“Online Signature Verification System”

End-Semester Report of
7th Semester Mini Project
FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

BY

N. Lavan Kumar (IIT2014078)
B. Sharath Chandra (IIT2014056)
B. Priyatham (IIT2014057)

UNDER THE SUPERVISION OF

Dr. Sonali Agarwal
Assistant Professor
IIIT-ALLAHABAD

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, ALLAHABAD

November, 2017

Declaration by the Candidates

We, hereby declare that the project titled "Online Signature Verification System" is a record of bonafide project work carried out by us under the guidance of **Dr. Sonali Agarwal** in fulfilment of the VII semester Mini-Project work for the B.Tech (IT) Course in Indian Institute of Information Technology, Allahabad.

N Lavan Kumar - IIT2014078
B Sharath Chandra - IIT2014056
B Priyatham - IIT2014057

Certificate

This is to certify that the project report entitled “Online Signature Verification System” submitted to Department of Information Technology, Indian Institute of Information Technology, Allahabad in fulfilment of the VII semester Mini-project work, is a record of bonafide work carried out by:

N Lavan Kumar	–	IIT2014078
B Sharath Chandra	–	IIT2014056
B Priyatham	–	IIT2014057

under my supervision and guidance.

This report has not been submitted anywhere else for any other purpose.

Submission Date : 22/11/2017

Dr. Sonali Agarwal
Assistant Professor
Department of Information Technology
Indian Institute of Information Technology
Allahabad - 211015

Acknowledgement

We would like to express our special thanks of gratitude to Dr. Sonali Agarwal who gave us the opportunity to do this project titled “Online Signature Verification System”. We appreciate her contribution, constant support and perseverance in this endeavour of ours. Her engagement through the process of this project has been precious and irreplaceable.

Table of Contents

1. Abstract

Biometric recognition and verification systems have been existed for many years and they are extremely important for every industry for keeping their data and information secure. These include but are not constrained to retina scanning, fingerprint scanning, voice analysis, palm vein authentication, iris scanning and facial recognition. With advancements and developments in biometric authentication technology, most of the biometric recognition and verification systems have improved however, the deploying cost of biometric systems are very high, by considering verification of very high usage. Not only are that but portability of most of these systems not good. Now a days smart phones are used very prevalent so they can be incorporated in biometric systems. In-order to find a middle ground between effectiveness, efficiency in portability and mass quantity, we take a deeper look into signature verification which can be deployed using mobile devices to produce fruitful results.

2. Introduction

Signature verification has been a challenging and interesting problem for biometric researchers from many years in the recent past. Signatures can be broadly divided into two categories

1) Offline Signatures and
2) Online Signatures.

Offline signatures are those that take place on paper and analysis of them can be carried out only when information of a signature on paper is somehow extracted through image processing and stored digitally. This information can be termed as features and are used primarily to distinguish between two signatures. In a way, it would be safe to say that the features provide a unique identity to a signature.

Fig. 1: Image showing offline signature (Image Source[7])

Online Signatures on the other hand are signatures that take place on an electronic device that are capable of recording the movement of signature at a fixed interval of time, digitally. Therefore, x-y coordinates of the user's signatures along with attributes like pressure, time and pen up/down are also acquired. It is because of these vast range of dynamic features that online signature verification system usually achieves a better accuracy than an offline system.

Fig. 2: Image showing online signature (Image Source[7])

3. Motivation

In recent times signature forgery crimes have been increased from earlier. The signature verification system becomes handy in government offices for document validation, banking applications, student mark sheet verification etc. As it is expensive to implement a biometric recognition system, we can implement this as everyone is using smart phones these days. We can say our online signature verification is no less than any recognition system currently in demand.

4. Problem Definition

The problem is to verify whether given signature is genuine or forgery of a particular user with an android application using machine learning algorithms.

5. Literature Survey

This paper performs online signature on touch sensitive devices. They represented online signature as a discriminative feature vector derived from histograms of attributes which is computed in linear time. A resulting template signature is generated for every user which is used to validate a user as genuine or forgery. The dataset used is MCYT-100 and SUSIG data sets. The features used are coordinates of x, y, pressure and timestamps. The steps involved in their methodology are pre-processing, histogram feature extraction, template generator, matcher and verification. Validation of their method is done using acceptance false rate and rejection false rate. [1]

This paper implements field-programmable gate arrays of an embedded system for online signature verification. Steps involved are Signature pre-processing,

Sigma-Lognormal Parameter Extraction, Signature Reconstruction, Generation of duplicate signatures and Verification of signatures. Their methodology consists of three stages. In pre-processing stage, removal of noise and normalizing x and y coordinates of signature. Later dynamic time warping algorithm is used to align this processed signature with its template previously stored in a database. Next features are extracted and Gaussian mixture model is applied, it gives degree of similarity of signatures. The algorithm was tested using a public database of 100 users, obtaining high recognition rates for both genuine and forgery signatures which gave good results. [2]

Usually online signature verification requires 5 -10 genuine signatures of a person. But this implemented online signature using single signature of user. Their methodology is generating duplicate signatures from one single signature of user. Duplicates are generated using sigma lognormal decomposition. Two methods are presented to create human-like duplicated signatures: the first varies the strokes lognormal parameters whereas the second modifies their virtual target points. Their results proved that their accuracy is matching with models that their 5 signatures for verification. [3]

In this paper a feature extraction approach based on Legendre series representation of the time functions associated with the signatures is proposed. A consistency factor is proposed to quantify the discriminative power of different combinations of the time functions. Initial step is feature extraction, in this step normalization, extended functions like x-velocity, y-velocity, log curvature radius, consistency computation are calculated. They have used classifiers like support vector machine and random forests. They have explored different combinations of feature vectors. Dataset used in this paper is SigComp2011, it consists of Chinese and western dataset. [4]

This paper implemented on offline signature verification by using image processing, geometric feature extraction, neural network technique, resilient back propagation and radical basic function. Steps involved in their methodology are training stage and testing stage. In training stage there are four major steps those are retrieval of a signature image from a database, image pre-processing feature extraction and neural network training. In the next stage testing is done to verify the model designed. The accuracy of their system is 86.25. Their database consists of 2106 signatures containing 936 genuine and 1170 forgeries. [5]

In this paper, a novel online signature verification technique based on discrete cosine transform (DCT) and sparse representation is proposed. They obtained a compact representation of an online signature using a fixed number of coefficients, leading to simple matching procedures and providing an effective alternative to deal with time series of different lengths using DCT. It is also used to extract energy features. Sparsity features are extracted using sparse representation. Finally, energy features and sparsity features are concatenated to form a feature vector. This paper implemented offline signature verification by using image processing, Error rate in their methodology is less than 1.7. [6]

In this paper, they analysed several approaches to score normalization for dynamic time warping and propose a new two-stage normalization which detects simple forgeries in a first stage and copes with more skilled forgeries in a second stage. Their methodology consists of two stages one is feature extraction and the other is two stage normalization. Dataset used by them is MCYT online signature corpus, which contains over three hundred users, and the SUSIG visual sub-corpus, which contains highly skilled forgeries. Their results showed that two stage normalization works better in signature verification. [7]

In this paper, they have used artificial neural network using back propagation algorithm for signature verification. For calculating the accuracy and effectiveness of the model they took parameters as False Reject Rate, Equal Error Rate and False Accept Rate. They used signatures of 20 individuals consisting 400 signature samples of forged and genuine signatures. Their aim is to limit computer singularity in signature verification process. The type of Neural Network they have used is Feed Forward Neural Network. They have attained comparable speed, throughput and accuracy to the benchmark algorithms in signature verification. [8]

They used Neuro-fuzzy system in this paper for classification of signatures to find whether they are forged of genuine. They have proposed a new method which consists of partitioning such that they represent high and low pen's pressure and high and low speed of signature. Initially templates are generated and then distance between template and signatures are computed for each partition. A decision boundary is created for each partition and they have applied fuzzy rules for classification. They have used SVC2004 and bio-secure database which consists of 2000 signatures. For classification of dynamic signature velocity signal plays an important role than pressure signal. They got good accuracy comparable to other algorithms. [9]

They considered this problem of signature verification as pattern recognition of two classes. Dynamic time wrapping is used to test authenticity of reference signature and claimed user. 3 dimensional feature vector is formed. They have used principal component analysis (PCA) and linear classifier for classification purposes. After data acquisition, data pre-processing is performed and then features are selected. Finally Classifier is trained and tested. 1.4 percent of error rate is measured after classification of tested samples. [10]

They have proposed a new method in which they represented signatures by interval valued symbolic features. Methodology consists of 2 steps. First is symbolic representation of online signatures and then symbolic matching for recognition and verification. In signature representation they used mean and variance of feature vectors. They got equal error rate of 3.8 which is nine percent better than the models formed using this type of methodology. The dataset used by them consists of signatures of 330 individuals which are online signatures from MCYT signature sub corpus. [11]

They have used MCYT and SVC2004 signature verification datasets. They have used Longest Common Subsequence to find the similarity of online signatures. They have also used Support Vector Machine for classification of signatures whether they are forged or genuine. They have also made Support Vector Machine with other algorithms of kernel like Dynamic time wrapping and concluded that Support Vector machine with longest common subsequence is more accurate than any of those methods consisting of support vector machine. They also considered inclination angles of pen while doing signature on graphic tablet. [12]

This paper implemented online signature verification using hidden Markov models (HMM). They have concentrated more on the pen-tilt using a digitizer tablet and investigated the verification reliability based on different forgery types. They compare the discriminative value of the different features based on a linear discriminant analysis (LDA) and found that pen-tilt as important feature .Their methodology based on hidden Markov models reached equal-error rates between 1.0% - 1.9%. The fact that the forgers had access to both the spatial and dynamic characteristics of genuine signatures made the verification more challenging. [13]
In this paper, they implemented new warping technique they proposed its name

as extreme points warping. It proves to be more adaptive in the field of signature verification than DTW, given the presence of the forgeries. Instead of warping the whole signal as DTW does, EPW warps a set of selected important points. A signature database comprising 25 users was built. Each user donated 30 genuine signature samples and 10 forgeries in two phases with one month interval. Overall 1000 signatures were collected and stored in the database. [14]

In this paper, they implemented signature verification using fuzzy sets theory, Neuro-fuzzy systems, interpretability. The steps involved in their methodology are Partitioning of signatures, templates generation, calculation of distances between signatures and template in each partition, computation of the weights of importance, creation decision boundary for each partition, determination of the fuzzy rules used in classification phase, classification. Their results showed that signature velocity was more important than the value of pen pressure. Pen pressure in the middle and final phase of signing process was more important than in initial phase. Database used is SVC 2004, Commercial Biosecure database. [15]

Table. 1: Table showing literature survey

Sl.No	Title/Broad Area and year of publication	Details about Frameworks/Algorithms used	Details about Tools, Datasets	Summary of the research outcome
1.	Online Signature Verification on Mobile Devices-2014	An online signature is represented with a discriminative feature vector derived from attributes of several histograms that can be computed in linear time. The resulting signature template is compact and is verified with other signature for genuine or forgery.	The algorithm was first tested on the well-known MCYT-100 and SUSIG data sets	The results demonstrate the problem of within-user variation of signatures across multiple sessions and the effectiveness of cross session training strategies to alleviate these problems
2.	Embedded System for Biometric Online Signature Verification -2014	1. Field-programmable gate arrays (FPU) 2. Vector floating-point unit (VFPU) 3. DTW 4. Gaussian mixture model (GMM)	MCYT-100 data set of signatures	Acceptance rate And rejection rate are calculated.
3.	Dynamic Signature Verification System Based on One Real Signature-2016	1. Dynamic time wrapping. 2. Hidden Markov model	MCYT-100 data set of signatures	Experimental results suggest that our system, with a single reference signature, is capable of achieving a similar performance to standard verifiers trained with up to five signature specimens
4.	Dynamic Signature Verification System Based on One Lengedre polynomials Based feature extraction for online signature verification .Consistency analysis of feature combinations-2014	1. Legendre polynomials 2. Support Vector Machines 3. Random Forests.	SigComp2011 dataset, it consists of Chinese and western dataset.	Results show that there is a good correlation between the consistency factor and the verification errors, suggesting that consistency values could be used to select the optimal feature combination.
5.	Offline Handwritten Signature Verification using Neural Network-2015	1.Neural network technique 2.Resilient back-propagation 3.Radical basic function	Using a database of 2106 signatures containing 936 genuine and 1170 forgeries	The accuracy of their system is 86.25. Their neural network says whether a signature is forged or genuine.

6.	Online Signature Verification Based on DCT and Sparse Representation-2015	1) Discrete Cosine Transform 2) Sparse Representation	Sabancı University's Signature Database (SUSIG)-Visual and SVC2004 databases,	Proposed method authenticates persons very reliably with a verification performance which is better than those of state-of-the-art methods on the same databases.
7.	Robust Score Normalization for DTW-Based On-Line Signature Verification-2015	1. Dynamic time wrapping 2. Score normalization	SUSIG and MCYT dataset of signatures	The results demonstrate that score normalization is a key component for signature verification and that the proposed two-stage normalization achieves some of the best results on these difficult data sets both for random and for skilled forgeries
8.	Offline Handwritten Signature Verification System Using a Supervised Neural Network Approach - 2014	1) Artificial neural network based on the well-known Back-propagation algorithm is used for recognition and verification. 2) To test the performance of the system, the False Reject Rate, the False Accept Rate, and the Equal Error Rate (EER) are calculated.	Dataset of 1524 signatures from various resources	The experimental results for the accuracy speed and throughput showed excellent measurements that are comparable to the benchmark algorithms in the domain.
9.	New method for the on-line signature verification based on horizontal partitioning - 2014	Flexible Mamdani-type neuro-fuzzy system is used	SVC2004 and BioSecure Database	Simulations have shown that velocity signal plays more important role than pressure signal in the classification process of dynamic signature.
10.	Identity authentication using improved online signature verification method - 2005	1) Dynamic time wrapping 2) Bayes Classifier 3) SVM 4) Principal component Analysis	A data set of 94 people and 619 test signatures (genuine signatures and skilled forgeries) from various resources.	1.4% overall error rate in classifying the signatures (genuine and forgeries).
11.	Online Signature Verification and Recognition: An Approach Based on Symbolic Representation - 2009	1)False Reject Rate 2)False Accept Rate, and 3)Equal Error Rate (EER) are calculated. 4)PCA 5)FLD	MCYT_signature subcorpus consists of online signature samples of 330 individuals	Attaining EER =3.8 against the best reported PCAD method EER= 4.2, and thus achieving 9percent reduction in EER.

12.	Online Signature Verification With Support Vector Machines Based on LCSS Kernel Functions - 2010	1) Longest common subsequence (LCS) 2) Support Vector Machine (SVM) 3) DTW	Public benchmark databases for online signature verification MCYT and SVC 2004 are used.	They showed that SVM with the LCSS kernel authenticate persons very reliably and with a performance which is significantly better than that of the best comparing technique, SVM with DTW kernel.
13.	Online Signature Verification with Hidden Markov Models - 1998	1) Hidden Markov models (HMM) for signature verification 2) Linear discriminant analysis (LDA) for comparing discriminative values of different features.	Database of 1530 genuine signatures, 3000 amateur forgeries written by 51 individuals, and 240 professional forgeries.	This online signature verification system based on hidden Markov models reached equal-error rates between 1.0% and 1.9%.
14.	Online signature verification using a new extreme points warping technique - 2003	1) Dynamic time warping (DTW) 2) Extreme points warping (EPW)	Database comprises 25 users and their 1000 signatures	The EER is improved by a factor of 1.3 over using DTW and the computation time is reduced by a factor of 11.
15.	Online signature verification using vertical signature Partitioning - 2014	1) Fuzzy set theory 2) Flexible Mamdani-type neuro-fuzzy systems	Public SVC 2004 database, Commercial BioSecure database (BMDB)	Achieved good accuracy when compared to other works.

6. Methodology

We implemented two methods in our Online Signature Verification system.

6.1. Method 1

The following are the steps of first method in the methodology of Online Signature Verification system.

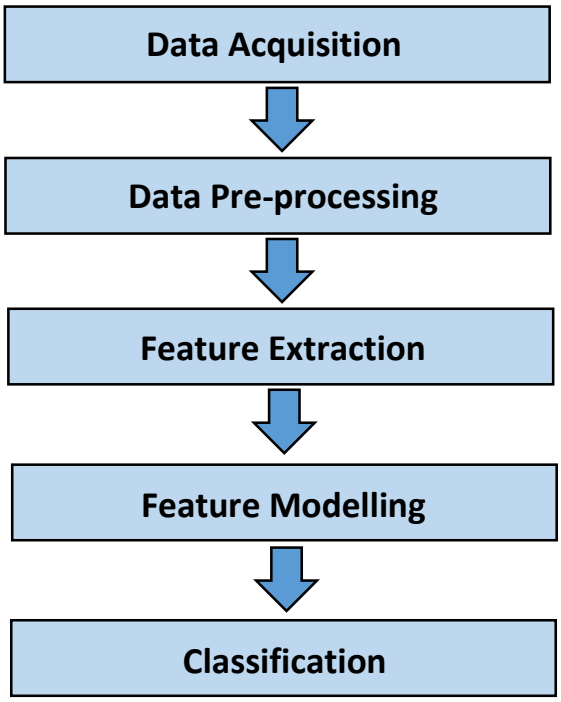


Fig. 3: Image shows the flow chart of proposed methodology

6.1. Data Acquisition

We did our signature verification process on two datasets. The first is SUSIG online dataset which is available online and the other one is dataset to be collected by us using Android application.

SUSIG online dataset which consists of signatures of 100 people containing 3000 genuine and 2000 skilled forgery signatures. The dataset contains following information i.e., X, Y, Time Stamp, Pressure, Pen Up/Down. Where X and Y are x and y coordinates, Time stamp is the time at that instant from start time, Pen Up/down indicates whether finger is touching screen or not at that instant, Pressure indicates its usual meaning.

Data Set	*Type*	*Users*	*Samples/User*	*Size*
SESSION1	Genuine	100	10	1000
SESSION2	Genuine	100	10	1000
SKILLED FORGERY	Forgery	100	5	500
HIGHLY SKILLED FORGERY	Forgery	100	5	500
VALIDATION	Genuine/Forgery	10	10/10	200

Fig. 4 Image showing summary of the SUSIG visual subcorpus dataset. (ImageSource:[17])

6.2. Data Pre-processing

Since any type of data which we collect may not be perfect or it may not be in the way in which we want Data pre-processing is necessary step before extracting features from dataset. So we are performing five steps in data pre-processing to make our dataset suitable for feature extraction.

The following are the steps in data pre-processing:

- Filtering.
- Linear Interpolation.
- Normalization.

6.2.1. Filtering

Firstly the basic and common thing which haunts any researcher after collecting the data is the noise it contains. Any type of data might contain noise even if it is collected with utmost care and exemplary conditions. So we are filtering the data which we got. Filtering is done to remove noise from the signature, with filtering the no of coordinates in the signature will decrease and signature looks crisp.

Ramer-Douglas-Peucker algorithm is applied for filtering. After application of this algorithm, signature looks crisper and noise is removed.

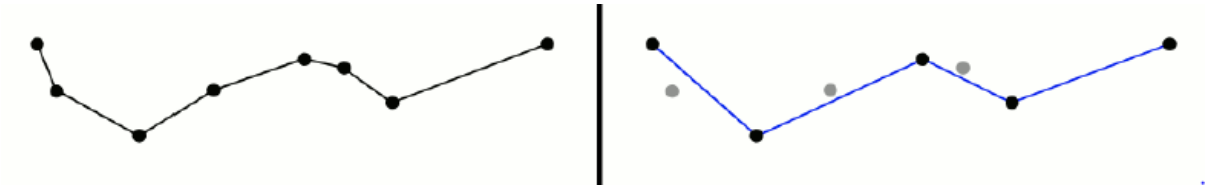


Fig. 5 (Left) Curve before applying Ramer-Douglas-Peucker Algorithm
(Right) Curve after applying Ramer-Douglas-Peucker Algorithm (ImageSource:[16])

We can notice that there is reduction in number of points after applying Ramer-Douglas-Peucker algorithm.

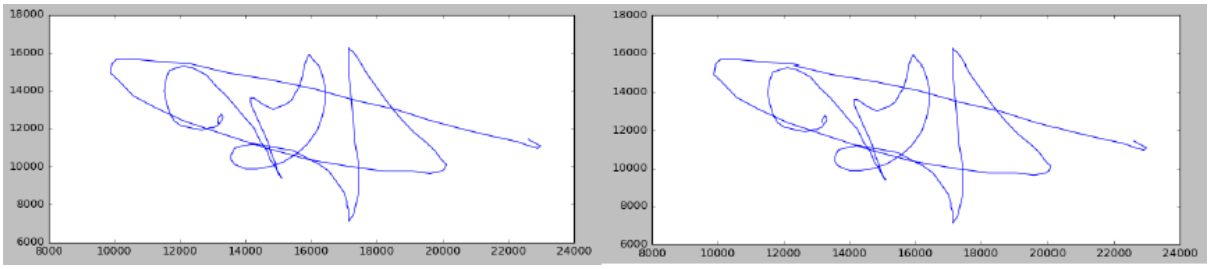


Fig. 6 (Left) Signature before applying Ramer-Douglas-Peucker Algorithm
(Right) Signature after applying Ramer-Douglas-Peucker Algorithm

It can be clearly observed that there is less or no differences between the two signatures i.e., RDP algorithm removes noise and decreases points but it doesn't change actual shape or features of original signature which can be clearly notices from the above figures.

6.2.2. Linear Interpolation

We are considering 256 points in the signature of every individual. Homogeneity is to be considered for the dataset, so we are taking 256 points which are equally spaced in a signature to maintain equal number of points for every signature of a particular user. Since to maintain constant length of feature vector the coordinates

representing a signature are interpolated. Linear interpolation is to be used to interpolate points. Linear Interpolation actually finds unknown value at a point given the values in its surrounding points. After applying linear interpolation all the obtained values are then floored to integers to avoid complexity and for convenience.

6.2.3. Normalization

Since signatures can be located at any position and they can be of any size, so they are normalized in terms of size, location and time. Three types of normalizations are used.

- **Size Normalization** is essential because any individual cannot make the signature which are exactly equal in size. So we are normalizing the signatures such that coordinates after normalization lies between 0 and 1 which advantages, how lengthy the signature could be, we are just normalizing between fixed points helping in verification.

6.3. Feature Extraction

After data pre-processing we will collect 256 x coordinates, 256 y coordinates, 256 pressure coordinates and pen up and downs. The present data is considered as the feature vector with the above mentioned columns as features.

6.4. Feature Modelling

After feature extraction we will have 10 genuine and 10 forgery signature files. Now we combine all the genuine and forgery files into one file and add an additional feature at the end which represents whether the signature is genuine or forgery by setting a 1 or 0 for that feature. Now for all users we make each file and make this as a final feature vector.

6.5. Classification

After forming feature vector final step in the verification process is classification which can be done by many machine learning algorithms like SVM, Neural Networks etc. After experimenting with the classifiers we found that the best model for our Online Signature Verification Process.

6.5.1. Support Vector Machine

Support Vector Machine is a classification algorithm which is used to find the optimal separating hyper-plane (example, a plane in a 3-dimensional space) that maximizes the margin (distance between the hyper-plane's and the closest data point) of the training data because it classifies unseen data to the best of its abilities. SVM finds a linear separating hyper-plane with the maximum margin in the higher dimensionally. It has been observed that SVM doesn't work best when classification is to be done in higher degree's, however, since the classification over here is binary (ie, forgery or genuine) SVM seems to be an ideal classifier that would fit our dataset. SVM also suits our dataset better as the features mentioned above which are a part of the feature vector as numerically less when compared to the training sample. It is also known as a black box classifier as the complex data transformations and resulting boundary plane are difficult to interpret.

Finding the optimal hyper-plane based on SVM:

The equation of hyper-plane can be written as $\mathbf{w}^{\mathbf{T}}\mathbf{x}=0$. Any hyper-plane can be written as the set of points $\mathbf{x}$ satisfying $\mathbf{w}\cdot\mathbf{u} + b = 0$ (Decision rule: if ≥ 0, then positive samples).

Considering 2-dimentional vectors let us assume that the two hyper-planes *h1* and *h2* separate the data and have the following equations.

$$\mathbf{w}\cdot\mathbf{x} + b = 1$$

$$\mathbf{w}\cdot\mathbf{x} + b = -1,$$

so that h_0 (h_0 is the optimal hyper-plane)is equidistant from *h1* and *h2*. The hyper-planes that meet the following constraints are selected where for each vector $\mathbf{x_i}$ either:

$$\mathbf{w}\cdot\mathbf{x_i} + b \geq 1 \text{ for } \mathbf{x_i} \text{ having the class 1, or,}$$

$$\mathbf{w}\cdot\mathbf{x_i} + b \leq -1 \text{ for } \mathbf{x_i} \text{ having the class } -1$$

Let us assume a function y_i such that

$y_i = 1$ for positive samples

$y_i = -1$ for negative samples

Therefore, $y_i\ (\ \mathbf{w}\cdot\mathbf{x_i} + b\ ) \geq 1$ and $y_i\ (\ \mathbf{w}\cdot\mathbf{x_i} + b\ ) \leq -1$. Using both the hyper-plane's equation, the margin calculated is

$$\boldsymbol{m} = \frac{\mathbf{2}}{||\boldsymbol{w}||}$$

Maximizing the margin is known to be the same as minimizing the norm of $\boldsymbol{w}$.

$$minimize\ \frac{1}{2}\ ||\boldsymbol{w}||^2$$

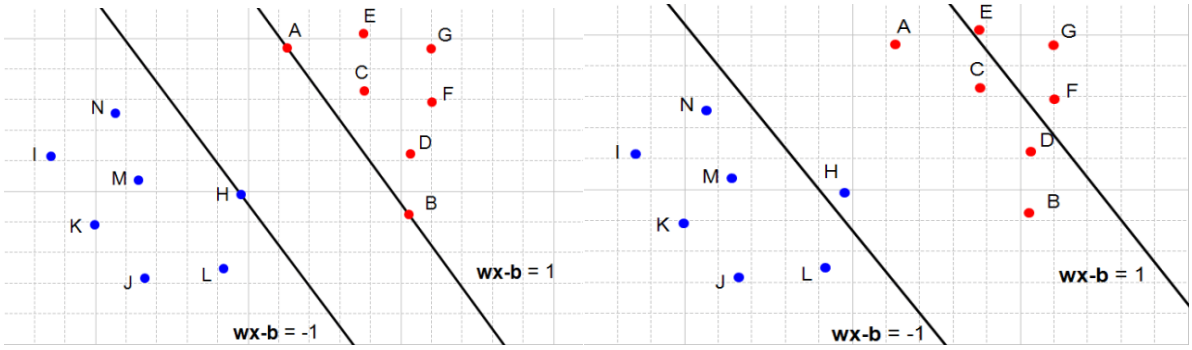


Fig. 7 (Left) Two hyperplanes satisfying the constraints

(Right) Two hyperplanes satisfying the constraints (ImageSource:[18])

If we are going to find the minimum of a function with constraints then we are going to have to use Lagrange multipliers that would give us a new expression which we can maximize or minimize without thinking of the constraints anymore.

$$L = \frac{1}{2}\ ||\mathbf{w}||^2 - \sum \propto_i\ [y_i(\boldsymbol{w}\cdot\boldsymbol{x}\mathbf{i}\ +\ \boldsymbol{b}) - 1] \quad -(1)$$

For the points that don't lie on the hyper-planes *h1* and *h2*, the value of $\propto$ is going to be 0. The partial derivatives are

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum \propto_i y_i x_i = 0$$

$$\mathbf{w} = \sum \propto_i y_i . x_i \qquad -(2)$$

$$\frac{\partial L}{\partial b} = -\sum \propto_i y_i = 0$$

$$\sum \propto_i y_i = 0 \qquad -(3)$$

Using equations (1), (2) and (3), we get

$$L = \sum \propto_i - \frac{1}{2} \sum_{i,j} \propto_i \propto_j y_i y_j x_i x_j \qquad -(4)$$

The optimization depends on the dot product of pairs in the samples, ie, $x_i . x_j$. Thus, if the decision rules follows the behavior-

$\sum \propto_i y_i x_i . \boldsymbol{u} + b \geq 0$ then, positive sample

$\sum \propto_i y_i x_i . \boldsymbol{u} + b < 0$ then, negative sample

This never gets stuck to the local maximum and keeps on evaluating its options for a better fit. Let us define a function $\varphi(\mathbf{x})$ which transforms the space to another perspective space. Support Vector Machine can also be generalized to non-linear cases. If a kernel function *K* exists such that $K(x_i, x_j) = \varphi(x_i).\varphi(x_j)$ then all the data can be transformed into a high dimensional space and linear algorithms can be used without knowing the transformation φ explicitily.

In the case of Online Signature Verification which is a binary classification problem (ie, forgery or genuine), radial basis function kernel (rbf) is used. This can be written as

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0.$$

The RBF Kernel defines a function space that is a lot larger than that of a linear kernel and polynomial kernel.

6.5.2. Random Forest

Random forest grows several classification trees. It works as follows

1. Let D be the training Dataset and it generates k bootstrap samples of D. Each bootstrap sample D(i) consists same number of tuples as that of D by sampling and replacing from D(i.e, some original tuples and some duplicated tuples of D).
2. For each D(i) a decision tree is constructed and k decision trees are formed.
3. Decision tree algorithm uses Gini index for tree generation.

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2$$

Where, pi is the probability of a tuple that belong to a class in training dataset D. Gini index gives the impurity of a dataset D.

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2).$$

This gives the Gini index of d by binary split on A(attribute) and splits into D1 and D2.

$$\Delta Gini(A) = Gini(D) - Gini_A(D).$$

This gives the reduction in impurity by splitting D on attribute A.

To classify a given test tuple X, each tree classification result is counted and X is classified with a class having the maximum count.

Finally, compare the different classification results based on precision, recall and accuracy.

Method 2

As in the above method we use the data but without performing pre-processing we directly split the data into bins and make feature vectors. Feature vector is concatenation of bins of different attributes of the signature. The attributes taken are k^{th} derivative of x and y coordinates, as well as k^{th} derivatives of pressure. We are finding the kth derivative of each attribute so as to make the feature vector as position invariant. i.e., a user cannot start the signing the signature from the same point and in the same orientation all the times he signs. So we are finding the kth derivatives of the points to solve this problem.

Now each attribute is made in to bins of regular intervals using max and min values of every attribute. For above attributes min and max values are found from the data directly. Now this entire range is divided into regular intervals called as bins. Initially the counts of all the bins are zero. Then by iterating through each tuple from the data, counts in the bins are incremented by checking to which bin that particular value belongs. Now we have bins of all attributes. Then we concatenate the bins of all attributes to form a feature vector of a single signature.

$$X^k = \{ x_i^k \mid x_i^k = x_{i+1}^{k-1} - x_i^{k-1} \}$$

$$Y^k = \{ y_i^k \mid y_i^k = y_{i+1}^{k-1} - y_i^{k-1} \}$$

$$P^k = \{ p_i^k \mid p_i^k = p_{i+1}^{k-1} - p_i^{k-1} \}$$

$$F = \{ B_1 \mid |B_2| \mid |B_3| \mid \ldots . \mid | B_k | \}$$

Where k represents total no of histograms and vector B_i contains bin frequencies of an i^{th} histogram. Therefore, an online signature is represented by a feature vector F which consists of bin information.

While performing verification on SUSIG online dataset of signatures, we considered x coordinates and y coordinates, first pen-up and maximum pressure while doing signature as features in feature vector as they are only available in the dataset.

On the other hand, when verification is performed using dataset which is to be collected from android application we can collect x and y coordinates, pressure applied at every point, average pressure, number of pen-ups and time taken for doing signature. But pressure in this case is not the actual pressure applied because to collect actual pressure applied on the screen we need to have resistive screens for our phones which are very rare when compared to capacitive screen which are very prevalent. But in capacitive screen phones it is not possible to collect actual pressure but it gives the number of pixels covered by our finger at that particular instant which in turn points to pressure applied because if more pressure is applied then number of pixels covered by our finger is more and if less pressure is applied then number of pixels covered by finger will be less on majority cases.

After getting the feature vector we can use this to classify as we did in above method. Remaining process is same as in the above method.

7. Software and Hardware Requirements

7.1. Software Requirements

- Python 2

7.2. Hardware Requirements

- 5.0” or 5.5” LCD display smart-phone with high processing power.
- OS: Ubuntu

7.3. Dataset Requirements

- SUSIG Online Signature Dataset (5000 Signatures)
- Dataset Link : http://biometrics.sabanciuniv.edu/susig.html [17]
- Self-developed data set (100-150 Signatures)

8. Implementation

SUSIG online dataset which consists of signatures of 100 people containing 3000 genuine and 2000 skilled forgery signatures. The dataset contains following information i.e., X, Y, Time Stamp, Pressure, Pen Up/Down. Where X and Y are x and y coordinates, Time stamp is the time at that instant from start time, Pen Up/down indicates whether finger is touching screen or not at that instant, Pressure indicates its usual meaning. After data collection we pre-processed the data to remove redundancy. In data pre-processing we performed filtering, interpolation, normalization. After pre-processing we used two methods for further processing. In one method we used the data directly as features. In second method, instead of using them directly we find the k^{th} derivative for the data as it helps in orientation of the signature. Now we combine all the data of a particular user in a file which contains all his genuine and forgery signatures in two methods. We add an additional feature at the end to represent whether it is a genuine or forgery by keeping 1 or 0 respectively for genuine and forgery. Now we generated the feature vectors which is used for

classification process.

We used SVM classifier to classify whether the given signature is genuine or forgery. The data in feature vectors are divided for training and testing the classifier. The classifier gets trained using the training data and draws a hyperplane which divides the classes which are genuine and forgery. After training the classifier tests the testing data and matches the predicted class with the actual class. In our case, the classifier takes the testing data as input and checks whether it is genuine or forgery and matches it with its actual feature. In this way accuracy is calculated for the overall model. For our model, the accuracy is 88 percent in first method and 80 percent in second method. It means if a signature is given it will find whether it is a genuine or forgery 88 percent accurately.

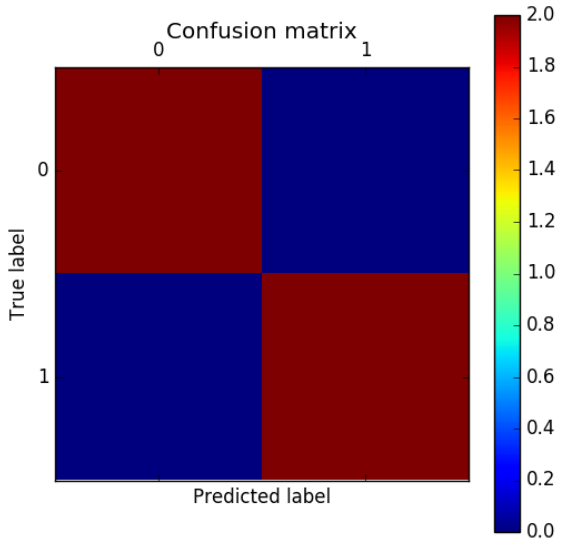
9. Observations & Results

Mentioned below are two sets of results, results obtained from the SUSIG Online Dataset and results obtained from the dataset collected for an Android Application. The SUSIG Online Dataset consisted of 20 genuine and 10 forgery signature of 115 people. Therefore, each user has 30 signatures associated with him, irrespective of genuine or forgery. In the case of the dataset collected from the Android Application, a dataset of 5 people were collected. Each person has 10 genuine signatures and 5 forgery signatures.

9.1. Results for SUSIG Online Dataset

Below is the confusion matrix of one random user out of 115 people. Since each user had a separate classifier implemented, 115 confusion matrices were generated. Shown below is the variation of the confusion matrix with respect to the training and testing data statistics. In cases where either 20%, 30% or 40%

of the data is used for testing and 80%, 70% and 60% of the data is used for training respectively, the below confusion matrices were generated.



20% testing – 80% training

Matrix: $\begin{matrix} 2 & 0 \\ 0 & 2 \end{matrix}$

30% testing – 70% training

Matrix: $\begin{matrix} 3 & 0 \\ 0 & 3 \end{matrix}$

40% testing – 60% training

Matrix: $\begin{matrix} 4 & 0 \\ 0 & 5 \end{matrix}$

Fig. 8: Image showing confusion matrix of an individual user

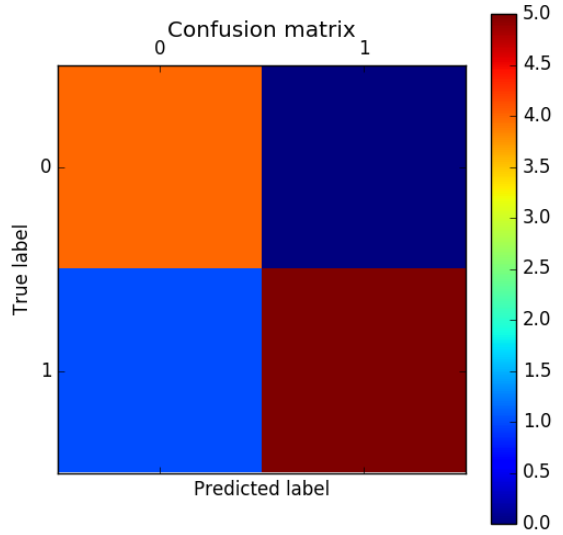


Fig. 9: Image showing confusion matrix of an individual user

In the case where 50% of the data was used for testing and 50% for training-

50% testing – 50% training

Matrix: $\begin{matrix} 4 & 0 \\ 1 & 5 \end{matrix}$

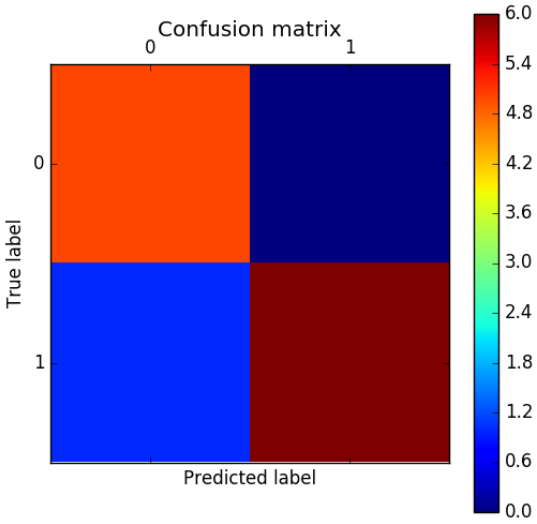


Fig. 10: Image showing confusion matrix of a user

In the case where 60% of the data was used for testing and 40% for training-

60% testing – 40% training

Matrix: $\begin{matrix} 5 & 0 \\ 1 & 6 \end{matrix}$

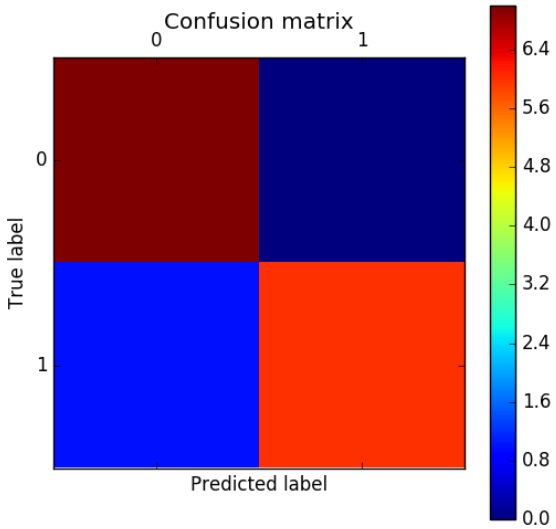


Fig. 11: Image showing confusion matrix of a user

In the case where 70% of the data was used for testing and 30% for training-

70% testing – 30% training

Matrix: $\begin{matrix} 7 & 0 \\ 1 & 6 \end{matrix}$

```
For File 107.sig 0.833333333333
Confusion Matrix :
[[2 1]
 [0 3]]
SCORE for user 107 : 0.833333333333
Actual Value :  [ 1.  0.  1.  0.  1.  0.]
Predicted Value :  [ 1.  1.  1.  0.  1.  1.]
For File 108.sig 0.666666666667
Confusion Matrix :
[[1 2]
 [0 3]]
SCORE for user 108 : 0.666666666667
Actual Value :  [ 1.  0.  1.  0.  1.  0.]
Predicted Value :  [ 1.  0.  1.  0.  1.  0.]
For File 109.sig 1.0
Confusion Matrix :
[[3 0]
 [0 3]]
SCORE for user 109 : 1.0
Actual Value :  [ 1.  0.  1.  0.  1.  0.]
Predicted Value :  [ 1.  0.  1.  0.  1.  0.]
For File 110.sig 1.0
Confusion Matrix :
[[3 0]
 [0 3]]
SCORE for user 110 : 1.0
Actual Value :  [ 1.  0.  1.  0.  1.  0.]
Predicted Value :  [ 1.  1.  1.  0.  1.  0.]
```

Fig. 12: Image showing details of some users processing in method 2

In the above image we can see the confusion matrix for a particular user, accuracy, actual and predicted class values for his signatures.

9.1.1. Cumulative Results:

1) For SVM:

Total number of users = 115

Total number of testing files = 690

Accuracy = 88.77

Matrix: $\begin{matrix} 225 & 60 \\ 4 & 281 \end{matrix}$

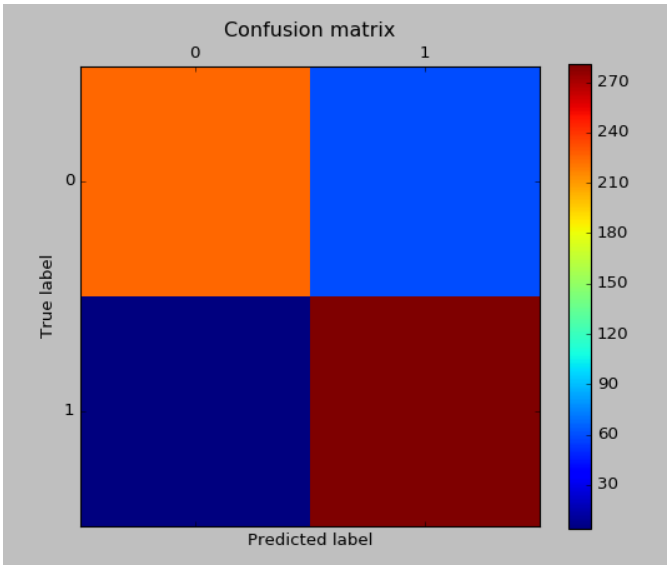


Fig. 13: Image showing confusion matrix for SVM classifier

```
Accuracy for SVM algorithm :  88.7719298246
225
60
4
281
[[225  60]
 [  4 281]]
```

Fig. 14: Image showing accuracy and confusion matrix for SVM classifier

2) For Random Forest classifier:

Total number of users = 115

Total number of testing files = 690

Accuracy = 91.75

Matrix: 257 28
19 266

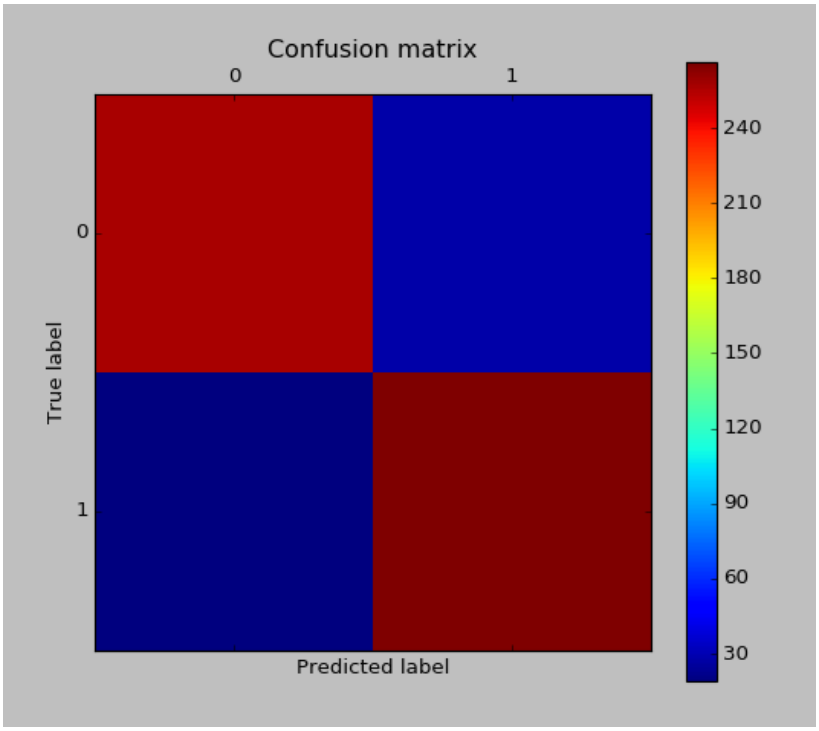


Fig. 15: Image showing confusion matrix for Random Forest classifier

```
Accuracy for Random Forest classifier :  91.5789473684
255
30
18
267
[[255  30]
 [ 18 267]]
```

Fig. 16: Image showing accuracy and confusion matrix for Random Forest classifier

9.2. Dataset collected by our Android Application:

Below is the confusion matrix of one random user out of 5 people. The dataset was accumulated using 2 Android devices. Since each user had a separate classifier implemented, 5 confusion matrices were generated. Shown below is the variation of the confusion matrix with respect to the training and testing data statistics.

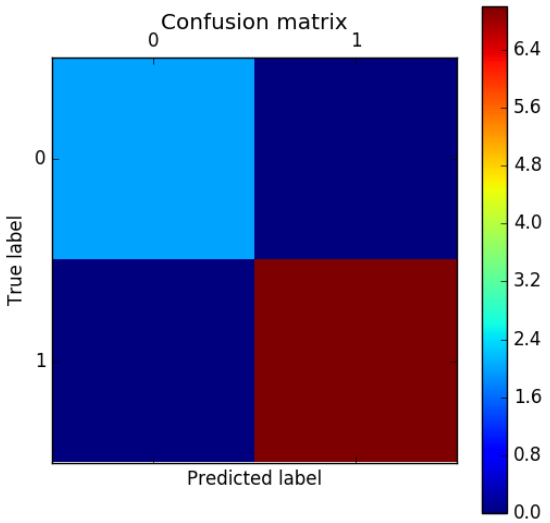


Fig. 17: Image showing confusion matrix of a user

In the case where 30% of the data was used for testing and 70% for training-

30% testing – 70% training

Matrix: $\begin{matrix} 1 & 0 \\ 0 & 4 \end{matrix}$

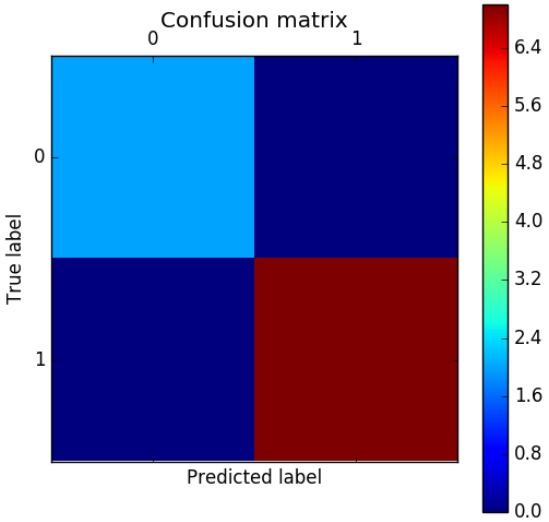


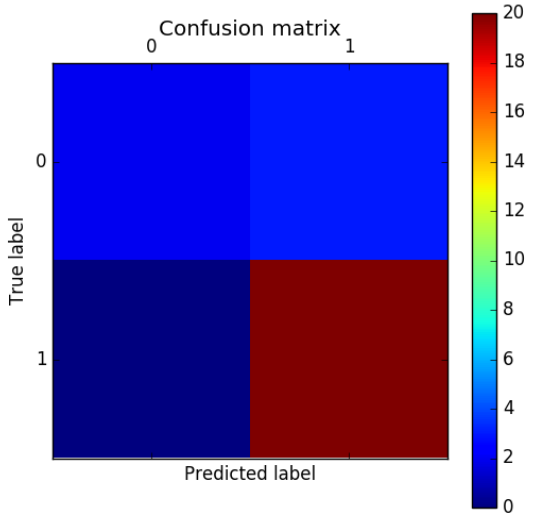
Fig. 18: Image showing confusion matrix

In the case where 60% of the data was used for testing and 40% for training-

60% testing – 40% training

Matrix: $\begin{matrix} 2 & 0 \\ 0 & 7 \end{matrix}$

Cumulative Results:



Total number of users = 5

Total number of testing files= 25

Accuracy= (2+20)/25 = 0.88

Matrix: $\begin{matrix} 2 & 3 \\ 0 & 20 \end{matrix}$

Fig. 19: Image showing confusion matrix

10. Comparison

We have used two methods for feature extraction and two methods for classification. Two methods in feature extraction are one with direct coordinates and the other using kth derivative and bins. Two methods in classification are SVM and Random Forest.

Result of SVM:

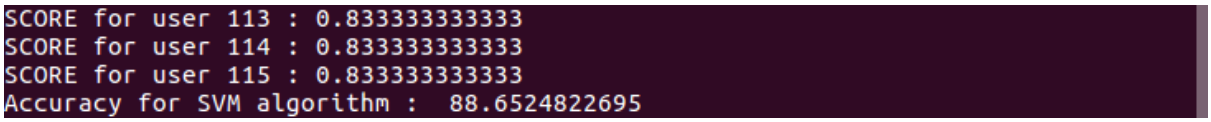


Fig. 20: Image showing accuracy of SVM classifier for method 2

Result of Random Forest Classifier:

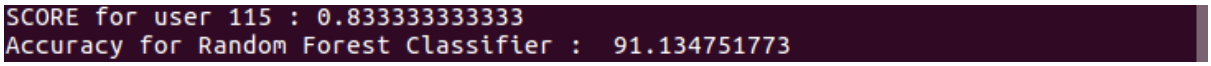


Fig. 21: Image showing accuracy of Random Forest classifier for method 2

Table. 2: Table showing accuracy using different methods and classifiers

Methods	SVM	Random Forest
Method-1	80.1	83.3
Method-2	88.64	91.13

Both the classifiers are showing comparable results with Random Forest getting higher accuracy than SVM and also running time of both the methods and classification algorithms doesn't have notable difference.

11. Conclusion

It can be concluded from the above table that Method-2 got higher accuracies than Method-1 and also Random Forest out performs SVM in both the methods. Hence it is better to use Random Forest algorithm for online signature verification. We are acquiring good accuracies comparable to other algorithms in this field.

12. References

[1] Napa Sae-Bae and Nasir Memon, “Online Signature Verification on Mobile Devices”, IEEE transactions on information forensics and security, Vol. 9, No. 6, June 2014.

[2] Mariano López-García, Rafael Ramos-Lara, Oscar Miguel-Hurtado, and Enrique Cantó-Navarro, “Embedded System for Biometric Online Signature Verification”, IEEE Transactions on industrial informatics, Vol. 10, NO. 1, February 2014.

[3] Moises Diaz, Andreas Fischer, Miguel A. Ferrer, Réjean Plamondon, “Dynamic Signature Verification System Based on One Real Signature”, IEEE Transactions on Cybernetics 2016.

[4] Marianela Parodi, Juan C.Gómez, “Legendre polynomials based feature extraction for online signature verification. Consistency analysis of feature combinations”, Elsevier Volume 47, Issue 1, January 2014, Pages 128-140.

[5] Pallavi V. Hatkar, Prof.B.T.Salokhe, Ashish A.Malgave, “Offline Handwritten Signature Verification using Neural Network”, International Journal of Innovations in Engineering Research and Technology [IJIERT], Vol 2, Issue 1, Jan 2015.

[6] Yishu Liu, Zhihua Yang, and Lihua Yang, “Online Signature Verification Based on DCT and Sparse Representation”, IEEE Transactions on Cybernetics,Vol 45, No. 11, Nov 2015.

[7] Andreas Fischer, Moises Diaz, Rejean Plamondon, Miguel A. Ferrer, “Robust Score Normalization for DTW-Based On-Line Signature Verification”, 2015 13th International Conference on Document Analysis and Recognition (ICDAR).

[8] Krzysztof Cpałka, Marcin Zalasiński, “On-line signature verification using vertical signature partitioning”, Elsevier Volume 41, Issue 9, July 2014, Pages 4170-4180.

[9] Mujahed Jarad, Dr. Nijad Al-Najdawi, Dr. Sara Tedmori, “Offline Handwritten Signature Verification System Using a Supervised Neural Network Approach”, IEEE 2014 6th International Conference on CSIT.

[10] Alisher Kholmatov, Berrin Yanikoglu, “Identity authentication using improved online signature verification method”, Elsevier Volume 26, Issue 15, November 2005, Pages 2400-2408.

[11] Krzysztof Cpałka, Marcin Zalasiński, Leszek Rutkowski, “New method for the on-line signature verification based on horizontal partitioning”, Elsevier Volume 47, Issue 8, August 2014, Pages 2652-2661.

[12] D.S. Guru and H.N. Prakash, “Online Signature Verification and Recognition:An Approach Based on Symbolic Representation”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 31, No. 6, June 2009.

[13] Christian Gruber, Thiemo Gruber, Sebastian Krinninger, and Bernhard Sick, “Online Signature Verification with Support Vector Machines Based on LCSS Kernel Functions”, IEEE Transactions on Systems, Man and Cybernetics – Par: B Cybernetics, Vol. 40, No. 4, Aug 2010.

[14] J.G.A. Dolfing, E.H.L. Aarts and J.J.G.M. van Oosterhout, “On-line Signature Verification with Hidden Markov Models”, Fourteenth International Conference on Pattern Recognition, 20 Aug 1998.

[15] Hao Feng, Chan Choong Wah, “Online signature verification using a new extreme points warping technique”, Elsevier Volume 24, Issue 16, December 2003, Pages 2943-2951.

[16] Mokrzycki, Wojciech & M, Samko. (2012). New version of Canny edge detection algorithm. 533-540.

[17] @article{SUSIG, author="Kholmatov, A. and Yanikoglu, B.", Title="SUSIG: an on-line signature database, associated protocols and benchmark results", journal="Pattern Analysis and Applications", volume = {12}, number = {3}, year = {2009}, pages = {227-236}}

[18] @article{author = “Alexandre KOWALCZYK”,link = “https://www.svm-tutorial.com/2015/06/svm-understanding-math-part-3”, year = {2015}}