Signature Forgery Detection

A Mini Project Report Submitted by

Khushi S Bhimani
(4NM20AI021)

Krishna M S
(4NM20AI022)

UNDER THE GUIDANCE OF

Mr. Mahesh B L
Associate Professor Gd-II

Department of Artificial Intelligence and Machine Learning Engineering

In partial fulfillment of the requirements for the

Pattern And Visual Recognition–
20AM503

NITTE
(Deemed to be University)
NMAM INSTITUTE
OF TECHNOLOGY

Nitte(DU) established under Section 3 of UGC Act 1956 | Accredited with 'A+' Grade by NAAC
:

☎ 08258 - 281039 – 281263, Fax: 08258 – 281265

December 2022

NITTE
EDUCATION TRUST

N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India
(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC
☎ : 08258 - 281039 - 281263, Fax: 08258 - 281265

Department of Artificial Intelligence and Machine Learning Engineering

B.E. CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021

CERTIFICATE

Certified that the mini project work entitled

"Signature Forgery Detection"

is a bonafide work carried out by

Khushi S Bhimani
(4NM20AI021)

Krishna M S
(4NM20AI022)

in partial fulfilment of the requirements for the award of Bachelor of Engineering Degree in Artificial Intelligence and Machine Learning Engineering prescribed by Visvesvaraya Technological University, Belgaum during the year 2022-2023.

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library.

The mini project report has been approved as it satisfies the academic requirements in respect of the mini project work prescribed for the Bachelor of Engineering Degree.

13/12/2022

Signature of Guide

Signature of HOD

Evaluation

Name of the Examiners **Signature with Date**

1. Disha D·N

2. Rakshitha 13/12/22

ACKNOWLEDGEMENT

We believe that our mini project will be complete only after we thank the people who have contributed to make this mini project successful.

First and foremost, our sincere thanks to our beloved principal, **Dr. Niranjan N. Chiplunkar** for giving us an opportunity to carry out our mini project work at our college and providing us with all the needed facilities.

I acknowledge the support and valuable inputs given by**, Dr. Sharada U Shenoy** the Head of the Department, Artificial Intelligence and Machine Learning Engineering, NMAMIT, Nitte

We express our deep sense of gratitude and indebtedness to our guide **Mr. Mahesh B L**, Assistant Professor Artificial Intelligence and Machine Learning Engineering, for his inspiring guidance, constant encouragement, support and suggestions for improvement during the course for our mini project.

We also thank all those who have supported us throughout the entire duration of our mini project.

Finally, we thank the staff members of the Department of Artificial Intelligence and Machine Learning Engineering and all our friends for their honest opinions and suggestions throughout the course of our mini project.

Khushi S Bhimani

Krishna M S

TABLE OF CONTENTS

CHAPTER 1:

1.1 ABSTRACT

Signature plays an important role in banking, financial, commercial etc. Signature may be unique for each person. However, with signatures comes many challengessince any two signatures may look very similar with little to no differences written by the same person. As there are unique

and important variations in the feature elements of each signature, thus in order to match a particular signature with the database, the structural parameters of the signatures along with the local variations in the signature characteristics are used. In order to avoid any such identity crimes committed in banks and many other companies, forgery detection systems

is a solution to this problem along with the help of the concepts of machine learning algorithms and CNN. This software can be used to validate signatures across many platforms like loans, legal document signing, application signing, applying and much more.

Signature verification is one of the biometric techniques frequently used for personal identification. In many commercial scenarios, such as bank check payment, the signature verification process is based on human examination of a single known sample. Although there is extensive research on automatic signature verification, yet few attempts have been made to perform the verification based on a single reference sample.

1.2 INTRODUCTION

The need for signature verification is very much important because unlike passwords signatures cannot be changed or forgotten because it is unique for everyone and it is considered as the important method for verification. The techniques and system used to solve signature verification is divided into offline signature and online signature methods. In offline signature verification method more number of hardware's were not used and images were captured using camera, whereas in online verification method more hardware's were used and the hardware's were directly connected to the computer. The features used for offline verification are simpler. The signatures from the database are preprocessed using various preprocessing techniques then the preprocessed database featureswere extracted.The automatic off-line signature verification solutions can be classified into two categories: handcrafted feature extraction algorithms and deep learning methods. The deep learning methods are especially considered to be the most promising approach for its great capability for image recognition and detection. Although studies of deep learning with small-scale data are getting considerable attention in recent years, most deep learning methods still need a large number of samples to train their system. In other words, most ofthe studies still need several (more than one) signature samples to accomplish their training process. In this paper, we propose an off-line handwritten signature verification method using convolution neural network (CNN). Signature forgery detection finds its application in the field of net banking, passport verification system, credit card transactions and bank checks. Therefore, with the growing demand for protection of individual identity, the design of an automatic signature system is needed.

1.3 SYSTEM REQUIREMENTS

1.3.1 HARDWARE REQUIREMENTS:

- Graphics Processing Unit (GPU).
- Intel Core i3 processor or above

1.3.2 SOFTWARE REQUIREMENTS:

- Python 2.7 or above.
- Jupyter Notebook.
- OS (any one below)
  - Windows 8 (Desktop)
  - Windows 7
  - Windows Vista SP2
  - Windows Server 2012 (64-bit)
  - Windows Server 2008 R2 SP1 (64-bit)

CHAPTER 2:

2.1 PROBLEM STATEMENT

Signature verification is an important biometric technique that aims to detect whether a given signature is genuine or forged. It is essential in preventing falsification of documents in numerous financial, legal, and other commercial settings. This is a comparative analysis of different already known deep learning architectures to check which of those performs the best on the classification. It was solely for offline handwritten signatures. The datasets are not available publicly . But you can mail the publisher and they would provide the download links or else you can you the dataset available on kaggle.The handwritten signature is a behavioral biometric which is not based on any physiology characteristics of the individual signature but on the behavior that change over time. Since an individual's signature alters over time the verification and authentication for the signature may take a long period which includes the errors to be higher in some cases. Inconsistent signature leads to higher false rejection rates for an individual· who did not sign in a consistent way.In this work, the signature images are stored in a file directory structure which the Keras Python library can work with. Then the CNN has been implemented in python using the Keras with the TensorFlow backend as suggested in the research paper to learn the patterns associated with the signature.

2.2 PROPOSED SOLUTION

The handwritten signature is a behavioral biometric which is not based on any physiological characteristics of the individual signature but on the behavior that change over time. Since an individual's signal alters over time the verification and authentication for the signature may take a long period which include the for the errors to be higher in some cases. Inconsistent signature leads to higher false rejection rates for an individual who did not sign in a consistent way.

Data Acquisition: Handwritten signatures are collected and some unique features are extracted to create knowledgebase each and every individual. A standard database of signatures for every individual is needed for evaluating performance of the signature verification system and also for comparing the result obtained using other technical on the same database

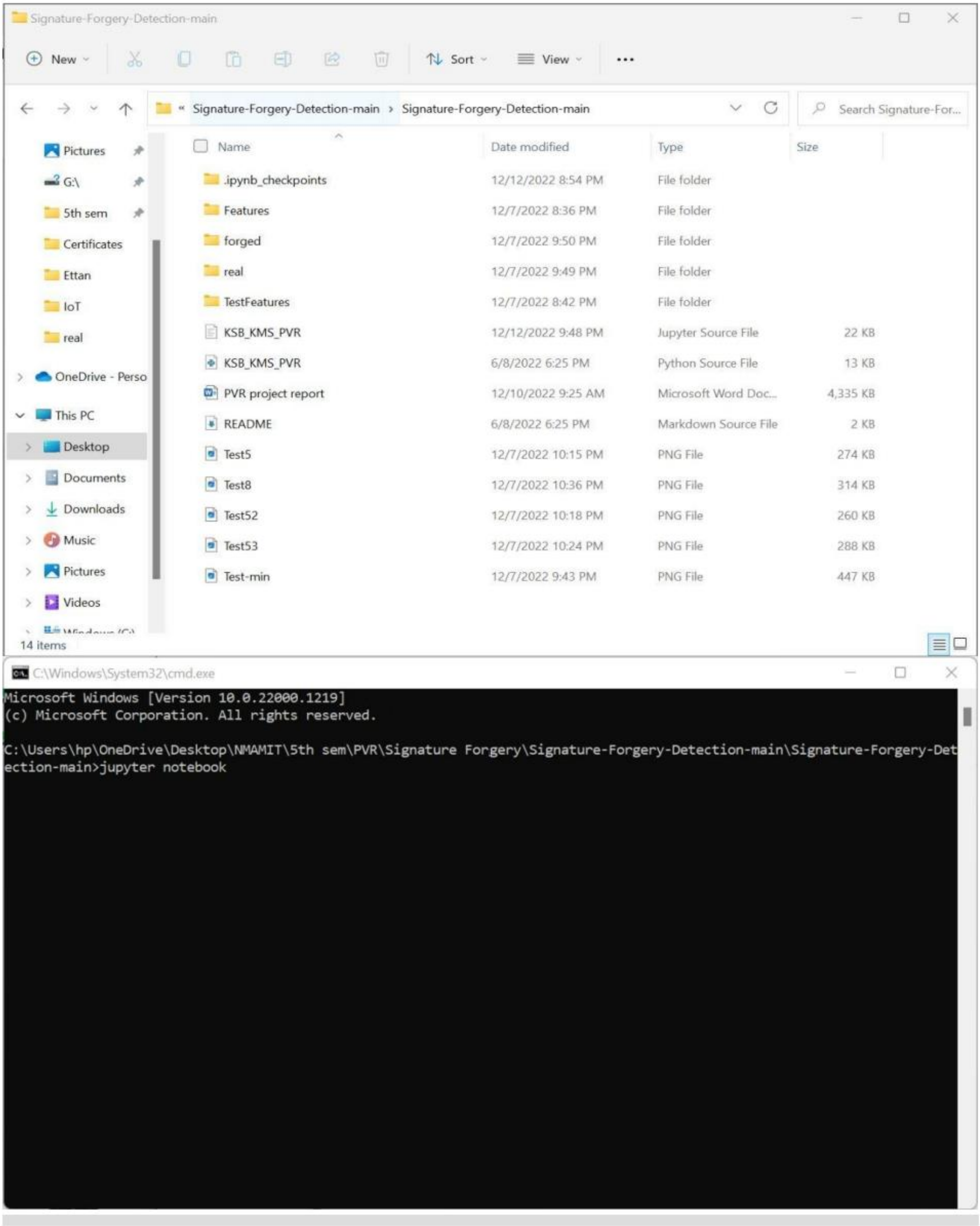
Pre-processing: RGB to Grayscale: In layman's terms, any RGB image is represented as matrix of X. Y dimensions and depth of 3 pla where each plane comprises of Red, Green and Blue values ranging from 0 to 255 Whereas, grays image is represented as matrix of X, Y dimensions and depth of only 1 plane. Each cell value ranges fre to 255 Any RGB (Red, Green, Blue) image which has to undergo Digital Image Processing (DIP) needed be converted to grayscale image. By doing this the computational complexity of DIP decreases drastic and helps to run image processing algorithms in much smoother way.

Noise Removal: Noise is the result of errors which is caused due to various types of acquisition process whic results in pixel values that do not reflect to the true intensities. The image which is taken from a camera contains the film grains in the source of noise, and also may be caused due to the damage when it introduced in the scanner itself, and also electronic transmission of image can also introduce noise. Any form of noise present in the image is unknown to the analyst and its quantity is undefined. In order improvise the noise removal process, a known form of noise is introduced in a very small quantity to the grayscale image. This ensures that the threshold of noise level in the image increases and henceforth easily detectable by de-noising filters. To remove the noise in the scanned image averaging and mean filters a used.

Grayscale to Bitmap: The Grayscale image format is converted into bitmap where image file format is used to store digital images. Raster image is in general is also referred to as a bitmap.

Resizing: The system must be able to maintain the high performance regardless of the size and slant given for the signature. It should be important that the system must be insensitive enough for the correction in the signature image. The image matrix is rescaled to standard resolution which is 256 X 256 in this case.

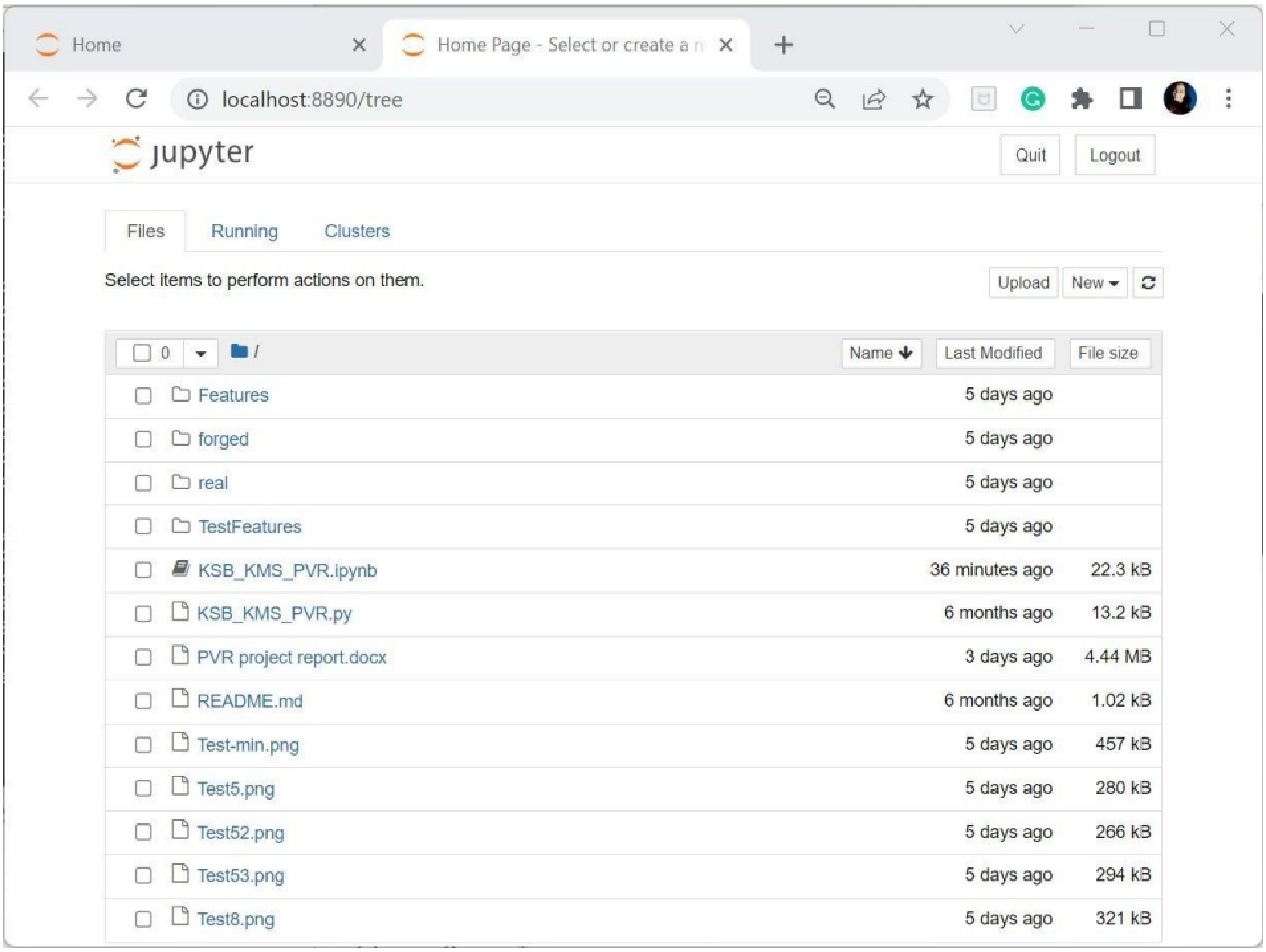
2.3 SOLUTION COMPOSITION



Name	Date modified	Type	Size
.ipynb_checkpoints	12/12/2022 8:54 PM	File folder	
Features	12/7/2022 8:36 PM	File folder	
forged	12/7/2022 9:50 PM	File folder	
real	12/7/2022 9:49 PM	File folder	
TestFeatures	12/7/2022 8:42 PM	File folder	
KSB_KMS_PVR	12/12/2022 9:48 PM	Jupyter Source File	22 KB
KSB_KMS_PVR	6/8/2022 6:25 PM	Python Source File	13 KB
PVR project report	12/10/2022 9:25 AM	Microsoft Word Doc...	4,335 KB
README	6/8/2022 6:25 PM	Markdown Source File	2 KB
Test5	12/7/2022 10:15 PM	PNG File	274 KB
Test8	12/7/2022 10:36 PM	PNG File	314 KB
Test52	12/7/2022 10:18 PM	PNG File	260 KB
Test53	12/7/2022 10:24 PM	PNG File	288 KB
Test-min	12/7/2022 9:43 PM	PNG File	447 KB

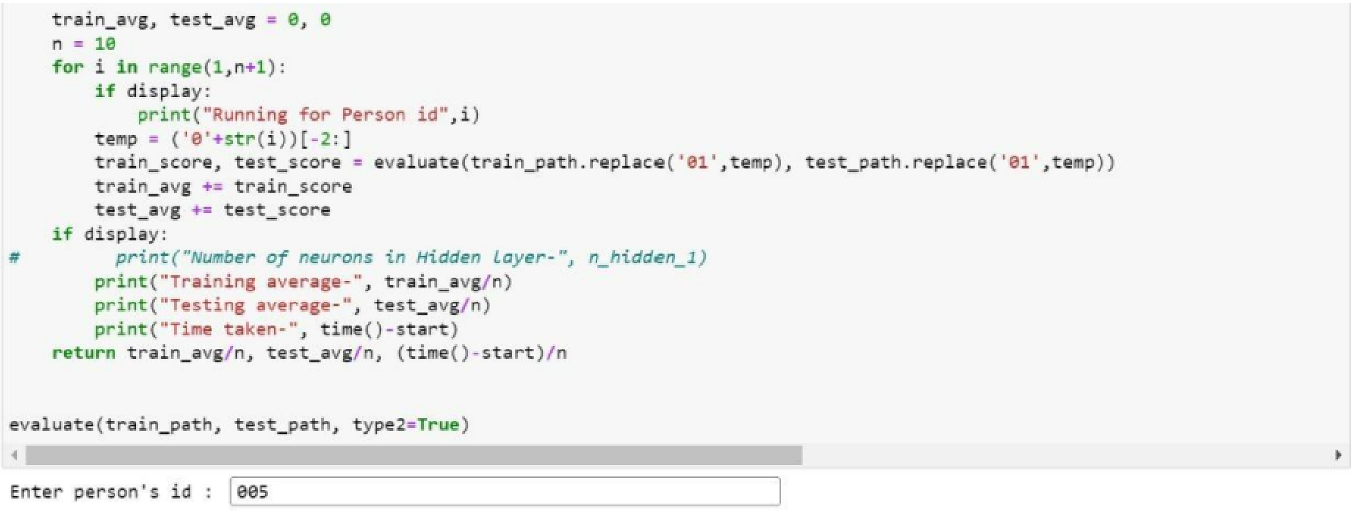
```
Microsoft Windows [Version 10.0.22000.1219]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp\OneDrive\Desktop\NMAMIT\5th sem\PVR\Signature Forgery\Signature-Forgery-Detection-main\Signature-Forgery-Det
ection-main>jupyter notebook
```



Select items to perform actions on them.

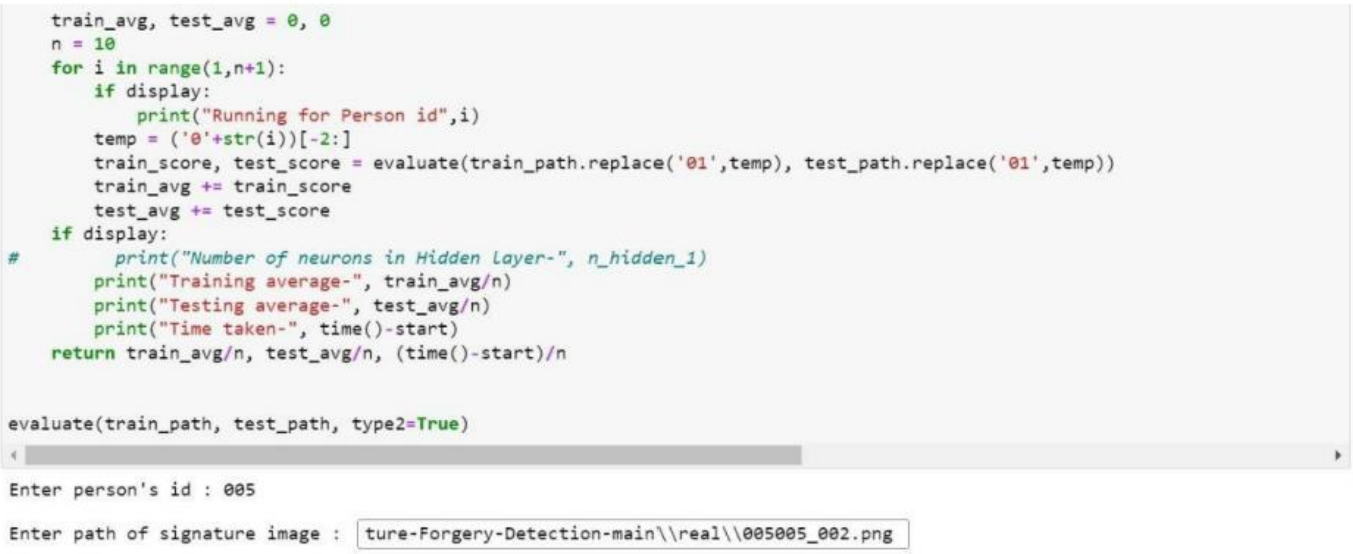
Name	Last Modified	File size
□ 0 /		
□ Features	5 days ago	
□ forged	5 days ago	
□ real	5 days ago	
□ TestFeatures	5 days ago	
□ KSB_KMS_PVR.ipynb	36 minutes ago	22.3 kB
□ KSB_KMS_PVR.py	6 months ago	13.2 kB
□ PVR project report.docx	3 days ago	4.44 MB
□ README.md	6 months ago	1.02 kB
□ Test-min.png	5 days ago	457 kB
□ Test5.png	5 days ago	280 kB
□ Test52.png	5 days ago	266 kB
□ Test53.png	5 days ago	294 kB
□ Test8.png	5 days ago	321 kB



```
    train_avg, test_avg = 0, 0
    n = 10
    for i in range(1,n+1):
        if display:
            print("Running for Person id",i)
        temp = ('0'+str(i))[-2:]
        train_score, test_score = evaluate(train_path.replace('01',temp), test_path.replace('01',temp))
        train_avg += train_score
        test_avg += test_score
    if display:
#           print("Number of neurons in Hidden Layer-", n_hidden_1)
        print("Training average-", train_avg/n)
        print("Testing average-", test_avg/n)
        print("Time taken-", time()-start)
    return train_avg/n, test_avg/n, (time()-start)/n


evaluate(train_path, test_path, type2=True)
```

```
Enter person's id :  005
```

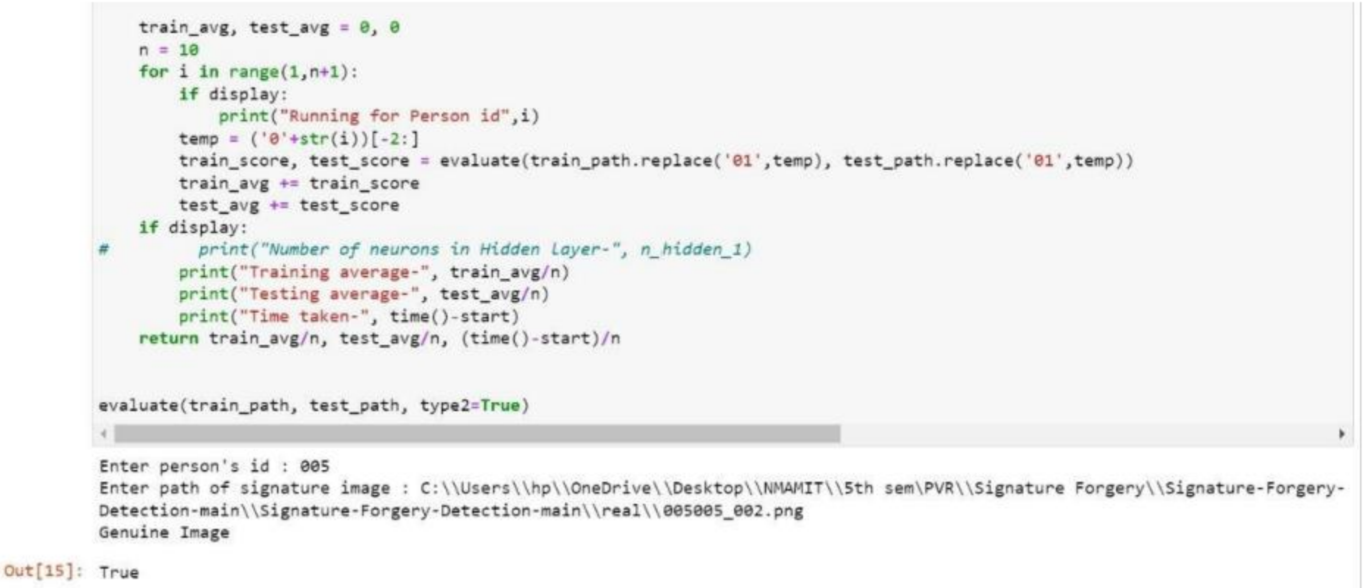


```
    train_avg, test_avg = 0, 0
    n = 10
    for i in range(1,n+1):
        if display:
            print("Running for Person id",i)
        temp = ('0'+str(i))[-2:]
        train_score, test_score = evaluate(train_path.replace('01',temp), test_path.replace('01',temp))
        train_avg += train_score
        test_avg += test_score
    if display:
#         print("Number of neurons in Hidden Layer-", n_hidden_1)
        print("Training average-", train_avg/n)
        print("Testing average-", test_avg/n)
        print("Time taken-", time()-start)
    return train_avg/n, test_avg/n, (time()-start)/n


evaluate(train_path, test_path, type2=True)
```

```
Enter person's id : 005

Enter path of signature image :  ture-Forgery-Detection-main\\real\\005005_002.png
```



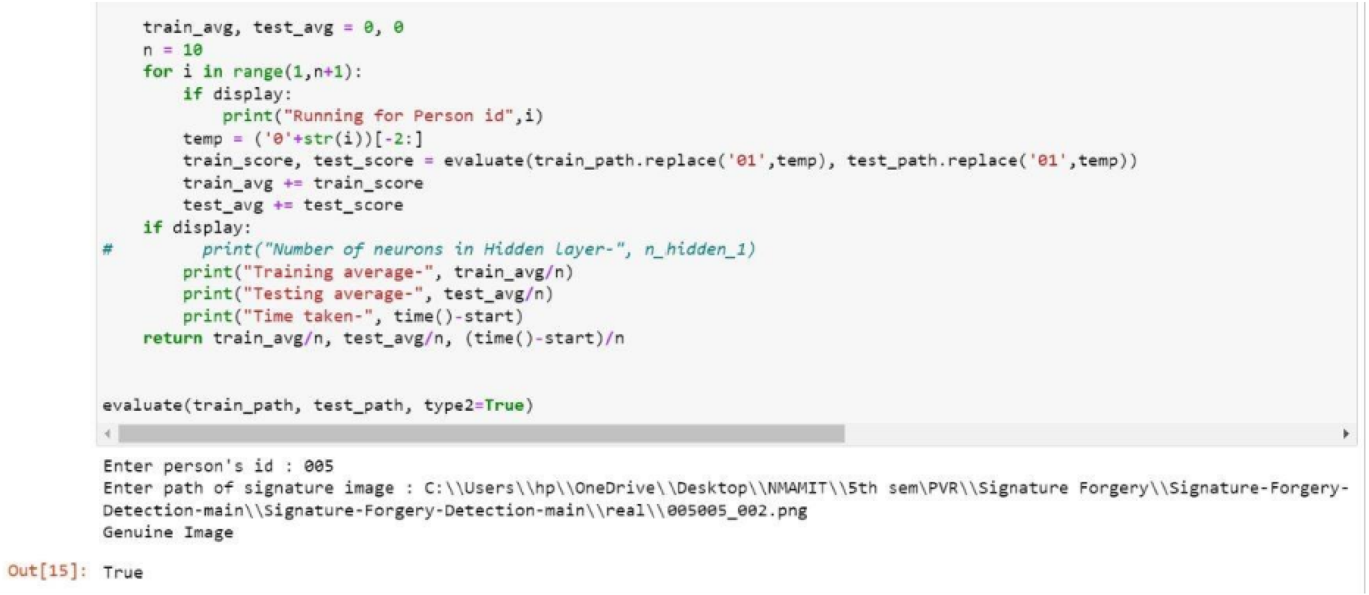
```
    train_avg, test_avg = 0, 0
    n = 10
    for i in range(1,n+1):
        if display:
            print("Running for Person id",i)
        temp = ('0'+str(i))[-2:]
        train_score, test_score = evaluate(train_path.replace('01',temp), test_path.replace('01',temp))
        train_avg += train_score
        test_avg += test_score
    if display:
#         print("Number of neurons in Hidden Layer-", n_hidden_1)
        print("Training average-", train_avg/n)
        print("Testing average-", test_avg/n)
        print("Time taken-", time()-start)
    return train_avg/n, test_avg/n, (time()-start)/n


evaluate(train_path, test_path, type2=True)
```

```
Enter person's id : 005
Enter path of signature image : C:\\Users\\hp\\OneDrive\\Desktop\\NMAMIT\\5th sem\PVR\\Signature Forgery\\Signature-Forgery-
Detection-main\\Signature-Forgery-Detection-main\\real\\005005_002.png
Genuine Image

Out[15]: True
```

2.4 OUTPUT



```
    train_avg, test_avg = 0, 0
    n = 10
    for i in range(1,n+1):
        if display:
            print("Running for Person id",i)
        temp = ('0'+str(i))[-2:]
        train_score, test_score = evaluate(train_path.replace('01',temp), test_path.replace('01',temp))
        train_avg += train_score
        test_avg += test_score
    if display:
#         print("Number of neurons in Hidden Layer-", n_hidden_1)
        print("Training average-", train_avg/n)
        print("Testing average-", test_avg/n)
        print("Time taken-", time()-start)
    return train_avg/n, test_avg/n, (time()-start)/n


evaluate(train_path, test_path, type2=True)
```

```
Enter person's id : 005
Enter path of signature image : C:\\Users\\hp\\OneDrive\\Desktop\\NMAMIT\\5th sem\PVR\\Signature Forgery\\Signature-Forgery-
Detection-main\\Signature-Forgery-Detection-main\\real\\005005_002.png
Genuine Image
```

```
Out[15]: True
```

CHAPTER 3:

3.1 CONCLUSION

A system that can learn from signatures and make predictions as to whether the signature in question is a forgery or not, has been successfully implemented. This system can be deployed at various government offices where handwritten signatures are used as a means of approval or authentication. While this method uses CNNs to learn the signatures, the structure of our fully connected layer is not optimal. According to [7], this implementation may be considered extreme. In the model created in this work, two classes are created for each user (genuine and forgery). If the genuine and forgery signatures of 100 people are given, then the model would have 200 classes to predict, which would make the learning process longer. One future enhancement would be to do comprehensive research on loss functions and derive a custom loss functions (preferably two) which would predict the user to which the signature belongs to, and whether it is a forgery or not.

3.2 APPLICATION AND FUTURE WORKS

Handwritten signatures are very important in our social and legal life for verification and authentication. A signature can be accepted only if it is from the intended person. The probability oftwo signatures made by the sameperson being the same is very less. Many properties of the signature may vary even when two signatures are made by the same person. So, detecting a forgery becomes a challenging task. Efficient methods of user verification are necessary in growing digitalization of various aspects of everyday life as well as new issues in offices and agencies. Parallel to new technology that is giving new possibilities a need for new and improved methods and algorithms is visible. The proposed method can be used as an effective signature verification system.Theproposed method successfully made the offline signature verification with improvement in the efficiency and accuracy and easily detected the skilled forgeries. We have used python and its libraries, in conjunction with a solution based on Convolutional Neural Network (CNN) successfully for signature forgery detection. Future works include improving the model by lowering the Fault Rejection rate. Another promising work can be to combine offline and online signature verification systems which will make the system more robust as it will take both speed of execution and genuine visual signature into consideration so it will become harder to forge signatures. This can be developed in to apps or web page or can be used in security systems in public places such as ATMs, official government institutions, colleges, legal institutions, etc.

3.3 BIBLIOGRAPHY

- https://www.kaggle.com/datasets
- https://github.com/
- https://www.researchgate.net/publication/277412469_Forged_Signature_Detection_Using_Artificial_Neural_Network
- https://www.sciencedirect.com/science/article/pii/S1877050918320301