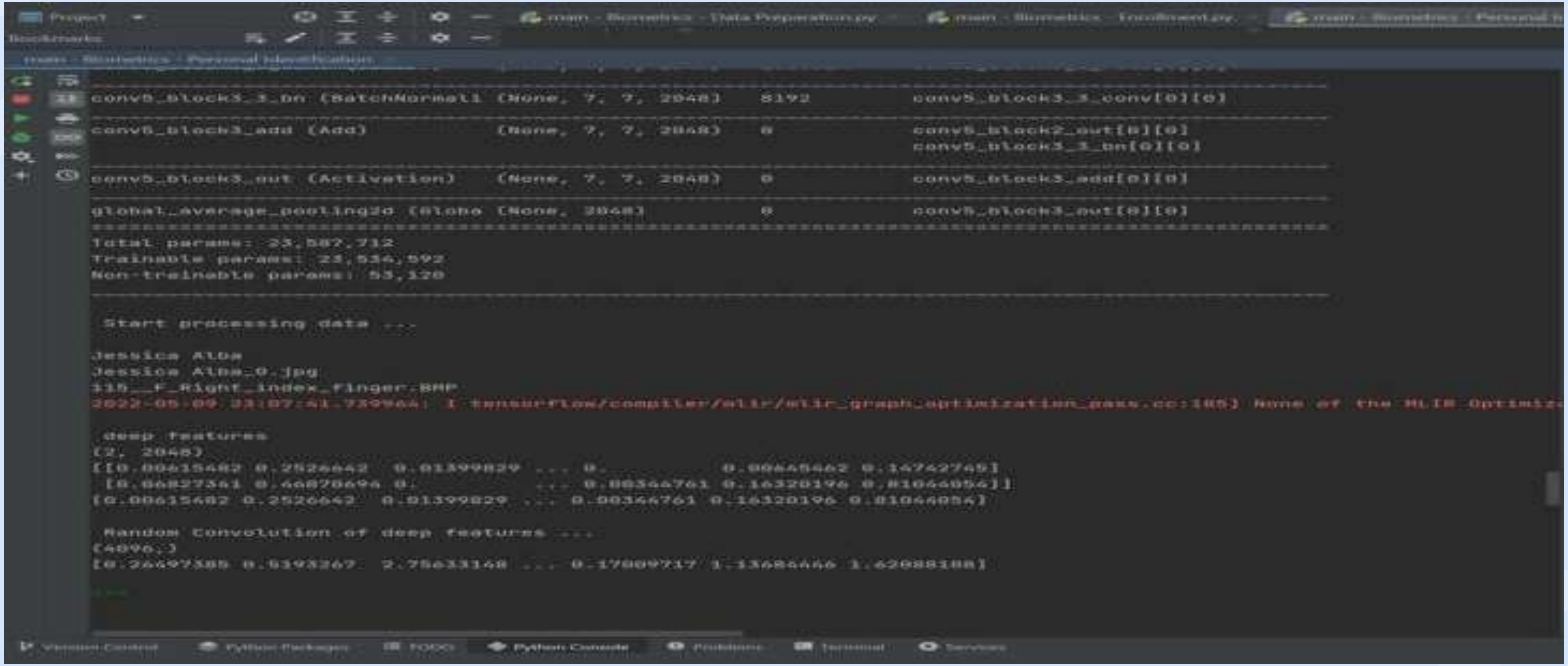
```
 cancelable template ...
cancelable template - Person ...
(4096,)
[0.00899971 0.01799942 4.14896822 ... 0.26911783 0.80267316 1.60534632]
Zac Efron


matching process
[0.05232857 0.13999171 1.62266976 ... 0.6043721  1.71275767 1.26955676]
89.15299631715642
[0.59305906 1.18611815 2.94788374 ... 0.06568688 1.39867006 2.53691259]
62.617214855630785
[0.02291724 0.         1.52958191 ... 0.05320633 1.23897035 2.38273454]
72.34450327375784
[0.14915297 0.29230374 1.76903379 ... 0.17857076 1.46012726 2.20597148]
78.1105419216756
[0.03868314 0.06136628 3.90776157 ... 0.47288373 1.83084178 1.82292726]
67.50603972973438
[0.07770432 0.05972686 2.40872544 ... 0.27096814 1.76765609 2.45143962]
100.48892459224581
[0.76297086 0.70611099 4.38888358 ... 0.08939952 0.78388053 1.310163  ]
77.35840652850982
[0.37672564 0.75345129 2.67622671 ... 0.45389555 1.23333395 0.65705963]
91.235261952074
[0.09222794 0.18645589 2.33694386 ... 0.18601362 3.85874653 5.86687565]
67.09338220791659
[0.01322721 0.02906968 4.05252296 ... 0.40660514 1.77810292 1.99771583]
75.36066682626614
[0.00345583 0.00942268 1.83865606 ... 0.31674784 1.09517565 0.92336315]
101.17475811115548
[0.11295865 0.21992682 2.28861514 ... 0.10284165 0.79508796 1.17864931]
70.88408583234186
```

Case 4-3


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

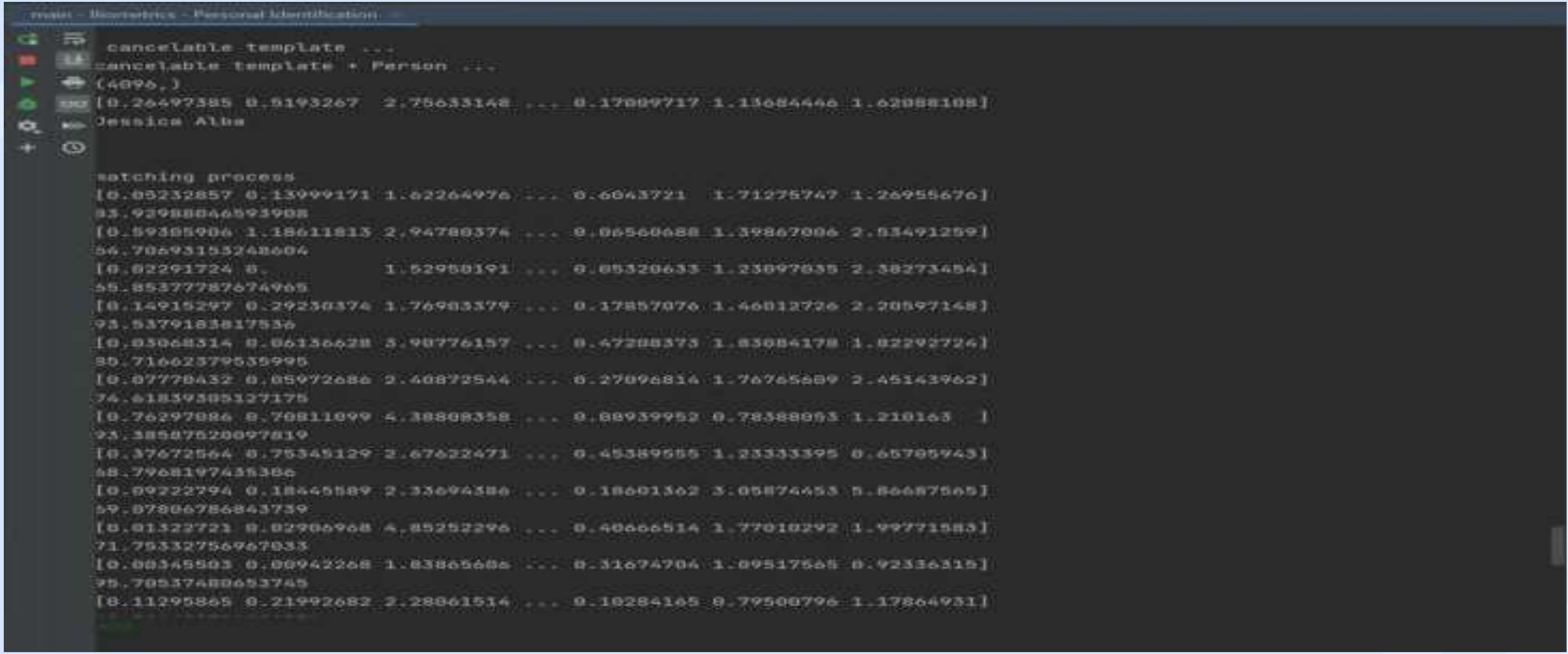
```
66.85065007816756
[0.11246356 0.09987104 3.61216807 ... 0.13578646 1.68522292 2.82730007]
71.82893141922749
[0.19690944 0.39585043 2.93596287 ... 0.17009717 1.13684446 1.62088108]
85.46783297886038
[0.0280576  0.0561152  1.88225365 ... 0.06046692 1.53766209 2.83345652]
66.33530943246599
[0.0823488  0.14861961 2.86696386 ... 0.57578145 1.28067493 1.08640265]
80.98648342131015
[0.04249915 0.06721048 3.42196369 ... 0.00876621 1.06324804 2.12649608]
82.96765167978342
[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]
98.16652171353056
[0.18107289 0.38098394 4.81147336 ... 0.01403251 0.77671276 1.4972955 ]
76.43719930258481
[0.13178271 0.26356542 0.74756545 ... 0.12098918 2.28711225 3.93058777]
70.51581568909658
[0.05241493 0.10482986 2.5723021  ... 0.14645371 0.37468345 0.78943832]
66.47455522141311
[0.1365278  0.2476297  3.85366627 ... 0.11365626 1.52789153 2.89803076]
66.31260255642178
[0.26925808 0.53851616 2.01339269 ... 0.26911783 0.80267316 1.60534632]
44.410792378086455
Zac Efron
44.410792378086455
>>>
```

Case 5-1


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

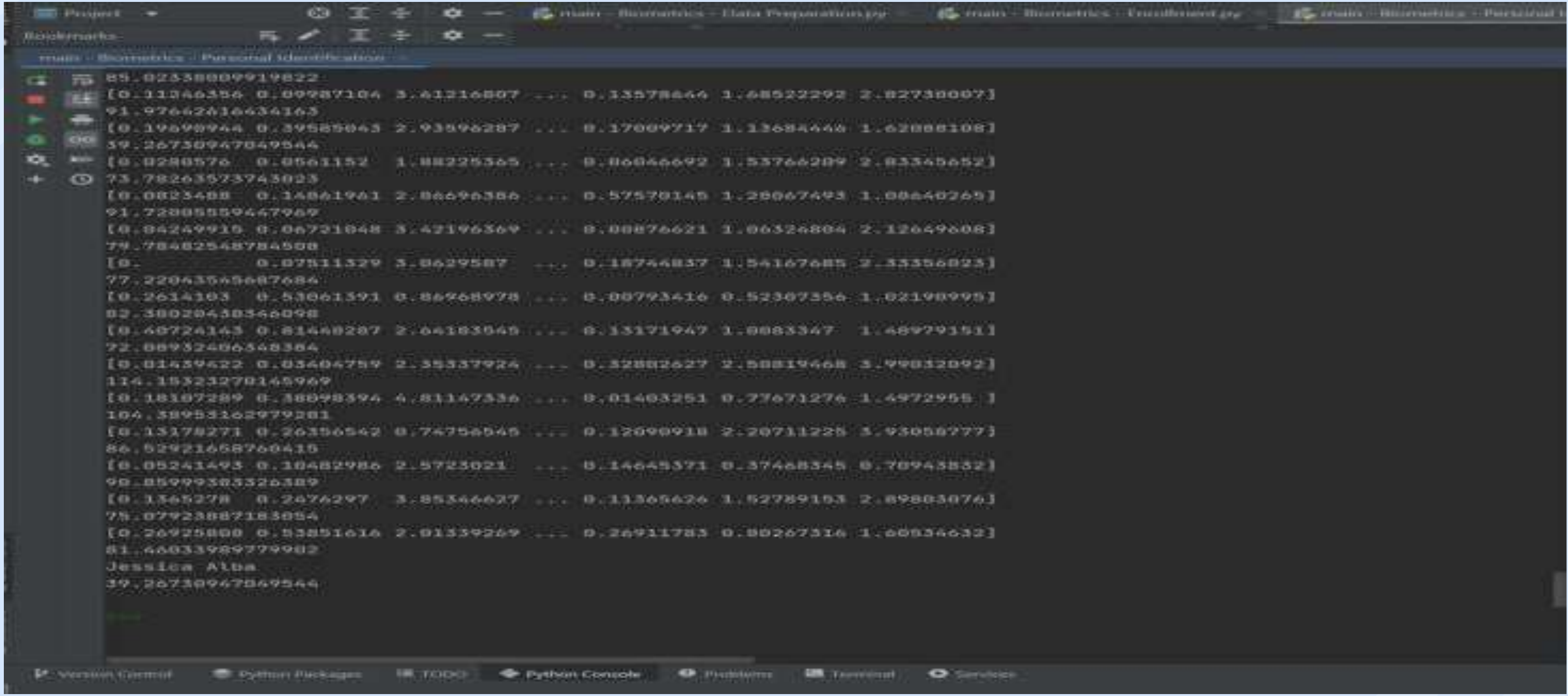
```
conv5_block3_3_bn (BatchNormali (None, 7, 7, 2048)    8192        conv5_block3_3_conv[0][0]
__________________________________________________________________________________________________
conv5_block3_add (Add)          (None, 7, 7, 2048)    0           conv5_block2_out[0][0]
                                                                  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 0.46878696 0.         ... 0.00364761 0.16320196 0.81044054]]
[0.00615482 0.2526642  0.01399829 ... 0.00364761 0.16320196 0.81044054]

 Random Convolution of deep features ...
(4096,)
[0.26497385 0.5193267  2.75633148 ... 0.17809717 1.13684466 1.62088188]
```

Case 5-2



```
 cancelable template ...
cancelable template + Person ...
(4096,)
[0.26497385 0.5193267  2.75633148 ... 0.17809717 1.13684446 1.62088108]
Jessica Alba


matching process
[0.05232857 0.13999171 1.62264976 ... 0.6043721  1.71275747 1.26955676]
83.92988046593908
[0.59385906 1.18611815 2.94788374 ... 0.06560688 1.39867006 2.53491259]
54.70693153248604
[0.82291724 0.         1.52958191 ... 0.05320633 1.23897035 2.38273454]
55.85377787674965
[0.14915297 0.29230374 1.76983379 ... 0.17857076 1.46812726 2.28597148]
93.53791838817536
[0.03068314 0.06136628 3.98776157 ... 0.47288373 1.83084178 1.82292724]
80.71662379535995
[0.07778432 0.05972686 2.40872544 ... 0.27896814 1.76765689 2.45143962]
74.61839385127175
[0.76297086 0.70811099 4.38808358 ... 0.88939952 0.78388893 1.218163  ]
93.38587528097819
[0.37672564 0.75345129 2.67622471 ... 0.45389555 1.23333395 0.65785943]
68.79681974353866
[0.09222794 0.18445589 2.33694386 ... 0.18601362 3.05874453 5.86687565]
59.87806786843739
[0.01322721 0.02986968 4.85252296 ... 0.40666514 1.77018292 1.99771583]
71.79332756967033
[0.08345503 0.08942268 1.83865686 ... 0.31674704 1.09517565 0.92336315]
75.70537488653745
[0.11295865 0.21992682 2.28861514 ... 0.10284165 0.79508796 1.17864931]
```

Case 5-3


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

```
85.02338809919822
[0.11346356 0.09987104 3.41216807 ... 0.13578644 1.68522292 2.82730007]
91.97642616434163
[0.19690944 0.39585043 2.93596287 ... 0.17009717 1.13684446 1.62088108]
39.267309470849544
[0.0280576  0.0561152  1.88225365 ... 0.86046692 1.53766289 2.83345652]
73.78263573743023
[0.0823488  0.14861961 2.86696386 ... 0.57570145 1.28067493 1.08640265]
91.72885559447969
[0.84249915 0.06721848 3.42196369 ... 0.08876621 1.06326804 2.12649608]
79.78482548784508
[0.         0.07511329 3.8629587  ... 0.18744837 1.54167685 2.33356823]
77.22043545687684
[0.2614183  0.53061391 0.86968978 ... 0.00793416 0.52307356 1.02198995]
82.38020438346098
[0.40724143 0.81440287 2.04183545 ... 0.13171947 1.0883347  1.48979151]
72.08932486348384
[0.01439422 0.03404759 2.35337924 ... 0.32882627 2.58819468 3.99832892]
114.15323278145969
[0.18187289 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 0.78943832]
98.85999383326389
[0.1365278  0.2476297  3.85346627 ... 0.11365626 1.52789153 2.89883876]
75.07923887183854
[0.26925808 0.53851616 2.01339269 ... 0.26911783 0.80267316 1.60534632]
81.46033989779982
Jessica Alba
39.267309470849544
```

Test results

Test results

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

04

Conclusion

Conclusion

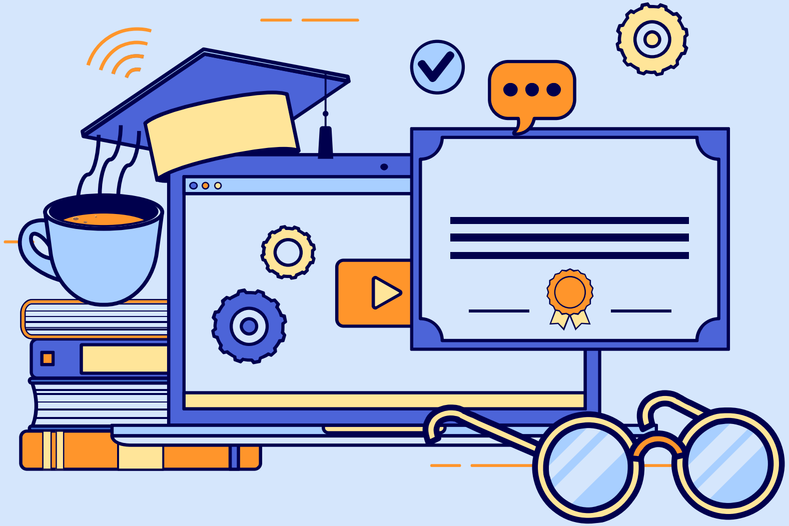
In this project, we have reviewed some of the latest research, which related to cancelable biometrics using deep learning .

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 cancelable 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.

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.



THANKS!