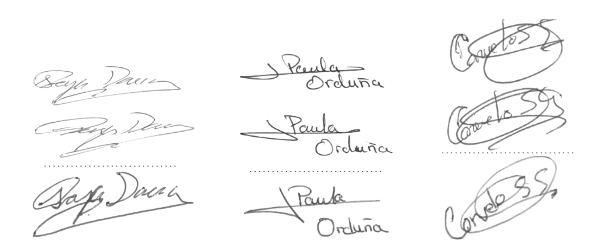
1. **INTRODUCTION**

Signatures are generally used for authentication of an individual. Aim of the signature verification system is to verify the uniqueness of an individual based on the analysis of the signature. Signature consists of graphical symbols on the surface in relation to a language. Signatures of the same person can vary with time. However, Signature is a unique feature for individual’s identification. Now a day’s massive number of transactions are authorizing via signature. A handwritten signature is a socially and legally accepted biometric trait for authenticating an individual. Typically, there are two types of handwritten signature verification systems: offline and online systems. In an off-line system, just an image of the user’s signature is scanned to obtain its digital image representation whereas, in an online system, a sequence of x-y coordinates of the user’s signature, along with associated attributes like pressure, time, etc., are also acquired. Like in other biometric verification systems, first, users are enrolled to the system by providing references. Later when a user presents a signature that is claimed to be a particular individual, the system compares this signature with the reference signatures for that individual. If the dissimilarity exceeds a certain threshold, the signature is rejected. The aim of this project is to apply the system identification technique to the analysis of the well-known personal signature characteristics.



To address both the issue of obtaining a good feature representation for signatures, as well as

Improving classification performance, we propose a framework for learning the representations directly from the signature images, using convolutional neural networks. In particular, we propose a novel formulation of the problem that incorporates knowledge of skilled forgeries from a subset of users using a multi-task learning strategy. The hypothesis is that the model can learn visual cues present in the signature images that are discriminative between genuine signatures and forgeries in general.

1. **PROBLEM STATEMENT**

Signature verification through inspection has really been a tough task and is creating several problems and also several forgery cases have occurred due to lack of good system for verification of signature. Also in several criminal cases, the failure of true verification of signature might have caused several incorrect decisions. Thus it’s the need of time to create a good system which can verify signature and thus differentiate a fraud signature with respect to that of original one.

1. **Literature Review**

The area of Handwritten Signature Verification has been broadly researched in the last decades, but remains an open research problem. The objective of signature verification systems is to discriminate if a given signature is genuine (produced by the claimed individual), or a forgery (produced by an impostor). This has demonstrated to be a challenging task, in particular in the offline (static) scenario that uses images of scanned signatures, where the dynamic information about the signing process is not available. Much advancement has been proposed in the literature in the last 5-10 years, most notably the application of Deep Learning methods to learn feature representations from signature images.

In [1] “Offline Signature Recognition Using Global Features”, the authors Ms. Pallavi Patil, Ms. Archana Patil, presents the method which is most popular biometric methods in the field of authentication of personal. In this Global features are extracting main features like area, height and breadth. Euclidean distance model is used while finding match between test signature and signature stored in the database. The algorithm gives 89% satisfactory results of the recognition by the proposed method in [1].

In [2] “Signature Verification system Based On Support Vector Machine Classifier”, the authors Ahmed Abdelrahman, Ahmed Abdallah, represents an offline signature verification system using Support Vector Machine technique, global features abstracted from the signatures using random transform. For every registered user, database is maintained. Two signatures are aligned usings dynamic time wrapping algorithm, 82% satisfactory results were achieved in [2].

In [3] “Signature Verification Using SVM”, the authors [3] highlights the development of online signature verification system using SVM to verify the input signature by comparing database. The signature is characterized as pen-strokes consisting x-y coordinates and the data will be stored in the signature database.

1. **OBJECTIVES**

The objectives of Signature verification system are as follows:

* To verify the input signature using SVM

1. **SCOPE AND LIMITATIONS**

The scopes of signature verification system are as follows:

* This system can be applied in banking sector where 20 specimen signatures of each account holder are taken. Using these methods we can verify fake and frauds in banks.
* A reliable signature verification system is an important part of law enforcement, security control and many business processes.
* It can be used in many applications like cheque, certificates, contracts etc.

As every invention and technology has the pros and cons. This application, too has some limitations along with many of its own limitations. Some of its limitations are mentioned below:

* Time consuming in banking sectors.
* Difficult to implement in rural areas.

1. **METHODOLOGY**

The research method for this module is discussed in this chapter. The different sections are:

* 1. **Data Collection**

The signatures are being collected from the different persons time and again at different time at different circumstances. Those collected data are kept at different databases to keep safely as the data provides good environment for future.

* 1. **Preprocessing**

Preprocessing is usually done manually by the humans in regard to the data sets. Here supervised machine learning technique is applied with the already gathered data sets and the result is to be expected as per the domains. The main problem of preprocessing is time, if there are large data sets processing time goes exponentially .So to reduce time some techniques of segmentation is to be used.

* 1. **Proposed Implementation Model**

In implementation model, we will be implementing the Support Vector Machine Algorithm (SVM) .The algorithm is simulated at different time for different domains data. The implementation model is given below:

Input Signature

Preprocessing

Feature Extraction

SVM Algorithm

Result Display

* 1. **Feature Extraction**

Extracted features in this phase are the inputs of training phase. The features in this system are global features, mask features and grid features. Global features provide information about specific cases of the signature shape. Mask features provide information about directions of the lines of the signatures. Grid features provide overall signature appearance information.

* + 1. **Global Features**

**Signature Area:** Signature area the number of pixels which belong to the  
signature. This feature provides information about the signatu  
re density.

**Signature Height and Width:** Signature obtained by dividing signatu  
re height to signature width. Signature height and width  
can change. Height-to-width ratios of one person’s signatures  
are approximately equal.

* + 1. **Mask Features**

Mask Features provides information about direction of the lines of the signatures. The angle of the signatures have interpersonal differences. In this features different masking protocols are applied with the same masking nature.

* + 1. **Grid Features**

Grid features are used for finding densities of the signature. Signature area is divided into different equal parts and the image area in each part is calculated.

* 1. **Learning and Training**

After the feature extraction phase process of learning and training begins. In the training phase recognition system learns patterns of different domains from input domains. The supervised learning method is applied as learning and training. System is then tested against testing domains and accuracy and efficiency of the system is calculated as per the correctness of the model. We tend to use Support Vector Model algorithm for all the process.

* + 1. **SVM**

We have a set of training data for a two-class problem: { (x1,y1),………,(xN, yN)} ,where **xi**∈RD is a feature vector of the ith sample in the training data and y ∈{+1 ,- 1 } is the class to which x belongs. In their basic form, a SVM learns a linear hyperplane that separates the set of positive examples from the set of negative examples with maximal margin(the margin is deﬁned as the distance of the hyperplane to the nearest of the positive and negative examples). In basic SVMs framework, we try to separate the positive and negative examples by hyperplane written as: *(****w*** *.****x****)+b =0* ***w*** *∈****Rn*** *,b∈* ***R*** .

If data are linearly separable then there exist a d-dimensional vector **w** and a scalar *b* such that

|  |  |
| --- | --- |
|  | (1) |

And

|  |  |
| --- | --- |
|  | (2) |

In compact form we may combine these two equations in

|  |  |
| --- | --- |
|  | (3) |

|  |  |
| --- | --- |
| Or | (4) |

Here (w, b) define the hyper plane that separates data in two class. The equation of the hyperplane is

|  |  |
| --- | --- |
|  | (5) |

Where *w*is normal to the plane, *b* is the minimum distance from the origin to the plane. In order to make each decision surface (*w, b*) unique, we normalize the perpendicular distance from the origin to the separating hyperplane by dividing it by |w| giving the distance as**.**

the perpendicular distance from the origin to hyper plane H1:

|  |  |
| --- | --- |
|  | (6) |

And the perpendicular distance from the origin to hyper plane H2:

|  |  |
| --- | --- |
|  | (7) |

The support vectors are defined as the training points on *H*1 and *H*2. Removing any points not on those two planes would not change the classification result, but removing the support vectors will do so. The margin, the distance between the two hyperplane *H*1 and *H*2 is . The margin determines the capacity of the learning machine which in turn determines the bound of the actual risk the expected test error. The wider the margin the smaller is *h*, the VC-dimension of the classifier. Therefore our goal is to maximize margin or equivalently minimize the.

Therefore the optimization problem can be formulated as follows

Minimize f= (8)

Subject to constraints (9)

The above SVM formulations require linear separation. The real life application data are not always linearly separable. To deal with nonlinear separation, the same formulation and techniques as for the linear case are still used.

To extend SVM to cases in which the data are not linearly separable, we introduce the hing loss function,

*max (0,1 –yi(w.xi-b)).*

Note that yi is the ith target and ( w → ⋅ x → i − b ) {\displaystyle ({\vec {w}}\cdot {\vec {x}}\_{i}-b)} (w.xi-b) is the current output.

We then wish to minimize by:



(10)  
max ( 0 , 1 − y i ( w → ⋅ x → i − b ) ) . {\displaystyle \max \left(0,1-y\_{i}({\vec {w}}\cdot {\vec {x}}\_{i}-b)\right).} w → ⋅ x → − b = 0 , {\displaystyle {\vec {w}}\cdot {\vec {x}}-b=0,\,}

**6**.**6 Performance**

**Precision:**

The number of correctly retrieved signatures by the system divided by the number signatures retrieved by the system. Mathematically,

**Recall:**

The number of signatures retrieved by the system divided by the number of signatures present in the test set. Mathematically,

**F-Measure:** Harmonic mean of precision and recall. Mathematically,

1. **Project Schedule**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Year | | 2018 | | | | | | | |
| Activity | 1 Week | 2 Week | 3 Week | 2 Week | 3 Week | 2 Week | 2 Week | 1 Week | 2 Week |
| Collection of Literature |  |  |  |  |  |  |  |  |  |
| Study of Literature |  |  |  |  |  |  |  |  |  |
| Analysis of proposed Scheme |  |  |  |  |  |  |  |  |  |
| Preparation And Abstract submission |  |  |  |  |  |  |  |  |  |
| Preparation Of Model |  |  |  |  |  |  |  |  |  |
| Implementation and debugging |  |  |  |  |  |  |  |  |  |
| Analysis and Simulation |  |  |  |  |  |  |  |  |  |
| Result Formulation |  |  |  |  |  |  |  |  |  |
| Final Write-up& Submission |  |  |  |  |  |  |  |  |  |

# **EXPECTED OUTCOME**

At the end of this project, the proposed model is to be expected to authenticate the signature using SVM model. More over the accuracy of the proposed model is calculated using precision and recall.

1. **REFERENCES**

[1] Ms Pallavi Patil, Ms Archana Patil, “Offline Signature Recognition Using Global Features,” International Journal of Emerging Technology and Advanced Engineering, vo.l3, Issue, Jan 2013.

[2] Ahmed Abdelrahaman, Ahmed Abdallah, “Signature Verification System based on Support Vector Machine Classifier,” International Arab Conference on Information Technology, vol.4, no.2, pp.521-555, 2013.

[3] “Signature Verification System Using SVM”, Mechatronics and its Applications, 2009.

[4] Larkins, R. Mayo, M., “Adaptive Feature Thresholding for Off-Line Signature Verification”, In: Image and vision computing New Zealand, 2008, pages: 1-6.

[5] C. Cortes and V. Vapnik Support-vector networks, Machine Learning, vol 20, pp, 273-297, Nov, 1995.

[6] Jain, F. Griess, and S. Connel, “Online Signature Recognition”, Pattern Recognition vol.35, pp 2963-2972, 2002.

[7] Y. Qiao, J. Liu, and X. Tang, “Offline signature verification using online  
handwriting registration,” 2013 IEEE Conference on Computer Vision and  
Pattern Recognition, vol. 0, pp. 1–8, 2007.

[8] G. S. Eskander, R. Sabourin, and E. Granger, “A bio-cryptographic system  
based on offline signature images.” *Inf. Sci.*, vol. 259, pp. 170–191, 2014.

[9] J. Coetzer, “Off-line signature verification,” Ph.D. dissertation, University of  
Stellenbosch, South Africa, 2005.

[10] “b2bedocuments - biometric signature,” http://www.b2bedocuments.com/  
html/biometricsignature02.htm.