Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 10, October-2020

Online Signature Recognition and Verification in Telugu Script

Aishwarya Sahai
Department of Computer Science and Engineering
National Institute of Technology, Patna
Patna, India

Akash Kant
Department of Electronics and Communication Engineering
International Institute of Information Technology
Hyderabad, India

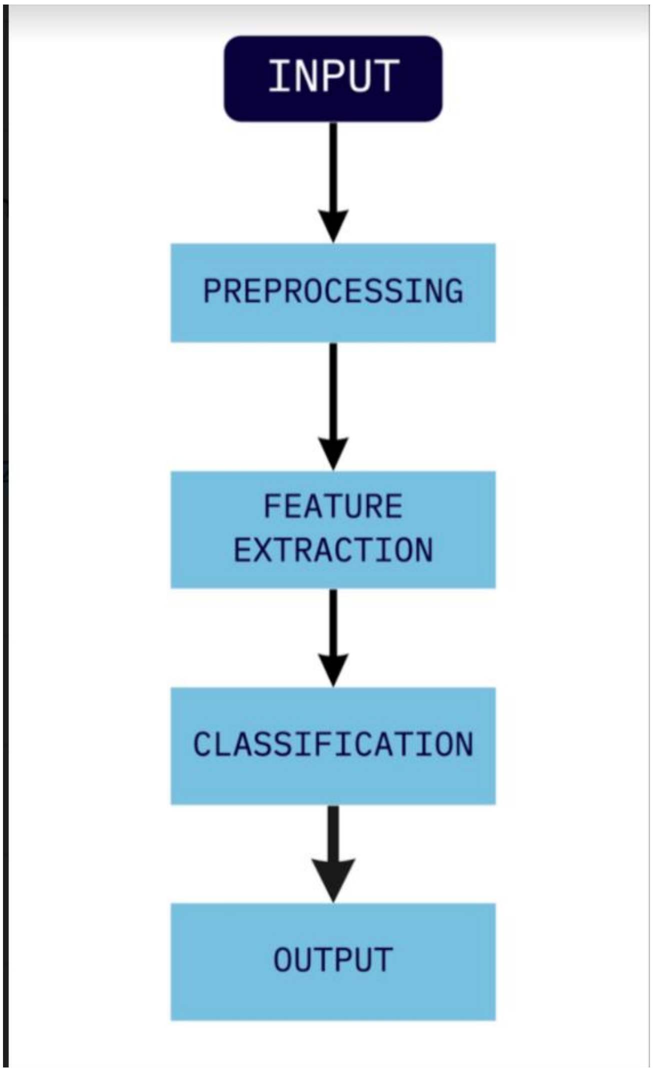
***Abstract*—Signature recognition and verification is an active field of research today, compromising of a variety of feature extraction techniques. Online signature recognition has gained enormous attention recently due to its vast application in various fields. In this article, the different techniques for the recognition and classification of signatures are analyzed. The signatures, in one of the most popular Indic scripts - Telugu, are examined dynamically at the time, when they are being fed as input to certain electronic digitizers, for example, a tablet. The current method uses multiple structural and directional features, analyzed in elliptical regions of the input image, to extract feature values from strokes of text and non-text data. The features are then studied in classification platforms based on Support Vector Machine (SVM) approach and Hidden Markov Model (HMM). The probabilistic outcomes of these two classification platforms are then combined using the Dempster–Shafer Theory (DST) to improve the accuracy rate.**

Keywords—Online signature; SVM; HMM; classification; DST

I. INTRODUCTION

Due to the recent advances achieved in hardware technology and the growing popularity of handheld devices, such as Personal Digital Assistant (PDA) , mobile phone , and Ultra Mobile PC (UMPC), new methods for the input of speech and handwriting have been developed. Thus, making character recognition an active field of research in the modern world. This paper deals with the online recognition, and verification, of signatures in Telugu – an Indian script. This mechanism of recognition uses a different feature space and concept of elastic matching in the form of Dynamic Time Warping (DTW); the classification among different classes has been done by a KNN classifier. Handwriting Recognition is a part of pattern Recognition. It has a varied use in modern scenario ranging from Signature recognition and verification to Handwriting recognition.

The remaining portions of this paper are arranged as follows. Section 2 details the related works. Section 3 discusses the Telugu script and method of dataset creation. Section 4 lists own the Preprocessing steps taken. Section 5 depicts the feature extraction technique. The processes of classifying text and non-text data separately using SVM and HMM as well as by combining them are discussed in section 6. Experimental results and analysis are discussed in section 7. Finally, conclusion of the paper is given in section 8.



II. LITERATURE SURVEY

Although limited studies are available on classification of text and non-text portions from within a single online handwritten document in a few non-Indian scripts, we were not able to find any research work in this specific area in Telugu script. A few studies, available in non-Indian scripts regarding this, are discussed here.

Kashi et al. [1] proposed ‘A Hidden Markov Model’ approach to online handwritten signature verification. A

Hidden Markov Model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states.

Zanauy et al. [2] proposed a multi-section codebook approach for online signature recognition. This algorithm considers the temporal evolution of the signature and obtains a significant speed improvement, which is estimated to be around 47 times.

Zhang et al. [3] presents a new three-stage verification system, which is based on three types of features - global features, local features of the corner points, and function features that contain information of each point of the signature. In this approach, signature verification is based on the comparison of input and reference signatures, involving both static and dynamic analysis.

In Barkoula et al. [4] approach, we see the signatures' TAS and TASS representations and their application to online signature verification. In the matching stage, a variation of the longest common sub-sequence matching technique has been employed.

III. Data Collection In Telugu Script

All the samples were collected from the students in National Institute of Technology, Patna via an electronic tablet. For Telegu script, a total of 600 samples were collected. Out of which, 300 were the original signatures, made by students. We collected 5 signature samples each from 60 students.

Original samples count: 60*5 = 300 samples

Also, we collected 300 forged signatures - 5 signature samples each from 60 individuals.

Fake samples count: 60*5 = 300 samples

This helped in bringing the sample data size to 600 samples.

Total samples count: 300 (original) + 300 (forged) = 600 samples

A tabular description of the data collected is given below:

Script	No. of Writers	Type of Sample	Samples	Samples for Testing (%)	Samples for training (%)
Telegu	60	Text	600	40	60

Table 1.

IV. PREPROCESSING

Online signature verification involves taking signature from the user using devices like touchpad and mobile device. The input sometimes might not be compatible with our algorithm. There can be issues like extra brightness, low intensity, sharp broken lines, incomplete points, and others.

To get this sorted, we pass the input through many stages of preprocessing. These stages have been shown here diagrammatically.

A. Interpolation

In the mathematical field of numerical analysis, interpolation is a method of constructing new data points within the range of a discrete set of known data points. In engineering and science, one often has a number of data points, obtained by sampling or experimentation, which represent the values of a function for a limited number of values of the independent variable. It is often required to interpolate (i.e. estimate) the value of that function for an intermediate value of the independent variable. This may be achieved by curve fitting or regression analysis. This approach preserves time dependencies, keeping a high spatial density of samples in slow strokes and a sparser sample distribution in high speed strokes.

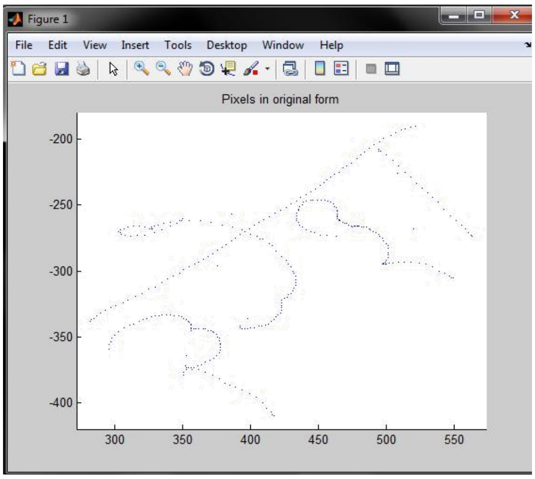


Figure 1. Image after interpolation

B. Smoothing

In statistics and image processing, to smooth a data set is to create an approximating function that attempts to capture important patterns in the data, while leaving out noise or other fine-scale structures/rapid phenomena. In smoothing, the data points of a signal are modified so individual points (presumably because of noise) are reduced, and points that are lower than the adjacent points are increased leading to a smoother signal. Smoothing may be used in two important ways that can aid in data analysis (1) by being able to extract more information from the data as long as the assumption of smoothing is reasonable and (2) by being able to provide analyses that are both flexible and robust. Many different algorithms are used in smoothing. Here we have used average method to get the average point between two consecutive points to get the smooth curve.

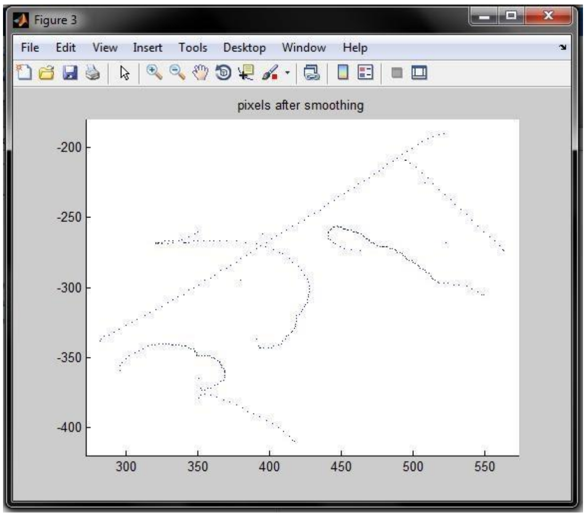


Figure 2. Image after Smoothing.

C. *Resampling*

Image resampling is the process of transforming a sampled image from one coordinate system to another. The two coordinate systems are related to each other by the mapping function of the spatial transformation. The inverse mapping function is applied to the output sampling grid, projecting it onto the input. The result is a resampling grid, specifying the locations at which the input is to be resampled. The input image is sampled at these points and the values are assigned to their respective output pixels.

Resampling implies changing the sample rate of a set of samples. In the case of an image, these are the pixel values sampled at each pixel coordinate in the image.

D. *Normalization*

The main goal of this preprocessing step is to ensure that the same characters have the same height for every handwritten signature. This is accomplished by transforming every signature to a given corpus height, where the corpus height is the distance between the baseline and the corpus line. Note that the total height of signatures may still vary after normalizing size, even for the signatures having same corpus height.

V. FEATURE EXTRACTION

The feature extraction plays an important role in overall process of signature recognition. Many feature extraction techniques have been proposed to improve overall recognition rates, however most of them are dependent on the size and slope of the signature. They require very accurate resizing, slant, correction procedure of technique otherwise they achieve very poor recognition rates. To extract features like standard deviation, curliness, writing direction, curvature, slope we have used two methods - Elliptical method, and NPEN on global basis method. These are described in detail, below.

A. *Elliptical Technique*

The feature extraction phase gives a matrix containing feature values of each sample identified by row. This sample is either training data set or test data set. These data sets are fed as an input to the classification phase, or further processing. The following five features were used - Standard deviation, Curliness, Writing direction, Curvature, and Slope.

In this technique, we make multiple concentric ellipses on the plane of the signature. The outer most ellipse covers the plane fully, touching the vertices of the plane. Then other ellipses are drawn having lesser radii and sharing same center point. The properties are extracted from the area between two consecutive ellipses. The feature values are saved in a vector matrix in the form of rows and columns.

Each elliptical region is divided into octants called bins. The number of feature values obtained from each region is five. Figure 1 shows the partition of plane by four number of ellipses and divided into octants.

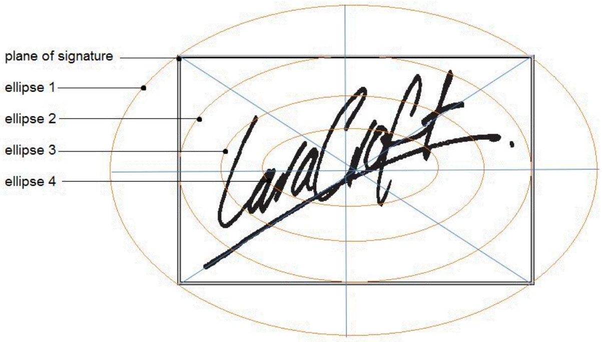


Figure 3.

B. *NPEN on Global Basis*

We have also used NPEN approach to extract features, like standard deviation, curliness, writing direction, curvature, and slope. In this approach, we directly extract the features from the whole signature without dividing it into further regions (Figure 2). The feature vector size for each sample was 50. Size=8(slope)+2(std_dev)+16(curvature)+16(writing_dir)+8(curliness)=50

Figure 4.

VI. CLASSIFICATION

Classification in online signature verification and recognition refers to the classification of the test data or sample into their respective class labels. These class labels tell the class of the sample. There should be a high level of accuracy during classification because it lays the basis for further recognition phase. If classification is not correct, then recognition will be even further bad. So, there is utmost need of better classification. However, it is impossible to get 100% accuracy during classification. It is evident that there will be some amount of incorrect classification, but it should be as less as possible.

There are two types of data set that needed classification phase in two different manners:

1. Training set data
2. Test set data

The classifier used here is Support Vector Machine (SVM), Hidden Markov Model (HMM) and Hidden-state Conditional Random Field Library (HCRF). These are described in detail below.

A. *Support Vector Machine(SVM)*

In machine learning, SVMs are supervised learning models with associated learning algorithms that analyze data and recognize patterns, and are often used for classification and regression analysis. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

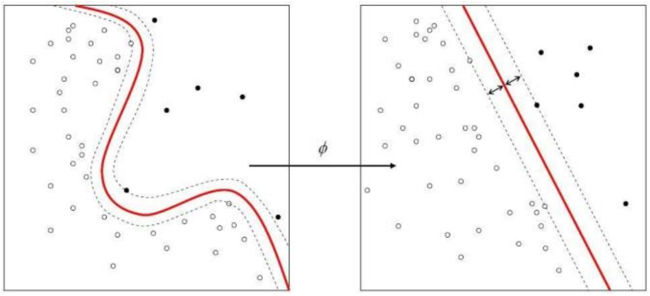
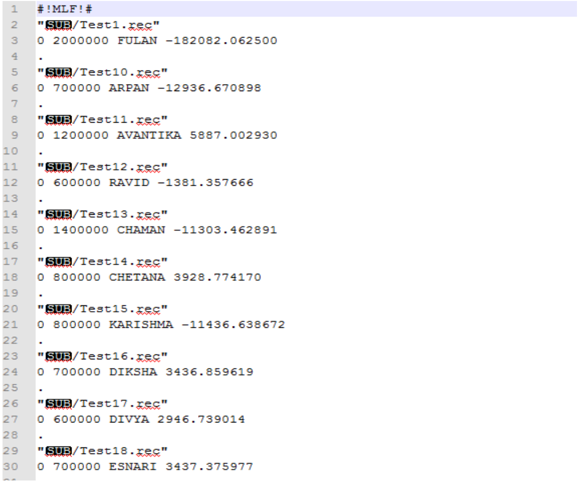


Figure 5.

In machine learning, kernel methods are a class of algorithms for pattern analysis, whose best known member is the support vector machine (SVM). The following kernels have been used here - Linear kernel, Polynomial kernel, and Radial Basis Function kernel.

B. *Hidden Markov Model(HMM)*

A Hidden Markov Model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. A method for the automatic verification of online handwritten signatures using both global and local features. A HMM can be presented as the simplest dynamic Bayesian network.We get the names of the person and their index number in the result file created in the process. If the index number and the corresponding name is incorrect, as in original state, then the sample is discarded.



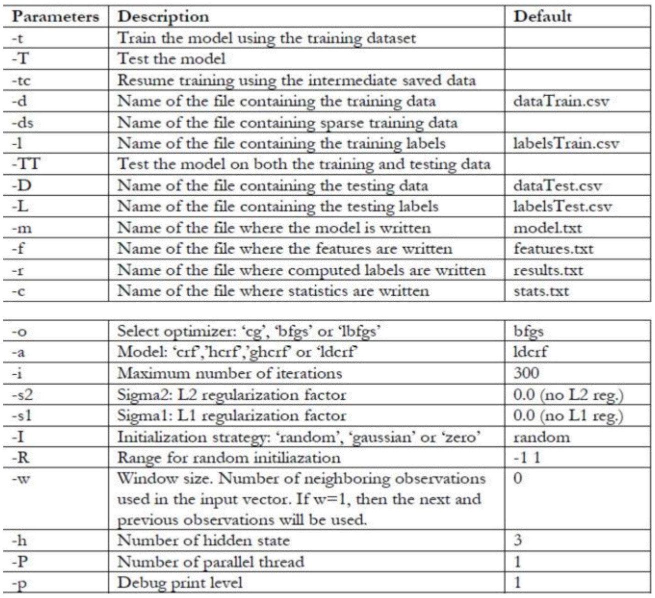
```
#!MLF!#
"SUB/Test1.rec"
0 2000000 FULAN -182082.062500
.
"SUB/Test10.rec"
0 700000 ARPAN -12936.670898
.
"SUB/Test11.rec"
0 1200000 AVANTIKA 5887.002930
.
"SUB/Test12.rec"
0 600000 RAVID -1381.357666
.
"SUB/Test13.rec"
0 1400000 CHAMAN -11303.462891
.
"SUB/Test14.rec"
0 800000 CHETANA 3928.774170
.
"SUB/Test15.rec"
0 800000 KARISHMA -11436.638672
.
"SUB/Test16.rec"
0 700000 DIKSHA 3436.859619
.
"SUB/Test17.rec"
0 600000 DIVYA 2946.739014
.
"SUB/Test18.rec"
0 700000 ESNARI 3437.375977
```

C. *Hidden State Conditional Random Field Library(HCRF)*

HCRF is a C++ library for training and inference of Conditional Random Field (CRF), Hidden-state CRF (HCRF) and Latent-dynamic CRF (LDRCF) models. This library implements three main models: Conditional Random Field (CRF), Hidden-state Conditional Random Fields (HCRF) and Latent-Dynamic Conditional Random Fields (LDCRF). We implemented two variants of the HCRF model. The CRF and LDCRF models can be applied to unsegmented sequences while the HCRF (and GHCRF) should be apply to pre-segmented sequences (only one label per sequence).

The HCRF library can be easily installed on a Microsoft Windows system using the IntallShield installation package. The HCRF library comes with a demo program called TestHCRF.exe. This program can be used to train and test CRF, HCRF and LDCRF models from the command prompt. The demo program TestHCRF.exe is called using the following syntax:

TestHCRF.exe [-t] [-T] [-d filename] [-l filename] [-D filename] [-L filename] [-m filename] [-f filename] [-r filename] [-o cg|bfgs|asa] [-a crf|ldcrf|hcrf|ghcrf]



Parameters	Description	Default
-t	Train the model using the training dataset	
-T	Test the model	
-tc	Resume training using the intermediate saved data	
-d	Name of the file containing the training data	dataTrain.csv
-ds	Name of the file containing sparse training data	
-l	Name of the file containing the training labels	labelsTrain.csv
-TT	Test the model on both the training and testing data	
-D	Name of the file containing the testing data	dataTest.csv
-L	Name of the file containing the testing labels	labelsTest.csv
-m	Name of the file where the model is written	model.txt
-f	Name of the file where the features are written	features.txt
-r	Name of the file where computed labels are written	results.txt
-c	Name of the file where statistics are written	stats.txt
-o	Select optimizer: 'cg', 'bfgs' or 'lbfgs'	bfgs
-a	Model: 'crf','hcrf','ghcrf' or 'ldcrf'	ldcrf
-i	Maximum number of iterations	300
-s2	Sigma2: L2 regularization factor	0.0 (no L2 reg.)
-s1	Sigma1: L1 regularization factor	0.0 (no L1 reg.)
-I	Initialization strategy: 'random', 'gaussian' or 'zero'	random
-R	Range for random initiliazation	-1 1
-w	Window size. Number of neighboring observations used in the input vector. If w=1, then the next and previous observations will be used.	0
-h	Number of hidden state	3
-P	Number of parallel thread	1
-p	Debug print level	1

VII. RESULTS

Different metrics can be used to rate the performance of a biometric factor, solution or application. The most common performance metrics are the False Acceptance Rate FAR and the False Rejection Rate FRR. The two indicators used for measuring performance of any signature verification and recognition system are - FAR (False Acceptance Rate), and FRR (False Rejection Rate)

a. *FAR*: The FAR or False Acceptance rate is the probability that the system incorrectly authorizes a non-authorized person, due to incorrectly matching the biometric input with a template. A system's FAR typically is stated as the ratio of the number of false acceptances divided by the number of identification attempts.

b. *FRR*: The FRR or False Rejection Rate is the probability that the system incorrectly rejects access to an authorized person, due to failing to match the biometric input with a template. It is stated as the ratio of number of false rejections to number of identification attempts.

Kernel	RBF				
Ellipses	3	4	5	6	7
Accuracy (%)	19.07	19.07	23.03	19.73	18.76

Kernel	Poly				
Ellipses	3	4	5	6	7
Accuracy (%)	17.1	19.73	26.36	20.39	21.4

Kernel	Linear				
Ellipses	3	4	5	6	7
Accuracy (%)	43.421	46.71	48.684	47.36	51.2

Table 1: Accuracy rate using different SVM kernels with varying number of ellipses. Train: Test = 3:2

Kernel	RBF				
Ellipses	3	4	5	6	7
Accuracy (%)	14.07	18.4	22.36	15.7	27.51

Kernel	Poly				
Ellipses	3	4	5	6	7
Accuracy (%)	13.15	18.4	19.73	19.73	23.26

Kernel	Linear				
Ellipses	3	4	5	6	7
Accuracy (%)	43.42	39.4	47.36	40.78	61.02

Table 2: Accuracy rate using different SVM kernels with varying number of ellipses. Train: Test = 4:1

Kernel	RBF				
Ellipses	3	4	5	6	7
Accuracy (%)	15.76	17.46	17.34	15.83	17.27

Kernel	Poly				
Ellipses	3	4	5	6	7
Accuracy (%)	16.67	17.46	16.46	16.25	14.12

Kernel	Linear				
Ellipses	3	4	5	6	7
Accuracy (%)	18.67	19.95	24.08	27.5	24.32

Table 3: Accuracy rate of Forged signatures using different SVM kernels with varying number of ellipses. Train: Test = 3:2

Kernel	RBF				
Ellipses	3	4	5	6	7
Accuracy (%)	14.47	18.42	22.36	15.78	19.25

Kernel	Poly				
Ellipses	3	4	5	6	7
Accuracy (%)	13.15	18.42	19.7	19.7	17.08

Kernel	Linear				
Ellipses	3	4	5	6	7
Accuracy (%)	43.42	39.47	47.368	40.789	29.01

Table 4: Accuracy rate of Forged signatures using different SVM kernels with varying number of ellipses. Train: Test = 4:1

Script	RBF Kernel	Poly Kernel	Linear Kernel
Telugu	75.71%	75.71%	92.85%

Table 5: Accuracy rate of Genuine signatures using different SVM kernels with varying number of ellipses for 70 samples: TRAIN:TEST=4:1

Script	Elliptical Approach	NPEN Approach
Telugu	95.71%	92.85%

Table 6: Accuracy rate of Genuine signatures using HMM classifier (Linear Kernel)

Script	Elliptical Approach	NPEN Approach
Telugu	92.85%	78.57%

Table 6: Accuracy rate of Genuine signatures using HCRF.

VIII. CONCLUSION

Automatic signature recognition and verification is a very attractive field from both scientific and commercial points of view. It is an emerging and trending field of pattern recognition which finds its utility in applications which are used for recognizing signature as well as languages having large characters (Japanese, Chinese), script identification and so on.

In this project we tried to explore this field and novel our idea of feature extraction to gain reasonable and acceptable accuracy. With our efforts we were able to achieve promising results. The best results for a common threshold are obtained with the feature set consisting of the local and global features. We obtained accuracy of 80-90 % for Telegu. These are quite promising results. Thus, we can say that are proposed approach can be used at it is for Telegu.

To conclude we can say that our proposed approach(system) works quite promisingly and can be further explored to enhance its capabilities so that give even better results. As a part of future scope, we can include word recognition, script recognition, application on other scripts, (Indian as well as Western), with structural features and so on.

Based on the above feature vector classification was performed using Support Vector Machine as classifier. The platform was identical for both the approaches before classification i.e., the same dataset was used for training as well as testing and comparative analysis of both approaches to obtain results. We have so far obtained 87% accuracy in linear kernel in Telegu signatures.

From the above result, it seems to be evident that our proposed approach works well as far as classification of Telegu strokes is concerned, based on spatiotemporal features.

REFERENCES

[1] Kui Zhang, Edgard Nyssen and Hichem Sahli,"A Multi-stage Online Signature Verification System" ETRO-IRIS, Vrije Universiteit Brussel, Brussels, Belgium

[2] K. Barkoula · G. Economou · S. Fotopoulos "Online Signature Verification Based on Signatures' Turning Angle Representation Using Longest Common Subsequence Matching", International Journal On Document Analysis And Recognition. Volume 4, Issue 16, pp 261– 272

[3] S. Jaeger, S. Manke, J. Reichert, A. Weibel, "Online handwriting recognition: the NPen++ recognizer", International Journal on Document Analysis and Recognition, Volume 3, Issue 3

[4] R. Kashi, J. Hu, W.L. Nelson, W. Turin, "Hidden Markov Model", International Journal On Document Analysis And Recognition. Volume 2, Issue 2, pp 102-109.

[5] Marcos Faundez-Zanuy • Juan Manuel Pascual-Gaspar , "Efficient Online Signature Recognition Based on Multi-section Vector Quantization", Pattern Anal Applic (2011),Volume 3,Issue 3,pp 37-45