International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 3, Issue 6, November-December 2014 ISSN 2278-6856

Enhanced Online Signature Verification System

Joslyn Fernandes[1], Nishad Bhandarkar[2]

[1]F/8, Malinee Apt., Mahakali Caves Road, Andheri east, Mumbai 400093.

[2]303, Meena CHS. LTD., 7 bungalows, Andheri west, Mumbai-400061.

Abstract

Signature has always been and still is widely used as a validation tool to identify the user or his intent in civil society. Online signature is nothing but the dynamic process of handwriting as a feature vector sequence along time. Dynamic time warping (DTW) has been used as a popular method to compare sequence data. This can be used as a security system for verification of a person's identity, for assessing entry application and also as a password substitution system. The technology used for signature verification requires mainly a digitizing tablet and a particular pen connected to the universal serial bus port (USB port) of a computer. When an individual signs on the digitizing tablet using this specialized pen, the features of the signature are captured and a data set is obtained, irrespective of his signature size and position. The signature is represented as pen-strokes and they consist of x-y coordinates and pressure. The data set thus obtained will be stored in the signature database in the form of a .txt file. These characteristics uniquely identify a person and are extremely difficult to mimic and impossible to steal. This system when compared to offline signature verification is much more reliable. It has a tendency of rejecting 10% of true signatures and accepting 30% of forged signatures using linear discriminant analysis whereas it rejects 5% of true signatures and accepts 2% of forged signatures using quadratic discriminant analysis.

Keywords:- Online Signature Verification, C#, MATLAB, Biometrics

1. INTRODUCTION

Signature is a biometric attribute. It is used to confirm various financial transactions, officially approve contents of some documents, etc. Signature verification is generally done by considering just the visual appearance of the signature. However the image or the visual appearance of the signature is not enough for verification. Signature is the only biometric entity that provides dynamic characteristics along with morphological characteristics. It is difficult to construct a signature with similar shape and identical dynamic characteristics at the same time. This paper aims at creating a module which can verify a signature on the basis of its time based features by digitizing the signature. This will reduce forgeries and fraud cases. Online signature verification can be used in various applications where handwritten signatures are currently collected such as cashing a check, authenticating legal documents and other banking transactions where PIN codes are used.

1.1 Literature Survey

There are two major aspects of signature present and they are static or offline features and the other dynamic or online features [1]. On weighing both, comparatively it was realized that the information content in online signature is more significant as compared to the offline signature. Thus the dynamism of signature was explored upon. At start, the most vital need was the hardware or sensor that was to be selected. This hardware called as a digitizer tablet or pressure pad is the foundation over which the entire project depends. The basic necessity for the hardware would be highly pressure sensitive, sleek and small. The other part of the project demanded a study or research of different methods of online signature verification and deciding which would be an appropriate one. Thus the following methods which have been used for online signature verification in recent years were studied.

A. Feature Based Approach

It is a vector representation method where the parameter model proposed a maximum distance for a subject, by ordering the feature.

B. Function Based-Local Method

Here time sequences describe local properties of the signature by Dynamic Time Warping (DTW) and differential geometric shape analysis [3] which matches the time function of different signatures by using elastic distance measures.

C. Function Based-Regional Method

A Hidden Markov model (HMM) performs stochastic matching of model and a signature, using a sequence of probability distributions of the features along the signature [2].

D. Hybrid Approach Method

The system proposed was a combination of function and parameter models of a signature.

1.2 Types of Signature Verification

Signature verification can be split into two categories as shown below

1. Static or Off-line
2. Dynamic or On-line

When we consider the 1st type i.e. off –line mode, users sign on a paper, and that signature is digitized by using a camera or an optical scanner and the biometric system verifies the signature after analyzing its shape.

On the other hand, in on-line mode, users sign on a digitizing tablet such as a pressure tablet device [3], which acquires the particular signature in real time. Another way of acquiring signatures by means of a stylus- operated PDA. There are three types of forgeries that can be established for any signature verification system, depending on the environment and the testing conditions [4].

1) Simple forgery- where the forger makes no attempt at all to simulate or trace a genuine signature.
2) Random forgery- where the forger uses his/her own signature as a forgery.
3) Skilled forgery- where the forger tries to purposely imitate the static and dynamic information of the signature to be forged.

An "Online Signature" is a biometric modality that uses, for verification purposes, the anatomic and behavioral characteristics that an individual exhibits when signing his or her name.

2. DATA-ACQUISITION

2.1 Choice of Software

C# and MATLAB are chosen as the main programming languages. C# because it is a object oriented language derived from C++ and java. Also it is a very simple and modern language that aims to combine the raw power of C++ and the high productivity of Visual Basic. Windows applications and web application can be developed using C#. In C#, Microsoft has taken care of C++ problems such as Memory management, pointers, etc. It supports garbage collection, automatic memory management. C# is very powerful and simple for building interoperable scalable and robust applications. MATLAB is a simple, easy to learn language. MATLAB is not only a programming language but programming environment as well. It has various inbuilt functions to perform basic mathematical operations. It is a powerful tool for implementing basic algorithms too. The important feature of MATLAB interface is the 'help' using which one can learn anything related to MATLAB programming. It is also possible to interface MATLAB with C#.

2.2 Choice of hardware

A digitizing tablet for capturing the signature in real time is needed. So after correspondence with the various digitizing tablet like Wacom Intous, Wacom Bamboo [3], Cintique and Bamboo-fun, it was found that the feature and cost of the Wacom Bamboo Pen tablet was most suitable for our requirements.

2.3Working of Digitizer Tablet

WACOM stylus makes use of electro-magnetic resonance technology. In operation, a grid of wires below the screen alternates between transmitting and receiving mode.

2.4 GUI using C#

Online signature verification uses digitizer tablet (WinTab compatible digitizer) to acquire dynamic data for verification. It is possible to interface the digitizer tablet in C# using VB Tablet component [5]. VB Tablet is a COM component used to interface digitizer devices. Once the digitizer is connected to the PC it is interfaced as a VB Tablet object in C# code. Visual basic is required to debug and interface digitizer using this code [6].

2.5 Integrating MATLAB with C#

Integrating digitizer with C#, we get context, x component, y component, pressure at each sampling point. As we have selected MATLAB as another programming language to align and verify signatures, we need to pass the parameters obtained from the C# interface into MATLAB. It is required to interface MATLAB with C# for the above purpose.

2.6 GUI using MATLAB

The GUI using C# allows the user to enter its signature and save it if it's fine by the user on button click. The extracted parameters of the signature are saved in a .txt file. C# is interfaced with MATLAB using .NET assembly. As soon as the user clicks the save button C# gets interfaced with MATLAB and opens another GUI window in MATLAB using MCR. The C# GUI is for input purpose while the MATLAB GUI is for the verification and database creation purpose.

3. FEATURE EXTRACTION

3.1 Feature Selection

The raw signature data available from the tablet consists of three dimensional series data as represented by equation (1.1), where x(t), y(t) is the pen position at time (t) and p(t) ϵ {0,1,…,1024} represents the pen pressure.

$$S = [x(t), y(t), p(t)]^T \quad t = 0,1,2,\ldots,n \qquad (1)$$

Most of the existing systems resample the signature so that the redundant points can be removed. This speeds up the comparisons and also helps in obtaining a shape-based representation, removing the time dependencies. However, re-sampling also results in significant loss of information since the seemingly redundant data incorporates speed characteristics of the genuine signer. Another problem with re-sampling is that the critical points of the signature may be lost; critical points are sometimes added separately to the set of equidistant points obtained after re-sampling to solve this problem. The benefits of not re-sampling significantly outweigh the disadvantage of not normalizing for speed [1]. For this system we have used following three local features and three features derived from the local features to obtain minimum error rate.

Δx: Change of x between two successive sampling points
Δy: Change of y between two successive sampling points
Δp: Change of p between two successive sampling points
$\Delta y/\Delta x$: Ratio of Δy to Δx
$\Delta p/\Delta x$: Ratio of Δp to Δx
v: Change of speed between two successive sampling points

The x and y coordinate differences between two consecutive points (Δx, Δy) give the lowest error rates [7].

Note that the Δx, Δy features are invariant with respect to translation.

3.2 Enrollment

While enrollment the user supplies a number of signatures which is used in measuring the variation in a user's signature and it is later used for training and verification process. Firstly the supplied signatures are aligned pairwise to find the distance between each pair. Using these alignment scores, we first select the reference signature with the minimum average distance to all other supplied signatures and designate it as template signature. Specifically, for a reference set R_{ID}, we compute average values over the reference set, for

Distances of reference signatures to their nearest neighbor $(\overline{d_{min}}(R_{ID}))$

Distances of reference signatures to their farthest neighbor $(d_{max}(R_{ID}))$

Distances of reference signatures to the template signature $\left(\overline{d_{temp}}(R_{ID})\right)$

The computed average distance values describe the user's variation to some extent and are selected so as to be used in normalizing a signature's min, max and template distances to its reference set. By normalizing these distances by the corresponding reference set averages, the need to calculate user-dependent thresholds is eliminated.

3.4Dynamic Time Warping (DTW)

In both training and verification stages, the need to align the signatures to calculate their distances is present. In order to compare and calculate distance between signatures of different lengths, the Dynamic Time Warping (DTW) algorithm is used, which is a widely used algorithm for aligning vectors of different lengths.

4. Training Classifiers

A training data set consisting of 12 genuine signatures and 3 skilled forgery signatures per person is collected in order to train classifiers used in the verification process. First, each training signature (Y) is compared to the signatures in the reference set (R_{ID}) it claimed to belong to, giving a three distance values ($d_{min}(Y, R_{ID})$, $d_{max}(Y, R_{ID})$, and $d_{template}(Y, R_{ID})$). Note that the same normalization process is done both for training signatures during the training process, and for testing signatures during the testing process. These distance values are then normalized by the corresponding averages of the reference set, to give the three-dimensional feature vector (F_y)

$$F_y = \begin{bmatrix} d_{min}(Y, R_{ID}) / \overline{d_{min}}(R_{ID}) \\ d_{max}(Y, R_{ID}) / \overline{d_{max}}(R_{ID}) \\ d_{template}(Y, R_{ID}) / \overline{d_{template}}(R_{ID}) \end{bmatrix} \quad (2)$$

In order to find the separating boundary between genuine and forgery signatures, comparisons of two classifiers were done: a linear discriminant analysis classifier and a quadratic discriminant analysis classifier.

4.1 Discriminant Analysis

Discriminant analysis is used to predict a categorical dependent variable also called as a grouping variable by one or more binary or continuous independent variables also called as predictor variables. Sir Ronald Fisher developed the original dichotomous discriminant analysis in the year 1936. It is different from an analysis from variance or multi-variant analysis of variance, which is used to predict continuous dependent variables by independent categorical variables. By using discriminant analysis we can determine if a set of variables are truly effective in predicting category membership. When groups are known a priori, discriminant analysis is used (which is not the case in cluster analysis). It is necessary for the cases to have a score on the quantitative predictor measures, and also on a group measure. In other words, Discriminant analysis is nothing but classification - the act of distributing things into classes, groups or categories of the same type. It is an extremely useful follow-up procedure to a multi-variant analysis of variance instead of doing one-way analysis from variance, for determining the differences of the groups on the composite of dependent variables. What is necessary to note is that in multi-variant analysis of variance, the independent variables are actually nothing but the grouping variables and similarly the dependent variables are the predictor variables. The types of discriminant analysis used for signature verification are:

1) Linear Discriminant Analysis
2) Quadratic Discriminant Analysis

4.1.1Linear Discriminant Analysis (LDA)

A linear classifier works by taking a decision by checking the linear combination of the features present. The characteristics of an object are also called as feature values and are presented to the machine in a vector function called a feature vector. A classifier can be viewed as a function of block. A classifier assigns one class to each point of the input space. The input space is thus partitioned into disjoint subsets, called decision regions, each associated with a class. This type of classifier is often used in places where we are forced to consider the speed of classification because most of the time linear classifier is the fastest classifier. This is especially true when x (input feature vector) is sparse. Another reason for choosing this classifier when speed is under consideration is that linear classifiers almost always work extremely well when the number of dimensions in the input feature vector is large. An example of this would be document classification where every element in the input feature vector, is generally the number of occurrences of a word.

4.1.2 Quadratic Discriminant Analysis (QDA)

QDA is very similar to LDA. The only difference being that in QDA we presume that the covariance matrix can possibly be different for each class. This is why the estimation of the covariance matrix Σk is done separately for each class k, k =1, 2,… K so that discrimination is based on quadratic and rather than linear functions of X. With QDA, however, there are no natural variates, and no general methods for displaying the analysis graphically. When there are just two feature variables, it is possible to visualize the quadratic classification regions in a scatterplot, but there is no clear extension to analyze with three or more features.

Now, because QDA allows the covariance matrix to be more flexible, it has a tendency of fitting the data better than Linear Discriminant Analysis. But if this is the case, unfortunately it automatically has more number of parameters to estimate. We also observe a significant rise in the number of parameters with QDA if a separate covariance matrix for every class is present. If many classes are there and sample points are very few, this can be a problem. The trade-off is between having a simple model to work with and the training data fitting well. Sometimes a simple model works just as well as a complicated model because it fits the data well. Even if the simple model does not fit the training data better than a complex model, it can still be better simply because the simple model is more robust.

5. CHALLENGES FACED

1) Pressure VB tablet (digitizer)

Pressure VB tablet used for signature capture did not give input to MATLAB serially, so an intermediate code had to be used, hence the c# interface is used and this is connected to MATLAB.

2) Orientation

The system initially could not handle any variation in orientation as pre-processing of orientation would disrupt the dynamics of the signature, but through bounding of features this problem was resolved, but to a limit of 10 degrees clockwise/anti-clockwise change of signature orientation.

3) False Reject

Small changes in signature string values due to jerk or small fundamental signature changes gave a FR (false reject) of the signature, with the use of QDA (quadratic discriminant analysis) we overcame this problem.

4) Hovering Time

The hovering time while signing was initially considered into the signature string, but this gives redundant information about the signature and messes with the other feature priorities as well(an error rate increase of about 50% was observed).

5) Pressure

Pressure applied while signing is not always the same, but the time variation of pressure is usually constant, by quantizing this feature, pressureturns into a biometric quality which can be and is used to verify signatures.

6 CONCLUSION AND FUTURE SCOPE

In online signature verification the user must provide a set of reference signatures to enroll in the system. Features are then extracted from the signatures. To verify a test signature, the same processes are applied. The test signature is then matched to all the other reference signatures. The method used to match signatures is based on the concept of string matching using DTW (Dynamic Time Warping). The classification is done with the help of two types of classifiers namely LDA (Linear Discriminant Analyzer) and QDA (Quadratic Discriminant Analyzer). Best results were obtained by using Quadratic Classifiers with an error rate of 1%. The dissimilarity values obtained is then compared to a threshold to decide whether the signature is genuine or a forgery. With the improvement in the forgery signatures enrolled, the overall system performance can be increased. The time complexity of the system can further be reduced using different feature combinations; by using a better hardware (pressure tablet) with more accurate and precise readings, an improvement in the system could be accomplished.

References

[1] Alisher Kholmatov, Berrin Yanikoglu, “Identity authentication using improved online Signature verification method”, Pattern Recognition Letters, vol.26, pp. 2400–2408, June 2005.

[2] Julian Fierrez. Javier Ortega-Garcia. Daniel Ramos, Joaquin - Rodriquez, "HMM—based on-line signature verification: Feature extraction and signature modeling", Pattern Recognition, vol. 28, Issue 16, Pages 2325–2334, December 2007.

[3] Wacom Co. Ltd., http://www.wacom.com/in/en, Wacom Bamboo pen tablet, September 2013.

[4] A. K. Jain, F. D. Griess, D. Connell, “On-line signature verification”, vol. 35 Issue 12, pp. 2963–2972, 2002.

[5] VBTablet.dll a WinTab Tablet API wrapper. Source Code available at http://sourceforge.net/projects/vbtablet/

[6] V. Bharadi, "Digitizer interface in C# using VBTablet", The Code Project. Available at: http://www.codeProject.com/KB/system/DigitizerDevice.aspx.

[7] Musa Mailah, Lim Boon Han, “Biometric signature verification using pen position, time, velocity and pressure parameters”,JurnalTeknologi, vol. 48(A), pp. 35 – 54, June 2008.

AUTHORS

Fernandes Joslyn received his B.E. degree in Electronics and Telecommunication Engineering from St. Francis Institute of Technology in 2014. He now works with Capgemini India Pvt. Ltd. as a Software Engineer.

Nishad Bhandarkar received his B.E. degree in Electronics and Telecommunication Engineering from Dwarkadas J. Sanghvi College of Engineering in 2014. He now works with Capgemini India Pvt. Ltd. as a Software Engineer.