`` **Signature Detection**



B.Sc. (Engineering) PROJECT

*A research project paper submitted to the Department of Information And Communication Engineering, University of Rajshahi, in partial fulfillments of the requirements for the Degree of B.Sc. Engineering*

**Department of Information and Communication Engineering**

**University of Rajshahi, Bangladesh**

**ABSTRACT**

A basic way to offline signature verification has been presented in this report, where the signature is composed on a paper and transferred into an image format or captured using mobile phone. For recognizing the signature, initially I doing some geometrical and statistical calculation planning to extract special features from the signatures then an artificial neural network has been changed on these features from various people.

At last, the extracted features from the tested signature are compared with the previously trained features. In oﬄine (static) signature verification, the dynamic information of the signature writing process is lost, and it is diﬃcult to design good feature extractors that can distinguish genuine signatures and skilled forgeries. This verification task is even harder in writer independent scenarios which is undeniably fiscal for realistic cases.

In this report, an Ensemble model for oﬄine writer, independent signature verification task with Deep learning have been proposed. Two CNNs for feature extraction have been used, after that RGBT for classification & Stacking to generate final prediction vector. Extensive experiments on various datasets from various sources have been done to maintain a variance in the dataset. The state of the art performance on various datasets have been achieved.

**CONTENTS**

Chapter Titles

Abstract

Chapter 1 Introduction

1.1 Introduction to the Project: 1

1.2 Projected Methodology: 3

Chapter 2 Signature Obtained

2.1 Problem Sets: ………………………………………………………………………………4

2.2 Data Sets: …………………………………………………………………………………..5

Chapter 3 Signature Preprocessing

3.1 Introduction: ………………………………………………………………………………..7

3.2 Color Inversion: …………………………………………………………………………….8

3.3 Image Filtering: ……………………………………………………………………………..9

3.4 Binarization: ……………………………………………………………………………… 9

Chapter 4 Feature Extraction

4.1 Introduction: …………………………………………………………………………… 12

4.2 Geometric Features: ………………………………………………………………………13

4.3 Graphometric Features: …………………………………………………………………...13

4.4 Directional Features: …………………………………………………………………… 13

4.5 Mathematical transformations: …………………………………………………………14

4.6 Texture Features: ………………………………………………………………………..14

4.7 Pseudo-dynamic features: ……………………………………………………………….15

4.8 Eccentricity: ……………………………………………………………………………..15

4.9 Skewness: ………………………………………………………………………………..16

4.10 Kurtosis: ………………………………………………………………………………..16

4.11 Orientation: …………………………………………………………………………….17

4.12 Entropy: ……………………………………………………………………………… 17

4.13 Euler Number: ………………………………………………………………………….17

4.14 Solidity: ………………………………………………………………………………..18

4.15 Mean: …………………………………………………………………………………..18

4.16 Standard Deviation: ……………………………………………………………………18

Chapter 5 Methodology

5.1 Processing of signature: 19

5.2 Signature Verification: 19

Chapter 6 Matlab Code and Results

6.1 Matlab Code: ……………………………………………………………………………..20

6.2 Results: 25

Chapter 7 Graphical User Interface

7.1 Notes for Using GUI: 27

Chapter 8 Conclusion and Future Work

8.1 Conclusion: 29

8.2 Future Work: 30

8.3 References: 30

**Introduction**

1.1 Introduction to the Project

In the present, there are several ways to check the validity of one’s personal data, starting from using signature to fingerprint. Signature is a sign as a symbol of the name written by the hand and by the person himself as a personal marker.

Signatures are often used in data verification either in schools, banks, corporations, hospitals, government and much more. Due to the importance of signature function, there are many parties who want to manipulate the signatures of others. Duplicate signatures can be detrimental and included in the criminal realm.

Traditional bank checks, bank credits, credit cards and different authoritative archives are an integral part of the modern economy. They are one of the essential mediums by which people and associations transfer money and pay bills. Even today all these transactions especially financial require our signatures to be authenticated [1].

The inevitable side-effect of signatures is that they can be misused to fake an archive's legitimacy. Hence the need for research in efficient automated solutions for signature recognition and verification has increased in recent years to avoid being vulnerable to fraud.

Signature verification can be viewed as a special instance of pattern recognition. Like in any pattern recognition issue, in signature verification particular features can be extracted from a set of original signatures. However, Approaches to signature verification fall into two classes: On-line and Off-line [2].

On-line data records the motion of the stylus while the signature is produced, and includes location, and online information records the movement of the pointer while the signature is created, and includes location, speed, increasing speed and pen pressure, as elements of time. Online systems use these data captured during signing.

**1 |** P a g e

Online systems could be used in real time applications like credit cards transaction or resource access. Then again, Off-line signatures are scanned from paper documents, where they were written in ordinary way. Off-line signature verification systems take as input the 2-D image of a signature [3].

Offline systems are useful in automatic verification of signatures found on bank checks and documents. A robust system has to be designed which should not only be able to consider these factors but also detect various types of forgeries.

In signature verification, forged signatures can be broken up into three different categories. These categories are based on how similar a forgery is in relation to the genuine signature and are known as random, simple and skilled. In random forgery the forger does not know the signer’s name or signature shape [5].

In simple forgery or unskilled forgery, the forger knows the name of the original signer but not what his signature looks like. While in skilled forgery, a close imitation of the genuine signature is produced by a forger who has seen and practiced writing the genuine signature. It is these skilled forgeries that this paper will focus on for signature verification.

Off-line Signature analysis can be completed with a filtered picture of the signature utilizing a standard camera or scanner, and they are helpful in programmed confirmation of signatures found on bank checks and archives.

We approach the problem in two steps. Initially the scanned signature image is preprocessed to be suitable for extracting features. Then the preprocessed image is used to extract relevant geometric parameters that can distinguish forged signatures from exact ones using the ANN approach. Part2 deals with the preprocessing steps and explains the features that are extracted followed by ANN construction and training procedures in Part 3. Implementation details and simulation results are listed in Part 4. Finally the conclusions are drawn in Part 5.

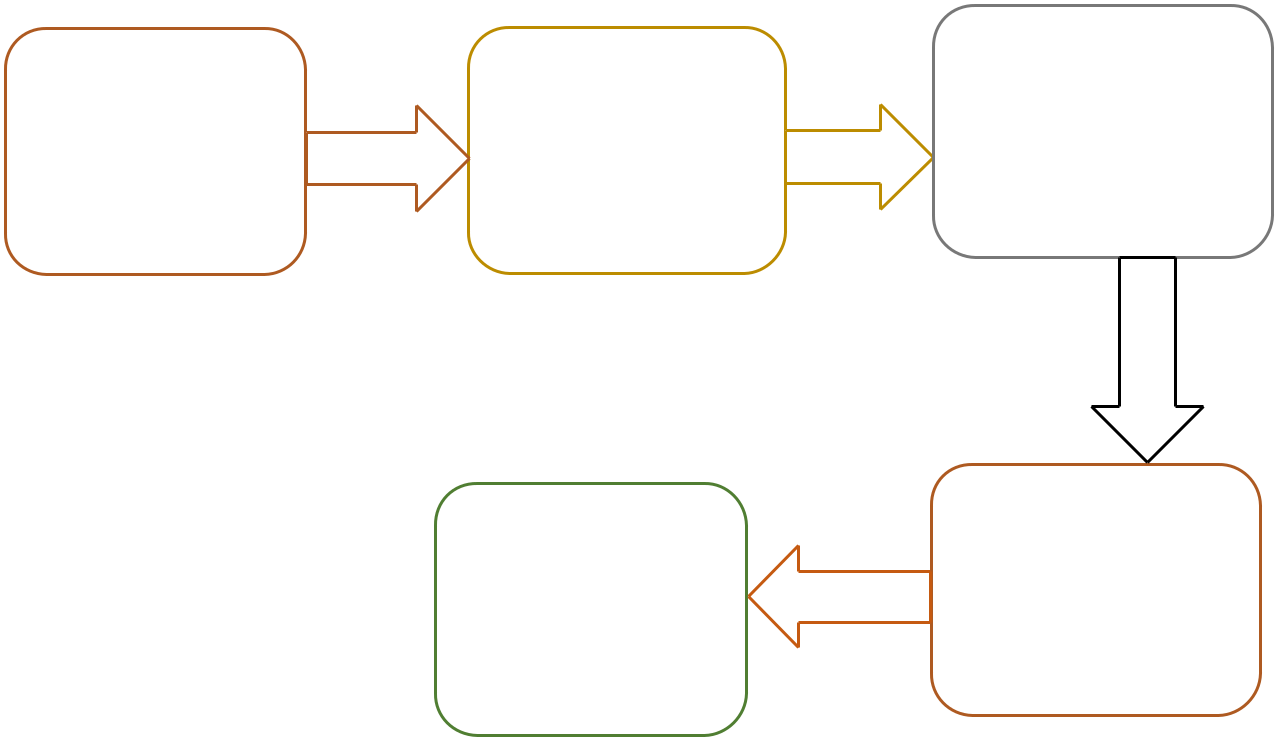
In this project I choose to implement the Off-line signature using Matlab.

First we will discuss about algorithm and how I solve the issue then we move on to discuss the results.

**2 |** P a g e

1.2 Projected Methodology

This segment depicts our way to solve the problem as I separated it into five steps as appeared in the diagram below:



|  |  |  |  |
| --- | --- | --- | --- |
| Signature | Signature | Characteristics |  |
| Extraction |  |
| Obtained | Preprocessing |  |
|  |  |

Signature Signature

Verification Processing

**3 |** P a g e

**Signature Obtained**

2.1 Problem set:

There are two ways to get the signature:

1. Using the phone or tablet .
2. By using scanned images of signatures from scanner or camera.

The issue of programmed manually written signature verification is normally demonstrated as a verification task: given a learning set L, that contains certified signatures from a lot of clients, a model is prepared. This model is then utilized for verification: a client asserts a personality and gives a question signature Xnew.

The model is utilized to characterize the signature as authentic (having a place with the guaranteed individual) or imitation (made by another person). To assess the presentation of the framework, we consider a test set 'Test' comprising of genuine signatures and forgeries.

The signatures are obtained in an enrollment stage, while the subsequent stage is referred to tasks (or classification) stage. If a single model is used to classify images from any user, we refer to it as a writer-independent (WI) system.

In the event that one model is prepared for every client, it is referred as an essayist subordinate (WD) framework. For WI frameworks, the normal practice is to prepare and test the framework with an alternate subset of clients.

**4 |** P a g e

For this situation, we consider an improvement set D (which is utilized to prepare the WI model), and a misuse set E, which speak to the clients enlisted to the framework (and is additionally partitioned in L and Test, as demonstrated previously).

Most work in the literature do not use skilled forgeries for training. Other work use skilled forgeries for training writer-independent classifiers, testing these classifiers in a different

arrangement of clients; finally, a few papers utilize gifted falsifications for preparing essayist subordinate classifiers, and test these classifiers in a different arrangement of certifiable marks and frauds from a similar arrangement of clients.

We limit our investigation to techniques that don't depend on talented frauds for the clients took a crack at the framework (the set E), since this isn't the situation in down to earth applications.

We do consider, however, that a dataset consisting of genuine signatures and forgeries is available for training writer-independent classifiers (the set D), where the users from this dataset are not used for evaluating the performance of the classifier.

This is reasonable for a practical application, since it is possible for an institution to collect forgeries for some users (e.g. by detecting actual forgery attempts), that could be used for training WI systems.

2.2 Data Sets

A large amount of research in automated signature verification has been conducted with private datasets. This makes it difficult to compare relate work, since an improvement in classification performance may be attributed to a better method, or simply to a cleaner or simpler database.

In the last decade, however, a few signature datasets were made available publicly for the research community, addressing this gap. The process to acquire the signature images follows similar steps for most of the public datasets.

**5 |** P a g e

The genuine signatures are collected in one or more sessions, and require the user to provide several samples of their signatures. The user receives a form containing many cells, and provide a sample of his/her signature in each cell. The cells often have sizes to match common scenarios such as bank cheques and credit card vouchers.

The collection of forgeries follows a different process: the users receive samples from genuine signatures and are asked to imitate the signature one or more times. It is worth noting that the users that provide the forgeries are not experts in producing forgeries. After the forms are collected, they are scanned (often at 300 dpi or 600 dpi), and pre-processed [6].

Table I presents a summary of the most commonly used signature datasets.

|  |  |
| --- | --- |
| Users | Signature Number |
|  |  |
| Atika | 5 |
|  |  |
| Fatin | 5 |
|  |  |
| Quick | 5 |
|  |  |
| Fox | 5 |
|  |  |
| Saju | 5 |
|  |  |

**6 |** P a g e

**Signature Preprocessing**

3.1 Introduction

**Signature extraction** -This is an initial step that consists in finding and extracting a signaturefrom a document. This is a specific testing issue in bank checks, where the signature is frequently composed on head of a complex background. This progression is, in any case, not considered in most signature verification examines, that as of now consider signatures extracted from the records.

**Noise Removal** - Scanned signature images often contain noise. A common strategy to addressthis problem is to apply a noise removal filter to the image, such a median filter. It is also common to apply morphological operations to fill small holes and remove small regions of connected components.

**Size normalization and centering** - Depending on the properties of the features to be used,different size normalization strategies are adopted. The least complex methodology is to edit the signature pictures to have a tight box on the signature.

Another methodology is to client a smaller bounding box, for example, cutting strokes that are a long way from the picture centroid, that are frequently dependent upon more change in a client's signature . Different creators utilize a fixed outline size (width and stature), and focus the signature in this edge.

**Signature representation** - Besides just using the gray level image as input to the featureextractors, other representations have been considered. For example, utilizing the signature's skeleton, framework, and ink conveyance, high pressure regions and directional frontiers.

**Signature Alignment** - Alignment is a common strategy in online signature verification,yet notextensively applied for the offline situation. The signature is first captured and changed into a course of action that can be handled by a PC.

**7 |** P a g e

By and by its set up for preprocessing. In preprocessing stage, the RGB picture of the signature is changed over into grayscale and a while later to binary picture. The motivation behind this stage is to get ready signatures prepared for attributes extraction. The preprocessing stage includes two steps: Color inversion, Filtering and Binarization.

3.2 Color Inversion

Color inversion, also known as the negative effect, is one of the easiest effects to achieve

in image processing. Color inversion is achieved by subtracting each RGBcolor value from the maximum possible value (usually 255).The true color image RGB is converted to the grayscale intensity image by eliminating the hue and saturation information while retaining the luminance.

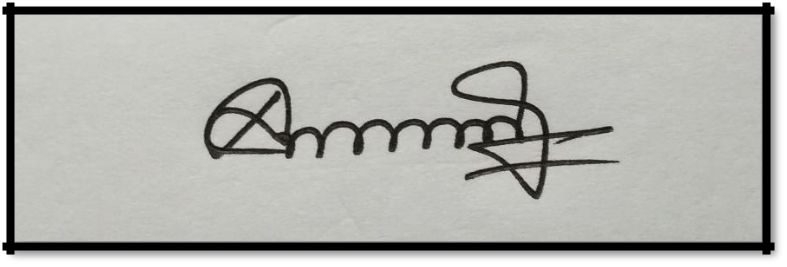
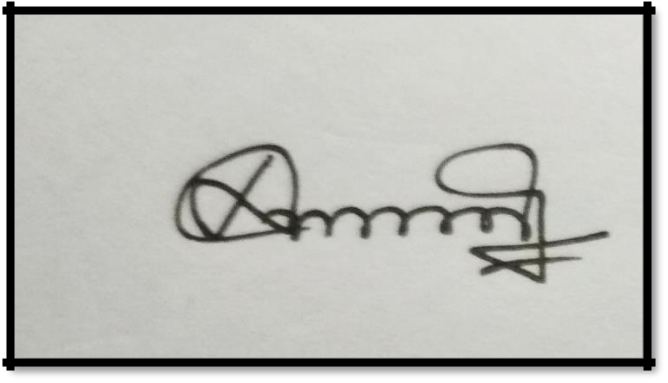


Fig. 1. (a) A sample signature to be processed;



(b) A Grayscale Intensity Image;

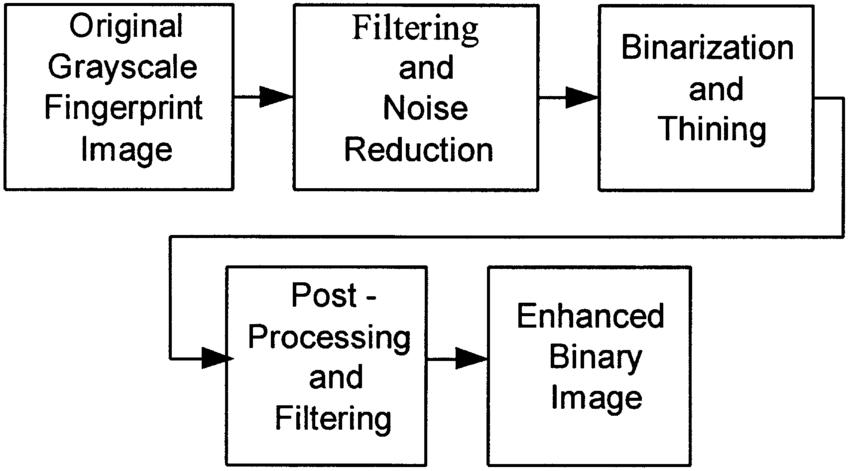
A grayscale image is a data matrix whose values represent intensities within some range where each element of the matrix corresponds to one image pixel.

**8 |** P a g e

3.3 Image Filtering

Any image when resample is filtered by a low pass FIR filter. This is done to avoid aliasing. This aliasing occurs because of sampling the data at a rate lower than twice the largest frequency component of the data.

So a low pass filter will remove the image high frequency components. And for this reason the filter used.



3.4 Binarization

The conversion of a gray scale image into black or white, so called binary image is

called **binarization.** The simplest way of binarization is **thresholding:** setting pixels to white (or

1. if the gray value is equal or greater than the threshold or setting to black (0) if smaller. When using adaptive thresholding, the level of threshold is determined automatically based on the content of the image or image sequence.

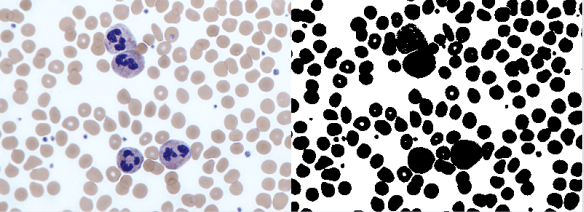
A well-established way of automatic threshold determination is Otsu’s method. An alternative method is using a given percentile of the intensity histogram as threshold value.

**9 |** P a g e

In contrast **locally adaptive thresholding** uses a level that varies object by object in the image.

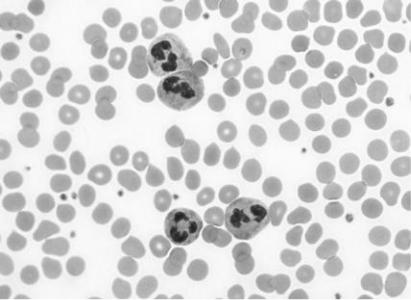
This can distinguish bright spots, shapes over varying background.

Binarization is the process of converting a pixel image to a binary image:



In the old days binarization was important for sending faxes. These days it’s still important for things like digitalizing text or segmentation.

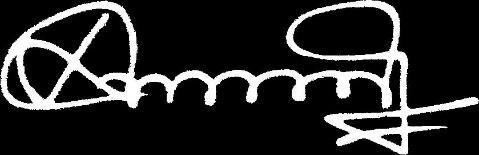
At first the image is converted into grayscale:



Then a threshold gets applied:



**10 |** P a g e



**11 |** P a g e

**Feature Extraction**

4.1 Introduction

Feature plays a very important role in the area of image processing. Before getting features, various image preprocessing techniques like binarization, thresholding, resizing, normalization etc. are applied on the sampled image.

After that, feature extraction techniques are applied to get features that will be useful in classifying and recognition of images. Feature extraction techniques are helpful in various image processing applications e.g. character recognition. As features define the behavior of an image, they show its place in terms of storage taken, efficiency in classification and obviously in time consumption also.

The features extracted from handwritten signature assume an imperative job in authentication process. An enormous number of features are extracted from signatures but not all features can be used in the feature set.

A good feature set would bring about an effective framework. It is really important to have a significant feature set for affirmation of appropriate learning by NN.

Offline signature verification has been studied from many perspectives, yielding multiple alternatives for feature extraction. Broadly speaking, the feature extraction techniques can be classified as Static or Pseudo-dynamic, where pseudo dynamic features attempt to recover dynamic information from the signature execution process (such as speed, pressure, etc.). Another broad categorization of the feature extraction methods is between Global and Local features.

Global features describe the signature images as a whole - for example, features such as height, width of the signature, or in general feature extractors that are applied to the entire signature image.

**12 |** P a g e

In contrast, local features describe parts of the images, either by segmenting the image (e.g. according to connected components) or most commonly by the dividing the image in a grid (of Cartesian or polar coordinates), and applying feature extractors in each part of the image.

Recent studies approach the problem from a representation learning perspective: instead of designing feature extractors for the task, these methods rely on learning feature representations directly from signature images. .

4.2 Geometric Features:

Geometric features measure the overall shape of a signature. This includes basic descriptors, such as the signature height, width, caliber (height-to-width ration) and area. More complex descriptors include the count of endpoints and closed loops.

Besides using global descriptors, several authors also generate local geometric features by dividing the signature in a grid and calculating features from each cell. For example, using the pixel density within grids [7][8].

4.3 Graphometric features:

Scientific report inspectors utilize the ideas of graphology and graphometry to look at handwriting for several purposes, including distinguishing genuineness and forgery.They chose a subset of graphometric features that could be portrayed algorithmically, and proposed a lot of feature descriptors.

They thought about the accompanying static features: Caliber the proportion of Height/Width of the picture; Proportion, referring to the symmetry of the signature, Alignment to baseline describing the angular displacement to a horizontal baseline, and Spacing - portraying void spaces between strokes.

4.4 Directional features:

Directional features try to describe the picture as far as the bearing of the strokes in the signature.

**13 |** P a g e

This descriptor utilized this strategy for include extraction utilizing networks of numerous scales.

This descriptor speaks to neighborhood shapes in a picture by a histogram of edge directions, likewise in different scales.

4.4 Mathematical transformations:

Researchers have used a variety of mathematical transformations as feature extractors and investigated the usage of a fast Hadamart transform and spectrum analysis for feature extraction. Pourshahabi used a Contourlet transform as feature extraction, stating that it is an appropriate tool for capturing smooth contours.

Broadened Shadow Code for signature verification. A network is overlaid on head of the signature picture, containing level, vertical and corner to corner bars, each bar containing a fixed number of receptacles. Every pixel of the signature picture is then anticipated to its nearest bar toward every path, actuating the particular container. The include of dynamic receptacles in the projections is then utilized as a descriptor of the signature.

This feature extractor has been utilized with various goals, along with directional highlights, to accomplish promising outcomes on author free and essayist subordinate classification, separately.

4.5 Texture features:

Texture features, in particular variants of Local Binary Patterns (LBP), have been used in many experiments in recent years. The LBP operator describe the local patterns in the image, and the histogram of these patterns is used as a feature descriptor.

LBP variations have been used in many studies and have demonstrated to be among the best hand-crafted feature extractors for this task. Another important texture descriptor is GLCM (Gray Level Co-occurrence Matrix). This feature uses relative frequencies of neighboring pixels, and was used in a few papers [9].

Interest point matching methods, such as SIFT (Scale-Invariant Feature Transform) and SURF

**14 |** P a g e

(Speeded up Robust Features) have been largely used for computer vision tasks. Ruiz-del-Solar used SIFT to extract local interest points from both query and reference samples to build a writer-dependent classifier.

After extracting interest points from both images, they generated a set of 12 features, using information such as the number of SIFT matches between the two images, and processing time. During classification, only the stable interest points are used for matching. The number of key points in the query image, and the number of matched key points were used to classify the signature as genuine or forgery [10].

4.6 Pseudo-dynamic features:

Pseudo-dynamic features, based on graphometric studies: Distribution of pixels, Progression - that measures the tension in the strokes, providing information about the speed, continuity and uniformity, Slant and Form - measuring the concavities in the signature.

More recently, Bertolini proposed a descriptor that thinks about the arch of the signature.

This was cultivated by fitting Benzier bends to the signature layout (all the more uncommonly, to the biggest portion of the signature), and utilizing the boundaries of the bends as features [11].

4.7 Eccentricity

It can be thought of as a measure of how much the conic section deviates from being circular. In particular, the eccentricity of a circle is zero. Ellipses, hyperbolas with all possible eccentricities from zero to infinity. In [mathematics,](https://en.wikipedia.org/wiki/Mathematics) the eccentricity of a [conic section](https://en.wikipedia.org/wiki/Conic_section#Eccentricity) is a non-negative real number that uniquely characterizes its shape [4].

More formally two conic sections are [similar](https://en.wikipedia.org/wiki/Similarity_(geometry)) [if and only if](https://en.wikipedia.org/wiki/If_and_only_if) they have the same eccentricity.One can think of the eccentricity as a measure of how much a conic section deviates from being circular. In particular:

* The eccentricity of a [circle](https://en.wikipedia.org/wiki/Circle) is [zero.](https://en.wikipedia.org/wiki/0)
* The eccentricity of an [ellipse](https://en.wikipedia.org/wiki/Ellipse) which is not a circle is greater than zero but less than 1. **15 |**P a g e

* The eccentricity of a [parabola](https://en.wikipedia.org/wiki/Parabola) is 1.
* The eccentricity of a [hyperbola](https://en.wikipedia.org/wiki/Hyperbola) is greater than 1

4.8 Skewness

It refers to something that is out of line or distorted on one side. “Skewness is a measure of symmetry, or more precisely, the lack of symmetry.

A distribution, or data set, is symmetric if it looks the same to the left and right of the center point”. Skewness can range from minus infinity to positive infinity. Skewness refers to distortion or asymmetry in a symmetrical bell curve, or [normal distribution,](https://www.investopedia.com/terms/n/normaldistribution.asp) in a set of data. If the curve is shifted to the left or to the right, it is said to be skewed.

Skewness can be quantified as a representation of the extent to which a given distribution varies from a normal distribution. A normal distribution has a skew of zero, while a [lognormal](https://www.investopedia.com/articles/investing/102014/lognormal-and-normal-distribution.asp) [distribution,](https://www.investopedia.com/articles/investing/102014/lognormal-and-normal-distribution.asp) for example, and would exhibit some degree of right-skew.

Skewness is used along with kurtosis to better judge the likelihood of events falling in the tails of a probability distribution.

4.9 Kurtosis

In a similar way to the concept of skewness, Kurtosis is a descriptor of the shape of a probability distribution and, just as for skewness; there are different ways of quantifying it for a theoretical distribution and corresponding ways of estimating it from a sample from a population [4].

The measurement of skewness allows us to determine how bowed are the lines in each segment of the signature.

**16 |** P a g e

4.10 Orientation

It allows us to know how the signer wrote down the signature. Which letters came first emphasizing the direction of angles and peaks?

Orientation' — Scalar; the angle (in degrees ranging from -90 to 90 degrees) between the x-axis and the major axis of the ellipse that has the same second-moments as the region [4].

4.11 Entropy

Entropy is a statistical measure of randomness that can be used to characterize the texture of the input image [4].

4.12 Euler Number

The Euler number for the binary image BW is the total number of objects in the image minus the total number of holes in those objects. BW can be numeric or logical and it must be real[4].

The bweuler function returns the Euler number for a binary image. The Euler number is a measure of the topology of an image. It is defined as the total number of objects in the image minus the number of holes in those objects.

We can use either **4**- or 8-connected neighborhoods. This example computes the Euler number for the circuit image, using 8-connected neighborhoods.

BW1 = imread('circbw.tif');

eul = bweuler(BW1,8)

eul = -85

In this example, the Euler number is negative, indicating that the number of holes is greater than the number of objects.

**17 |** P a g e

4.13 Solidity

Scalar specifying the proportion of the pixels in the convex hull that are also in the region.

Computed as Area/Convex Area [4].

4.14 Mean

A mean is the simple mathematical average of a set of two or more numbers. The mean for a given set of numbers can be computed in more than one way, including the

arithmetic mean method, which uses the sum of the numbers in the series, and the geometric mean method, which is the average of a set of products.

Average or mean value of the elements in the image.

4.15 Standard Deviation

Standard deviation is the measure of dispersion of a set of data from its mean. It measures the absolute variability of a distribution; the higher the dispersion or variability, the greater is

the standard deviation and greater will be the magnitude of the deviation of the value from their mean.

Standard deviation of matrix elements is the square root of an unbiased estimator of the variance of the population from which X is drawn.

**18 |** P a g e

**Methodology**

5.1 Processing of signature

Artificial Neural Network or ANN takes after the human brain in learning through training and data storage. The ANN is made and trained through a given input/target data training pattern. During the learning procedure, the neural network output is compared with the target value and a network weight correction via a learning algorithm is acted in such a manner to limit a mistake work between the two qualities.

The mean-squared error (MSE) is a commonly used error function which tries to minimize the average error between the network's output and the target value. Five genuine signatures from each signer where taken to prepare the network and they were sufficient to give generally excellent outcomes in confirmation

5.2 Signature Verification

Signature preprocessing is an important step to improve the exactness of the latter algorithm, and to reduce their computational needs. Following preprocessing steps are thought about:

1. Transformation from color to grayscale.
2. Transformation from grayscale to black and white using suitable threshold.
3. Color Inversion.
4. Apply opening and shutting morphological procedures with little structure component to remove the inward and external noise.

**19 |** P a g e

**Matlab Code and Results**

7.2 Matlab Code

Signature Template:

clc

* clear all;

global clients global strOp;

clients = ["Saju","FOX","Fatin","qui","Atika"]; qntt = 5;

dat = [];

for i = 1:size(clients,2)

for j = 1:qntt

img = imread(sprintf('DataSets/%s%d.jpg',clients(i),j)); img = imresize(img, [80 300]);

dat = [dat img];

end

end

cells = qntt \* size(clients,2);

colums = 300 + zeros(1,cells);

templates=mat2cell(dat,80,colums);

save ('templates','templates')

% clear all

**20 |** P a g e

Edge Cutting:

strInp = "16.jpg";

strOp = "Fatin5";

dat = imread(sprintf('%s',strInp)); dat = rgb2gray(dat);

dat =

imbinarize(dat,'adaptive','ForegroundPolarity','dark','Sensitivi

ty',0.4);

dat = 1-dat;

dat = bwareaopen(dat,100);

[tRow, tCol] = size(dat);

%Beginnings Horizontal Edge Cutting while 1

tmp = 0;

for j = 1:tRow

tmp = tmp + dat(j,1);

end

if tmp == 0

dat(:,1) = [];

else

break

end

end

%Endings Horizontal Edge Cutting

dat = imrotate(dat,180);

while 1

tmp = 0;

for j = 1:tRow

tmp = tmp + dat(j,1);

end

if tmp == 0

dat(:,1) = [];

else

break

end

end

dat = imrotate(dat,180);

**21 |** P a g e

%Top vertical edge cutting

[tRow, tCol] = size(dat);

while 1

tmp = 0;

for j = 1:tCol

tmp = tmp + dat(1,j);

end

if tmp == 0

dat(1,:) = [];

else

break

end

end

dat = imrotate(dat,180);

while 1

tmp = 0;

for j = 1:tCol

tmp = tmp + dat(1,j);

end

if tmp == 0

dat(1,:) = [];

else

break

end

end

dat = imrotate(dat,180);

imshow(dat);

imwrite(dat,sprintf('%s%s.jpg','DataSets/',strOp)); % imwrite(dat,strOp);

**22 |** P a g e

Signature Detection:

clear all;

dat = imread('test/test4.jpg');

dat = rgb2gray(dat);

dat =

imbinarize(dat,'adaptive','ForegroundPolarity','dark','Sensitivi ty',0.4);

dat = 1-dat;

dat = bwareaopen(dat,100);

[tRow, tCol] = size(dat);

%Edge Cutting

while 1

tmp = 0;

for j = 1:tRow

tmp = tmp + dat(j,1);

end

if tmp == 0

dat(:,1) = [];

else

break

end

end

dat = imrotate(dat,180);

while 1

tmp = 0;

for j = 1:tRow

tmp = tmp + dat(j,1);

end

if tmp == 0

dat(:,1) = [];

else

break

end

end

dat = imrotate(dat,180);

**23 |** P a g e

[tRow, tCol] = size(dat);

while 1

tmp = 0;

for j = 1:tCol

tmp = tmp + dat(1,j);

end

if tmp == 0

dat(1,:) = [];

else

break

end

end

dat = imrotate(dat,180);

while 1

tmp = 0;

for j = 1:tCol

tmp = tmp + dat(1,j);

end

if tmp == 0

dat(1,:) = [];

else

break

end

end

dat = imrotate(dat,180);

%End of edge cutting

dat = imresize(dat, [80 300]);

imshow(dat);

load templates;

comp=[];

sz = size(templates,2);

for n=1:sz

sem=corr2(templates{1,n},dat);

comp=[comp sem];

%pause(1)

end **24 |** P a g e

average = [];

sum = 0;

threshold = 0.15;

count = 0;

pos = 0;

for i = 1:5:sz

for j = i:i+5

pos = (floor(i/5)+1);

sum = sum+comp((i-1)+pos);

end

average = [average (sum/5)];

sum = 0;

if average(pos)>=threshold

% sprintf("Signature matched with %d",pos); count = count + 1;

break;

end

end

Sig\_Template;

global clients

if count == 0

sprintf("Signature did not match");

else

name = clients(pos);

sprintf("Signature matched with %s",name);

end

ans

7.1 Results

The data base of about 25 signatures was tested. I test the system on five of my friends taking 5 signature of each one then and adding some data set from the internet and the result was about 92% for a classification ratio. It could distinguish 23 out of 25 signatures. The precision of signature verification systems can be expressed by two types of error:

False Acceptance Ratio (FAR): The false acceptance ratio is given by the number of fake signatures accepted by the system with respect to the total number of comparisons made.

**25 |** P a g e

False Rejection Ratio (FRR): The false rejection ratio is the total number of genuine signatures rejected by the system with respect to the total number of comparisons made.

Both FAR and FRR depend on the threshold variance parameter taken to decide the genuineness of an image. If I choose a high threshold variance then the FRR is reduced, but at the same time the FAR also increases. If I choose a low threshold variance then the FAR is reduced, but at the same time the FRR also increases.

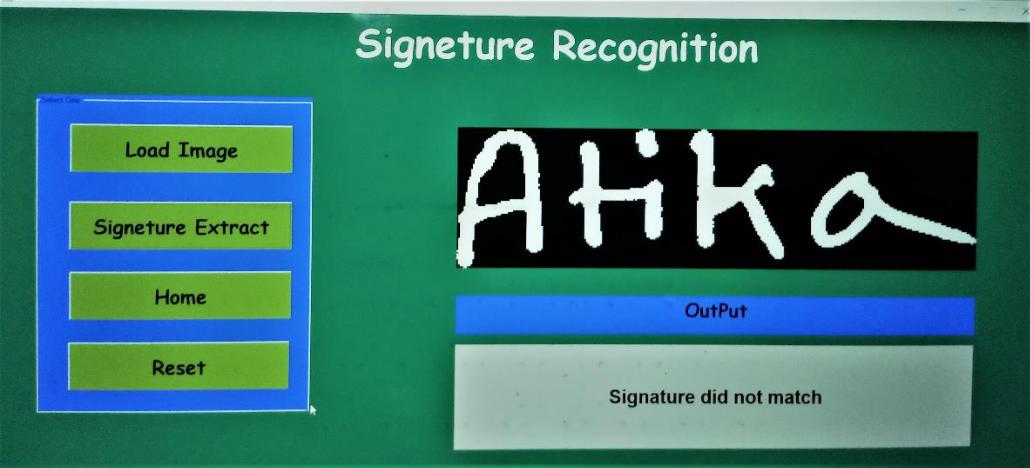
I obtain taking a threshold of 0.15 .The network was tested and it was capable of classifying the signatures of the taken database: exact or forged. And the minimized error percentages constitute an additional factor for the success of the verification system.

**26 |** P a g e

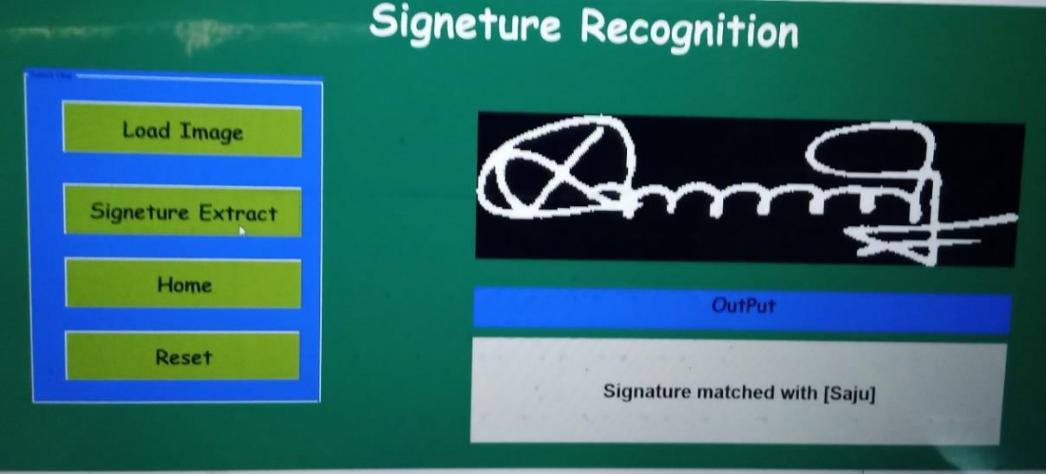
**Graphical User Interface**

6.1 Notes for Using GUI

I built a simple GUI shown to facility inserting the signature images and add the likelihood to draw your own signature using the phone and lastly test your signature to verify the signer.



**27 |** P a g e



**28 |** P a g e

**Conclusion and Future Work**

8.1 Conclusion

Neural systems have shown their accomplishment in numerous applications because of their capacity to take care of certain issues without any difficulty of utilization and the model-free property they enjoy. One of the primary features, which can be attributed to ANN, is its capacity to learn nonlinear issue disconnected with specific preparing, which can prompt adequately exact reaction.

Utilization of Artificial Neural Network (ANN) to the previously mentioned issue has accomplished expanding significance for the most part because of the productivity of present day PCs. Furthermore, the hours of reproduction and testing in the ANN application are insignificant. What's more, the check framework dependent on ANN can learn various types of signature datasets, by utilizing just mathematical disconnected features.

In addition, the utilization of enormous databases isn't required to show the ability of learning for such an issue, we have picked just five certified signatures and three manufactured ones for preparing, and we get excellent outcomes. Anyway for genuine practice use, bigger preparing information can build the robustness of the framework.

In the wake of preparing, the best order exact nesses were accomplished.

The algorithm we supported uses simple geometric features to characterize signatures that effectively serve to classify signature as exact or forged.

The framework is strong and can recognize arbitrary, straightforward and semi-talented forgeries. We have no away from about its exhibition in the event of talented fabrications since we are not skillful imitating signatures to the degree being considered as skilled forgeries.

**29 |** P a g e

8.2 Future Work

We can improve the work to check on-line signature which could be more precise or solid framework and we could manufacture an android application that could likewise some other biometric confirmation framework like handwriting examination when joined with other biometric angles, for example, discourse and face acknowledgment can introduce a far superior outcome than any individual framework.

8.3 References

1. Ali Karounia, Bassam Dayab, Samia Bahlakb, “Offline signature recognition using neural networks approach,” science direct, p. 7, 2010.
2. Julio Martínez-R.,Rogelio Alcántara-S., “On-line signature verification based on optimal feature representation and neural-network-driven fuzzy reasoning,” IEEE, 2002.
3. Harpreet Anand,Prof. D.L Bhombe, “ENHANCED SIGNATURE VERIFICATION AND RECOGNITION USING MATLAB,” IJIRAE, vol. 1, no. 4, p. 7, 2014.
4. Saeed Al.Sabbagh, Eng. Basel Shanwar, Dr. Nizar Zarka, “Signature Verification”.
5. L.Basavaraj and R.D Sudhaker Samuel, “Offline-line Signature Verification and

Recognition: An Approach Based on Four Speed Stroke Angle”, International Journal of Recent Trends in Engineering, Vol 2, No. 3, November 2009.

1. J.F. Vargas, M.A. Ferrer, C.M. Travieso, and J.B. Alonso. Off-line Handwritten Signature GPDS-960 Corpus. In Ninth International Conference on Document Analysis and Recognition, 2007. ICDAR 2007, volume 2, pages 764–768, September 2007.
2. H. Baltzakis and N. Papamarkos. A new signature verification technique based on a two-stage neural network classifier. Engineering applications of Artificial intelligence, 14(1):95–103, February 2001.

**30 |** P a g e

1. A. El-Yacoubi, E. J. R. Justino, R. Sabourin, and F. Bortolozzi. Offline signature verification using HMMs and cross-validation. In Neural Networks for Signal Processing X, 2000. Proceedings of the 2000 IEEE Signal Processing Society Workshop, volume 2. IEEE, 2000.
2. Alceu S. Britto, Robert Sabourin, and Luiz E. S. Oliveira. Dynamic selection of classifiers - a comprehensive review. Pattern Recognition, 47(11):3665–3680, November 2014.
3. M. Malik, M. Liwicki, A. Dengel, S. Uchida, and V. Frinken. Automatic Signature Stability Analysis and Verification Using Local Features. In Frontiers in Handwriting Recognition, Int. Conference on. IEEE, 2014.
4. D. Bertolini, L. S. Oliveira, E. Justino, and R. Sabourin. Reducing forgeries in writer-independent off-line signature verification through ensemble of classifiers. Pattern Recognition, 43(1), January 2010.

**31 |** P a g e