S	
E1	
E2	
For Office Use Only	

Masters Project Final Report

(MCS)

2020

Project Title	Offline Handwritten Signature Verification System
Student Name	B.U.D Thambugala
Registration No. & Index No.	2016/MCS/106 & 16441068
Supervisor's Name	Dr. D.A.S Atukorale

For Office Use ONLY

Offline Handwritten Signature Verification System

A dissertation submitted for the Degree of Master of Computer Science

B.U.D Thambugala
University of Colombo School of Computing
2020

DECLARATION

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: B.U.D Thambugala

Registration Number: 2016/MCS/106

Index Number: 16441068

Signature Date 13/11/2020

This is to certify that this thesis is based on the work of

Mr. B.U.D Thambugala

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Dr. D.A.S Atukorale

Signature Date 13/11/2020

ABSTRACT

Even before the era of computers, handwritten signature was being used as a unique biometric. There are two methods that extensively deliberated the signature verification. They are Offline method and Online method. Even if it is considered somewhat difficult than online method due to the need of dynamic information, offline systems are straightforward to make use of when compared to online systems. Because of its importance for use in day-to-day life, the offline verification system has taken more attraction.

This document presents an offline handwritten signature verification system using the support vector machine approach. Features are extracted from the signature images by calculating, Grey level Co-occurrences Matrices (GLCM). Then the texture feature calculations are performed and the SVM model gets trained with them. The appropriate SVM parameters (Gamma and C) are obtained by performing a k-Fold Cross-Validation by trying out different parameter combinations. In the verification process the texture feature calculation of the disputed signature image is performed and obtained the feature vector to be verified with the trained SVM Model. And finally, in the classification, verification result is classified as genuine or forged.

This method takes care of skilled forgeries. The main objective of the solution is to minimize the two important parameters False Acceptance Rate (FAR) and False Rejection Rate (FRR) usually used in any kind of signature verification system.

The proposed system has achieved a performance of approximately 86% by using a dataset of 768 signatures (genuine signatures and skilled forgeries) from 32 writers.

Keywords: signature verification; support vector machine; grey level co-occurrences matrices; k-fold cross-validation

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my supervisor Dr. D.A.S Atukorale for providing his invaluable guidance, comments, and suggestions throughout the course of the project.

I sincerely convey my thankfulness to Dr. Randil Pushpananda, the project coordinator for their immense support, timely guidance, and valuable instructions.

University of Colombo, School of computing helped me throughout this thesis, and I am grateful to it.

Finally, an honorable mention goes to my parents and friends for their understandings and supports on me in completing this project.

Finally, I am responsible for any errors that remain in this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

CID	Criminal Investigation Department
FAR	False Acceptance Rate
FRR	False Rejection Rate
GLCM	Grey Level Cooccurrences Matrices
HMM	Hidden Markov Model
NN	Neural Network
DTW	Dynamic Time Warping
DRT	Discrete Radon Transform
RBF	Radial Basis Function

INTRODUCTION

1.1 Motivation

The basic accepted methods for a person to authenticate himself to another are based on at least one of the following three general principles.

- What he/she knows?
- What he/she has? or
- What he/she is?

Based on the assumption that a person's signature changes slowly, the handwritten signature is treated as the general way of recognizing the signer of a written document. And also, handwritten signature is very hard to remove, change or forge without detection [1].

Signature is a special case of handwriting which consisted of special symbols and hence hard to be identified even by a human [1]. Signature is not one of the physiological properties of a person, like face or fingerprint. Hence it is a behavioral biometric. Therefore, one's signature may change time to time, and it is not nearly as easy to forge as iris patterns or fingerprints [2].

The signature is appropriate for some lower-security authentication needs because the public widely accepting it. Signature has a primary benefit in that it is the accepted way of claiming the identity of a person in day-to-day operations like electronic fund transfers and automated banking transactions. When someone is mindful or willing to write in the typical way only, signature analysis is possible. To give an opposite example, even when someone is in an unconscious state of mind his or her fingerprint can be used. [2]

Also, due to intrapersonal and interpersonal variances, it is required to analyze signature as a whole image instead of characters and words. The requirement for study in effective auto-mated solutions for signature verification has improved in recently, because signature is the main approach for both authentication and authorization in legal transactions. [3]

Depending on the data obtaining approach, handwritten signature verification can be categorized as online handwritten signature verification (dynamic) and offline handwritten signature verification (static).

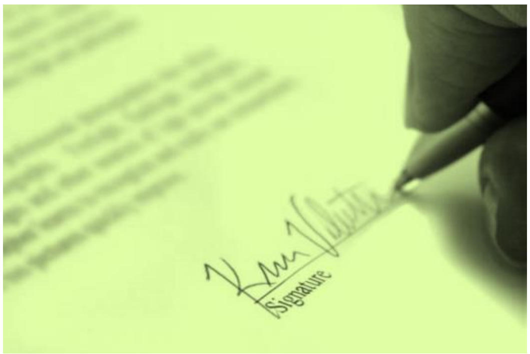


Figure 0.1: Offline signatures [4]

Figure 0.2: Online signatures [5]

The offline approach contains fewer electronic control [Figure 1.1]. It uses images of the signature captured by a scanner or camera. This approach is bit challenging than online handwritten signature verification because of the unavailability of the dynamic information and the difficulty to recover them from the images [6]. In the online approach, it provides more information about a signature with the dynamic properties of the signature. It requires special instruments and devices to track the pen movements and pressure [Figure 1.2]. Then the device associated with the pen will extract information on typing speed, emphasis points, shocks, acceleration and other important static information to verify the signatures [6].

Because of scanning hardware or paper background, signatures in offline systems generally may have noise and may hold fewer discriminative information, since the image of the signature is the only input to the system. An automated offline signature verification makes a very challenging problem, because genuine signatures of the same individual may a bit differ while variances among a forged and a genuine signature may be unnoticeable [2].

1.2 Problem Overview

Biometrics measure an individual's unique behavioral or physical features with the intent of identifying or verifying their identity [7]. And fingerprints, hand or palm geometry, retina, iris, or facial features are the most used physical biometrics and signature, voice, keystroke pattern, and gait are the behavioral characteristics used. Handwritten signatures are widely used, and it is one of the popular social and legal attributes known to use in person identification. Handwritten signature is a symbol of consent and authorization in banks and other financial organizations, and a well concerned mark for fraud for a long time [7].

A signature can be properly written when someone is mindful and in the potential of write as normal, even if it is likely that individual may be forced to sign. A forged signature can accurately be produced only by a well skilled and an experienced forger [7]. Manual signature verification can vary from person to person and depends on the signature verification skills, the mood of the person, the level of concentration for the task, etc. A signature verified by someone could be considered as a forged signature by another and vice versa. When a person is required to verify a large number of signatures per day, he or she can easily get exhausted by the process and will lead to a higher error rate. In such cases, there is a greater likelihood of rejecting legitimate signatures and more likely to accept forged signatures. Such things can have serious consequences.

1.2.1 Context and preliminary investigation of the problem

The problem area was identified by conducting an investigation through document reviews, observations, and interviews. The author recognized the issues which are mostly applicable to Sri Lanka that arise when signature matching tasks are performed.

1.2.1.1 Investigation methods

Investigation is an extremely vital process to carry out any sort of project, successfully. To gather data and other information, there are several approaches for the investigation which helps out. [8]

- Document review
- Observation
- Interview

□ **Document review**

The followings are concerns that increase the importance of conducting document review to gather information.[8]

- Eases to obtain information which may be impossible or difficult to explore.
- Eases to obtain large samples of data.
- Eases to overcome problems of encouraging participation by users.
- The cost of conducting document review is low.

While conducting the document review research papers and relevant articles were analyzed very carefully and it was identified some issues which cause difficulties in the signature verification process.

Following are some of the issues in matching signatures which cause difficulties to the signature verification process [9] [10] [11] [12].

- The complexity of signature patterns makes the signature verification process harder.
- Difficulty to match signatures due to the wide intrapersonal differences.
- Uncertainty in pattern structure and the interaction among components.
- The minimal variances of skilled forgeries with the genuine signatures.
- The quality of the signatures relies on the various circumstances at the time of signing.
- The signature matching gets complex by random variations, due to the writer's pauses or hesitations.
- Short signatures could carry fewer information than long signatures, resulting in poor accuracy in verification outcomes.
- Individuals with same names share similar signatures with others. At least concerning shape characteristics.
- Difficulty to eliminate forgeries created by tracing or photocopying.

□ Observation

Following are some concerns that made the importance to conduct observations to gather information [8].

- Eases to obtain live data from real situations.
- Eases to obtain data in a physical environment and its organization.
- Eases to find out things that other parties might not talk openly in interviews.
- Eases to find inefficiencies.

Set of six individuals were asked to sign on a white color A4 paper. A signature per day and in two days 12 signatures were collected. The signing processes were cautiously observed while they are signing on the paper.

The images of the collected signatures for the observation are given below.

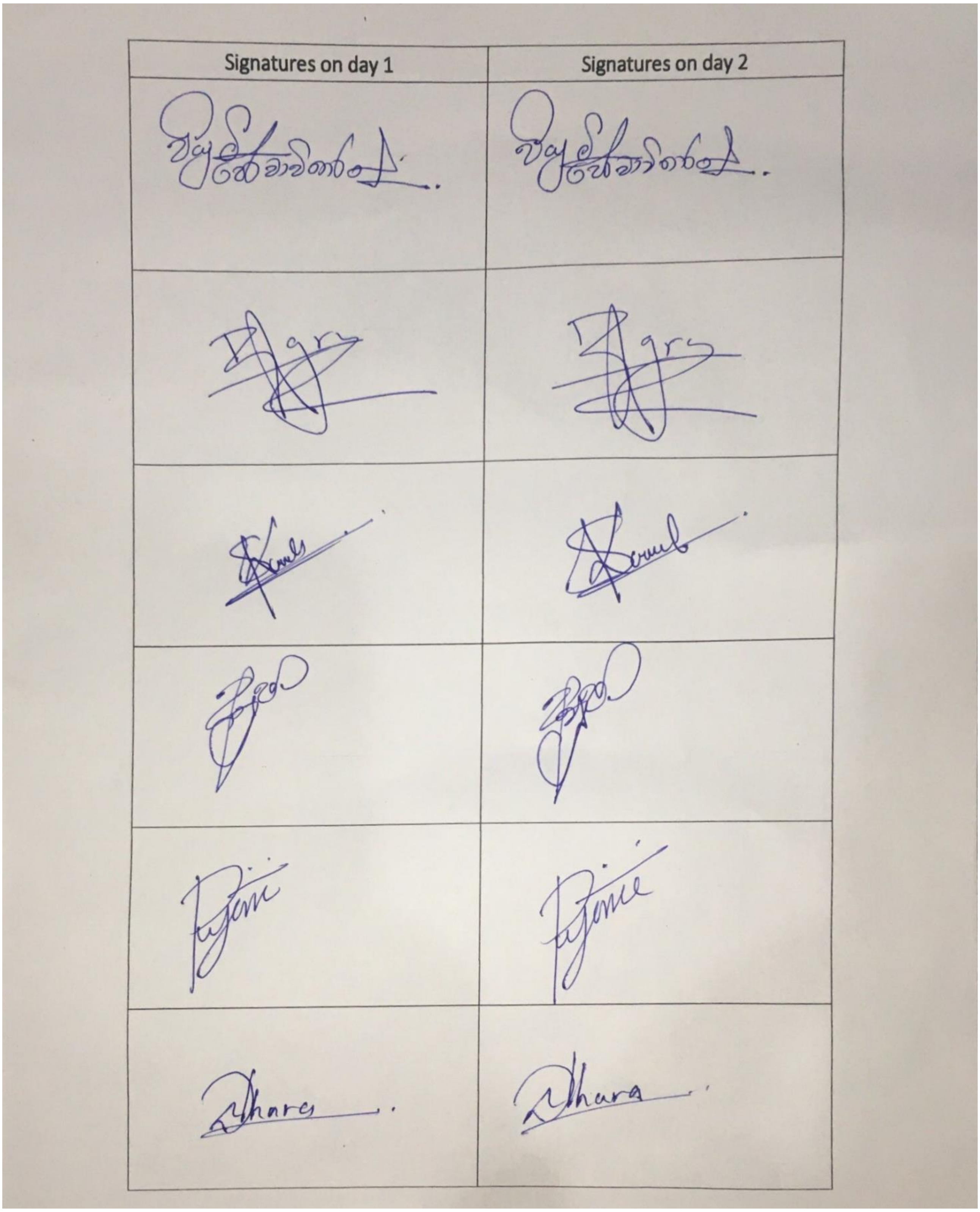


Figure 0.3: Signatures collected for observation

It was identified that the signatures collected on the first day are slightly different from the signatures collected on the second day of the same individual. This shows that the signatures of the same individual slightly vary from one signature to another.

- □ **Interview**

The followings are some concerns that increase the importance of conducting interviews to gather information [8].

- Provides an environment to ask more detailed questions and to obtain detailed information.
- Maintains a suitable situation to achieve a high response rate.
- Possibility of recording Interviewee’s own words.
- A well-known strategy to handle difficult and open-ended questions.
- Eases to clear-up uncertainties.

A sample of 3 individuals who are regularly involved in the signature verification process from the banking sector has been interviewed to gather relevant data and information regarding the signature matching process. (Please refer to APPENDIX A to view the interview document used for preliminary investigation)

From the interviews conducted, the probabilities and the frequencies of the issues in the signature verification process which were identified by the document review were recorded. This would help to identify the issues, more likely to occur and define the project objectives and its scope to address those issues in the proposed system. The following table shows the frequencies of the identified issues occur in the signature verification process. The mean values of the frequencies have been calculated from the values of frequencies recorded by the interviews conducted.

Issue	**Frequency**
The complexity of signature patterns makes the signature verification process harder	1.33 %
Difficulty to match signatures due to the wide intrapersonal differences	2.33 %
The minimal variances of skilled forgeries with the genuine signatures	2.66 %
The signature matching gets complex by random variations, due to the writer’s pauses or hesitations	0.33 %
Short signatures could carry fewer information than long signatures, resulting in poor accuracy in verification outcomes	3.33 %
Individuals with same names share similar signatures with others. At least concerning shape characteristics	1.33 %
Difficulty to eliminate forgeries created by tracing or photocopying	0.33 %

Table 0.1: Frequencies of the issues in matching signatures

1.3 Project Objective(s)

- Identify what makes different among genuine and forged signature, which is correlated to intrapersonal and interpersonal variability while achieving an acceptable level of accuracy in the signature verification process. Intrapersonal difference is the dissimilarity among the signatures of the same person and interpersonal is the difference among the genuine signatures and the forgeries.

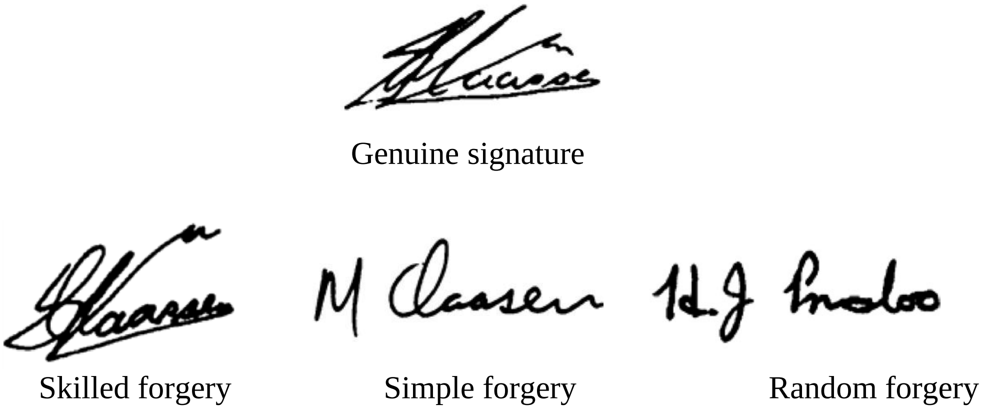
- The proposed solution will be trained with sample signatures by accepting the signatures with a white background from scanned images for the verification process.

1.4 Scope

There are three types of signature forgeries [13].

- **Random forgery**
  - The forger does not have access to genuine signatures and there is no known information about the name of the person which the forger tries to imitate the signature and the forger produces a random signature.
- **Simple forgery**
  - The forger does not have access to the samples of the signatures but is aware about the name of the person which the forger tries to imitate the signature and, produces the signature in the forger's own style.
- **Skilled forgery**
  - The forger has access to the samples of the genuine signatures and reproduces it.

Following are some of the sample genuine and forged signatures



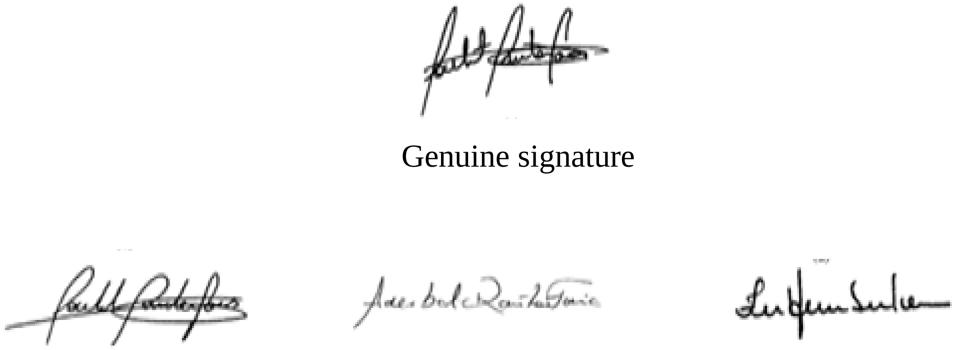
Genuine signature

Skilled forgery

Simple forgery

Random forgery

Figure 0.4: Genuine & forged signatures – Sample 1 [13]



Genuine signature

Skilled forgery

Simple forgery

Random forgery

Figure 0.5: Genuine & forged signatures – Sample 2 [14]

Although it is not easy to plan, still important to identify the signer's identity, as most current financial transactions are still on paper.

Therefore, the project scope is to analyze Sinhala, Tamil, and English signatures in the Sri Lankan context using an offline handwritten signature verification system and to identify skilled forgeries which have a minimal difference to the genuine signatures.

1.4.1 Limitations

Signature images should be extracted from a white color background. Accuracy of the classification depends on the noise in the background of the signature image. Scanned images are ideal for the verification process.

1.5 Structure of the Dissertation

After went through a fully comprehensive description and understanding about the problem domain and scope of the project in the Introduction chapter, the next chapters of this thesis contains important details of the study much deeper as given below.

In the second chapter contains a comprehensive elaboration about the literature review of the problem domain which has been referred during the project. The mentioned studies under this chapter are the existing knowledge and new methods which related to the research.

The third chapter is the critical analysis where decide what approaches should use for further stages in the project by summarizing the information gathered in the literature review. Moreover, it has a descriptive evaluation of the approaches through a trial and error process by doing experiments.

The fourth chapter describes about the proposed solution by elaborating the methodology which has taken in order to achieve the targets. Moreover, explains the main phases of signature verification related to the proposed system.

In the fifth chapter of this document contains the details of different evaluation approaches and how they are performed and the results of each approach.

The final chapter, chapter six contains the general conclusion of the research and the important discoveries grasped. It also explains on the possible areas of future research related to the study.

LITERATURE REVIEW

This chapter of the report presents an overview of characteristics of forged signatures, different phases of a signature verification system, what are the approaches taken by previous researchers in the past few years and the techniques used for their solutions.

2.1 Characteristics of forged signatures

Followings are some of the characteristics of the forged signatures [15].

- **Larger in term of size**

A forged signature is generally large in term of size when comparing with a genuine signature. This happens because the forger cautiously observes the genuine signature while imitating. And the response system of the forger's brain is slower than the genuine signer's and because of that makes the forged signature larger in terms of the size.

- **Curves become angular**

It is frequent that the curved letters are more angular than in the genuine signature. The forger uses a slower speed to produce curves accurately to obtain the correct letter shapes. Since more time is spent on curves, these areas become more angular than in the genuine signature.

- **Retouching**

Retouching results once the signature has been imitated, but some additions are made after the imitation.

- **Poor line quality**

The ink lines make known the differences in speed and pressure with different amount of ink appearing on the paper. It is often visible that more pressure is applied on forged signatures than the genuine signatures.

- **Hesitation**

During the signature forging process, the person who forge may pause the process to observe the genuine signature and then continue. When the pen leaves, it often creates blobs on the paper.

- **Punctuation**

There could be full stops, dots on letters such as “i, j” placed in incorrect places, missing, or added.

- **Different pressure**

This happens because different pressure applied by the signer, while signing. Pressure variances occur in several regions of the signature. It is hard to imitate the pressure differences as identical to the genuine signature.

- **Sudden endings**

Usually lines of forged signatures just stop whereas the lines of genuine signatures fade away.

- **Spacing**

There could be strange spaces between letters, words, and punctuations.

- **Forger’s characteristics**

The forger unintentionally exposes features of their own handwriting such as spacings, basic letter shapes and positions of letters in relation to baseline while doing the forgery.

- **Baseline error**

The forger regularly not attend to care that the baseline of the forged signature is similar to the baseline of the genuine signature.

- **Bad line quality**

This happens when the forged signature has been produced too slowly and shows hesitant or shaky pen strokes.

- **Forming characters not appearing in genuine signature**

Forgeries created by not seeing the genuine signature but knowing only the name which is used to sign could include letters which doesn’t appear in the genuine signature.

2.2 Phases of an Offline Handwritten Signature Verification System

There are many offline handwritten signature verification systems proposed and designed by various authors. An offline handwritten signature verification system has the following four main phases [3][14].

- Image acquisition
- Preprocessing
- Feature extraction
- Verification

Image acquisition is the phase which captures the signature images. For offline signature verification systems, signature image is scanned using a digital scanner and stored digitally after feeding into the system for preprocessing [14]. In the preprocessing stage the signature images get altered in order to generate an appropriate input for the feature extraction phase [14] [16]. In the feature extraction phase different features of the signatures are extracted and stored in a database. And during the verification phase, features extracted from an inputted signature are compared against the information in the database to judge whether a signature is genuine or forged [17].

Although there are several stages of offline handwriting verification, these steps are not always separate. This is because the whole system is an algorithm, and the intermediate stages rely on the results of previous stages and interconnect with the other stages [Figure 5]. Designing an efficient algorithm for an offline handwritten signature verification system requires loads of research and analysis. Therefore, this section provides a brief overview of the stages involved in the system and describes what the various authors have accomplished under each of their algorithms.

Workflow of an offline signature verification system is as follows [14].

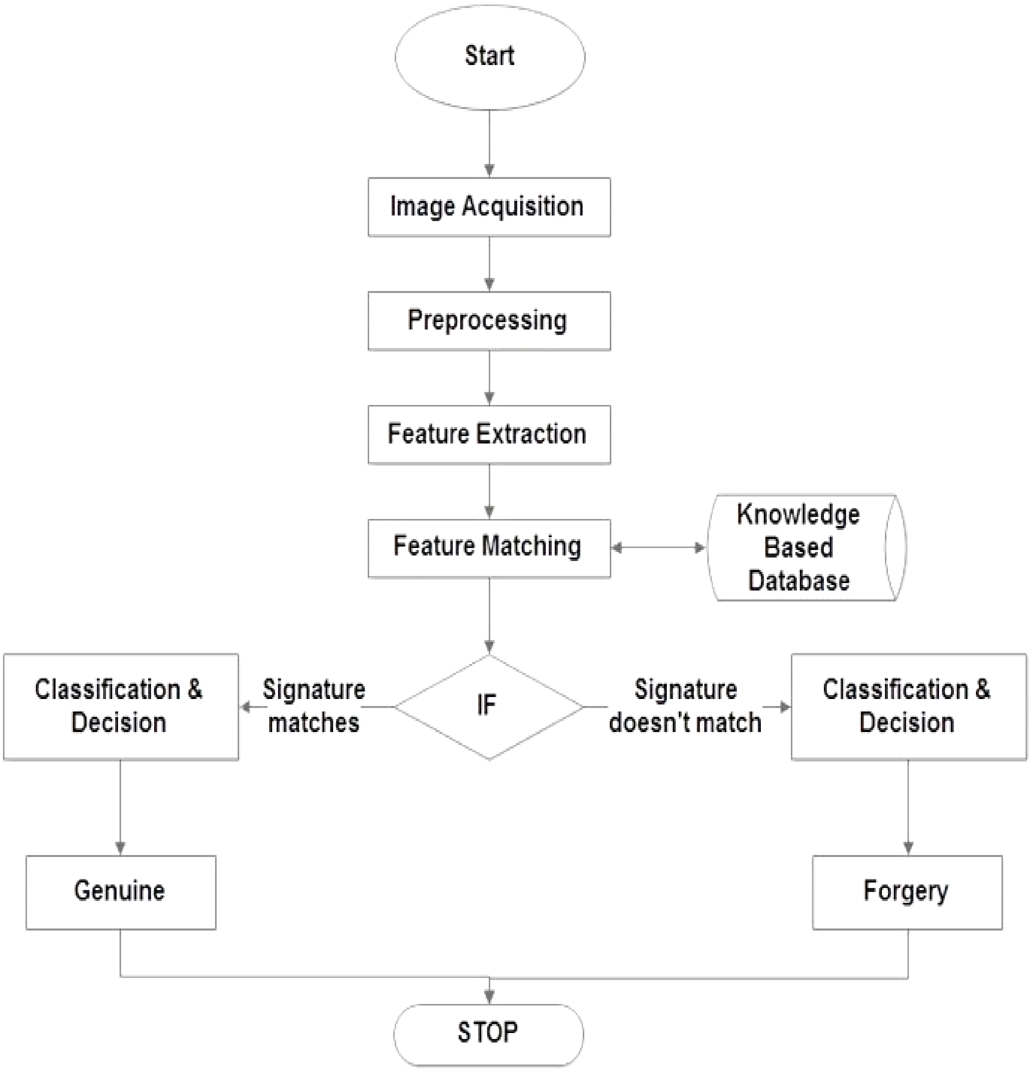


Figure 0.1: Workflow of an offline signature verification system

2.2.1 Preprocessing

Preprocessing is an important step in order increase the accuracy of the latter algorithms of the system and to minimize their computational requirements [3]. This process is done in both training and testing phases of the system. In preprocessing phase makes the signature up to a standard and prepares the signature image for the feature extraction. Different preprocessing steps can be involved in different systems according to the requirements [18].

Following are some of the preprocessing steps involved [1][11][18][19][20][21].

- **Noise removal**

Noise removal is important to remove the unnecessary pixels contained in the signature image which are not a part of the signature. When a signature is scanned from a paper some unnecessary pixels come with the scanned image and these unnecessary parts must be removed before the feature extraction process.

Signature image with noise

Signature image without noise

Figure 0.2: Signature noise removal [20]

- **Background Elimination**

A lot of image processing applications need separation of objects from the background and Thresholding is the appropriate method for the background removal [1].

- **Converting image to binary**

Converting a grayscale image into binary makes feature extraction simpler [18].

- **Image resizing**

Signature images fed to the system could be of different sizes and it is required to bring them into a standard size [18].

- **Thinning**

The thinning process reduces binary objects or shapes to strokes that are single pixel wide [18].

Signature image before thinning

Signature image after thinning

Figure 0.3: Signature thinning [18]

- **Bounding box of the signature**

This process decreases the area around the signature to be used for further processing and saves time [18].

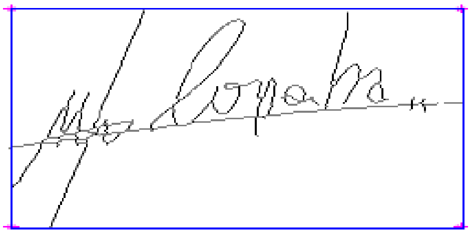


Figure 0.4: Signature image with a bounding box [18]

- **Dilating along all the contours of a signature image**

Generally, signatures signed by the same individual may differ among various trials and there could be gaps between components which may not exist in another trial. And proposed a 3x3 morphological mask to dilate along all the contours of the signature image to compensate for the variations resulting from different trials [19].

Signature with a gap

Signature without gaps

Figure 0.5: Dilating signatures [19]

- **Area filter**

The area filter removes small dots and isolated pixels in the signature images. This process must be done because a signer makes no effort to place the dots included in the signature in the correct places. Usually these dots don't affect global features and needs to be eliminated to stop them from interfering with local features. [11].

Signature with dots

Signature without dots

Figure 0.6: Signature area filter [11]

2.2.2 Feature extraction

The feature extraction process is highly impacting to the success of a signature verification system [6]. A perfect feature extraction method extracts a minimal feature set that increases interpersonal distance among signatures of different individuals while decreasing intrapersonal distance for signatures owned by the same individual [6].

Features extracted for offline handwritten signature verification can be separated into following categories [6].

- Global features
- Local features
- Geometric features

2.2.2.1 Global features

In global feature extraction, the signature is treated as a complete image. Therefore, features are extracted from every pixel containing in the image. And various kinds of global features are extracted based on the style of the signature. The followings are some of the global features. [6]

1. Signature area
2. Signature height-to-width ratio
3. Signature height
4. Pure width and height
5. Number of closed loops
6. Baseline Slant Angle
7. Horizontal and vertical center of the signature
8. Maximum horizontal histogram and maximum vertical histogram
9. Edge point numbers of the signature
10. Vertical projection peaks

2.2.2.2 Local features

These features are extracted from a particular region of the image and assigned to an element gained once signature image segmentation and the features are computed to define the geometrical and topological features of local segments. Like local pixel density or slant, local features are usually derived from the distribution of pixels of a signature image. Furthermore, these features are highly responsive to noise inside the particular region under attention, but not affected by any other region of the signature. Considerably accurate than global features, though computationally complex [6].

Some of the Local features are as follows [6].

1. Number of black pixels
2. Length ratio of the two consecutive parts
3. Corner line features
4. Slant angle of the element
5. Unballistic motion and tremor information in stroke segments
6. Stroke elements
7. Local shape descriptors
8. Position relation between the global and local baseline
9. Upper central line features
10. Pressure and slant features

2.2.2.3 Geometric features

Geometric features define the characteristics geometry and topology of a signature image and reserve their global and local properties. These features can tolerate with distortions, degree of translation, rotation variations and style variations [14].

2.2.3 Verification

Verification is the phase where the genuineness of the test signatures is assessed by matching the extracted features against those stored in the database. The verification phase generates a feedback that states the genuineness of the test signature [17].

Following are the most popular verification approaches available [3][6][17].

- Template Matching Approach
    - Euclidean Distance Based Signature Model
- Statistical Approach
    - Hidden Markov Model (HMM)
    - Neural Network (NN)
    - Support Vector Machine (SVM)
- Structural / Syntactic Approach

2.2.3.1 Template Matching Approach

This technique is a pattern comparison process which can be used for offline handwritten signature verification [3]. Matching is a basic process to find out the similarities among two objects in pattern recognition [22]. A pattern class is denoted by a template and a template pattern is any curve or image. [23].

The Euclidean Distance Based Signature Model and Dynamic Time Warping (DTW) are the common Template Matching approaches used. The DTW can be used for online handwritten signature verification and Euclidean Distance Based Signature Model can be used for both online and offline signature verification system [17] [24].

- **Euclidean Distance Based Signature Model**

In this approach, first some predefined features are extracted from signatures of several individuals. Then calculate mean signature features for each individual by using the features extracted. Then the features of a signature, which needs to be verified are extracted to compute the Euclidean distance with regard to the mean signature features of the genuine signatures. Then the lowest and highest Euclidean distance values of genuine signatures are used to established the acceptance range and if the Euclidean distance of the signature which needs to be verified is accepted if it's within the acceptance range and is rejected if not [25].

Following are the advantages and disadvantages of the Template Matching approach [6] [26].

Advantage	**Disadvantages**
Simplest template matching approach.	Fail if the patterns are inaccurate due to the imaging process or large intra-class differences between the signatures.
Takes a less amount of time to verify the signatures.	Not appropriate for verification among skilled forgeries and genuine signatures.
Fewer requirements needed for the verification.	
Can detect casual forgeries.	

Table 0.1: Advantages and disadvantages of the Template Matching approach

- **Similar Systems**

In 2009 Schafer and Viriri proposed a system based on simple geometric features and the Euclidean Distance Based verification approach. For each individual, a centroid feature vector was calculated using the extracted features of the signer's genuine examples. They have tested the system using a database of 24 genuine signatures and 39 forgeries per writer from 39 writers. They have obtained a FAR of 18.5% and an FRR of 27% [27].

In 2006 Majhi, Reddy and Babu proposed an offline handwritten signature verification system based on a novel feature extraction scheme. The proposed technique used the geometric center of the signatures to extract features to reduce intra-personal variations. Signature features based on the geometric features were used and the Euclidean Distance Model was used for the signature classification. They have tested the system using a database of 30 forgeries and 21 genuine signatures per person. They have obtained an FRR of 14.58% and FARs of 16.36%, 9.75%, and 2.08% for skilled, simple, and random forgeries respectively [28].

2.2.3.2 Statistical Approach

Statistical knowledge easily allows to find out the relation, deviation, etc. among two or more datasets. To discover the relation among dataset items, usually the Correlation Coefficient concept is used [3]. The same technique can be used to validate a submitted signature with the support of an average signature gained from a dataset of previously collected signatures and by finding out the amount of deviation among them [29]. Based on some predetermined deviation, the choice of accepting or rejecting the submitted signature should be made [30].

Hidden Markov Model, Neural Network and Support Vector Machine are some of the Statistical approaches which are commonly used [6] [17].

□ **Hidden Markov Model (HMM)**

In signature verification, HMM is the most commonly used models for sequence analysis. Since these models have the ability to grip the unpredictability among patterns and their similarities, HMM models are stochastic [3].

In HMM stochastic matching is including and it's complete by phases of the probability distribution of features related to the signatures or the probability of how the genuine signature is computed. If the outcomes display a greater probability than the probability of the test signature, then the signature gets accepted, otherwise the signature gets rejected [3].

Following are some of the advantages and disadvantages of the Hidden Markov Model [17][22][31].

Advantage	**Disadvantages**
Ability to outperform human verifiers.	Needs huge training sets and computational cost.
Good for simple and random forgeries.	Low accuracy for skilled forgeries

Table 0.2: Advantages and disadvantages of the Hidden Markov Model

- **Similar Systems**

In 2001 Justino, Bortolozzi, and Sabourin have presented an offline handwritten signature verification method considering different forgery types in a Hidden Markov Model (HMM) framework and have used both static and pseudo dynamic features for the system. They have tested the system using a database with 40 signatures per writer using 100 writers of genuine signatures and 1200 forged signatures (casual, skilled, and random forgeries). They have obtained the FAR of 1.44%, FRR of 2.83%, 2.50%, and 22.67% for random, simple, and skilled forgeries correspondingly [32].

In 2004 Coetzer, Herbst, and Preez presented an offline handwritten signature using the discrete Radon Transform (DRT) and HMM by considering only global features. They have evaluated the system using a dataset of 6 random forgeries, 6 skilled forgeries and 20 genuine signatures per writer. They have gained an EER of 4.5%, and an EER of 18% for random and Skilled forgeries correspondingly [13].

□ **Neural Network (NN)**

The NN is a parallel computing system that contains a huge number of simple processors with various interconnections and capable of learn complex non-linear input-output relationships, use sequential training actions and adjust itself to the data.

The key reason for the common usage of NNs in pattern recognition is their power which comes from the modern methods have in NNs which are capable of building quite complex functions and due to the easiness of use [3].

The NN can be presented with test signatures that can be classified as belonging to a particular signer once the relationship has been learned [3].

Followings are the advantages and disadvantages of Neural Network approach [14][17] [22][31].

Advantage	**Disadvantages**
When want to insert a set of signatures to the system, it is only required to train three new small NNs and not the whole NN.	Requires large amounts of learning data.
Shows acceptable outcome for skilled and random forgeries.	

Table 0.3: Advantages and disadvantages of Neural Network approach

- **Similar Systems**

In 2012 Pansare and Bhatia presented a system using image processing techniques, geometric feature extraction, and NN training with extracted features and verification. The verification phase of their system contains using the extracted features of the test signatures to a trained NN, which classifies them as genuine or not. They have tested the system using a database of 24 genuine signatures, 24 forgeries per writer from 30 writers. They have obtained a FAR of 14.66%, and the FRR of 20% [18].

In 1997 Huang and Yan presented a system based on a NN classification technique. In their system, geometric features of submitted signature images are examined at the same time under some scales by a NN classifier and a complete match rating is produced by merging the results at each scale. They have tested the system using 504 genuine signatures and 3024 of forged. They have obtained a FAR of 11.8% and the FRR of 11.1% [9].

- **Support Vector Machine (SVM)**

SVMs use a high dimensional feature space and guess variances among classes of given data to generalize unobserved data [14]. Basically, SVM defined for separating linearly two classes. A kernel function is used as polynomial function, radial basis function (RBF) or multilayer perceptron, when data are none linearly separable. The signature verification based on SVM includes training and testing phases. The training phase involves finding the optimal parameters and they are found experimentally based on the dataset [33].

A decision rule needs to be produced on the results of the SVMs where the values are either positive or negative, to predict whether a signature is genuine or forged [33]. In generally, there is no ordinary formula for the membership degree and the only concern is that it should be restricted in the range of 1, 0 where SVM generate one result [34].

Followings are some of the advantages and disadvantages of the Support Vector Machine approach [23] [26].

Advantage	**Disadvantages**
Better performance even when amount of data is limited.	Consumes a large amount of time to train and test an SVM.
Suitable for simple, skilled, and random forgeries.	

Table 0.4: Advantages and disadvantages of the Support Vector Machine approach

- **Similar Systems**

In 2011 Vargas, Ferrer and Travieso have proposed a way for offline handwritten signature verification based on grey level information using texture features. They say that the local binary patterns and co-occurrence matrix are examined and used as features in their system. They further say that their system begins with by a background removal of the signatures and a histogram is also handled to decrease the impact of various writing ink pens used. The genuine and forged signature examples used to train the SVM model and both random and skilled forgeries used to test the system. They have used 2 databases to test the system (MCYT-75 database– 15 genuine signature & 15 forgeries per user from 75 signers and GPDS-100 database– 24 genuine signatures & 24 forgeries of 100 individuals). They have obtained 9.02% of an EER with GPDS-100 database and 8.80% of an EER with MCYT-75 database [7].

In 2006 Audet, Bansal and Baskaran proposed a system using SVM by using global, directional, and grid features of signatures. Their system used a virtual SVM to validate and classify the signatures and skilled forgeries have been used to test the system. They have tested the system using a database of 24 genuine signatures and 20 forgeries per writer of 160 writers. They have obtained a FAR of 16.00%, and the FRR of 13% [35].

2.2.3.3 Structural / Syntactic Approach

In these techniques, a pattern is observed as being composed of simple sub-patterns which are constructed from simpler sub-patterns [24]. Also, structural/syntactic pattern recognition is the representation of patterns by means of symbolic data structures like strings, trees, and graphs [3].Moreover, this approach is grounded on the relational organization of the low-level to higher-level features structures, and then matching these structures with models stored in the database [23]. The MDF uses the location of transitions of the boundary representation of an entity [39].

Followings are the advantages and disadvantages of the Structural/Syntactic approach [6][14] [37] [38].

Advantage	**Disadvantages**
High accuracy when detecting skilled forgeries.	Requires large training sets which leads to extensive computational time.
Useful when the signature is treated as a complete entity.	
Defines a description of the given pattern	

Table 0.5: Advantages and disadvantages of the Structural/Syntactic approach

- □ **Similar Systems**

In 2009 Raja, Ramachandra, Ravi, Patnaik and Venugopal presented a system by following the structural approach and used graph matching and cross-validation principle algorithms. Identical measures between signatures are determined by using Bipartite graph and complete matching techniques. They have tested the system using a database of 21 genuine signatures, 24 random forgeries, and 30 skilled forgeries per writer of 5 individuals. They have obtained an EER of 29.00% and 15.00% for skilled and random forgeries [39].

In 2007 Abuhaiba implemented a system by following the structural approach and used graph matching techniques. Their system based only on the raw binary pixel intensities of the signatures and has avoided the use of a complex set of signature features. They have tested it using a database of 12 genuine signatures, 60 random forgeries, and 15 skilled forgeries per writer of 5 individuals. They have obtained an EER of 26.7% and 5.6% for skilled forgeries and random forgeries [40].

2.3 Measures of performance evaluation

To evaluate the performance of a signature verification system, the following measures will be used [38].

- False Rejection Rate (FRR)
- False Acceptance Rate (FAR)
- Average Error Rate (AER)
- Equal Error Rate (EER)

❖ **False Rejection Rate (FRR)**

False Rejection is that a genuine signature gets rejected as a forged signature [41]. FRR is the ratio of genuine test signatures rejected to the total number of genuine test signatures observed [38].

$$\mathrm{FRR} = \frac{\text{Total number of genuine signatures rejected}}{\text{Total number of genuine signatures observed}} \times 100$$

❖ **False Acceptance Rate (FAR)**

False Acceptance is that a forged signature is accepted as a genuine signature [41]. FAR is the ratio of the number of forgeries accepted to the number of forgeries observed [38].

$$\mathrm{FAR} = \frac{\text{Total number of forged signatures accepted}}{\text{Total number of forged signatures observed}} \times 100$$

❖ **Average Error Rate (AER)**

AER is the average of FAR and FRR [13]

❖ **Equal Error Rate (EER)**

EER is a value where FAR and FRR is equal. [13] [38] [40].

2.4 Performances of similar systems

Approach		Authors	Signature Database	Performance
Template Matching	Euclidean Distance	Schafer & Viriri (2009)	24 - genuine 39 - *forgeries* per signer from 39 signers	FAR 18.5% FRR 27%
		Majhi, Reddy & Babu (2006)	21 - genuine 30 - forgeries per signer	FRR 14.58% FAR 16.36% (skilled) FAR 9.75% (simple) FAR 2.08% (random)
Statistical Approach	Hidden Markov Model	Justino, Bortolozzi & Sabourin (2001)	40 - genuine 12 - forgeries Per signer from 100 signers	FRR 2.83% FAR 1.44% (random) FAR 2.50% (casual) FAR 22.67% (skilled)
		Coetzer, Herbst & Preez (2004)	20 - genuine 6 -skilled 6- casual per signer	EER 18% (skilled) EER 4.5% (casual)
	Neural Network	Pansare & Bhatia (2012)	24 - genuine 24 - forgeries per signer from 30 signers	FAR 14.66% FRR 20%
		Huang & Yan (1997)	504 - genuine 3024 - forgeries (random, skilled)	FAR 11.8% FRR 11.1%
	Support Vector Machine	Vargas, Ferrer, Travieso & Alonso (2011)	15 - genuine 15 - forgeries per signer from 75 signers	EER 8.80% (skilled)
			24 - genuine 24 - forgeries per signer from 100 signers	EER 9.02% (skilled)
		Audet, Bansal & Baskaran (2006)	24 - genuine 20 - forgeries per signer from 160 signers	FAR 16.00% FRR 13%
Structural/Syntactic		Ramachandra, Ravi, Raja, Venugopal & Patnaik (2009)	21-genuine 24 -random 30 - skilled per signer from 5 signers	EER 29.00% (skilled) EER 15.00% (random)
		Abuhaiba (2007)	12 - genuine 60 - random 15- skilled per signer from 5 signers	EER 26.7% (skilled) EER 5.6% (random)

Table 0.6: Performances of similar systems

2.5 Research Gap

According to the literature survey, identified that there is no research performed to verify Sinhala and Tamil signatures with an offline handwritten signature verification system. Even there is no system relevant to verify Sinhala or Tamil signatures.

Since the project scope is to analyze Sinhala, Tamil, and English signatures in the Sri Lankan context using an offline handwritten signature verification system, there are some gaps between the proposed solution and currently existing systems.

CRITICAL ANALYSIS

3.1 Research paper analysis

In the previous section, it was discussed a wide variety of offline handwritten signature verification systems proposed and designed by various authors. It was identified that; generally offline handwritten signature verification systems are categorized according to the verification approaches which are used in the systems. Similar systems covering all the popular verification approaches were discussed along with their performances. Having studied the advantages and disadvantages of all the verification approaches, it was realized that Euclidean Distance Based Signature Model and Hidden Markov Model verification approaches are not suitable to detect skilled forgeries. So, the offline handwritten signature verification systems based on these two signature verification methods will not be evaluated since the project scope is to identify the skilled forgeries in the system to be implemented.

In 2013 Kovari, analyzed the surveys covering more than 500 papers from the last 30 years and the latest developments. And he presented that in 2011 Vargas, Ferrer, Travieso and Alonso achieved the best performance for offline signature verification systems when skilled forgeries take place by their system based on the SVM approach [7] [16].

Having studied the performances when skilled forgeries take place on the systems based on the NN, SVM and Structural/Syntactic verification approaches, it was identified that the system proposed by Vargas, Ferrer, Travieso and Alonso in 2011 using the Support Vector Machine approach, achieved the best performance among the other systems which were discussed and reached to the same decision which Kovari made in 2013.

3.2 Trial and error analysis

When selecting algorithms there are a lot of rules of thumb.

E.g. If an algorithm looking for data of an exact distribution and the data has the expected distribution, then maybe the algorithm is a decent fit for the problem [46].

But there are two issues:

1. Several algorithms might have hopes that make them appropriate for the problem.
2. From time to time good and even better results can be reached when the expectations of an algorithm are violated.

Rules of thumb are great way to start, but it is not the optimal version of the algorithm. The best version for the problem is found empirically by trial and error [48].

Since the verification phase of the proposed solution is a binary classification approach, the author had to evaluate popular algorithms that can be used for binary classification. So, in this section it elaborates the reasons which have been taken for select the appropriate classification algorithm by conducting a trial and error evaluation on below mentioned classification algorithms using the Weka experiment environment.

1. k-Nearest Neighbors
2. Decision Trees
3. Support Vector Machine
4. Naive Bayes
5. Logistic Regression

3.2.1 Compare Algorithm Performance in Weka

The Weka Experiment Environment allows to create, run, modify, and analyze experiments in a more convenient manner. Therefore, the author created an experiment that runs different schemes of above-mentioned binary classification algorithms against a sequence of data and then evaluated the results to find which one of the schemes is best.

Initially, each algorithm has been evaluated using the default algorithm configurations. And later on, added more variations of each algorithm with common or standard algorithm configurations by tuning different hyperparameters. Different hyperparameter values proposed by machine learning practitioners were used for related algorithms.

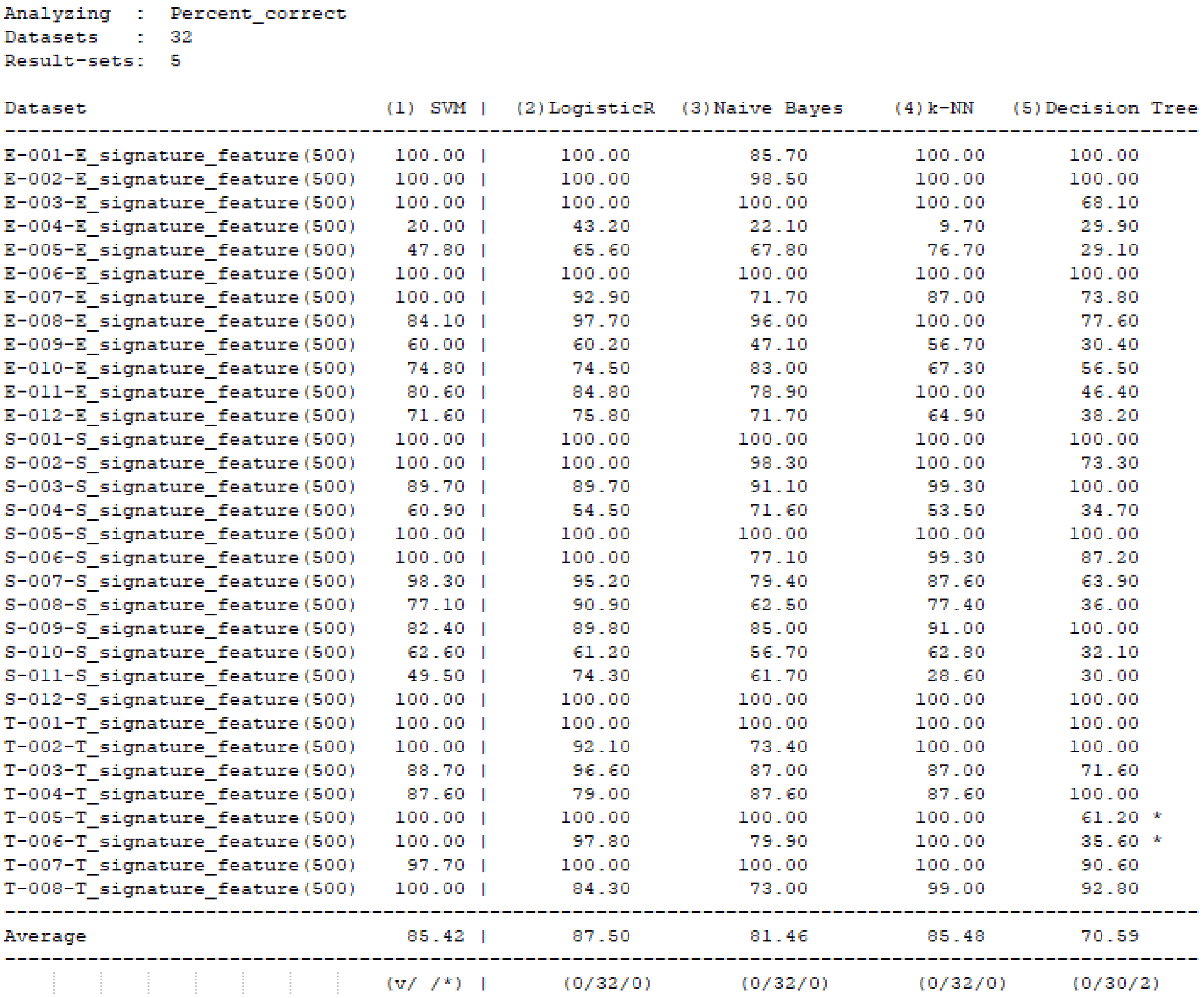
Since research paper analysis suggested that the SVM algorithm has good classification performance, it was used as the baseline algorithm for the initial experiment. Hence, selecting the SVM algorithm caused the other algorithms to be compared individually to the SVM algorithm.

The experiment performed 5-fold cross-validation using 32 datasets which contain signatures belongs to 32 individuals (4 genuine signatures and 4 forged signatures for each dataset).

As a result of initial experiment 80000 records were loaded. This is because the 5 algorithms that were each evaluated 500 times, 5-fold cross validation multiplied by 100 repeats for 32 datasets (No. of algorithms × No. of folds × No. of repeats × No. of datasets).

In here the process which did up to now, results were collected for several performance measures, like accuracy of classification. The Weka experiment environment has used to take decisions from the experiment by performing statistical tests on the several performance measures that. From here onwards focused on two factors. First is “Which algorithm has the best performance, after the evaluation in the experiment?” This was beneficial when needed to generate a good performance model instantly. And second is “what is the rank of algorithms by performance?” This was beneficial when needed more investigate and tune the two to three algorithms that performed the best on the problem [48].

Therefore, first compared each and every algorithm result to one base result, which in this case is Support vector machine. Followings are the results of experiment.



```
Analyzing  :  Percent_correct
Datasets   :  32
Result-sets:  5

Dataset                          (1) SVM |  (2)LogisticR  (3)Naive Bayes    (4)k-NN   (5)Decision Tree
------------------------------------------------------------------------------------------------------
E-001-E_signature_feature(500)    100.00 |      100.00          85.70        100.00       100.00
E-002-E_signature_feature(500)    100.00 |      100.00          98.50        100.00       100.00
E-003-E_signature_feature(500)    100.00 |      100.00         100.00        100.00        68.10
E-004-E_signature_feature(500)     20.00 |       43.20          22.10          9.70        29.90
E-005-E_signature_feature(500)     47.80 |       65.60          67.80         76.70        29.10
E-006-E_signature_feature(500)    100.00 |      100.00         100.00        100.00       100.00
E-007-E_signature_feature(500)    100.00 |       92.90          71.70         87.00        73.80
E-008-E_signature_feature(500)     84.10 |       97.70          96.00        100.00        77.60
E-009-E_signature_feature(500)     60.00 |       60.20          47.10         56.70        30.40
E-010-E_signature_feature(500)     74.80 |       74.50          83.00         67.30        56.50
E-011-E_signature_feature(500)     80.60 |       84.80          78.90        100.00        46.40
E-012-E_signature_feature(500)     71.60 |       75.80          71.70         64.90        38.20
S-001-S_signature_feature(500)    100.00 |      100.00         100.00        100.00       100.00
S-002-S_signature_feature(500)    100.00 |      100.00          98.30        100.00        73.30
S-003-S_signature_feature(500)     89.70 |       89.70          91.10         99.30       100.00
S-004-S_signature_feature(500)     60.90 |       54.50          71.60         53.50        34.70
S-005-S_signature_feature(500)    100.00 |      100.00         100.00        100.00       100.00
S-006-S_signature_feature(500)    100.00 |      100.00          77.10         99.30        87.20
S-007-S_signature_feature(500)     98.30 |       95.20          79.40         87.60        63.90
S-008-S_signature_feature(500)     77.10 |       90.90          62.50         77.40        36.00
S-009-S_signature_feature(500)     82.40 |       89.80          85.00         91.00       100.00
S-010-S_signature_feature(500)     62.60 |       61.20          56.70         62.80        32.10
S-011-S_signature_feature(500)     49.50 |       74.30          61.70         28.60        30.00
S-012-S_signature_feature(500)    100.00 |      100.00         100.00        100.00       100.00
T-001-T_signature_feature(500)    100.00 |      100.00         100.00        100.00       100.00
T-002-T_signature_feature(500)    100.00 |       92.10          73.40        100.00       100.00
T-003-T_signature_feature(500)     88.70 |       96.60          87.00         87.00        71.60
T-004-T_signature_feature(500)     87.60 |       79.00          87.60         87.60       100.00
T-005-T_signature_feature(500)    100.00 |      100.00         100.00        100.00        61.20 *
T-006-T_signature_feature(500)    100.00 |       97.80          79.90        100.00        35.60 *
T-007-T_signature_feature(500)     97.70 |      100.00         100.00        100.00        90.60
T-008-T_signature_feature(500)    100.00 |       84.30          73.00         99.00        92.80
------------------------------------------------------------------------------------------------------
Average                            85.42 |       87.50          81.46         85.48        70.59
------------------------------------------------------------------------------------------------------
                                 (v/ /*) |     (0/32/0)       (0/32/0)      (0/32/0)     (0/30/2)
```

Figure 0.1: Classification accuracy of experiment 1

As shown in the above [Figure 12] that SVM, the base for comparison marked as (1) has the average accuracy of 85.42% on the problem. This result is calculated by comparing to the other 4 algorithms for all 32 datasets.

Note the "*" next to the (5) Decision Tree results for few datasets. This shows that the outcomes are significantly different from the SVM, but the scores are lower. (2) k-NN, (4) Logistic Regression do not have any character next to their results in the table, showing that the outcomes are not significantly different from SVM, but results are larger than the SVM algorithm (If any algorithm has results larger than the base algorithm and the variance was significant, a **lowercase 'v'** would show next to the outcomes [48].) If need to build a model directly with this outcomes, Logistic Regression is the best option here, but anyone might pick k-NN, Naive Bayes or SMV as their outcomes were not significantly different.

3.2.1.1 Calculate Precision, Recall, and F1 score (F-Measure)

The number of correct predictions divided by the number of predictions made for a data sample is called classification accuracy. For imbalanced classification problems accuracy is an unsuitable performance measure. The reason is the large number of data instances from the majority class will beat the number of data instances in the minority class, sense that even unskilled models can reach accuracy rate of above 90%. Therefore, use precision and recall metrics as an alternative to classification accuracy [49].

The precision and recall system of measurement is formed based on the cells in the confusion matrix. The confusion matrix offers additional vision into not only the performance of a predictive model, but also which classes are being predicted correctly or incorrectly, and what sort of errors are being made [49].

	Positive Prediction	Negative Prediction
Positive Class	True Positive (TP)	False Negative (FN)
Negative Class	False Positive (FP)	True Negative (TN)

- **Precision for Binary Classification**

The precision is calculated as in following manner.

Precision = TruePositives / (TruePositives + FalsePositives)

The outcome is a value in the range of 0.0 for no precision and 1.0 for full or perfect precision [49].

- □ **Recall for Binary Classification**

The recall is calculated as in following manner.

Recall = TruePositives / (TruePositives + FalseNegatives)

The result is a value in the range of 0.0 for no recall and 1.0 for full or perfect recall [49].

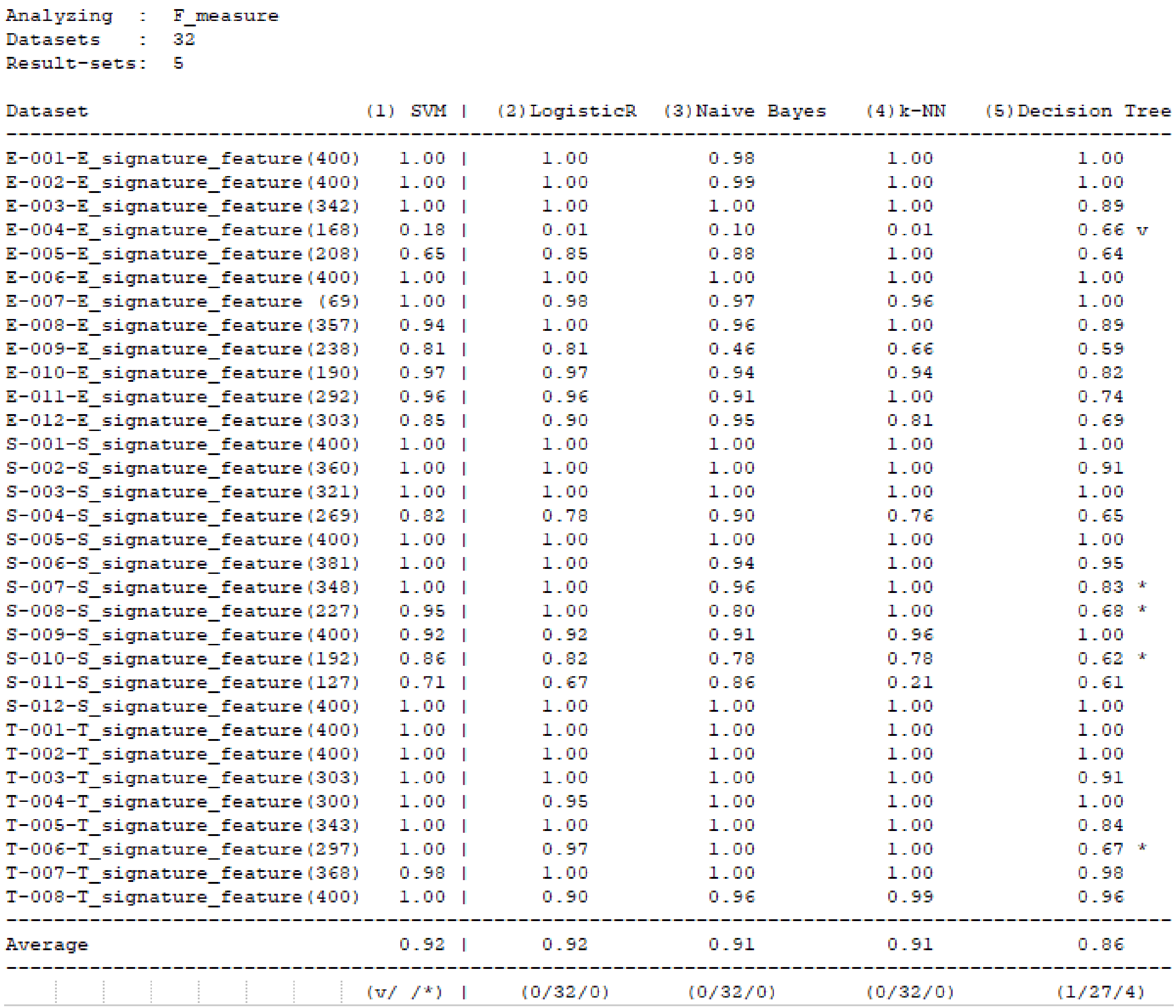
- □ **F-Measure for Binary Classification**

Classification accuracy is commonly accepted measure used to evaluate model performance. More importantly F-Measure offers a way to link precision and recall into a one measure that captures both properties. It’s not telling the entire story with separate precision or recall. It could be an excellent precision with poor recall, or alternately, poor precision with excellent recall. The F-Measure is calculated as in following manner:

F-Measure = (2 * Precision * Recall) / (Precision + Recall)

A poor F-Measure score is 0.0 and a best or perfect F-Measure score is 1.0 [49].

In the Weka experiment environment, there is the possibility of calculating F-Measure. Therefore, F-Measure was calculated for all the five algorithms.



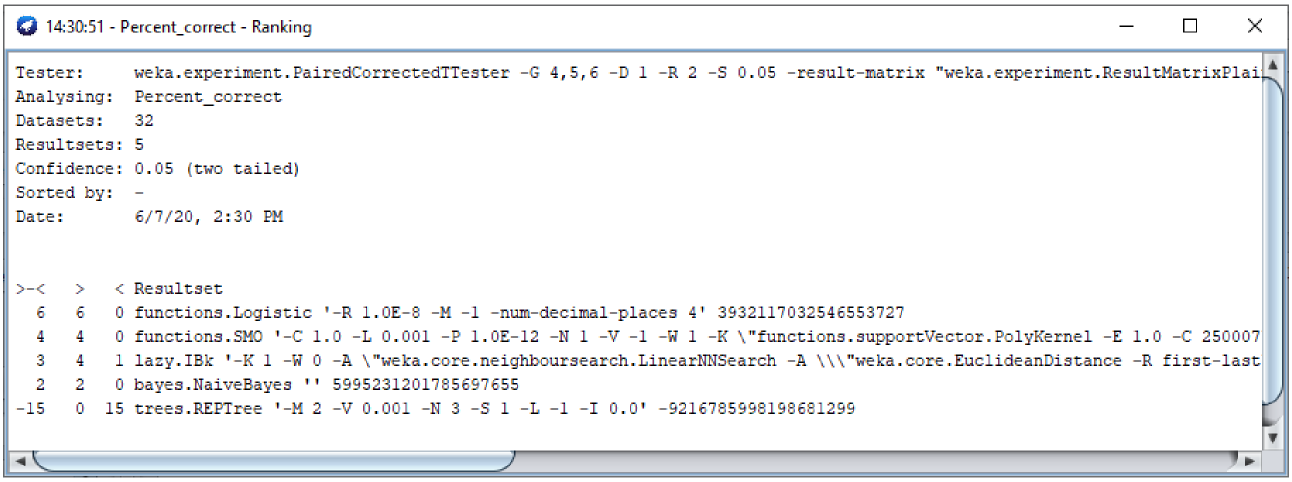
```
Analyzing  :  F_measure
Datasets   :  32
Result-sets:  5

Dataset                        (1)  SVM |  (2)LogisticR  (3)Naive Bayes   (4)k-NN   (5)Decision Tree
---------------------------------------------------------------------------------------------------
E-001-E_signature_feature(400)   1.00 |      1.00           0.98           1.00            1.00
E-002-E_signature_feature(400)   1.00 |      1.00           0.99           1.00            1.00
E-003-E_signature_feature(342)   1.00 |      1.00           1.00           1.00            0.89
E-004-E_signature_feature(168)   0.18 |      0.01           0.10           0.01            0.66 v
E-005-E_signature_feature(208)   0.65 |      0.85           0.88           1.00            0.64
E-006-E_signature_feature(400)   1.00 |      1.00           1.00           1.00            1.00
E-007-E_signature_feature (69)   1.00 |      0.98           0.97           0.96            1.00
E-008-E_signature_feature(357)   0.94 |      1.00           0.96           1.00            0.89
E-009-E_signature_feature(238)   0.81 |      0.81           0.46           0.66            0.59
E-010-E_signature_feature(190)   0.97 |      0.97           0.94           0.94            0.82
E-011-E_signature_feature(292)   0.96 |      0.96           0.91           1.00            0.74
E-012-E_signature_feature(303)   0.85 |      0.90           0.95           0.81            0.69
S-001-S_signature_feature(400)   1.00 |      1.00           1.00           1.00            1.00
S-002-S_signature_feature(360)   1.00 |      1.00           1.00           1.00            0.91
S-003-S_signature_feature(321)   1.00 |      1.00           1.00           1.00            1.00
S-004-S_signature_feature(269)   0.82 |      0.78           0.90           0.76            0.65
S-005-S_signature_feature(400)   1.00 |      1.00           1.00           1.00            1.00
S-006-S_signature_feature(381)   1.00 |      1.00           0.94           1.00            0.95
S-007-S_signature_feature(348)   1.00 |      1.00           0.96           1.00            0.83 *
S-008-S_signature_feature(227)   0.95 |      1.00           0.80           1.00            0.68 *
S-009-S_signature_feature(400)   0.92 |      0.92           0.91           0.96            1.00
S-010-S_signature_feature(192)   0.86 |      0.82           0.78           0.78            0.62 *
S-011-S_signature_feature(127)   0.71 |      0.67           0.86           0.21            0.61
S-012-S_signature_feature(400)   1.00 |      1.00           1.00           1.00            1.00
T-001-T_signature_feature(400)   1.00 |      1.00           1.00           1.00            1.00
T-002-T_signature_feature(400)   1.00 |      1.00           1.00           1.00            1.00
T-003-T_signature_feature(303)   1.00 |      1.00           1.00           1.00            0.91
T-004-T_signature_feature(300)   1.00 |      0.95           1.00           1.00            1.00
T-005-T_signature_feature(343)   1.00 |      1.00           1.00           1.00            0.84
T-006-T_signature_feature(297)   1.00 |      0.97           1.00           1.00            0.67 *
T-007-T_signature_feature(368)   0.98 |      1.00           1.00           1.00            0.98
T-008-T_signature_feature(400)   1.00 |      0.90           0.96           0.99            0.96
---------------------------------------------------------------------------------------------------
Average                          0.92 |      0.92           0.91           0.91            0.86
---------------------------------------------------------------------------------------------------
                               (v/ /*) |    (0/32/0)      (0/32/0)       (0/32/0)        (1/27/4)
```

Figure 0.2: F-Measure for experiment 1

As shown in the above [Figure 13] SVM and Logistic Regression has average F-Measure value of 0.92, while k-NN and Naïve Bayes has average F-Measure value of 0.91.

The second factor which was wanting to know was which algorithm was the best. In the Weka experiment environment, there is the possibility of ranking the algorithms by the number of times a given algorithm beat the other algorithms.



```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -result-matrix "weka.experiment.ResultMatrixPlai
Analysing:  Percent_correct
Datasets:   32
Resultsets: 5
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       6/7/20, 2:30 PM


>-<   >   < Resultset
  6   6   0 functions.Logistic '-R 1.0E-8 -M -1 -num-decimal-places 4' 3932117032546553727
  4   4   0 functions.SMO '-C 1.0 -L 0.001 -P 1.0E-12 -N 1 -V -1 -W 1 -K \"functions.supportVector.PolyKernel -E 1.0 -C 250007
  3   4   1 lazy.IBk '-K 1 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.EuclideanDistance -R first-last
  2   2   0 bayes.NaiveBayes '' 5995231201785697655
-15   0  15 trees.REPTree '-M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0' -9216785998198681299
```

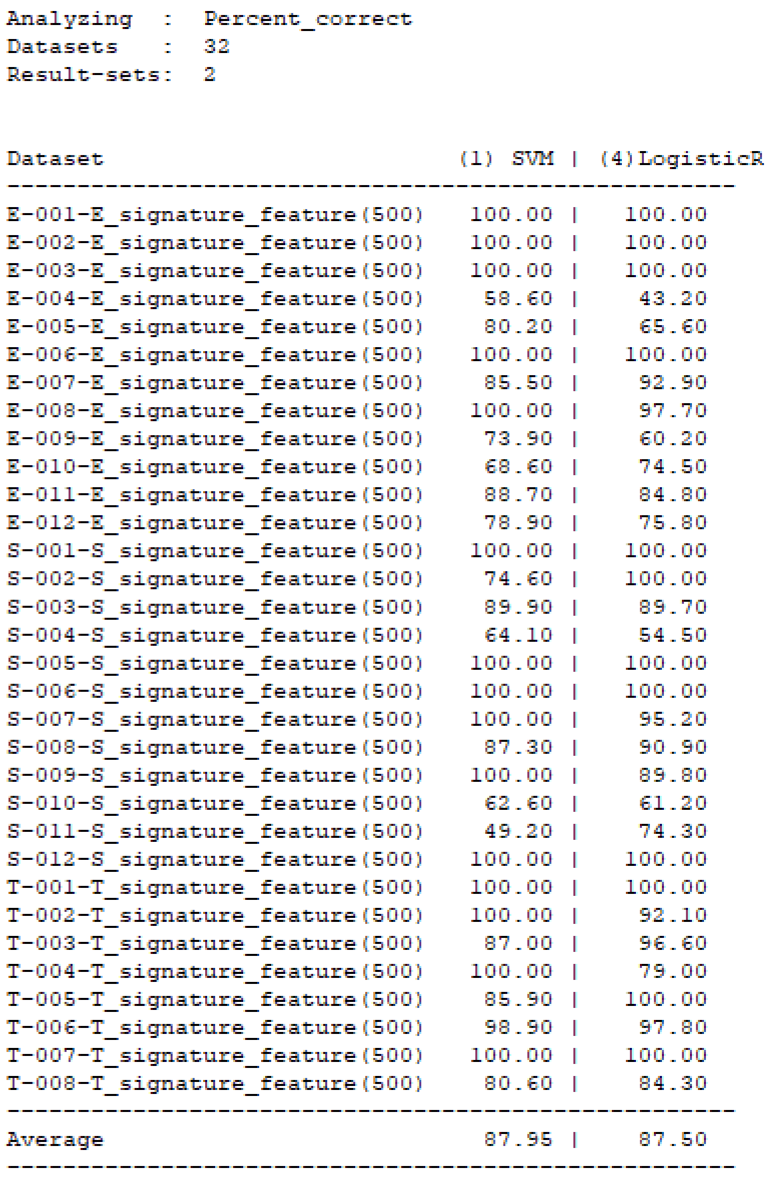
Figure 0.3: Ranking of classification accuracy in experiment 1

The above figure [Figure 14] displays the number of statistically important wins each algorithm has won against all other algorithms. As shown in the above figure that Logistic regression has six wins and no losses, while SVM has four wins and no losses. And also, k-NN has 4 wins and one lose. While Naïve Bayes has two wins. But Decision tree has 15 losses. And therefore, Decision Tree was eliminated from further experiments.

Since Logistic regression and SVM algorithms has good accuracy, F-Measure values and good ranking based on the performance, only these two algorithms were evaluated in the second round of experiments. The Logistic regression and SVM algorithms were evaluated using more variations of each algorithm with common or standard algorithm configurations tried by tuning hyperparameters. Different hyperparameter values suggested by machine learning practitioners were used for related algorithms.

Therefore, done several iterations of experiments with different configurations for each algorithm by tuning hyperparameters. And finally found very good rates for both the algorithms.

Followings are the final iteration's results of the "**Percent_correct**" metric (accuracy) of the round two of experiments.



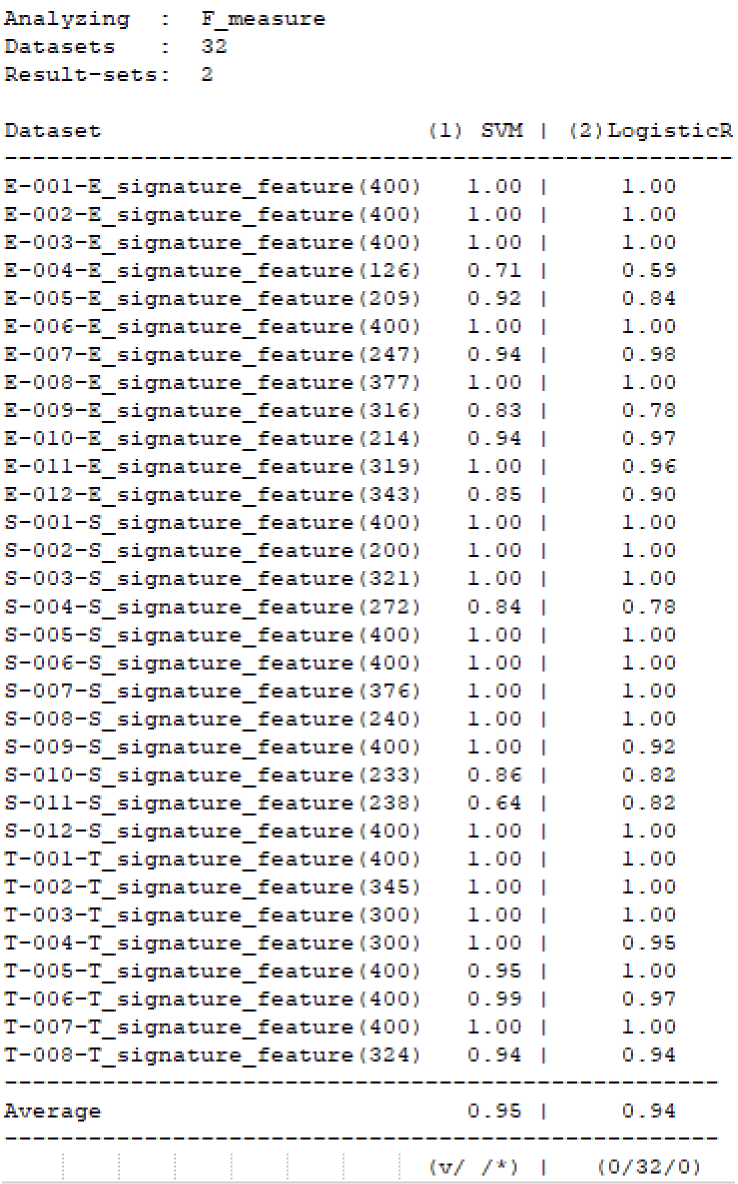
```
Analyzing  :  Percent_correct
Datasets   :  32
Result-sets:  2


Dataset                          (1) SVM | (4)LogisticR
----------------------------------------------------
E-001-E_signature_feature(500)   100.00 |   100.00
E-002-E_signature_feature(500)   100.00 |   100.00
E-003-E_signature_feature(500)   100.00 |   100.00
E-004-E_signature_feature(500)    58.60 |    43.20
E-005-E_signature_feature(500)    80.20 |    65.60
E-006-E_signature_feature(500)   100.00 |   100.00
E-007-E_signature_feature(500)    85.50 |    92.90
E-008-E_signature_feature(500)   100.00 |    97.70
E-009-E_signature_feature(500)    73.90 |    60.20
E-010-E_signature_feature(500)    68.60 |    74.50
E-011-E_signature_feature(500)    88.70 |    84.80
E-012-E_signature_feature(500)    78.90 |    75.80
S-001-S_signature_feature(500)   100.00 |   100.00
S-002-S_signature_feature(500)    74.60 |   100.00
S-003-S_signature_feature(500)    89.90 |    89.70
S-004-S_signature_feature(500)    64.10 |    54.50
S-005-S_signature_feature(500)   100.00 |   100.00
S-006-S_signature_feature(500)   100.00 |   100.00
S-007-S_signature_feature(500)   100.00 |    95.20
S-008-S_signature_feature(500)    87.30 |    90.90
S-009-S_signature_feature(500)   100.00 |    89.80
S-010-S_signature_feature(500)    62.60 |    61.20
S-011-S_signature_feature(500)    49.20 |    74.30
S-012-S_signature_feature(500)   100.00 |   100.00
T-001-T_signature_feature(500)   100.00 |   100.00
T-002-T_signature_feature(500)   100.00 |    92.10
T-003-T_signature_feature(500)    87.00 |    96.60
T-004-T_signature_feature(500)   100.00 |    79.00
T-005-T_signature_feature(500)    85.90 |   100.00
T-006-T_signature_feature(500)    98.90 |    97.80
T-007-T_signature_feature(500)   100.00 |   100.00
T-008-T_signature_feature(500)    80.60 |    84.30
----------------------------------------------------
Average                           87.95 |    87.50
----------------------------------------------------
```

Figure 0.4: Classification accuracy of experiment 2

As shown in the above figure that SVM, the base for comparison marked as (1) has the average accuracy of 87.95% on the problem and Logistic regression has the average accuracy of 87.50% on the problem.

Followings are the final iteration's results of the F1 score (F-Measure) metric of the round two of experiments.



```
Analyzing  :  F_measure
Datasets   :  32
Result-sets:  2

Dataset                          (1) SVM |  (2)LogisticR
--------------------------------------------------------
E-001-E_signature_feature(400)   1.00 |      1.00
E-002-E_signature_feature(400)   1.00 |      1.00
E-003-E_signature_feature(400)   1.00 |      1.00
E-004-E_signature_feature(126)   0.71 |      0.59
E-005-E_signature_feature(209)   0.92 |      0.84
E-006-E_signature_feature(400)   1.00 |      1.00
E-007-E_signature_feature(247)   0.94 |      0.98
E-008-E_signature_feature(377)   1.00 |      1.00
E-009-E_signature_feature(316)   0.83 |      0.78
E-010-E_signature_feature(214)   0.94 |      0.97
E-011-E_signature_feature(319)   1.00 |      0.96
E-012-E_signature_feature(343)   0.85 |      0.90
S-001-S_signature_feature(400)   1.00 |      1.00
S-002-S_signature_feature(200)   1.00 |      1.00
S-003-S_signature_feature(321)   1.00 |      1.00
S-004-S_signature_feature(272)   0.84 |      0.78
S-005-S_signature_feature(400)   1.00 |      1.00
S-006-S_signature_feature(400)   1.00 |      1.00
S-007-S_signature_feature(376)   1.00 |      1.00
S-008-S_signature_feature(240)   1.00 |      1.00
S-009-S_signature_feature(400)   1.00 |      0.92
S-010-S_signature_feature(233)   0.86 |      0.82
S-011-S_signature_feature(238)   0.64 |      0.82
S-012-S_signature_feature(400)   1.00 |      1.00
T-001-T_signature_feature(400)   1.00 |      1.00
T-002-T_signature_feature(345)   1.00 |      1.00
T-003-T_signature_feature(300)   1.00 |      1.00
T-004-T_signature_feature(300)   1.00 |      0.95
T-005-T_signature_feature(400)   0.95 |      1.00
T-006-T_signature_feature(400)   0.99 |      0.97
T-007-T_signature_feature(400)   1.00 |      1.00
T-008-T_signature_feature(324)   0.94 |      0.94
------------------------------------------------------
Average                          0.95 |      0.94
------------------------------------------------------
                              (v/ /*) |   (0/32/0)
```

Figure 0.5: F-Measure for experiment 2

As shown in the above figure that SVM, the base for comparison marked as (1) has the average F1 score of 0.95 on the problem and Logistic regression has the average F1 score of 0.94 on the problem.

According to above experiment results it shows both the SVM and Logistic regression are good models to select since it is well understood, simple and fast to train. But considering research paper analysis it shows that SVM approach has good performance when it comparing to other approaches.

Therefore, considering both of the analysis approaches it was safe to conclude and select SVM as the binary classification algorithm which is capable to detect skilled forgeries for the implementation.

PROPOSED SOLUTION

Under the critical evaluation, it was discussed that the algorithm proposed by Vargas, Ferrer, Travieso, and Alonso has been selected as the most suitable method to present an offline handwritten signature verification system which can detect skilled forgeries. Therefore, to address the problem Support Vector Machine approach is proposed by the author. Since this research is based on biometrical data, it is not easy to gather a huge number of signature samples. But using SVM models, it is possible to achieve better generalization performance even if the amount of data is limited as discussed in the literature review. And also, according to trial and error analysis, it is clear that SVM has better generalization performance.

4.1 Methodology

In this chapter, the main four phases of the proposed solution approach will be discussed in detail.

4.1.1 Image acquisition

This is the first phase of the proposed approach. As discussed in a previous section, the image acquisition task will be done by scanning the signature images from the signed papers by using a digital scanner and feeding the scanned images to the system for preprocessing.

4.1.2 Preprocessing

Preprocessing steps ensure that the signature images fed to the system reach to the feature extraction phase according to the appropriate standard which facilitates to accurately extract the signature features required to verify the signatures.

The preprocessing steps proposed by Vargas, Ferrer, Travieso and Alonso in 2011 are as follows and same steps will be used in proposed system.

- Greyscale
- Background removal (segmentation)
- Histogram displacement
- Crop
- Resize
- Interpolation
- Quantification

4.1.2.1 Greyscale

The scanned images will be converted to greyscale images in order to be used for further processing.

4.1.2.2 Background removal (segmentation)

According to (Vargas, Ferrer, Travieso, & Alonso, 2011), the signature features characterize the grey level distribution of the signature images and it is required to remove the background of the image. [7]. In 2011 Vargas, Ferrer, Travieso, and Alonso proposed a posterization procedure to avoid background influence of the signature images.

□ **Posterization procedure**

Let $I(x, y)$ be a 256-level grey scale image and $n_L + 1$ the number of grey levels considered for the posterization.
The posterized image $\boldsymbol{I_P(x,y)}$ is defined as follows.

$$\boldsymbol{I_P(x,y) = round\left(round\left(\frac{I(x,y)n_L}{255}\right)\frac{255}{n_L}\right)}$$

$\boldsymbol{round}$ – rounds up or down the elements to the nearest integers.

The interior $\boldsymbol{round}$ performs the posterization process, and the exterior $\boldsymbol{round}$ ensure that the resulting grey level of $\boldsymbol{I_P(x,y)}$ is an integer.

$\boldsymbol{n_L = 3}$ were used to obtain a 4-grey level posterized image.

In the posterized image the background looks white (255-grey level) and the signature strokes appear darker (0, 85, or 170 grey levels). To obtain black strokes and a white background, the image will be binarized by applying the following thresholding operation.

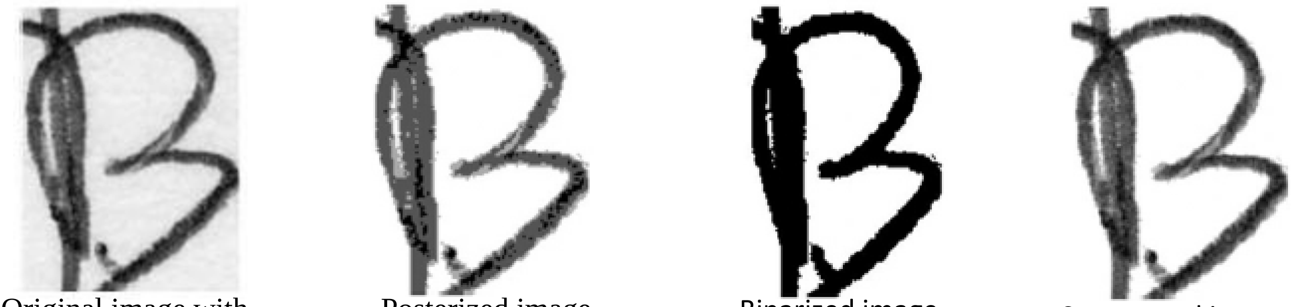
$$\boldsymbol{I_{bw}(x,y) = \begin{cases} 255 & if\ I_P(x,y) = 255 \\ 0 & otherwise \end{cases}}$$

The image with black strokes and white background $\boldsymbol{I_{bw}(x,y)}$ is used as a mask to segment the genuine signature. The segmented signature $\boldsymbol{I_S(x,y)}$ is obtained as follows.

$$\boldsymbol{I_S(x,y) = \begin{cases} 255 & if\ I_{bw}(x,y) = 255 \\ I(x,y) & otherwise \end{cases}}$$

A clear segmentation among background and the foreground is achieved, at this point.

The following image shows outcomes under each step of the above described Posterization procedure.



Original image with 256 grey levels $\boldsymbol{I(x,y)}$

Posterized image with $\boldsymbol{n_L} = \mathbf{3}$: 4 grey levels $\boldsymbol{I_P(x,y)}$

Binarized image $\boldsymbol{I_{bw}(x,y)}$

Segmented image $\boldsymbol{I_S(x,y)}$

Figure 0.1: Posterization procedure [7]

4.1.2.3 Histogram displacement

It is required to minimize the impact of the various writing ink pens on the segmented image. In 2011 Vargas, Ferrer, Travieso and Alonso achieved this by shifting the histogram of the signature image's pixels towards zero and keeping, white (255 grey level) color in the background. This can be done by deducting the minimum grey level in the signature image from the signature pixels as follows [7].

$$\boldsymbol{I_G(x,y)} = \begin{cases} \boldsymbol{I_S(x,y)} & \boldsymbol{if\ I_S(x,y) = 255} \\ \boldsymbol{I_S(x,y) - \min\{I_S(x,y)\}} & \boldsymbol{otherwise} \end{cases}$$

Where $\boldsymbol{I_G(x,y)}$ is the segmented image histogram displaced toward zero.

4.1.2.4 Crop

The image is cropped to fix the signature size and to remove any unnecessary parts of the signature image.

4.1.2.5 Resize

The cropped image is resized to N=512 and M=512 and this brings thee images to a standard size.

4.1.2.6 Interpolation

Resizing an image makes the points in the image mapped to a new set of points and after the mapping is done, the new image is left with two types of spurious points (point which have not been mapped into by any of the points in the original image and points which have been mapped into more than one point in the original image)[42].

4.1.2.7 Quantification

Quantification guarantees that the grey levels of the signature image are converted to specific values. These specific values are determined by the value which is used for the quantification. The quantified image $\boldsymbol{I_Q(x,y)}$ is obtained from $\boldsymbol{I_G(x,y)}$ as follows:

$$\boldsymbol{I_Q(x,y) = round\left(fix\left(\frac{I_G(x,y)8}{255}\right)\frac{255}{8}\right)}$$

Where $\boldsymbol{fix}$ rounds towards zero, and the exterior $\boldsymbol{round}$ is to ensure integer levels in the $\boldsymbol{I_Q(x,y)}$ image.

4.1.3 Feature Extraction

In the algorithm proposed by Vargas, Ferrer, Travieso and Alonso in 2011, under feature extraction phase, Grey level Co-occurrences Matrices (GLCM) are calculated from the signature images. Then Homogeneity, Contrast, Entropy, and Correlation texture feature calculations are performed to prepare the signature features for the verification. Finally, the range and the mean values of the texture features are obtained as the feature vector to be used for the verification [7].

4.1.3.1 Grey level Co-occurrences Matrices (GLCM)

A GLCM is a tabulation of how frequently various blends of pixel grey levels occur in an image. GLCM deliberates the relation among two pixels at a time called the reference pixel and the neighbor pixel. Then the occurrence of each pixel combination is recorded in a matrix.

Depending on the position of the considered neighbor pixel, 4 types of GLCM matrices can be calculated. The following image shows the four possible neighbor's locations.

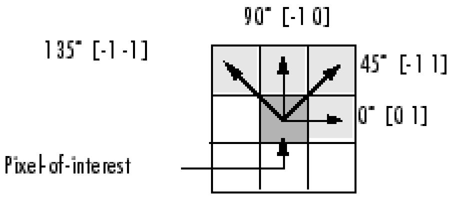
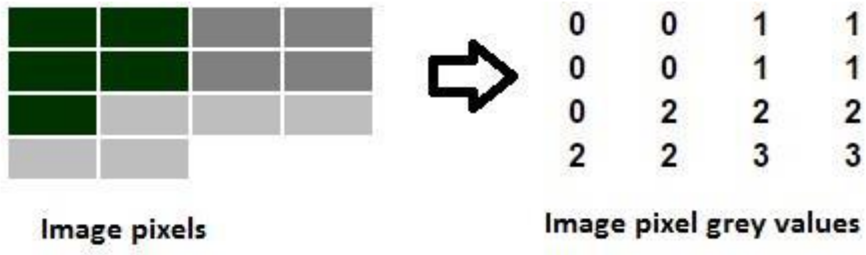


Figure 0.2: GLCM neighbor pixels [43]

The four possible neighbor locations which are shown in the image are the neighbor pixel to its right, neighbor pixel to its right and above, above neighbor pixel and the neighbor pixel to its left and above.

A simple demonstration of the GLCM matrix calculation when considering the neighbor pixel to its right side is as follows. When considering the following 4 x 4 sample image, first its pixel grey values are obtained.



0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

Figure 0.3: Image pixel values

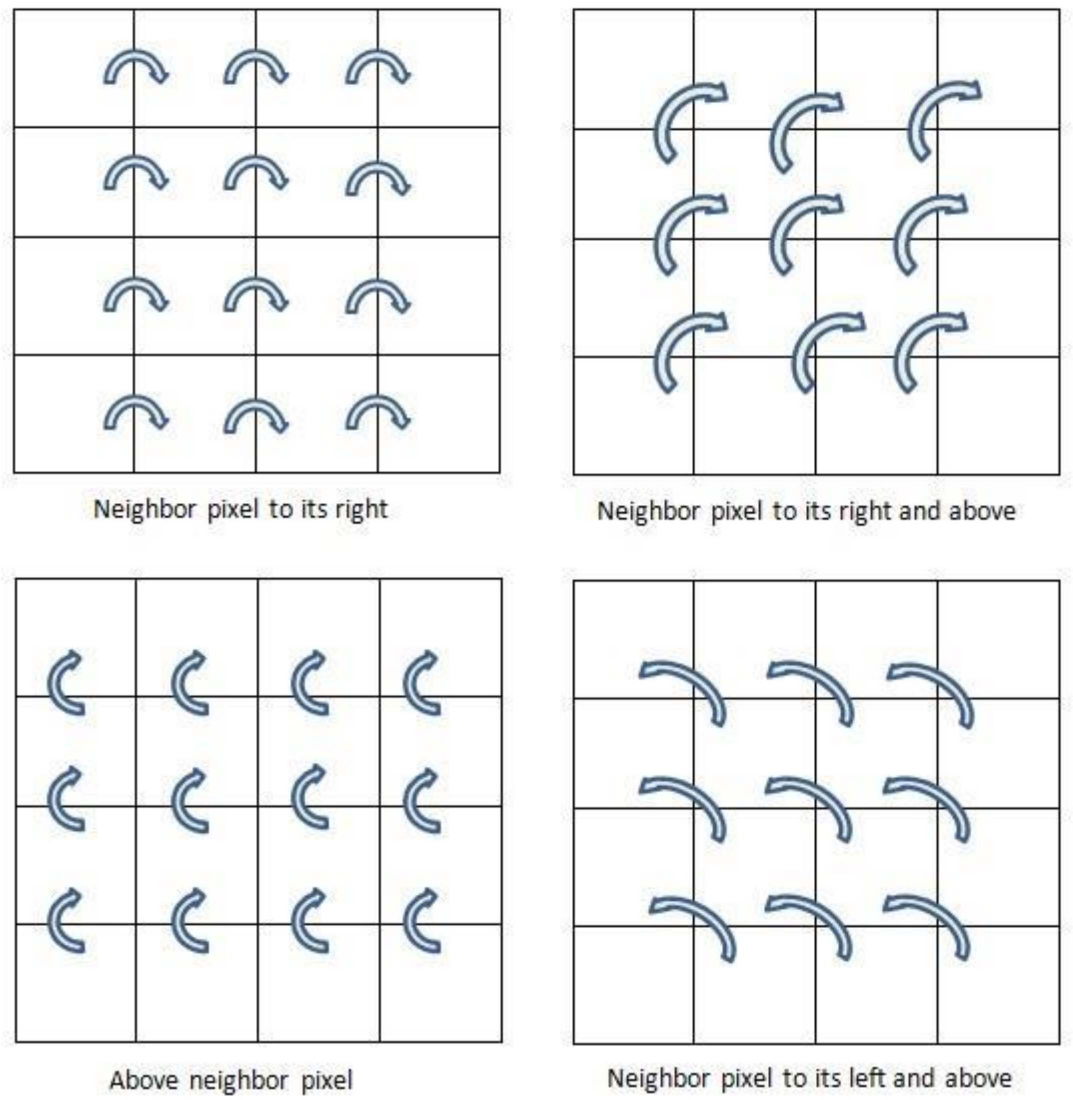
Since the sample image pixels contains only 4 grey levels, a 4 x 4 matrix is declared to record the grey level occurrence of each pixel with its right-side neighbor. The matrix calculation starts with the left most pixel of the top row of the image. Since the reference pixel (current pixel) and the right-side neighbor's grey levels are (0, 0) the matrix value of the element at the (0, 0) position is increased by 1. Then the reference pixel becomes the previous neighbor pixel. This process is continued for all the rows.

		Neighbor pixel values			
		0	1	2	3
Reference pixel values	0	2	2	1	0
	1	0	2	0	0
	2	0	0	3	1
	3	0	0	0	1

Table 0.1: Sample GLCM matrix

Later a symmetrical matrix is obtained, and the matrix values are normalized by separating every value by the total of all the matrix element values. Then the final matrix is obtained.

The following image shows how image pixel iterations of neighbor pixels are performed in order to calculate the 4 different matrices.



Neighbor pixel to its right

Neighbor pixel to its right and above

Above neighbor pixel

Neighbor pixel to its left and above

Figure 0.4: Pixel iterations for matrix calculation

As discussed, four GLCM matrices with four different neighbor pixel relationships (neighbor pixel to its right, neighbor pixel to its right and above, above neighbor pixel and the neighbor pixel to its left and above) are calculated to be used for the feature extraction.

4.1.3.2 Texture feature calculation

The textural measures obtained for each GLCM matrix are the following.

- Homogeneity - H
- Contrast - C
- Entropy - E
- Correlation - O

- □ **Texture Homogeneity**

Texture homogeneity can be computed by applying the following equation for the GLCM matrix [7].

$$\boldsymbol{H} = \sum_{\boldsymbol{i}=\boldsymbol{0}}^{\boldsymbol{G}-\boldsymbol{1}} \sum_{\boldsymbol{j}=\boldsymbol{0}}^{\boldsymbol{G}-\boldsymbol{1}} \{\boldsymbol{P}(\boldsymbol{i},\boldsymbol{j})\}^{\boldsymbol{2}}$$

Where $\boldsymbol{P}(\boldsymbol{i},\boldsymbol{j})$ is the matrix element value in the $(\boldsymbol{i},\boldsymbol{j})$ location and $\boldsymbol{G} = \boldsymbol{8}$

- □ **Texture Contrast**

Texture contrast can be computed by applying the following equation for the GLCM matrix [7].

$$\boldsymbol{C} = \sum_{\boldsymbol{i}=\boldsymbol{0}}^{\boldsymbol{G}-\boldsymbol{1}} \sum_{\boldsymbol{j}=\boldsymbol{0}}^{\boldsymbol{G}-\boldsymbol{1}} \boldsymbol{P}(\boldsymbol{i},\boldsymbol{j})(\boldsymbol{i}-\boldsymbol{j})^{\boldsymbol{2}}$$

Where $\boldsymbol{P}(\boldsymbol{i},\boldsymbol{j})$ is the matrix element value in the $(\boldsymbol{i},\boldsymbol{j})$ location and $\boldsymbol{G} = \boldsymbol{8}$

- □ **Texture Entropy**

Texture entropy can be calculated by applying the following equation for the GLCM matrix [7].

$$\boldsymbol{E} = \sum_{\boldsymbol{i}=\boldsymbol{0}}^{\boldsymbol{G}-\boldsymbol{1}} \sum_{\boldsymbol{j}=\boldsymbol{0}}^{\boldsymbol{G}-\boldsymbol{1}} \boldsymbol{P}(\boldsymbol{i},\boldsymbol{j}) - \mathbf{ln}\{\boldsymbol{P}(\boldsymbol{i},\boldsymbol{j})\}$$

Where $\boldsymbol{P}(\boldsymbol{i},\boldsymbol{j})$ is the matrix element value in the $(\boldsymbol{i},\boldsymbol{j})$ location and $\boldsymbol{G} = \boldsymbol{8}$

- □ **Texture Correlation**

Texture correlation can be calculated as follows [7]. First the $\boldsymbol{meanI}$ and $\boldsymbol{meanJ}$ values are calculated.

$$\boldsymbol{meanI} = \sum_{\boldsymbol{i}=\boldsymbol{0}}^{\boldsymbol{G}-\boldsymbol{1}} \sum_{\boldsymbol{j}=\boldsymbol{0}}^{\boldsymbol{G}-\boldsymbol{1}} \boldsymbol{i}\big(\boldsymbol{P}(\boldsymbol{i},\boldsymbol{j})\big)$$

$$\boldsymbol{meanJ} = \sum_{\boldsymbol{i}=\boldsymbol{0}}^{\boldsymbol{G}-\boldsymbol{1}} \sum_{\boldsymbol{j}=\boldsymbol{0}}^{\boldsymbol{G}-\boldsymbol{1}} \boldsymbol{j}\big(\boldsymbol{P}(\boldsymbol{i},\boldsymbol{j})\big)$$

Then the $\boldsymbol{StandardVarianceI^2}$ and $\boldsymbol{StandardVarianceJ^2}$ values are calculated.

$$\boldsymbol{StandardVarianceI^2} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \boldsymbol{P(i,j)(i - meanI)^2}$$

$$\boldsymbol{StandardVarianceJ^2} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \boldsymbol{P(i,j)(j - meanJ)^2}$$

Then the $\boldsymbol{StandardDeviationI^2}$ and $\boldsymbol{StandardDeviationJ^2}$ values are calculated.

$$\boldsymbol{StandardDeviationI^2} = \sqrt{\boldsymbol{StandardVarianceI^2}}$$

$$\boldsymbol{StandardDeviationJ^2} = \sqrt{\boldsymbol{StandardVarianceJ^2}}$$

Finally, the texture correlation is calculated.

$$\boldsymbol{O} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{\boldsymbol{(i - meanI)(j - meanJ)}}{\boldsymbol{StandardDeviationI} \times \boldsymbol{StandardDeviationJ}}$$

Where $\boldsymbol{P(i,j)}$ is the matrix element value in the $\boldsymbol{(i,j)}$ location and $\boldsymbol{G = 8}$

- □ **Calculating texture mean values**

The mean values of all the texture measures are calculated in the following manner.

E.g.:

$$\underset{\boldsymbol{1 \leq i \leq 4}}{\boldsymbol{mean\ H_i}} = \frac{\boldsymbol{1}}{\boldsymbol{4}} \sum_{i=1}^{4} \boldsymbol{H_i}$$

Then the four-element vector $\boldsymbol{M}$ comprising the average of each textural measure is obtained.

$$\boldsymbol{M} = \left\{ \underset{\boldsymbol{1 \leq i \leq 4}}{\boldsymbol{mean\ H_i}}, \quad \underset{\boldsymbol{1 \leq i \leq 4}}{\boldsymbol{mean\ C_i}}, \quad \underset{\boldsymbol{1 \leq i \leq 4}}{\boldsymbol{mean\ E_i}}, \quad \underset{\boldsymbol{1 \leq i \leq 4}}{\boldsymbol{mean\ O_i}} \right\}$$

☐ **Calculating texture range values**

The range values of all texture measures are calculated in the following manner.
E.g.:

$$\underset{\boldsymbol{1 \le i \le 4}}{\boldsymbol{range\ H_i}} = \underset{\boldsymbol{1 \le i \le 4}}{\boldsymbol{max\ H_i}} - \underset{\boldsymbol{1 \le i \le 4}}{\boldsymbol{min\ H_i}}$$

Where the range is the difference among the highest and the lowest values,

Then the four-element vector, comprising the range of each textural measure is obtained.

$$\boldsymbol{R} = \underset{\boldsymbol{1 \le i \le 4}}{\boldsymbol{range\ H_i}}, \quad \underset{\boldsymbol{1 \le i \le 4}}{\boldsymbol{range\ C_i}}, \quad \underset{\boldsymbol{1 \le i \le 4}}{\boldsymbol{range\ E_i}}, \quad \underset{\boldsymbol{1 \le i \le 4}}{\boldsymbol{range\ O_i}}$$

☐ **Calculating GLCM feature vector**

The eight-component feature vector is calculated by concatenating the $\boldsymbol{M}$ and $\boldsymbol{R}$ vectors.

$$\textbf{GLCM feature vector} = \{\boldsymbol{M}, \boldsymbol{R}\}$$

4.1.4 Verification

Once the eight-component feature vector is obtained, the SVM machine learning algorithm is used for the verification. The feature vector is mapped to the SVM’s high dimensional feature space. Since the obtained feature vectors are non-linearly separable the RBK kernel function is used. The appropriate SVM parameters (Gamma and C) are obtained by performing a k-fold Cross-Validation by trying out different parameter combinations. In the training phase, genuine signature features will be labeled as +1 and forgery signature features will be labeled as -1 to perform the decision rule. More information regarding SVM was discussed in the literature review.

4.1.5 k-Fold Cross-Validation

Mainly there are two reasons for doing cross-validation. As a testing method which gives a nearly unbiased estimate of the generalization power of the model. Also, to find the best C and gamma parameters over the training data.

Cross-validation is a technique used to evaluate machine learning models on a small number of datasets. The process has a variable called k that denotes the number of chunks that a given dataset is to be split into. Therefore, the process is named k-fold cross-validation. When any value for k is selected, it can be used in position of k in the reference to the model, such as k=5 becoming 5-fold cross-validation [47].

To guess the accuracy of a machine learning model on unobserved data, is mainly used Cross-validation. That is to use a small number of data so as to evaluate how the model is perform in common when used to get predictions on unseen data through the training of the model.

The overall process is as follows:

1. Randomly shuffle the dataset.
2. Divide the dataset into k chunks
3. For each unique chunk:
    - Keep the chunk as a hold out or test data set
    - Keep the remaining chunks as a training data set
    - Set a model on the training set and evaluate it on the test set.
    - Keep the evaluation score and remove the model
4. Review the accuracy of the model.

Every observation in the dataset is allocated to a fold and stays in that fold for the period of the process. This means that each dataset is given the chance to be used to train the model k-1 times. This method includes randomly separating the set of observations into k folds, of roughly equal size. The initial fold is considered as a validation set, and the process is fit on the k-1 folds [47].

Even though k-fold Cross-Validation is computationally expensive, does not waste too much data. This is a key benefit to problems like signature verification where the number of data instance is very limited.

4.1.6 Configuration of k in k-fold Cross-Validation

The value “k” must be selected cleverly, because poorly selected value for “k” may affect in an incorrect idea of the ability of the model. Also, k =5 or k =10 is commonly acceptable in the field and is recommended to select value 5 or 10 for the dataset. The 5 or 10 is a value that has been discovered through investigation. Therefore, choice of k is 5 or 10 [47].

4.2 Selection of the SVM library

There are various libraries for SVM. The most popular SVM libraries are LibSVMsharp, SVMlight and JNI_SVMlight. To select a better library, the author has done evaluation.

4.2.1 LibSVMsharp

LibSVMsharp is a C# wrapper for SVM and can be used for tasks such as classification and regression. This library is open source and supports for both 32-bit and 64-bit Cross-platform environments and the core (libsvm) is written using C++. All the popular Kernel options are provided by this library. LibSVMsharp is released under the MIT License and libsvm is released under the modified BSD License. LibSVMsharp is available as a NuGet package to download and install. Proper documentations are available [44].

4.2.2 JNI_SVMlight API

JNI_SVMlight is an implementation of the Support Vector Machine and can be used for tasks such as classification and regression. This API is open source and supports for both32-bit and 64-bit Windows and Linux environments and is written using Java. Since the precompiled libraries are also available, the API can be used by either using command line parameters or by directly integrating to a program using the source code. Also, all the popular Kernel options are provided by this API and the training data can be provided by using either file systems or directly from the code.

4.2.3 SVMlight API

SVMlight API is an implementation of the Support Vector Machines using C language and it is freely available. They further say that the API comes under a fast optimization algorithm and can be used for classification and regression problems. This API can be used with large amounts of training data and all the popular Kernel options are provided [45].

4.2.4 Evaluation and selection of SVM library

The JNI_SVMlight API is written using Java and the SVMlight API is written using the C language. Java is built upon C and when running a Java application, at some point it gets converted to C before its execution. And LibSVMsharp library is a C# wrapper of libsvm which is written using C++. Major added features in C++ are Object-Oriented Programming and Exception Handling.

Since all the APIs' and libraries considered are freely available to use, there is no cost applied when using them. The SVMlight and JNI_SVMlight API has poor support and lacks with good documentation about the API usage. The LibSVMsharp provides better support and provides helpful examples to the users about its usage. Unlike SVMlight, and JNI_SVMlight, the LibSVMsharp library allows the training data to be sent to the using either file systems or directly from the code. Using SVMlight with file systems may reduce the performance of the system. When considering these facts, it is safe to conclude that LibSVMsharp is the most suitable SVM library to use.

4.3 Selection of programming language

All programming languages that compatible with the shortlisted libraries has been evaluated and finally the most suitable programming language for the project was selected. All the shortlisted SVM libraries are compatible with NET Framework languages like C#. Even though JNI_SVMlight API, SVMlight API are written in java, C or C++, those can be used with C#, because of the language independence capability in NET Framework.

Both C# and Java are stable, platform independent languages and do not need excessive amount of care for the memory management. Also, applications written using both of these can be run on any operating system.

Even though the some of the libraries and APIs' can be used with Java language, it is clear that all of the libraries and APIs can be used with NET Framework languages like C#. Therefore, C# has been selected as the most suitable programming language due to these advantages.

EVALUATION PLAN

Testing is a vital procedure to be done so as to find whether the implemented system shows the intended results or not. A unit testing was done in order to evaluate the source code of the system and to evaluate accuracy of the system, the performance matrices which were discussed under the “Measures of performance evaluation” section was calculated.

Since this project is proposed and developed to address a frequently happening problem in most of the organizations such as banking and financial sectors, the concept must be evaluated by the respective end users. Because evaluation is a very important process which should be performed in order to ensure to that whether the proposed system produces the intended results or not. Hence, a set of professionals who are currently involved in the signature verification process from the banking sector can be taken to involve in the evaluation process.

5.1 Unit testing

Unit testing has done for the components according to the sequence of their action for possible various cases of the system. This allowed to determine any issues of the system modules since in the early stages.

Please refer to APPENDIX B for the detailed test cases.

5.2 Accuracy testing

Accuracy testing has done by calculating False Acceptance Rate (FAR) and the False Rejection Rate (FRR) performance matrices. In detailed information about these performance matrices have been discussed under the “Measures of performance evaluation” section.

The system will be tested using Sinhala, English, and Tamil signatures. The performance matrices will be calculated separately for each language in order to evaluate the system performance under each language.

The details of the collected signatures are as follows.

Language	No of persons	Genuine signatures		Skilled forgery signatures	
		Per person	Total	Per person	Total
Sinhala	12	12	144	12	144
English	12	12	144	12	144
Tamil	8	12	96	8	96

Table 0.1: Details of the collected signature database

In the training phase of the SVM machine learning algorithm, 5 genuine signatures and 5 skilled forgeries were used for each individual. All the signatures were collected using a black or blue color ballpoint pen in a white color A4 paper. The forgers were given unlimited amount of time to practice a signature and to produce the skilled forgeries. And, some signatures were collected from online resources like Kaggle.com. But the credibility of the signatures which gathered from online resources is uncertain.

The accuracy testing was performed by using 5 genuine signatures and 5 skilled forgery signatures for each individual. Skilled forgeries and genuine signatures, which will be used to test the system are not the signatures which were used to train the system.

The accuracy calculation was performed for several iterations with different Gamma and C parameters to improve the accuracy rates. The calculated error rates under each iteration are as follows.

5.2.1 Accuracy calculation - iteration 1

Language	Test data			
	Genuine signatures		Skilled forgery signatures	
	Success rate	FRR	Success rate	FAR
Sinhala	100%	0%	34%	66%
Tamil	100%	0%	16%	84%
English	100%	0%	8%	92%

Table 0.2: Accuracy calculation - iteration 1

5.2.2 Accuracy calculation - iteration 2

Language	Test data			
	Genuine signatures		Skilled forgery signatures	
	Success rate	FRR	Success rate	FAR
Sinhala	100%	0%	44%	56%
Tamil	98%	2%	28%	72%
English	100%	0%	18%	82%

Table 0.3: Accuracy calculation - iteration 2

5.2.3 Accuracy calculation - iteration 3

Language	Test data			
	Genuine signatures		Skilled forgery signatures	
	Success rate	FRR	Success rate	FAR
Sinhala	98%	2%	58%	42%
Tamil	90%	10%	62%	38%
English	98%	2%	26%	74%

Table 0.4: Accuracy calculation - iteration 3

5.2.4 Accuracy calculation - iteration 4

Language	Test data			
	Genuine signatures		Skilled forgery signatures	
	Success rate	FRR	Success rate	FAR
Sinhala	96%	4%	64%	36%
Tamil	90%	10%	72%	28%
English	97%	3%	40%	60%

Table 0.5: Accuracy calculation - iteration 4

5.2.5 Accuracy calculation - iteration 5

Language	Test data			
	Genuine signatures		Skilled forgery signatures	
	Success rate	FRR	Success rate	FAR
Sinhala	95%	5%	68%	32%
Tamil	88%	12%	78%	22%
English	96%	4%	46%	54%

Table 0.6: Accuracy calculation - iteration 5

5.2.6 Accuracy calculation - iteration 6

Language	Test data			
	Genuine signatures		Skilled forgery signatures	
	Success rate	FRR	Success rate	FAR
Sinhala	90%	10%	88%	12%
Tamil	89%	11%	81%	19%
English	92%	8%	77%	23%

Table 0.7: Accuracy calculation - iteration 6

Then concluded the accuracy testing with the final adjustments to the SVM parameters in iteration six. Thus, the final error rates were taken from the results calculated in iteration six.

Considering the FRR values recorded for all three languages, it's clearly shown that the system has a high success rate of identifying the genuine signatures. Because, the English language has the lowest FRR for genuine signatures, which was recorded as 8% and the Tamil language has the highest FRR for genuine signatures, was recorded as 11%.

After examining the obtained FAR values, it seems the FAR values for skilled forgeries are bit lower. The lowest FAR for skilled forgeries was for the Sinhala signatures, which is recorded as 12% and the highest FAR is for the English signatures is recorded as 23%. This seems the system is good at recognizing forged signatures.

The above-mentioned accuracy rates were obtained by using the proposed and implemented system for offline handwritten signature verification in the context of Sri Lanka for signatures of Sinhala, Tamil, and English.

The result is exceptional because no such research was conducted to test Sinhala and Tamil signatures with an offline handwritten signature verification system previously. After the enhancements done in each iteration offer helpful indications that the Offline Handwritten Signature Verification System's accuracy could be further optimized by recognizing the better Gamma and C parameters for the SVM Model.

Usually, a considerable number of signatures are weak and can be easily forged. The main reason for this is signatures are too simple, legible and varies broadly whenever they sign. Since weak signatures with wide differences minimize the level of the accuracy of the system, all the differences should to be considered for verifying the signatures. A forged signature produced by an unskilled person which differ a lot to the genuine signature can be accepted as genuine because of the differences.

This also encourages the users to use signatures which are complex, stylized, illegible and with less variations since they are hard to forge.

5.3 End user's evaluations and interviews

In the preliminary investigation, it was identified some of the issues in the signature verification process. Then in the interviews which were carried out in the preliminary investigation, three professionals who are regularly involved in the signature verification process from the banking sector were interviewed and the probabilities and the frequencies of the issues in the signature verification process which were identified are recorded accordingly.

(Please refer to APPENDIX A to view the interview document used for preliminary investigation)

Based on the identified issues, implemented solution can be evaluated by end users and can get their feedback to decide the effectiveness and whether the identified issues were overcome or not. Therefore, conducted another interview with the same professionals who interviewed in the preliminary investigation to get their feedback.

(Please refer to APPENDIX C to view the interview document using for evaluation)

The following table shows the mean values of the percentage that the proposed system is helpful to overcome each issue.

Issue	Percentage
The complexity of signature patterns makes the signature verification process harder.	60 %
Difficulty to match signatures due to the wide intrapersonal differences.	75 %
The minimal variances of skilled forgeries with the genuine signatures.	75 %
The signature matching gets complex by random variations, due to the writer's pauses or hesitations.	75 %
Short signatures could carry fewer information than long signatures, resulting in poor accuracy in verification outcomes.	40 %
Individuals with same names share similar signatures with others. At least concerning shape characteristics.	75 %
Difficulty to eliminate forgeries created by tracing or photocopying.	60 %

Figure 0.1: End user's evaluations and interviews results

CONCLUSION

The thesis elaborates a solution for an Offline Handwritten Signature Verification System by producing a result of classifying quality while mentioning the related theories and the used technologies.

In the preliminary investigation, a study has performed keenly and the problem domain was recognized by conducting document reviews, observations, and interviews. It turned out to be a good involvement to understand the problem domain by visiting to the real environment where such systems are in operation. As a result of that, the problems which are mostly relevant to Sri Lanka that occurs when signature verification processes are performed were able to identify. Addressing these issues turned into the motivation of the project.

The technologies and theories which are relevant were analyzed by conducting a solid research. As per the information gathered from the literature review, managed to get an idea that a binary classification algorithm is the best solution for the addressed problem. And as a result of studying similar systems, it was identified that the support vector machine algorithm has best performs when compared to other binary classification algorithms. Even though SVM has the ability of solving the problem, performed a trial and error analysis to evaluate some popular binary classification algorithms using collected datasets. Therefore, using Weka experiment environment, evaluated the algorithms based on different hyper parameters. After evaluating multiple iterations, considering accuracies, rankings and F1 score (F-Measure) of the models, SVM was selected as the solution for the problem. Even if this turned-out to be a stressful task, was able to recognize all the relevant areas subsequently.

Time management is an important factor for any kind of project. Therefore, since from the beginning author was conscious of the fact that the scope that he is going to be covered is practical enough to finish within the given time frame. Because of this reason author had to come up with time saving strategies while caring the quality of the results.

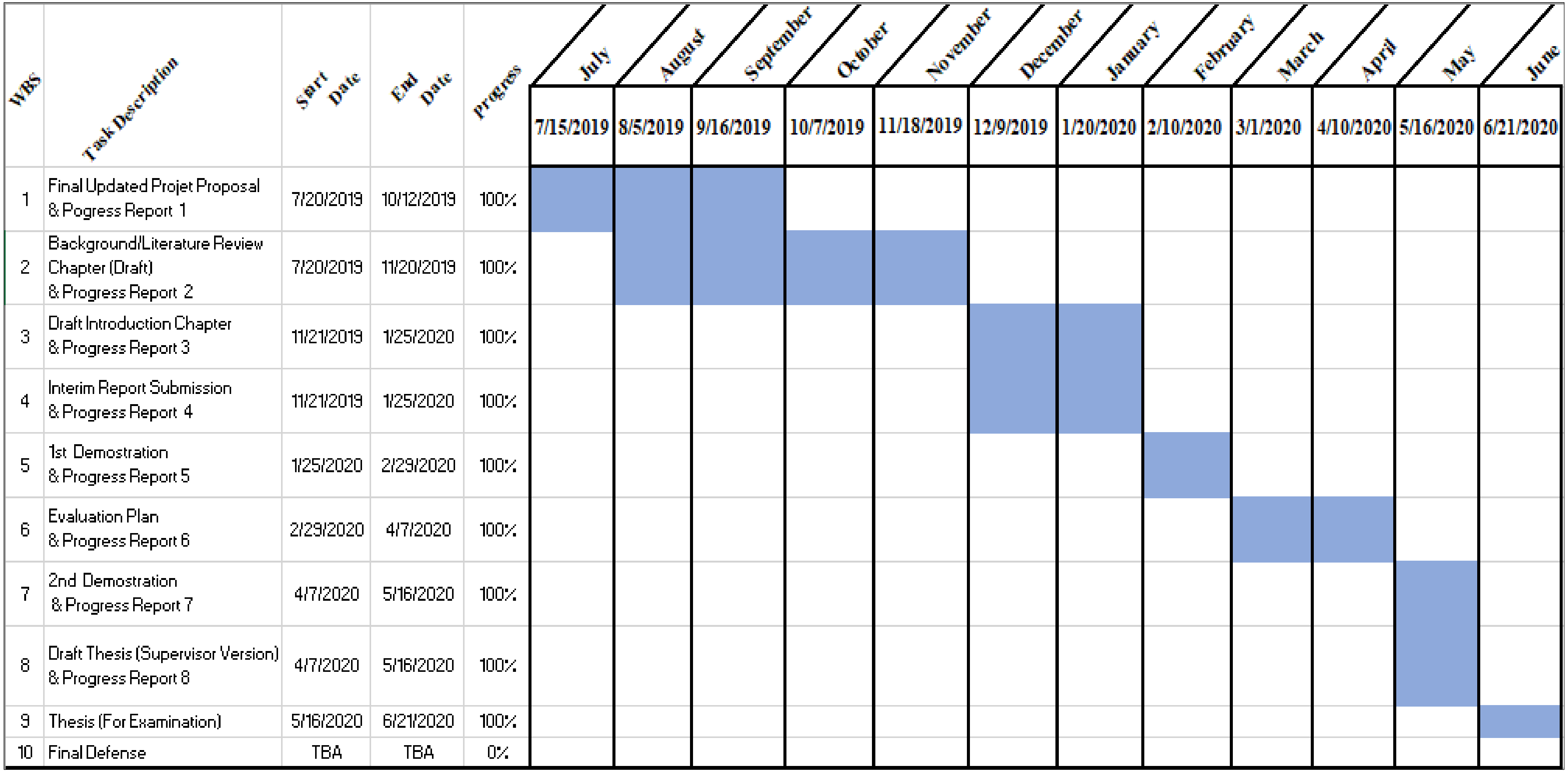
According to the project scope, the proposed solution should be evaluated with signatures belongs to Sinhala, Tamil and English languages, collecting test datasets turned-out to be a sort of challenging task. Some people refused to provide their signatures, while the majority agreed to give. And collection of Sinhala and Tamil signatures turned-out to be a difficult task since they are not commonly available.

Testing is a very important process which should be performed in order to ensure to that whether the proposed system produces the intended results or not. In order to test the system Unit testing and accuracy testing were done. It was observed that the system is efficient in identifying genuine signatures by performing accuracy testing. The result is exceptional because no such research was done to test Sinhala and Tamil signatures with an offline handwritten signature verification system previously.

6.1 Future work

Even if the proposed solution's shows high performance, it can be optimized by performing a further research to apply advanced multithreading functionality for the implementation. The image preprocessing techniques could be further optimized to a standard where the signatures could be extracted from any background. More signature features can be extracted and more suitable Gamma and C parameters for the SVM can be discovered to upgrade the system's accuracy. All the genuine and forged signatures were specifically created to evaluate the proposed solution. Therefore, credibility of the test data can be increased by collecting the signatures from Police or CID from criminal cases where forgeries were actually created by criminals to cheat.

PROJECT PLAN AND TIMELINE



WBS	Task Description	Start Date	End Date	Progress	July	August	September	October	November	December	January	February	March	April	May	June
					7/15/2019	8/5/2019	9/16/2019	10/7/2019	11/18/2019	12/9/2019	1/20/2020	2/10/2020	3/1/2020	4/10/2020	5/16/2020	6/21/2020
1	Final Updated Projet Proposal & Pogress Report 1	7/20/2019	10/12/2019	100%	🟦	🟦	🟦									
2	Background/Literature Review Chapter (Draft) & Progress Report 2	7/20/2019	11/20/2019	100%		🟦	🟦	🟦	🟦							
3	Draft Introduction Chapter & Progress Report 3	11/21/2019	1/25/2020	100%						🟦	🟦					
4	Interim Report Submission & Progress Report 4	11/21/2019	1/25/2020	100%						🟦	🟦					
5	1st Demostration & Progress Report 5	1/25/2020	2/29/2020	100%								🟦				
6	Evaluation Plan & Progress Report 6	2/29/2020	4/7/2020	100%									🟦	🟦		
7	2nd Demostration & Progress Report 7	4/7/2020	5/16/2020	100%											🟦	
8	Draft Thesis (Supervisor Version) & Progress Report 8	4/7/2020	5/16/2020	100%											🟦	
9	Thesis (For Examination)	5/16/2020	6/21/2020	100%												🟦
10	Final Defense	TBA	TBA	0%												

REFERENCES

[1] Abikoye, O. C., Mabayoje, M. A., & Ajibade, R. (2011, December). Offline Signature Recognition & Verification using Neural Network. International Journal of Computer Applications, 35(2), 44-51.

[2] Kumar, L & Babu, A. (2012). Genuine and Forged Offline Signature Verification Using Back Propagation Neural Networks.

[3] Arya, Meenakshi & Inamdar, Vandana. (2010). A Preliminary Study on Various Off-line Handwritten Signature Verification Approaches. International Journal of Computer Applications. 1. 10.5120/199-338.

[4] McManus, R. (2014, May 1). How are AIIM Professionals Using Signature Verification. [Online]. Available: https://www.parascript.com/blog/aiim-professionals-using-signature-verification/. [Accessed: 20- Oct- 2019].

[5] Amna. (2014). Computers and Technology. [Online]. Available: http://www.cssforum.com.pk/off-topic-section/computers-technology/63157-digital-glossary-2.html. [Accessed: 20- Oct- 2019].

[6] Saikia, H., & Sarma, K. C. (2012, March). Approaches and Issues in Offline Signature Verification System. International Journal of Computer Applications, 42(16), 47.

[7] Vargas, J. F., Ferrer, M. A., Travieso, C. M., & Alonso, J. B. (2011, February). Offline signature verification based on grey level information using texture features. Pattern Recognition, 44(2), 375-385.

[8] L. C., L. M., & K. M. (2007). Research Methods in Education (6th ed.). Oxon: Routledge.

[9] Huang, K., & Yan, H. (1997, April). Off-line signature verification based on geometric feature extraction and neural network classification. Pattern Recognition, 30(1).

[10] Brault, J., & Plamondon, R. (1993, March). A complexity measure of handwritten curves: modeling of dynamic signature forgery. IEEE Transactions on Systems, Man and Cybernetics, 23(2), 400-413.

[11] Ismail, M. A., & Gad, S. (2000, October). Off-line arabic signature recognition and verification. Pattern Recognition, 33(10), 1727-1740.

[12] Quek, C., & Zhou, R. W. (2002, March). Antiforgery: a novel pseudo-outer product based fuzzy neural network driven signature verification system. Pattern Recognition Letters, 23(14), 1795-1816.

[13] Coetzer, J., Herbst, B. M., & Preez, J. A. (2004, April 21). Offline Signature Verification Using the Discrete Radon Transform and a Hidden Markov Model. EURASIP Journal on Advances in Signal Processing, 559-571.

[14] Garhawal, S., & Shukla, N. (2013, August). A Study on Handwritten Signature Verification Approaches. International Journal of Advanced Research in Computer Engineering & Technology, 2(8), 2501

[15] Madasu, V. K., & Lovell, B. C. (2008). An Automatic Offline Signature Verification and Forgery Detection System. Pattern Recognition Technologies and Applications: Recent Advances. IGI Global, 63-89.

[16] Kovari, B. A. (2013). Models and Algorithms in Offline, Feature-Based, Handwritten Signature Verification. Budapest University of Technology and Economics, Department of Automation and Applied Informatics, Budapest.

[17] Impedovo, D., & Pirlo, G. (2008). Automatic Signature Verification: The State of the Art. Transactions on Systems, Man, and Cybernetics, 612

[18] Pansare, A., & Bhatia, S. (2012, January). Handwritten Signature Verification using Neural Network. International Journal of Applied Information Systems, 1(2), 44-49.

[19] Deng, P. S., Liao, H.-Y. M., Ho, C. W., & Tyan, H.-R. (1999, December). Wavelet-Based Off-Line Handwritten Signature Verification. Computer Vision and Image Understanding, 76(3), 173-190.

[20] Bansal, A., Gupta, B., Khandelwal, G., & Chakraverty, S. (2009, March). Offline Signature Verification Using Critical Region Matching. International Journal of Signal Processing, Image Processing and Pattern, 2(1), 57-70.

[21] Misra, P. (2013). An Offline Handwritten Signature Verification System. Jadavpur University, Faculty Council for UG and PG Studies in Engineering and Technology, Kolkata.

[22] Inglis, S., & Witten, I. H. (1994). Compression- Based Template Matching. Proc. IEEE Data Compression Conference, (pp. 106-115). Los Alamitos, CA

[23] Chavan, G. H., Adhiya, K. P., & Gharde, S. S. (2012). Offline Handwritten Signature Verification Approaches: A Review. International Journal of Computer Sci ence And Technology, 3, 2.

[24] Karwankar, A. R., & Bhosale, V. K. (2013). Automatic Static Signature Verification Systems: A Review. International Journal of Computational Engineering Research, 3(2), 9-10.

[25] Jana, R., Saha, R., & Datta, D. (2014). Offline Signature Verification using Euclidian Distance. International Journal of Computer Science and Information Technologies, 5(1), 709.

[26] Ferrer, M. A., Alonso, J. B., & Travieso, C. M. (2005). Offline Geometric Parameters for Automatic Signature Verification Using Fixed-Point Arithmetic. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(6), 995.

[27] Schafer, B., & Viriri, S. (2009). An Off-Line Signature Verification System. IEEE International Conference on Signal and Image Processing Applications (pp. 95-100). Kuala Lumpur: IEEE.

[28] Majhi, B., Reddy, Y. S., & Babu, P. (2006). Novel Features for Off-line Signature Verification. International Journal of Computers, Communications & Control, 1(1), 17-24.

[29] Kumar, P., Singh, S., Garg, A., & Prabhat, N. (2013, March). Handwritten Signature Recognition & Verification using Neural Network. International Journal of Advanced Research in Computer Science and Software Engineering, 3(3), 559- 560.

[30] Bhattacharyya, D., Bandyopadhyay, S. K., Das, P., Ganguly, D., & Mukherjee, S. (2008). Statistical Approach for Offline Handwritten Signature Verification. Journal of Computer Science, 3(4), 181.

[31] Gunjal, S. N., & Lipton, M. (2011, November). Robust Offline Signature Verification Based on Polygon Matching Technique. International Journal of Emerging Technology and Advanced Engineering, 1(1), 80.

[32] Justino, E. R., Bortolozzi, F., & Sabourin, R. (2001). Off-line Signature Verification Using HMM for Random, Simple and Skilled Forgeries. Document Analysis and Recognition (pp. 1031-1034). Seattle, WA: IEEE.

[33] Abdelrahman, A., & Abdallah, A. (2013). Signature Verification System Based on Support Vector Machine Classifier. The International Arab Conference on Information Technology, (pp. 1-5).

[34] Nguyen, V., Blumenstein, M., Muthukkumarasamy, V., & Leedham, G. (2007). Off-line Signature Verification Using Enhanced Modified Direction Features in Conjunction with Neural Classifiers and Support Vector Machines. Ninth International Conference on Document Analysis and Recognition. 2, pp. 734-738. Parana: IEEE.

[35] Audet, S., Bansal, P., & Baskaran, S. (2006, April 7). Offline Signature Verification Using Virtual Support Vector Machines. ECSE 526 – Artificial Intelligence.

[36] Blumenstein, M., Liu, X. Y., & Verma, B. (2004). A Modified Direction Feature for Cursive Character Recognition. International Joint Conference on Neural Networks, (pp. 2893-2987).

[37] Patel, S., Pandit, R., & Shete, S. S. (2014, January). Comparative Analysis of Different Approaches for Offline Signature Verification. International Journal of Advanced Electronics & Communication Systems, 3(1).

[38] Satyarthi, D., Maravi, Y. P., Sharma, P., & Gupta, R. K. (n.d.). Comparative Study of Offline Signature Verification Techniques. International Journal of Advancements in Research & Technology, 2(2), 3-4.

[39] Ramachandra, A. C., Ravi, J., Raja, K. B., Venugopal, K. R., & Patnaik, L. M. (2009, May). Signature Verification using Graph Matching and Cross-Validation Principle. International Journal of Recent Trends in Engineering, 1(1), 57-61.

[40] Abuhaiba, I. S. (2007). Offline Signature Verification Using Graph Matching. Turkish Journal of Electrical Engineering, 15(1), 89-104.

[41] Enturk, T. S., Zgunduz, E. O., & Karshgil, E. (2005). Handwritten Signature Verification Using Image Invariants and Dynamic Features. Proceedings of the 13th European Signal Processing Conference EUSIPCO. Antalya Turkey.

[42] Ramesh, V. E., & Murty, M. N. (1999, February). Off-line signature verification using genetically optimized weighted features. Pattern Recognition, 32(2), 217-233.

[43] Mathworks.in. n.d. Create Gray-Level Co-Occurrence Matrix from Image - MATLAB Graycomatrix- Mathworks India. [online] Available at: http://www.mathworks.in/help/images/ref/graycomatrix.html [Accessed 20 January 2020].

[44] GitHub. (2020). ccerhan/LibSVMsharp. [online] Available at: https://github.com/ccerhan/LibSVMsharp [Accessed 23 Jan. 2020].

[45] Svmlight.joachims.org. (2020). [online] Available at: http://svmlight.joachims.org/ [Accessed 23 Jan. 2020].

[46] J. Brownlee, "Machine Learning Mastery with Weka: Analyze Data, Develop Models, and Work Through Projects," Machine Learning Mastery, 2016.

[47] Brownlee, J., 2018. A Gentle Introduction To K-Fold Cross-Validation. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/k-fold-cross-validation/ [Accessed 18 May 2020].

[48] Brownlee, J., 2019. How To Compare The Performance Of Machine Learning Algorithms In Weka. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/compare-performance-machine-learning-algorithms-weka/ [Accessed 21 May 2020].

[49] Brownlee, J., 2020. How To Calculate Precision, Recall, And F-Measure For Imbalanced Classification. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/ [Accessed 20 May 2020].

APPENDIX A: Interview Document (Preliminary investigation)

1. Could you please explain the process of signature verification?
2. Do you use any computer system for the verification process?
3. What characteristics do you look for in a signature to match?
4. How do you check for forgeries?

5. What are the difficulties you find in signature verification?

Difficulty	Frequency

6. When do you fail to match and verify a signature even though it is a legitimate signature?

Occasion	Frequency

7. When do you fail to identify forged signatures?

Occasion	Frequency

8. What are the other technical errors and mistakes were made by people and/or systems in verifying signatures?

Errors/Mistakes	Frequency

APPENDIX B: Unit Testing – Test cases

1. Add new person

1.1. Input validations

Test case no: 1	Test scenario: Check for valid number of images
Test description	Test whether the system successfully identifies when submitting the required signature image number. (genuine and forged signature images)
Input data	5 signature images
Expected results	System detects the submitted number of images are 5
Actual results	System detects the submitted number of images are 5
Pass/ Fail	Pass
Changes required	No changes are required

Test case no: 2	Test scenario: Check for invalid number of images
Test description	Test whether the system successfully identifies when submitting less than 5 signature images.
Input data	A number of less than 5 signature images.
Expected results	System detects the submitted number of images are less than 5
Actual results	System detects the submitted number of images are less than 5
Pass/ Fail	Pass
Changes required	No changes are required

Test case no: 3	Test scenario: Check for invalid number of images
Test description	Test whether the system successfully identifies when submitting more than 5 signature images.
Input data	A number of greater than 5 signature images.
Expected results	System detects the submitted number of images are greater than 5
Actual results	System detects the submitted number of images are greater than 5
Pass/ Fail	Pass
Changes required	No changes are required

Test case no: 4	Test scenario: Check for valid image file inputs
Test description	Test whether the system successfully allows to submit image files.
Input data	5 signature images, Exe files, folders
Expected results	System only allows to submit image files.
Actual results	System only allows to submit image files.
Pass/ Fail	Pass
Changes required	No changes are required

1.2. Save person's details to database

Test case no: 5	Test scenario: Save person's details to database
Test description	Test whether the component successfully saves person's ID or name to the database
Input data	Person's ID or Person's name
Expected results	System successfully saves details to the database
Actual results	System successfully saves details to the database.
Pass/ Fail	Pass
Changes required	No changes are required

1.3. Preprocess signature images

Test case no: 6	Test scenario: Grayscale signature image
Test description	Test whether the system successfully convert the signature image to a grayscale image
Input data	Signature image
Expected results	Signature image with grey levels ranging from 0-255
Actual results	Signature image with grey levels ranging from 0-255
Pass/ Fail	Pass
Changes required	No changes are required

Test case no: 7	Test scenario: Posterize signature image
Test description	Test whether the system successfully posterize the grayscale signature image
Input data	Grayscale signature image
Expected results	Signature image with 0, 85, 170, 255 grey levels
Actual results	Signature image with 0, 85, 170, 255 grey levels
Pass/ Fail	Pass
Changes required	No changes are required

Test case no: 8	Test scenario: Binarize and segment signature image
Test description	Test whether the system successfully binarize and segment the posterized signature image
Input data	Posterized signature image
Expected results	Binarized and segmented signature image
Actual results	Binarized and segmented signature image
Pass/ Fail	Pass
Changes required	No changes are required

Test case no: 9	Test scenario: Histogram displacement
Test description	Test whether the system successfully displace the histogram of the binarized and segmented signature image.
Input data	Binarized and segmented signature image
Expected results	Histogram displaced signature image
Actual results	Histogram displaced signature image
Pass/ Fail	Pass
Changes required	No changes are required

Test case no: 10	Test scenario: Crop signature image to its bounding box
Test description	Test whether the system successfully crop the histogram displaced signature to the bounding box of the signature.
Input data	Histogram displaced signature image
Expected results	Signature image cropped to its bounding box
Actual results	Signature image cropped to its bounding box
Pass/ Fail	Pass
Changes required	No changes are required

Test case no: 11	Test scenario: Resize and perform nearest neighbor interpolation to signature image
Test description	Test whether the system successfully resizes and perform the nearest neighbor interpolation to the cropped signature image.
Input data	Cropped signature image
Expected results	Resized and nearest neighbor interpolation performed signature image
Actual results	Resized and nearest neighbor interpolation performed signature image
Pass/ Fail	Pass
Changes required	No changes are required

Test case no: 12	Test scenario: Test scenario: Quantify signature image
Test description	Test whether the system successfully quantifies the signature image
Input data	Resized and nearest neighbor interpolation performed signature image
Expected results	Signature image with 0, 31, 63, 95, 127, 159, 191, 255 grey levels
Actual results	Signature image with 0, 31, 63, 95, 127, 159, 191, 255 grey levels
Pass/ Fail	Pass
Changes required	No changes are required

1.4. Extract signature features

Test case no: 13	Test scenario: Calculate Grey Level Co-occurrence Matrix (GLCM) 1
Test description	Test whether the system successfully calculate GLCM 1
Input data	Preprocessed signature image
Expected results	System successfully calculates the GLCM matrix 1
Actual results	System successfully calculates the GLCM matrix 1
Pass/ Fail	Pass
Changes required	No changes are required

Test case no: 14	Test scenario: Calculate Grey Level Co-occurrence Matrix (GLCM) 2
Test description	Test whether the system successfully calculate GLCM 2
Input data	Preprocessed signature image
Expected results	System successfully calculates the GLCM matrix 2
Actual results	System successfully calculates the GLCM matrix 2
Pass/ Fail	Pass
Changes required	No changes are required

Test case no: 15	Test scenario: Calculate Grey Level Co-occurrence Matrix (GLCM) 3
Test description	Test whether the system successfully calculate GLCM 3
Input data	Preprocessed signature image
Expected results	System successfully calculates the GLCM matrix 3
Actual results	System successfully calculates the GLCM matrix 3
Pass/ Fail	Pass
Changes required	No changes are required

Test case no: 16	Test scenario: Calculate Grey Level Co-occurrence Matrix (GLCM) 4
Test description	Test whether the system successfully calculate GLCM 4
Input data	Preprocessed signature image
Expected results	System successfully calculates the GLCM matrix 4
Actual results	System successfully calculates the GLCM matrix 4
Pass/ Fail	Pass
Changes required	No changes are required

Test case no: 17	Test scenario: Normalize Grey Level Co-occurrence Matrix (GLCM)
Test description	Test whether the system successfully normalizes the calculated GLCM matrix
Input data	GLCM matrix
Expected results	System successfully normalizes the GLCM matrix
Actual results	System successfully normalizes the GLCM matrix
Pass/ Fail	Pass
Changes required	No changes are required

Test case no: 18	Test scenario: Calculate texture homogeneity
Test description	Test whether the system successfully calculates texture homogeneity
Input data	Normalized GLCM matrix
Expected results	System successfully calculate texture homogeneity
Actual results	System successfully calculate texture homogeneity
Pass/ Fail	Pass
Changes required	No changes are required

Test case no: 19	Test scenario: Calculate texture contrast
Test description	Test whether the system successfully calculates texture contrast
Input data	Normalized GLCM matrix
Expected results	System successfully calculate texture contrast
Actual results	System successfully calculate texture contrast
Pass/ Fail	Pass
Changes required	No changes are required

Test case no: 20	Test scenario: Calculate texture entropy
Test description	Test whether the system successfully calculates texture entropy
Input data	Normalized GLCM matrix
Expected results	System successfully calculate texture entropy
Actual results	System successfully calculate texture entropy
Pass/ Fail	Pass
Changes required	No changes are required

Test case no: 21	Test scenario: Calculate texture correlation
Test description	Test whether the system successfully calculates texture correlation
Input data	Normalized GLCM matrix
Expected results	System successfully calculate texture correlation
Actual results	System successfully calculate texture correlation
Pass/ Fail	Pass
Changes required	No changes are required

Test case no: 22	Test scenario: Calculate texture feature average
Test description	Test whether the system successfully calculates texture feature average
Input data	Texture feature values of a signature
Expected results	System successfully calculate texture feature averages
Actual results	System successfully calculate texture feature averages
Pass/ Fail	Pass
Changes required	No changes are required

Test case no: 23	Test scenario: Calculate texture feature range difference
Test description	Test whether the system successfully calculates texture feature range difference
Input data	Texture feature values of a signature
Expected results	System successfully calculate texture feature range
Actual results	System successfully calculate texture feature range
Pass/ Fail	Pass
Changes required	No changes are required

1.5. Save extracted features to database

Test case no: 24	Test scenario: Save extracted features to database
Test description	Test whether the system successfully saves extracted features to database
Input data	Texture feature average values, Texture feature range difference values
Expected results	System successfully saves details to the database
Actual results	System successfully saves details to the database
Pass/ Fail	Pass
Changes required	No changes are required

2. Verify a signature

2.1. Input validations

Test case no: 25	Test scenario: Check for valid image file input
Test description	Test whether the system successfully validates that the submitted file is an image file
Input data	1 signature image
Expected results	System validates that the submitted file is an image file
Actual results	System validates that the submitted file is an image file
Pass/ Fail	Pass
Changes required	No changes are required

Test case no: 26	Test scenario: Check for invalid image file input
Test description	Test whether the system successfully identifies that the submitted file is a non-image file
Input data	1 EXE file
Expected results	System detects that the submitted file is a non-image file
Actual results	System detects that the submitted file is a non-image file
Pass/ Fail	Pass
Changes required	No changes are required

2.2. Preprocess signature image

All the signature preprocessing components have been tested and the test cases are included under the “Preprocess signature images” in the previous section. (Test case no: 27)

2.3. Extract signature features

All the signature feature extraction components have been tested and the test cases are included under the “Extract signature features” in the previous section. (Test case no: 28)

2.4. Load signature features from database

Test case no: 29	Test scenario: Load signature features from database
Test description	Test whether the system successfully loads the signature features of the relevant person from the database
Input data	Person’s ID
Expected results	System loads the signature features of the relevant person from the database
Actual results	System loads the signature features of the relevant person from the database
Pass/ Fail	Pass
Changes required	No changes are required

2.5. Verification

Test case no: 30	Test scenario: Load SVMModel
Test description	Test whether the system successfully loads the SVMModel using the person’s loaded signature features from database.
Input data	Person’s loaded signature features from database
Expected results	System loads the SVMModel
Actual results	System loads the SVMModel
Pass/ Fail	Pass
Changes required	No changes are required

Test case no: 31	Test scenario: Load FeatureVector
Test description	Test whether the system successfully loads the FeatureVector using the features of the signature to be verified.
Input data	Calculated features from the signature to be verified
Expected results	System loads the FeatureVector
Actual results	System loads the FeatureVector
Pass/ Fail	Pass
Changes required	No changes are required

Test case no: 32	Test scenario: Verify signature
Test description	Test whether the system successfully gives the verification results using the SVMModel and the FeatureVector
Input data	SVMModel, FeatureVector
Expected results	System successfully gives the verification results
Actual results	System successfully gives the verification results
Pass/ Fail	Pass
Changes required	No changes are required

APPENDIX C: Interview Document (Evaluation)

Issue	**The software is helpful to overcome this issue. Rate your satisfaction.** (Very poor - 0%, Poor - 20%, Satisfactory - 50%, Good - 70%, Excellent - 100%)
The complexity of signature patterns makes the signature verification process harder.	□ Very poor □ Poor □ Satisfactory □ Good □ Excellent
Difficulty to match signatures due to the wide intrapersonal differences.	□ Very poor □ Poor □ Satisfactory □ Good □ Excellent
The minimal variances of skilled forgeries with the genuine signatures.	□ Very poor □ Poor □ Satisfactory □ Good □ Excellent
The signature matching gets complex by random variations, due to the writer's pauses or hesitations.	□ Very poor □ Poor □ Satisfactory □ Good □ Excellent
Short signatures could carry fewer information than long signatures, resulting in poor accuracy in verification outcomes.	□ Very poor □ Poor □ Satisfactory □ Good □ Excellent
Individuals with same names share similar signatures with others. At least concerning shape characteristics.	□ Very poor □ Poor □ Satisfactory □ Good □ Excellent
Difficulty to eliminate forgeries created by tracing or photocopying.	□ Very poor □ Poor □ Satisfactory □ Good □ Excellent

Comments: