ANALYSIS OF HANDWRITING AS A BIOMETRIC PARAMETER

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

In

Computer Science and Engineering/Information Technology

By

Devvrat Bhardwaj (161219)

Mridul Prashar (161232)

Under the supervision of

Mr. Prateek Thakral

Assistant Professor (Grade II)

Computer Science & Engineering and Information Technology

To

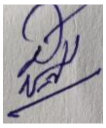
Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

CANDIDATE'S DECLARATION

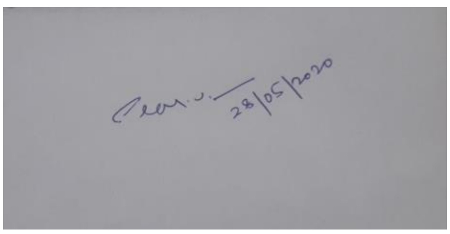
We hereby declare that the work presented in this report entitled **"ANALYSIS OF HANDWRITING AS A BIOMETRIC PARAMETER"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of our own work carried out over a period from August 2019 to May 2020 under the supervision of **Mr. Prateek Thakral** (Assistant Professor(Grade II), Computer Science and Engineering And Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Devvrat Bhardwaj, 161219 

Mridul Prashar, 161232 

This is to certify that the above statement made by the candidate is true to the best of my knowledge



Mr. Prateek Thakral

Assistant Professor (Grade II)

Computer Science Engineering and Information Technology

Dated: 28th May 2020

ACKNOWLEGEMENT

We would like to express our sincere gratitude to our supervisor Mr. Prateek Thakral for invaluable consultations, interest, initiative, and continuous support throughout the duration of project on the topic **“ANALYSIS OF HANDWRITING AS A BIOMETRIC PARAMETER”**. We are really thankful to him.

Secondly, we would like to thank lab assistants who helped us a lot in finalizing this project within the limited time frame. Also, we would like to thank all the people who took out time to provide us their text samples.

Devvrat Bhardwaj, 161219

Mridul Prashar, 161232

TABLE OF CONTENTS

LIST OF ABBREVATIONS

IDE: Integrated Development Environment

IDLE: Integrated Development and Learning Environment

GUI: Graphics User Interface

OS: Operating System

BSD: Berkeley Source Distribution

PIL: Python Imaging Library

LIST OF FIGURES

LIST OF GRAPHS

LIST OF TABLES

ABSTRACT

The rising requirement for robotization of frameworks has affected the advancement of content identification and acknowledgment from pictures to an enormous degree. Handwriting biometrics (which comes under the field of behavioral biometrics), focuses on identifying the writer based on features and characteristics in the handwriting sample. In this project, we describe a novel approach to improve the accuracy of writer detection of a given text. Most of the methods for author identification require image analysis of the text (i.e., are content dependent). Our approach is also content dependent, but it focuses on comparing the pressure versus time plots obtained rather than comparing the text images. The identification is regarded as finding similarity in the curve of the plot. We apply the techniques of Image Hashing and Dynamic Time Warping to fulfill the identification task. We conducted experiments using English handwriting samples from 20 people and achieved promising results.

CHAPTER – 1

INTRODUCTION

1.1 OBJECTIVE

This project focuses mainly on analysis of handwritten samples with main goal of identification of the author of a handwritten sample. Optical character recognition should not be confused with handwritten biometric recognition. While the goal of OCR is to recognize the context of the text, regardless of the author, the goal of handwritten biometrics is to identify the author of the given text. Also, handwriting recognition focuses on minimizing the discrepancies in writing style, whereas, in the identification of handwritten text, variations in the writing style are focused upon.

1.2 INTRODUCTION

Handwritten Biometric Recognition

Identification of a person based on biometrics is an important research area and finds applications in various day to day fields such as marking attendance for employees, establishing authenticity, ensuring security etc. The study of biometrics can be divided into two major areas:

1. Based on <u>physiological</u> characteristics such as iris of the eye or fingerprints of a person.
2. Based on acquired <u>behaviors</u>, such as voice/speech, handwriting, keystroke patterns etc.

We focus on identification of a person on the basis of his/her handwriting in the project which is a subfield of behavioral biometrics. We focused on handwriting as it is universal and easy/cheap to capture and analyze. Moreover, there are plenty of applications which can benefit from this study such as in identification of forgery and impersonation. Also, long term change in a person’s handwriting can indicate underlying neurological conditions and diseases such as the Parkinson's disease.

The features of a person's signature (time taken to write, coordinates, grip, pressure, stroke, and shape) ensure that handwriting is unique and can be used to identify an individual reliably.

Most of the existing techniques and methods rely on analysis of offline as well as static placement of text. We have focused on online and "dynamic" study of the written text i.e., in this project, we analyzed digitally collected handwritten samples and composed algorithms that help in correct identification of a writer.

1.3 PROBLEM DEFINITION

In the era of technology, where it has various pros but also have cons to it. Cons here not totally define the negative impact of the technology but some loopholes which got created and can be assumed to be threat to the technology. The project we are doing is considered to be one of the solutions to such potential threats in this we can verify the handwriting of the person and on that basis, we can verify whether the document belongs to that specific person or not in order to prevent forgery. This model is very useful for verifying a digital legal document.

1.4 SCOPE

With the onset of electronic communications and automatization of security techniques, identity theft has become a critical threat. Therefore, handwriting as a biometric can be used for identification by which we can prevent forgery. It can be used for forensic document examination which can be used in crime scene investigations. This technique can be used to scan the signatures present on the legal documents to verify the author. Moreover, it can be used to take digital signature of the author which can be useful in banking systems. Also, it can be used to identify underlying neurological disorders such as the Parkinson's disease.

1.5 METHODOLOGIES

Sample Collection:

Dataset for testing handwriting as a biometric comprised of 8 words, 1 sentence and 3 characters collected by 20 different individuals on a graphics tablet. Each writer also had a unique test case (word, character, or sentence). The test case was randomly selected from the collection in the dataset, and each writer was made to write the random selection again. Therefore, the size of the dataset was 20*(12+1(test)). Also, the features recorded against each sample were pressure (which was recorded in non-scaled units ranging from 0 to 1023), time and (x, y) coordinates. The effect of arm movements was excluded by placing the graphics tablet on a solid resting surface. Moreover, x and y coordinates recorded for each stroke nullified the effect of position (rotation and displacement), of the graphics tablet. The proposed algorithm was implemented in a computer with a Core i5 processor (4 GB RAM, 2.4 GHz). Experiments were performed after preprocessing the data in the .txt file. The preprocessing steps used in the project are discussed in detail in chapter 3.

CHAPTER – 2

LITERATURE SURVEY

2.1 LITERATURE REVIEW:

Ameur Bensefia and Hatem Asad Tamimi in paper [1] gave “a generic scheme for a system that can operate in verification or identification mode”. In the verification (training) mode, the system authenticated a user on the basis of respective identity. In the identification (testing) mode, the system recognized the user. Final evaluation of the handwriting was done around the following five properties: <u>Universal, Distinctive, Permanent, Measureable and Imitable</u>. It was noticed that the handwriting was permanent, universal, and quantifiable. However, the distinctiveness of the samples was relative to the size of the datasets It was also noticed that handwriting was hard to imitate hence, could be used as a biometric classifier.

Syed Faraz Ali Zaidi and Shahzaan Mohammed in paper [2], summarized some techniques to improve “biometric handwritten signature verification methods”. This was based on possible addition of some simple parameters which were:

- Total signing time.
- Pen down time.
- Root Mean Square speed.
- Acceleration.

- Length by width ratio.
- Horizontal-span ratio.
- Correlation in x, y speed.
- First moment

Milena Pugnaloni in [3], concluded that "handwriting expertise found its bases on graphic laws that articulate the rules of the analysis of graphic dynamic". Although the pen pressure results vary from study to study, one consistent finding in this research paper is that forging writing (whether text or signatures) does increase the demands on the processing system and this is observed in changes to pen motion speed. Moreover it was emphasized that "the differences in pen pressure between the forgers' genuine signatures and the forgeries they produced in combination with the non-dependence of the natural signing style of the forger on this change, does indicate that pen pressure can be a useful parameter in discriminating between genuine signatures and forgeries. In order to advance the use of pressure differentials in the forensic environment there is a need to develop pressure measurement techniques".

R. Venkatesan, S.-M. Koon, M. H. Jakubowski, and P. Moulin in [4], introduced an image hashing computation that changes over images into short strings. Utilizing this calculation, we can think about two pictures by checking no good strings for careful uniformity, as opposed to endeavoring the issue of looking at picture information prone to editing.

Hashing was utilized instead of watermarking, with the advantage that nothing was added to pictures. From a hypothetical perspective, hashing appeared to be more grounded than watermarking, since hashing calculations were adjustable without the need to change and pre-discharge sent pictures, with watermarking.

Namboodiri, Anoop and Gupta, Sachin in paper [5] proposed a "text independent writer identification system for online documents". This method was advantageous as it only needed small amount of testing data and was text independent. The classification was fast, and it was observed that confidence in the results improved with increase in the size of dataset ("evidence accumulation"). High confidence about the identity of the writer was achieved even with a single line taken into consideration. They used "sub character level features for writer identification". To improve on the accuracy and robustness of the system the authors recommended the use of "other high-level primitives based on character, word, line, and paragraph. Different primitives like shape, size and other higher-level features also could be used in combination to improve the system".

Bashir, Muzaffar and Kempf, Jürgen in [6], found out that "the RDTW technique applied to down sampled BiSP data was well suited to classify between human individuals and handwritten items like PIN words or just a short sequence of isolated characters". The RDTW method complied efficiently and performed well in online recognition system. "Single characters and PIN words, handwritten by the same person were recognized at an extremely high score (better 99%) with a response time of less than 0.5 seconds". Such accurate results lead to a potential application, where "biometric person and PIN code recognition were combined". Moreover, the authors suggested for coping with the computing time problem in character recognition by application of RDTW in a "hierarchical classification scheme". This approach was effective and reduced the computational time without degrading the performance.

2.2 EXISTING SYSTEMS

Signature verification:

Signature is a special instance of handwriting which is perfected by practice over the years and is imbibed in the muscle memory of personnel. The pattern(s) for each user are analyzed and recorded. Signature features such as velocity of the pen, and pressure put on the writing surface, grip pressure etc. result in unique formation of signatures for every individual.

CHAPTER – 3

SYSTEM DEVELOPMENT

3.1 TOOLS AND TECHNIQUES USED:

HARDWARE:

Wacom Intuous Art tablet and stylus with 1024 levels of pressure.

SOFTWARES:

Eye and Pen 3.

The Eye and Pen programming records the developments of the pen during times of composing and stops.

On the off chance that owning an eye tracking gadget, eye and Pen can likewise synchronously record the eye developments. For analysts in instructive sciences, etymology, or psychological brain science, the capacity to follow high accuracy the handling of visual data during composing will contribute tremendously to their comprehension of compositional systems and the working of composed language.

Examinations in the working environment and in proficient composing ought to furnish specialists with understanding into the obtaining and advancement of skill. Studies directed with youngsters should give fascinating pictures of spelling handling and troubles. Ergonomics and instructive applications are various.

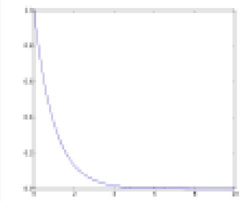
Python IDLE:

IDLE is the abbreviation for Integrated Development and Learning Environment.

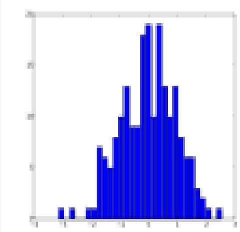
IDLE is simple and fast IDE and was used to implement all the experiments in this project. The following libraries were used for data/image processing:

Matplotlib library:

Matplotlib is used to plot two dimensional images (graphs, histograms etc.) in Python. Here are some examples of matplotlib library function:



Line plot



Histogram

Fig 3.1. Sample figures obtained using matplotlib library.

NumPy Library:

NumPy is a numerical computing library in Python which allows us to operate on multidimensional matrices and arrays and apply advanced mathematical functions to manipulate the data variables. It is highly efficient and is open source.

PIL (Python Image Library):

Python Imaging Library is used for processing and manipulating images using the Python language.

Image Hashing:

Hash value extracted from a picture by a hash function. Each picture has a unique hash value and is easier to process/compare as compared to a picture, in terms of both time and memory. Moreover, the similarity of images can be determined on the closeness of their hash values. We studied four types of image hashing in this project (perceptual hashing, average hashing, difference hashing and wavelet hashing). Out of these 4, the best results were obtained using difference and average hashing, and only these techniques were used for further analysis and comparison.

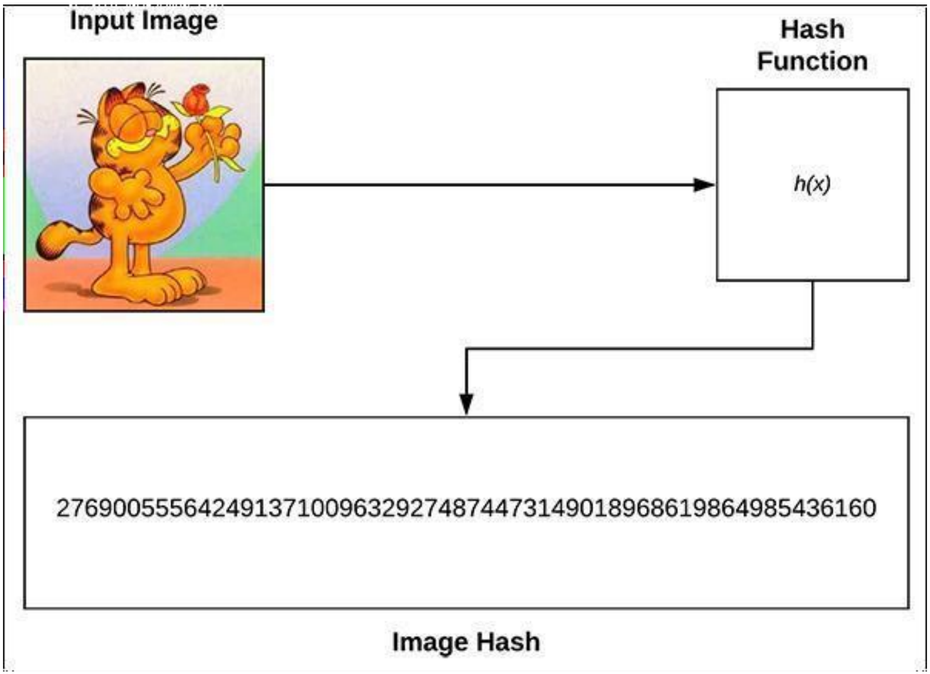


Fig 3.2. Sample Image Hashing

Scikit-Image Library:

Scikit-picture is a picture handling Python bundle that works with numpy clusters which is an assortment of calculations for picture preparing.

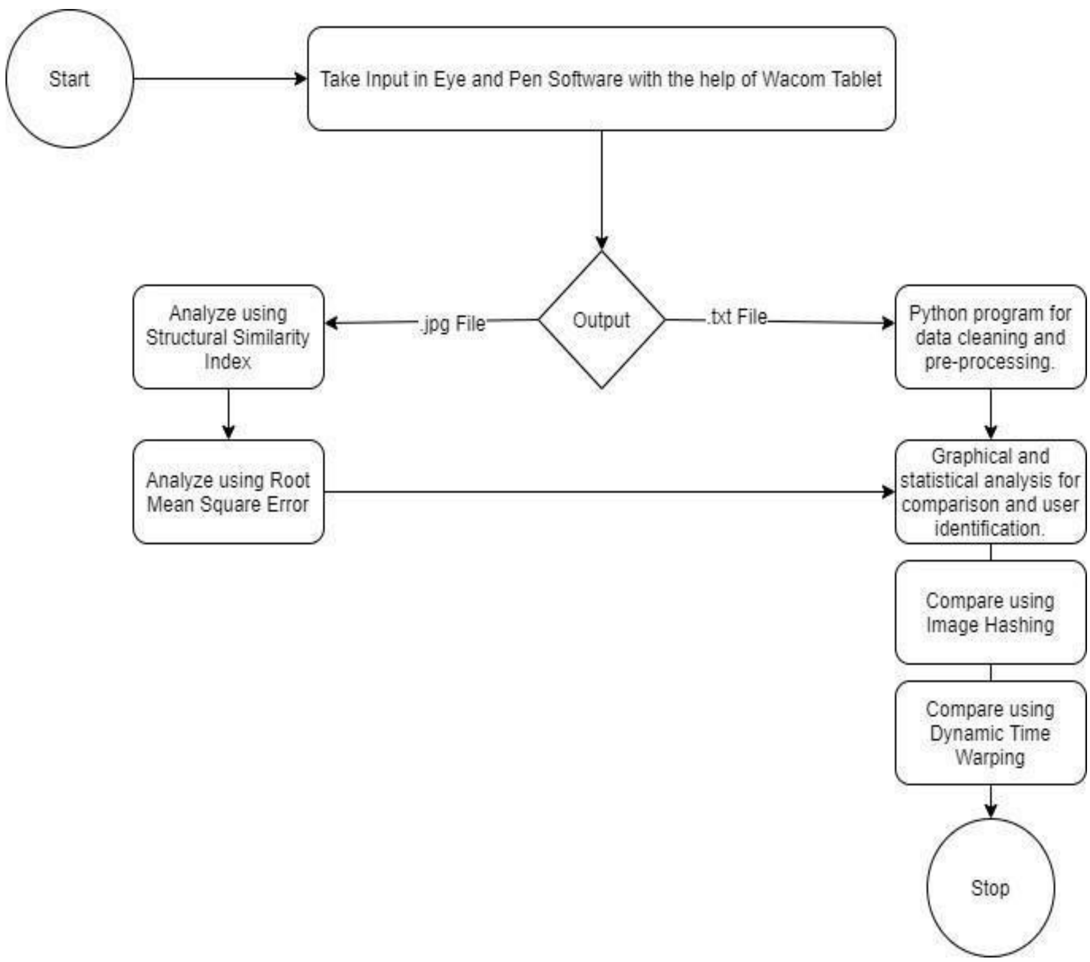
Fig 3.3. Sample Scikit Image

Pearson Coefficient:

Pearson's coefficient is used to quantify relationship between two factors and was used in this study to measure similarity in text samples.

3.2 FLOW CHART:

The flow chart here depicts the overall flow of the handwriting sample collection and analysis, i.e., the basic activities in a procedural format.



Graph 3.1. Flow diagram depicting procedures to be followed

3.3 EXTRACTION OF DATA FROM HANDWRITING SAMPLES:

The data from .tab file received by a writing sample in the software Eye and Pen 3 was extracted in the form of a text file, which included pressure, x coordinate, y coordinate, time (milliseconds). Moreover, an image in .png format was also extracted so as to use it in the dataset for recognition of text. It was graphically represented, and image samples were saved for further analysis. Also, different parameters were extracted at this level such as horizontal length, vertical length, maximum pressure, frequency of maximum pressure.

3.4 COMPUTATIONAL DEVELOPMENT

PREPROCESSING:

The data in text file was read in python and cleaned i.e., removing missing and null values, and also removing data collected before the first contact. Also, the string values were converted into integers so that graphs could be plotted, and mathematical operations could be performed.

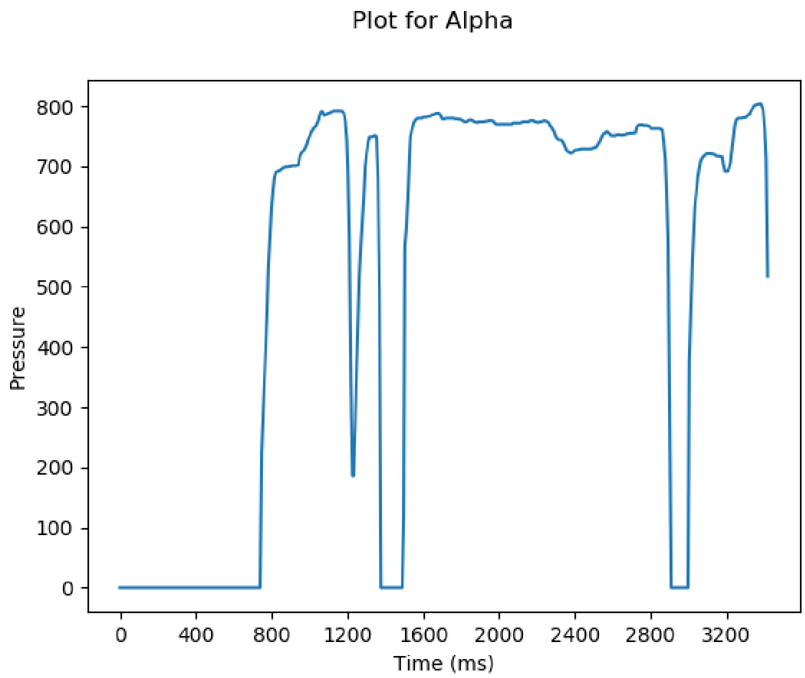


Fig 3.4. Pressure v/s Time Graph before preprocessing

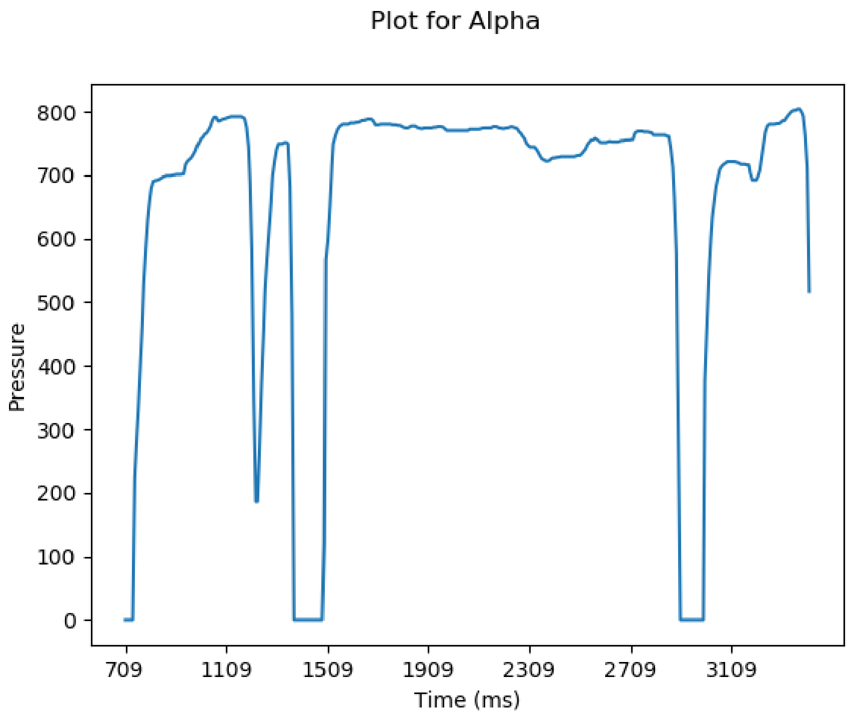


Fig 3.5. Pressure v/s Time Graph after preprocessing

Image Hash Difference

There are hash collisions if images are similar in case of image hashing. The analysis was done by plotting pressure v/s time graph of the data obtained from acquisition of different words and sentences in the samples collected. The test case word was compared with every other reference graph of the same word by applying image hashing (average hashing). Values of the axes were hidden as we were only concerned with comparing the plot. The hash differences were recorded against respective writers. This was done with every test case and respective words/characters/sentence.

```
 RESTART: C:/Users/bhard/AppData/Local/Programs/Python/Python37-32/Single case avg hash.py
Average Hash Difference for 'Original Writer's Text - Original Writer's Text':    0
>>>
```

The hamming distance is zero when the image hash of a picture is compared to its image hash. Therefore, the least hamming distance indicates maximum similarity between the images. Similarity can also be seen by visually analysing the blended graphs.

Case 1: Random v/s Test

Average Hash Difference is 20 between two plots.

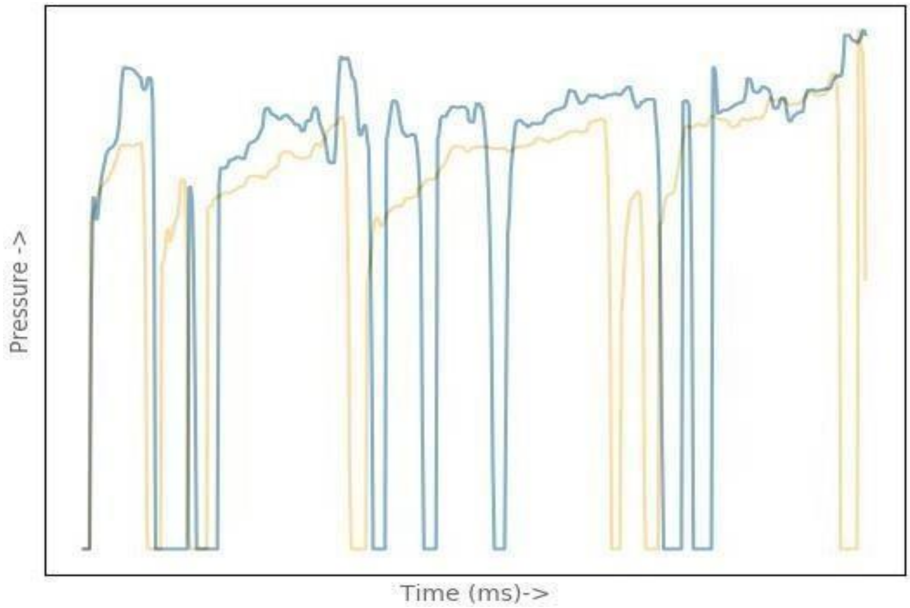


Fig 3.6. Blended graph where blue line is the random writer and orange line is the test case pressure v/s time plot.

Case 2: Original v/s Test

Average Hash Difference is 20 between two plots.

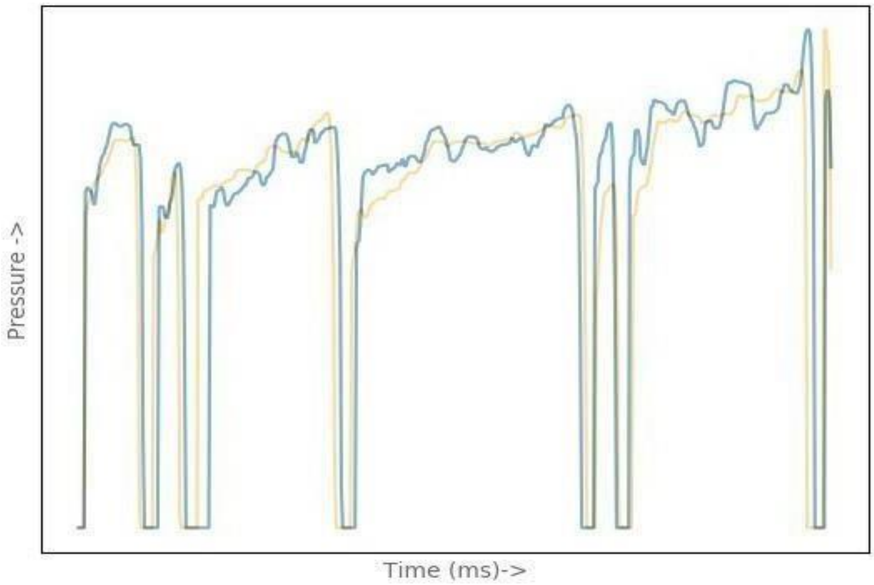


Fig 3.7. Blended graph where blue line is the original writer and orange line is the test case pressure v/s time plot.

Dynamic Time Warping

As different samples may differ in speed, dynamic time warping was used to measure the similarity between the two sequences (Pressure/Time plots for test sample and writer's sample). The minimum distances between the sequences were obtained by DTW and were stored against the writer as a true measure of similarity between them (writer's sample and test sample).

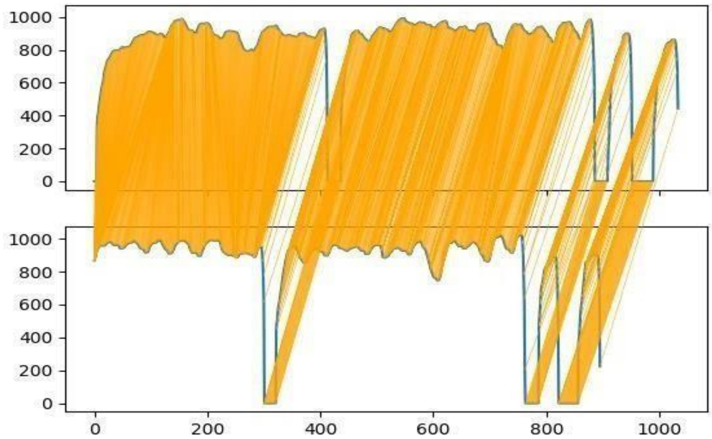


Fig 3.8. DTW path between writer's sample (top) and test sample (bottom) for writer 17 and testcase 17 with DTW distance 2593.

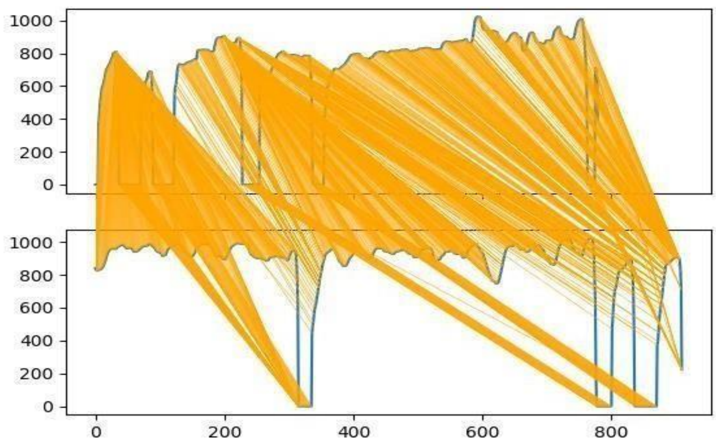


Fig 3.9. DTW path between writer's sample (top) and test sample (bottom) for writer 20 and test case 17 with DTW distance 4553.

3.5 STATISTICAL DEVELOPMENT

The data and graphs were tested against each other with different statistical measures such as root mean square error and structural similarity index to identify the writer. For testing the graphs against such tests, axes and title were cleared so that they don't give garbage values and interfere with our results.

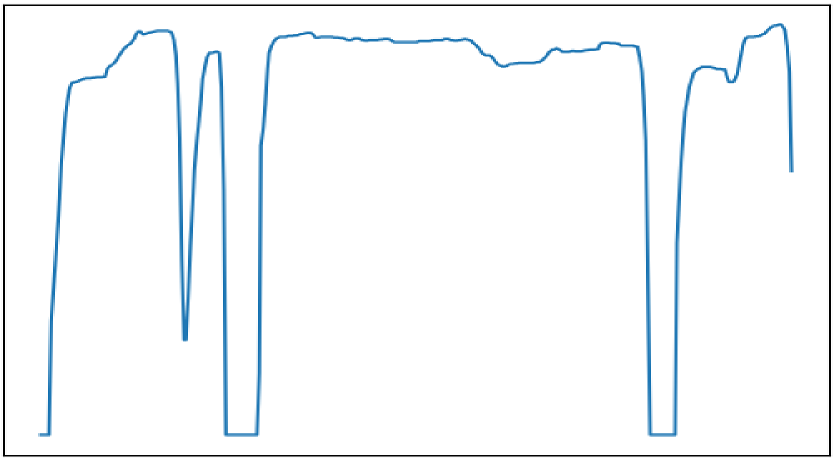


Fig 3.10. Pressure v/s Time Graph after preprocessing and clearing the axes

```
Maximum pressure:  804
Frequency of max pressure:  2
Horizontal Distance:  2502
Vertical Distance:  2121
>>> 
```

Fig 3.11. Parameters calculated for statistical comparison and tests

```
'''
= RESTART: C:\Users\bhard\AppData\Local\Programs\Python\Python37-32\ssim.py =
SSIM: 0.9926309772351769
>>>
= RESTART: C:\Users\bhard\AppData\Local\Programs\Python\Python37-32\rmse.py =
Root Mean Square Error 958.1134084077868
>>>
```

Fig 3.12. Structural Similarity Index and Root Mean Square Error after comparing two pressure-time graphs.

```
= RESTART: C:\Users\bhard\AppData\Local\Programs\Python\Python37-32\stats.py =
LinregressResult(slope=0.07366882528995997, intercept=301.58924580477276, rvalue=0.38096421947783193,
pvalue=1.5215635070519968e-18, stderr=0.00805239714220152)
>>>
```

Fig 3.13. Slope, intercept, revalue, pvalue and standard error of a user sample as tested against test sample.

3.6 Algorithms:

(1.) Algorithm for Data Acquisition:

1. Connect Wacom Graphics Tablet.
2. Open Eye and Pen 3 software.
3. Go to simple acquisition, name your file and write the sample on the tablet.
4. Press Esc key to save the sample.
5. The sample will be saved in .tab format.

(2.) Algorithm for Data Extraction:

1. Open Eye and Pen 3 software.
2. Open the respective .tab file whose data is to be extracted.
3. Export the file as .txt file. (The exported .txt file will contain pressure, x-coordinate, y- coordinate and time in milliseconds)

(3.) Algorithm for processing extracted data in python:

1. Import matplotlib library.
2. Open the .txt file using open function and giving read only preference. F $\leftarrow$ Text File.
3. Convert each line in text file as a comma separated value and store in a list. F’$\leftarrow$CSV
4. Split the comma separated file into list of lists. F’’$\leftarrow$List of Lists

5. Close the text file.
6. Remove the first list from list of lists, i.e., the header.
7. Make four lists for each entry in a list from F’’ i.e., first (time), second(x), third(y) and last (pressure) and append them respectively.
8. T←Time X←x-coord. Y←y-coord. P←pressure
9. Find max pressure by using max function on list P. Pmax← max(P)
10. Initialize a counter.
11. Counter←0
12. Go through the list P and increase Counter when Pmax is observed. for Pmax in P:
13. Counter++
14. Calculate maximum and minimum horizontal distance. Hmax←max(X)
15. Hmin←min(X)
16. Max horizontal length is difference between Hmax and Hmin Xlen←Hmax - Hmin
17. Calculate maximum and minimum vertical distance. Vmax ←max(Y)
18. Vmin←min(Y)
19. Max vertical length is difference between Vmax and Vmin Ylen← Vmax – Vmin
20. Print and record these observations.
21. Plot the graph using P and T via matplotlib.
22. Save the graph (image file) for further analysis.

(4.) Algorithm for calculating SSIM:

1. Import OpenCv, Python Imaging Library.
2. Import metrics module from skimage libraries.
3. Load the images to be compared in OpenCv.
4. Convert the images to grayscale.
5. Apply structural similarity function of the metrics module.
6. Display the SSIM value.

(5.) Algorithm for calculating RMSE:

1. Import Image and ImageChops modules from the Python Imaging Library.
2. Import math and operator libraries.
3. Load the images via Image.
4. Calculate difference using ImageChops. Plot histogram using this difference.
5. For each entity in the histogram, square and add to the sum of squares.
6. Divide by total entities and take square root.
7. Display RMSE value.

(6.) Algorithm for calculating Average Image Hash Difference:

1. Import imagehash module and form PIL import image library.
2. For every writer, open the pressure v/s time plot and apply average hash function of the imagehash library.
3. Open the test plot and calculate its avg. hash value.
4. Find the absolute difference b/w original avg. hash value and test avg. hash value and print it.

(7.) Algorithm for Dynamic Time Warping

1. Import matplotlib, numpy, pandas and from dtaidistance library, import the dtw and dtw_visualisation module.
2. Open the text file (that contains input data) of the word written by the required writer that is to be compared against the test. Open the text file of test also.
3. Extract pressure feature from both of these files and store it in a list.
4. Use the distance function from the dtw module that takes the parameters as two pressure lists.
5. Print the dtw distance rounded to two decimal digits.
6. Calculate the warping path using dtw.warping_path() function.
7. Plot the warping path using dtw_visualisation.plot_warping() function, that takes the parameters as pressure lists and warping path and save the image of warping path.

CHAPTER-4

PERFORMANCE ANALYSIS

4.1 Statistical Comparison on distance parameters and features.

Author	Max. Horizontal Distance	Max. Vertical Distance	Maximum Pressure	Frequency Of Maximum Pressure	Pressure Peaks
Devvrat	4118	2395	727	2	11
Mridul	7719	3566	749	2	15
Anubha	12190	4798	1023	12	17
Pratyush	20660	9884	999	2	13
Test	4191	2468	746	2	11

Table 4.1 Parameters obtained by statistical analysis for the word sample "serially".

Hence, on comparison of similar parameters obtained by different authors, it can be concluded that the test sample "serially" is written by Devvrat, which is true. See Appendix for graphs.

4.2 Analysis using RMSE and SSIM values:

```
>>>
= RESTART: C:\Users\bhard\AppData\Local\Programs\Python\Python37-32\rmse.py =
Root Mean Square Error between original image and test 957.864291014129
>>>
= RESTART: C:\Users\bhard\AppData\Local\Programs\Python\Python37-32\rmse.py =
Root Mean Square Error between different image and test 958.1134084077868
>>>
```

Fig 4.1 Root Mean Square Error should be relatively smaller

```
= RESTART: C:\Users\bhard\AppData\Local\Programs\Python\Python37-32\ssim.py =
Between original image and original image SSIM: 1.0
>>>
= RESTART: C:\Users\bhard\AppData\Local\Programs\Python\Python37-32\ssim.py =
Between original image and test image SSIM: 0.992624174894585
>>>
>>>
= RESTART: C:\Users\bhard\AppData\Local\Programs\Python\Python37-32\ssim.py =
Between different image and test image SSIM: 0.9918298467036416
>>>
```

Fig. 4.2 SSIM should be as near to 1 for similarity.

It is to be noted that both SSIM and RMSE gave inaccurate results from which no conclusion could be devised. Hence, both these approaches were discarded in image comparison.

4.3 Analysis using Slope, Intercept, and Pearson Correlation Coefficient:

```
= RESTART: C:\Users\bhard\AppData\Local\Programs\Python\Python37-32\stats.py =
LinregressResult(slope=0.0373207512524963 4, intercept=434.41843953250407, rvalue=0.215608738149501 8,
pvalue=2.019411220451331 3e-07, stderr=0.00709207193038152 4)
>>>
```

Fig. 4.3 Statistical Results when Original writer's text is compared against test text.

```
...
= RESTART: C:\Users\bhard\AppData\Local\Programs\Python\Python37-32\stats.py =
LinregressResult(slope=-0.03486667822588006, intercept=989.8252242282389, rvalue=-0.26842334288745967
, pvalue=3.0658080528719307e-17, stderr=0.004051148270743267)
>>>
```

Fig. 4.4 Statistical Results when random (other than original) writer's text is compared against test text.

After calculating and recording these results, we compared them. However, an inference could not be reached with these parameters as there was not any distinguishable pattern. Hence, this approach was discarded.

4.4 Image Hashing:

Two types of image hashes were calculated and recorded:

- **4.4.1 Average Image Hash**

After scaling the image to 8x8 pixels, all values of image are averaged. Then, each pixel is examined in left to right fashion. If the value is greater than average, 1 is added, if less than average than 0 is added to the hash.

```
Average Hash Difference for 'Test case 1':
1 - Test : 10
2 - Test : 18
3 - Test : 18
4 - Test : 17
5 - Test : 20
6 - Test : 18
7 - Test : 19
8 - Test : 16
9 - Test : 13
10 - Test : 15
11 - Test : 17
12 - Test : 13
13 - Test : 13
14 - Test : 18
15 - Test : 17
16 - Test : 13
17 - Test : 21
18 - Test : 15
19 - Test : 14
20 - Test : 17
```

Fig.4.5. Writer 1 has the least hamming distance in this case followed by writer 9.

- **4.4.2 Difference Image Hash**

Difference hash uses gradients (difference between adjacent pixels) instead of the average value. The rest of the approach is similar as average hash.

```
Difference Hash difference for 'Test Case 1':
1 - Test : 6
2 - Test : 17
3 - Test : 20
4 - Test : 13
5 - Test : 9
6 - Test : 16
7 - Test : 20
8 - Test : 18
9 - Test : 8
10 - Test : 15
11 - Test : 16
12 - Test : 11
13 - Test : 19
14 - Test : 17
15 - Test : 14
16 - Test : 16
17 - Test : 21
18 - Test : 14
19 - Test : 12
20 - Test : 6
```

Fig.4.6. Difference Hash Difference for Test Case 1

The average hash value of test writer was then subtracted from each writer's average hash value for that given text sample. The writer's having the least hamming distance and second to least hamming distance were recorded in results for further analysis:

Test Case	Writer with least image hash difference.	Writer with second to least image hash difference.
1	1	9
2	2	16
3	6	19
4	11	4
5	5	13
6	6	18
7	7	8
8	8	12
9	9	8
10	10	4
11	11	14
12	12	6
13	13	17
14	14	11
15	15	3
16	16	11
17	17	1
18	2	14
19	18	19
20	20	17

Table No. 4.2 Difference amid least hamming distance and second to least hamming distance.

The writer having the least hamming distance was the original writer. This approach alone gave an accuracy of 80%. The entries coloured red in the table indicate the writers that have been incorrectly identified i.e., 20% failure.

4.5 Dynamic Time Warping:

Application of DTW for test entry 19 i.e., against writers 18 and 19 respectively:

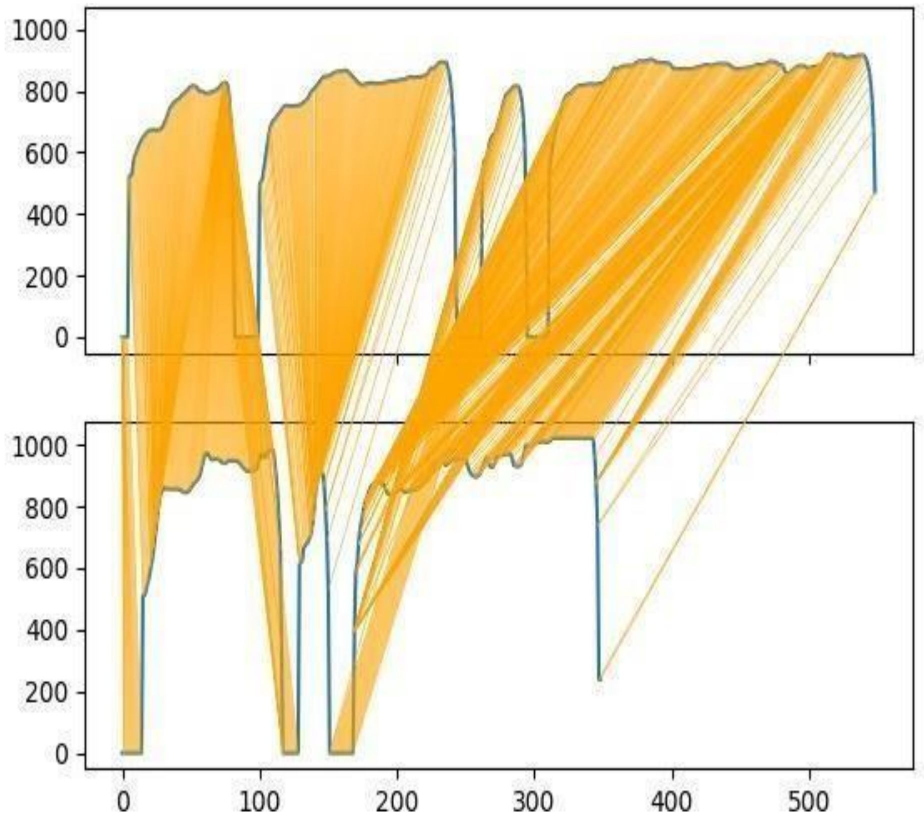


Fig.4.7. DTW Plot for Writer 19 vs Writer 18

```
 RESTART: C:\Users\bhard\AppData\Local\Programs\Python\Python37-32\dtwdist.py
DTW distance b/w 18 and 19(test) for word 'Serially':  2774.8
>>>
```

Fig.4.8. DTW Distance for plot 19 v/s 18

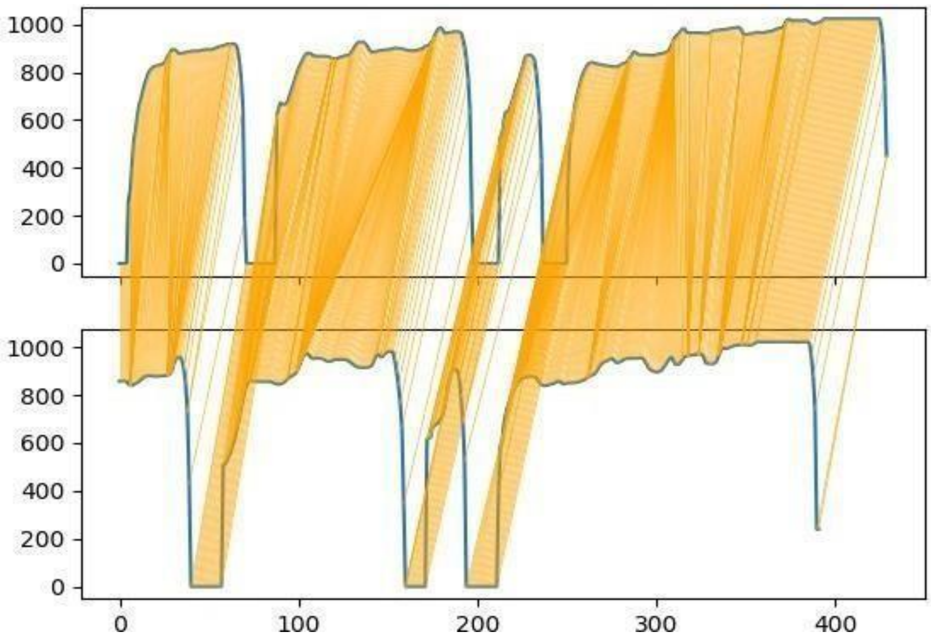


Fig 4.9. DTW Plot for Writer 19 vs Writer 19

```
>>>
 RESTART: C:\Users\bhard\AppData\Local\Programs\Python\Python37-32\dtwdist.py
DTW distance b/w 19 and 19(test) for word 'Serially':  2268.17
>>>
```

Fig.4.10. DTW Distance for plot 19 v/s 19

After recording DTW distance against the respective cases, the table obtained is:

Test Case	Writer with least image hash difference.	DTW distance with least image hash difference.	Writer with second to least image hash difference.	DTW distance with second to least image hash difference.
1	1	1995.94	9	2263.94
2	2	2041.94	16	3580.83
3	6	3864.79	19	4662.33
4	11	2790.67	4	1548.98
5	5	1588.74	13	2661.23
6	6	2551.41	18	4604.2
7	7	1337.99	8	3556.21
8	8	1155.18	12	7240.1
9	9	1247.08	8	1351.61
10	10	1691.25	4	3769.13
11	11	1644.02	14	2137.55
12	12	1780.93	6	4544.89
13	13	1712.54	17	5636.98
14	14	1893.71	11	3768.79
15	15	2069.35	3	3731.31
16	16	3470.0	11	4274.0
17	17	2593.48	1	9366.59
18	2	2209.54	14	1206.58
19	18	2774.8	19	2268.17
20	20	2012.59	17	3271.14

Table No. 4.3. DTW distances against respective cases.

Red entries denote incorrect identification of writer. Black entries are accurate identifications. Blue denotes the entries that were incorrect on the application of avg. image hashing alone, but that gave correct results on application of DTW.
Application of DTW after avg. image hashing gave an accuracy of 90%.

4.6 Results:

DTW was applied on the pressure signals in succession to Average Hashing for the writers having the least avg. hash difference and the second least avg. hash difference, against the test signal. The min DTW distance among them indicated the writer of the text.

Case	Average Image Hash	DTW Distance
7 v/s Test	22	3195.47
9 v/s Test	7	3520.96
10 v/s Test	4	1691.25
14 v/s Test	15	4582.37

Table 4.4. Average Image Hash and DTW distance values for sample cases compared against test case of writer “10”.

After analyzing every test case with the respective samples, the following results were obtained:

Technique used	**Identification Accuracy**
Dynamic Time Warping Only	65%
Difference Hash Difference Only	65%
Average Hash Difference Only	80%
Dynamic Time Warping after Average Hash Difference	90%

Table 4.5. Accuracy of different techniques in identification of the original writer.

CHAPTER-5

CONCLUSION

5.1 CONCLUSION

We have presented a novel algorithm for improving accuracy to detect writer of a handwritten text sample. The proposed method was tested on 8 words, 1 sentence and 1 alphabet, which were collected digitally using a graphics tablet. We found out certain advantages to the proposed approach as compared with other similar techniques, which are discussed one by one in detail. Firstly, the method functions on the plot obtained after pre-processing. Therefore, size variations in the text samples are not of concern. Secondly, this method needs no complex computation and is fast. It may be easily applicable in real-world applications. Finally, the accuracy for writer identification is high. The accuracy can further be enhanced by using high-grade hardware that can measure parameters such as angle of writing, velocity of the pen and the pressure with which the pen is held, which can be used in combination with the technique proposed in this project.

All these advantages demonstrate that the combination of Image Hashing and Dynamic Time Warping, is efficient in accurate identification of a writer.

5.2 FUTURE SCOPE

In future, the above proposed idea may be used on a huge scale to provide cyber security, personal identification and as a method to digitally sign. Further, the recognition method can be used as efficient and faster way for small scale recognition. Checking the authenticity of the writer must be possible with this process. The database executive for verification depends on collection from person to person which can be taken into consideration by the interested parties. The database can be collected by provision of a graphics tablet that records the parameters as presented in the above report. Such a device can be made available at government offices where personal biometrics are captured such as driving license (RTO office) or Aadhar card centers.

REFERENCES:

[1] : Bensefia, A., Tamimi, H.: “Validity of Handwriting in Biometric Systems”, PRAI 2018: Proceedings of the International Conference on Pattern Recognition and Artificial Intelligence, (2018).

[2] : Zaidi,S., Mohammed,S.: “Biometric Handwritten Signature Recognition”, TDDD17: Information Security Course, Linköpingsuniversitet, Sweden, (2009).

[3] : Pugnaloni, M., Federeconi, R.: “Forensic Handwriting Analysis: A Research by means of Digital Biometrical Signature”, Criminalistics and Forensic Examination : Science, Studies, Practice” Vilnius, Lithuania , (2013).

[4] : Venkatesan,R., Koon,S., Jakubowski,M.H., Moulin,P.: "Robust Image Hashing", Proceedings 2000 International Conference on Image Processing , Vancouver, Canada,(2000), pp. 664-666.

[5] : Namboodiri, A., Gupta, S.: “Text Independent Writer Identification from Online Handwriting”, Tenth International Workshop on Frontiers in Handwriting Recognition, Université de Rennes 1, La Baule France (2006).

[6] : Bashir, M., Kempf, J.: “Reduced Dynamic Time Warping for Handwriting Recognition based on Multidimensional Time Series of a Novel Pen Device”, World Academy of Science, Engineering & Technology International Journal of Electrical and Computer Engineering, Vol 2 (2008).

Appendix

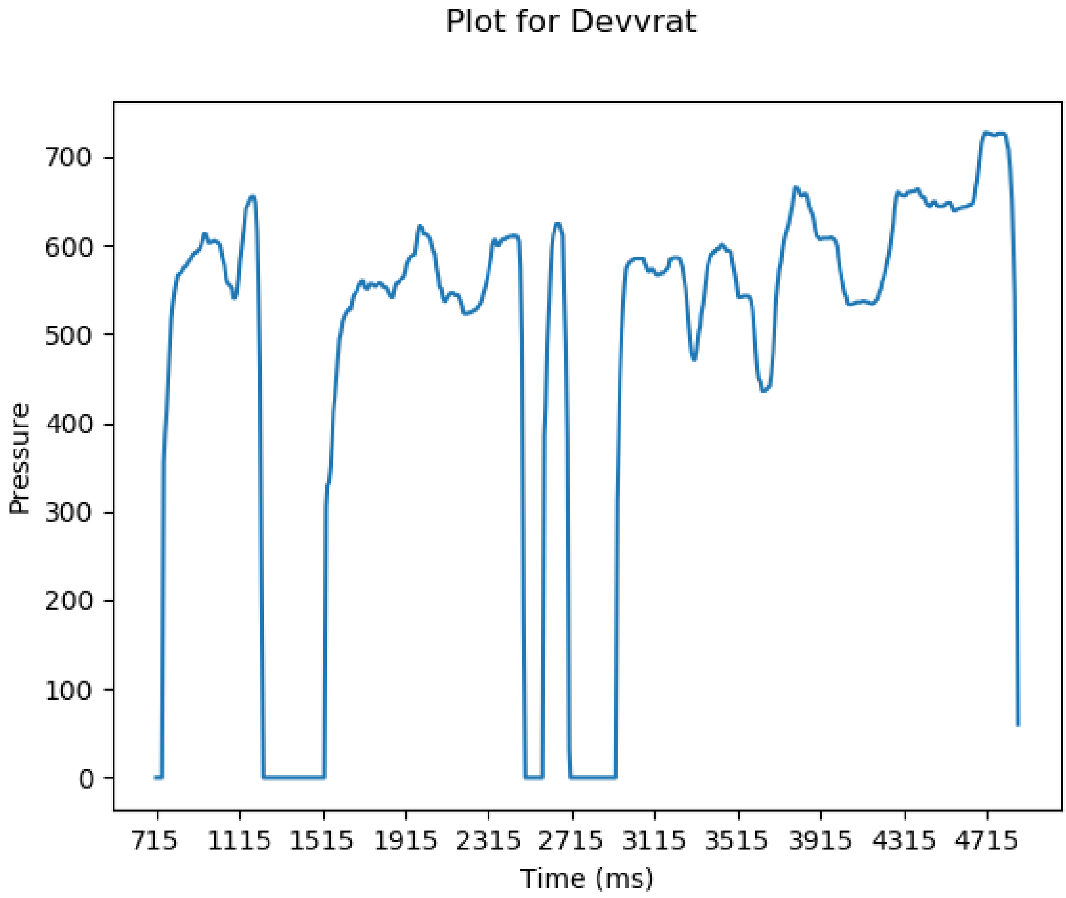


Fig. A1 Pressure v/s time plot for the word "serially" written by Devvrat.

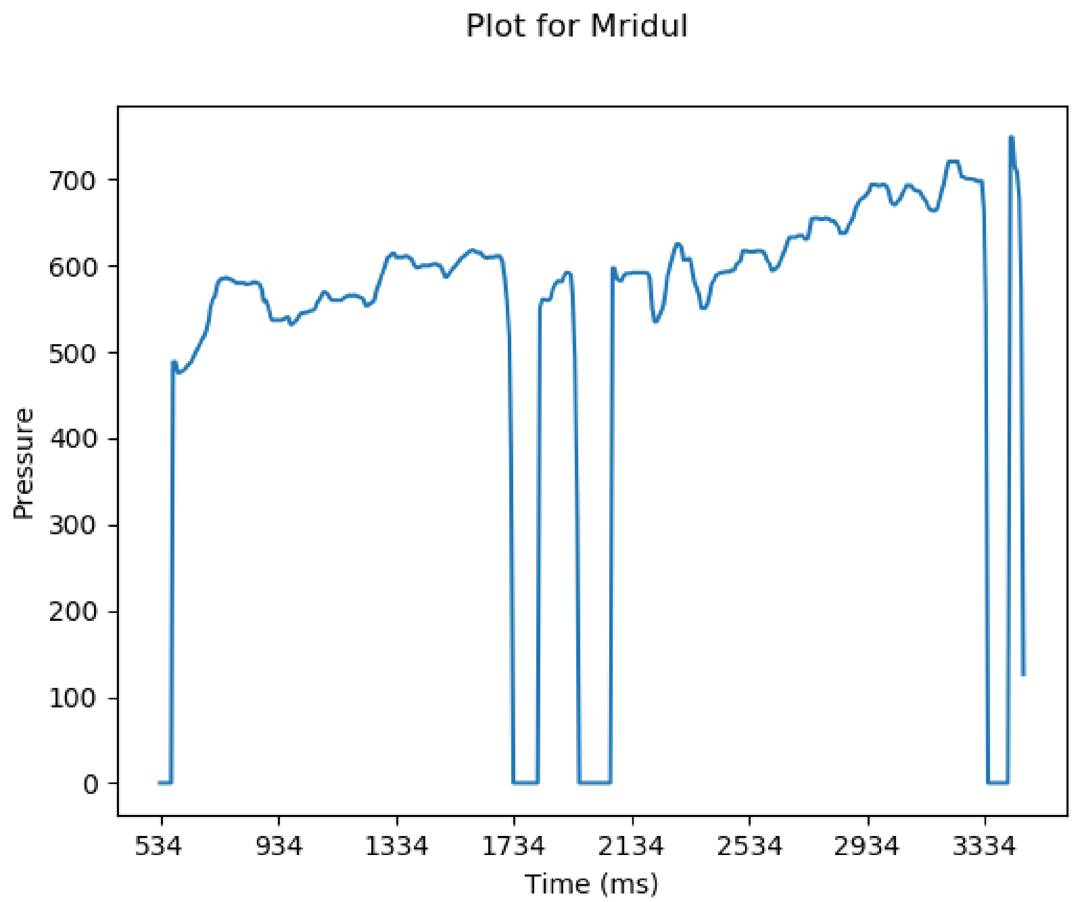


Fig. A2 Pressure v/s time plot for the word "serially" written by Mridul.

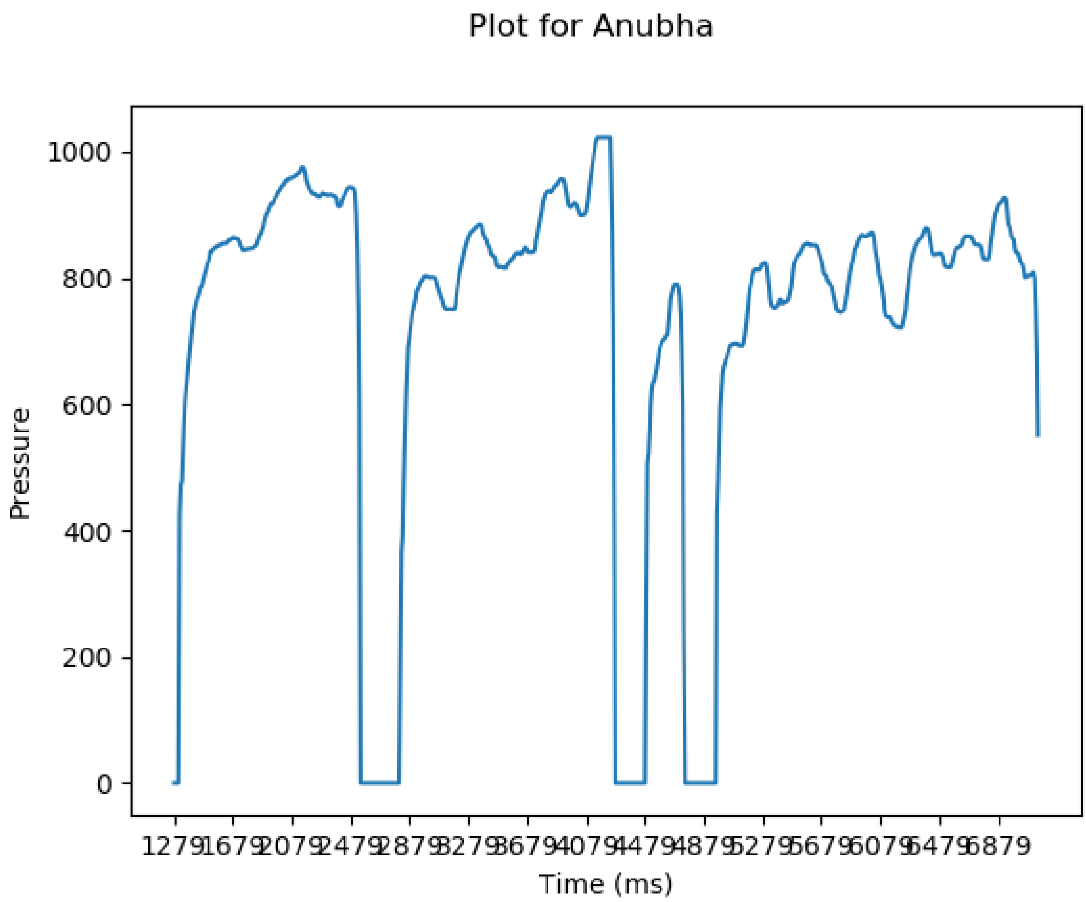


Fig. A3 Pressure v/s time plot for the word "serially" written by Anubha.

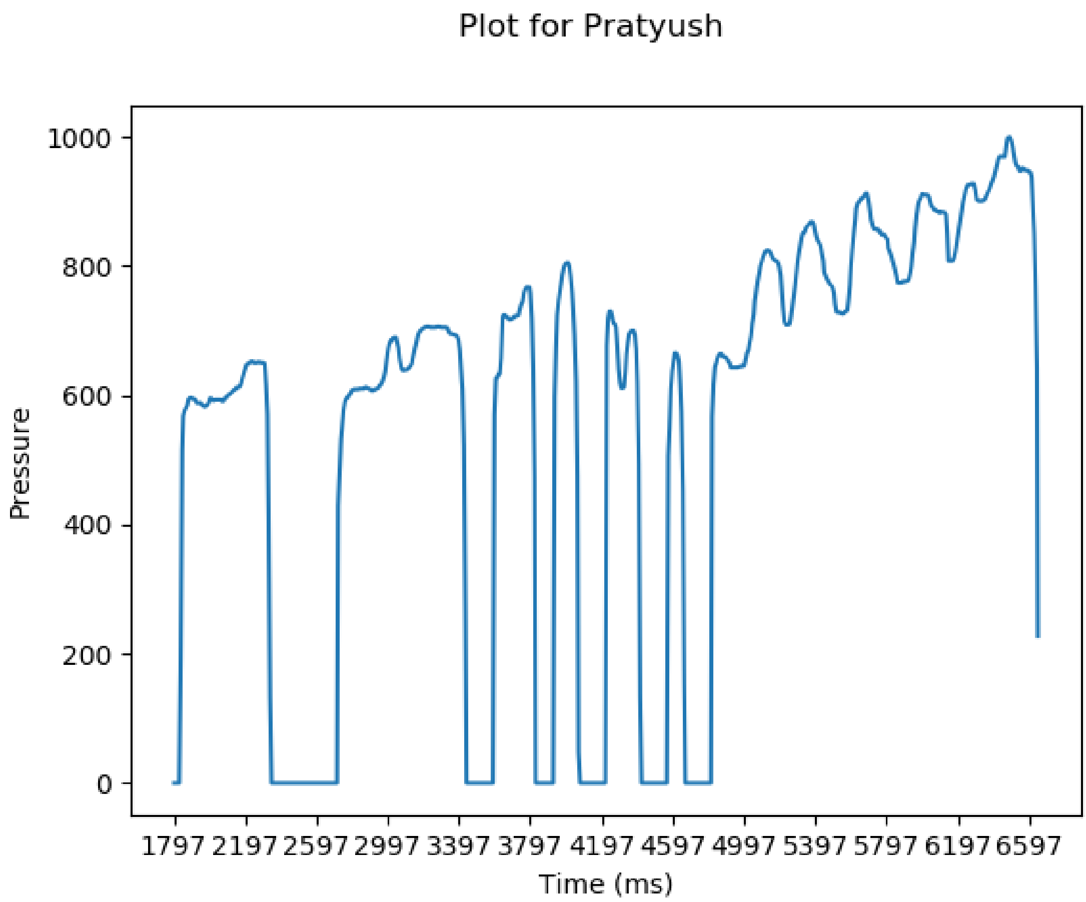


Fig. A4 Pressure v/s time plot for the word “serially” written by Pratyush.

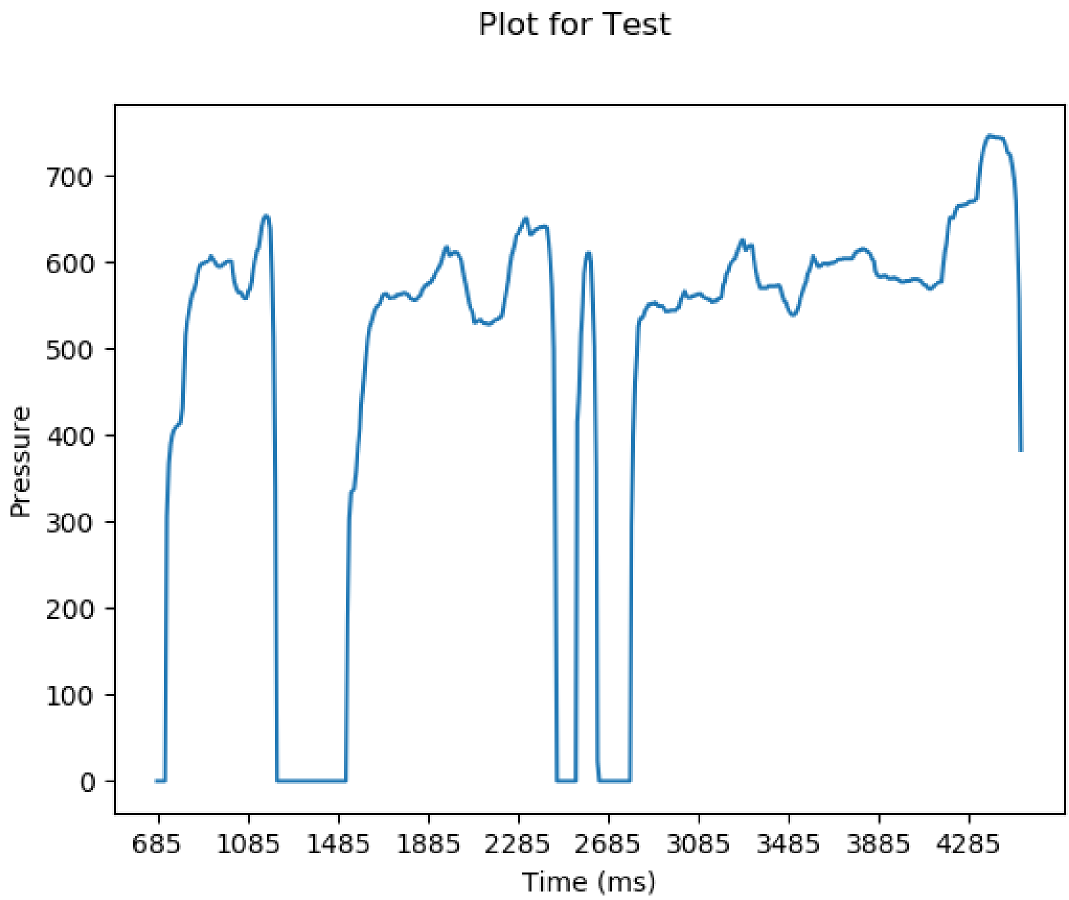


Fig. A5 Pressure v/s time plot for the word “serially” for test.

Final Report

ORIGINALITY REPORT

7%	6%	2%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Jaypee University of Information Technology Student Paper	3%
2	www.ir.juit.ac.in:8080 Internet Source	1%
3	Ameur Bensefia, Hatem Tamimi. "Validity of Handwriting in Biometric Systems", Proceedings of the International Conference on Pattern Recognition and Artificial Intelligence - PRAI 2018, 2018 Publication	<1%
4	www.waset.org Internet Source	<1%
5	cvit.iiit.ac.in Internet Source	<1%
6	ethesis.nitrkl.ac.in Internet Source	<1%
7	repository.its.ac.id Internet Source	<1%

8	scholarworks.wm.edu Internet Source	<1%
9	Submitted to Universiti Teknologi Malaysia Student Paper	<1%
10	Submitted to CSU, San Jose State University Student Paper	<1%
11	krishikosh.egranth.ac.in Internet Source	<1%
12	search.ndltd.org Internet Source	<1%
13	Mace, S.. "Eager interpretation of on-line hand-drawn structured documents: The DALI methodology", Pattern Recognition, 200912 Publication	<1%
14	etd.lib.metu.edu.tr Internet Source	<1%
15	Studies in Computational Intelligence, 2010. Publication	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		