2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)

SVM Based Feature Set Analysis in Dynamic Malayalam Handwritten Character Recognition

Steffy Maria Joseph
M.Tech
Govt.Engineering College,Kozhikode
Email: steffymaria89@gmail.com

Abdu Rahiman V
SignalProcessing & Communication Lab
Govt.Engineering College,Kozhikode
Email: vkarahim@gmail.com

Abdul Hameed K M
SignalProcessing & Communication Lab
Govt.Engineering College,Kozhikode
Email:erhameed@yahoo.com

***Abstract*—Dynamic or Online handwritten character recognition is a challenging field in Human Computer Interfaces. The classification success rate of current techniques decreases when the dataset involves the similarity and complexity in stroke styles, number of strokes and stroke characteristics variations. Malayalam is a complex south indian language spoken about 35 million people especially in Kerala and Lakshadweep islands. In this paper, a classification scheme based on support vector machines (SVM) is proposed to improve the accuracy in classification and recognition of online malayalam handwritten characters. SVM Classifier is a popular one in academy as well as in industry. This Classifiers are more suitable in a real world applicative problem, if we have major concern on the speed of recognition per character. The contribution of various features towards the accuracy in recognition is analyzed. Performance for different kernels of SVM are also studied. A graphical user interface has developed for reading and displaying the character. Different writing styles are taken for each of the 44 alphabets. Various features are extracted and used for classification after the preprocessing of input data samples. Feature Selection is carried out by choosing of different combinations of extracted features versus accuracy. Highest recognition accuracy of 97% is obtained for the best selected features in SVM with polynomial kernel. Recognition speed of a single stroke is obtained 0.52 secs.**

I. Introduction

Online Handwritten Character Recognition has been studied from the past several years which demands to replace a large keyboard because of the huge alphabet size for texting applications. Also it makes easy to write in our own style and can be used for signature verifications[1]. An Online Real time Dynamic handwritten character recognition generally consists of preprocessing the raw x, y coordinates or the position coordinates tracked from a digitizer/smartphones, relevant feature extraction, classification and display[2]. This paper focused on an application based novelty which experiments on a very complex south indian language - Malayalam. Malayalam is a language having complex script in which alphabets consist multiple loops, cusps and similar character styles. There are some major works reported on Malayalam Online Handwritten Character Recognition from 2003 onwards. Preprocessing methods like normalization, duplicate point elimination, dot detection, dehooking, smoothing and equidistant resampling are commonly used. The accuracy of recognition per character can be varied based upon the type of dataset, extracted features and the classifiers used.

In 2009, Sreeraj and Sumam[3] used Kohonen Network to classify the context bitmap of normalized coordinates and achieved an accuracy of 88.75%. Later they experimented with KNN Classiifer using time domain features, writing direction and curvature of strokes[4]. The accuracy and speed were 98.125% and 0-16ms correspondingly. In 2011, Amit Arora and M. Namboodiri[5] achieved an accuracy of 78.07% and speed 10.67 ms with time domain and fourier coefficients of strokes using Multi-layer Markov Model. Binu P Chacko and Babu Anto P[6] used Online Sequential Extreme Learning Machine based (OS-ELM) to classify the division point features(statistical & structural features) in the year 2011. The accuracy reported was 96.83%. Prime and Sumam[7] again came out with a new proposal where the wavelet transform of the input co-ordinates and the angle features like direction and curvature are used to form the feature vector in compressed form. Wavelet transform used were db1 and haar and gave good frequency time resolution The classifier used was Simplified Fuzzy ARTMAP network(SFAM), which requires comparatively very less time for training than HMM and Backpropogation networks. In 2012, Indhu T. R. and Bhadran V. K.[8] presented an online handwriting recognition system for Malayalam using simplified Fuzzy ARTMAP technique. In their approach, individual strokes were identified by comparing the features of the unknown stroke with that of the stroke feature database. The accuracy was 98.26%. Feature extraction method based on discrete wavelet transform was used with SVM in the year 2013[9]. Works were done in 2014 to compare the accuracies and speed of recognition related to One Vs One and One Vs all Muliticlassification of malayalam handwritten datasets in SVM[1]. System using SVM require very much less training time compared to those using HMM. A comprehensive literature study was referred on online handwriting recognition given in[10]. In all works reported in Online Malayalam Handwritten Character Recognition from the year 2009 to 2014, it is understood that there are no large variations in the methods preprocessing, feature extraction and classification used. Since there are no standard datasets for Malayalam, the data collection modifies the accuracy to an extent. The training dataset of a classifier should include various styles, various speed, various angle of directions to generalize the results.

A. Malayalam Script and its Challenges

Malayalam is a dravidian language originated from Brahmi Script. It has the largest number of alphabets about 128 characters among the Indian languages. Modern Malayalam Script includes vowels (15), consonants (36), chillu (5), anuswaram (1), visargam (1), chandrakkala (1), consonant signs (3), vowel signs (9), conjunct consonants (57). Malayalam scripts have syllabic alphabet in which all

978-1-4799-8996-6/15/$31.00

consonants have an inherent vowel. Diacritics can appear above, below, before or after the consonant they belong to, are used to change the inherent vowel. When they appear at the beginning of a syllable, vowels are written as independent letters. When certain consonants occur together, special conjunct symbols are used which combine the essential parts of each letter. Almost all the characters are circular by themselves. They consist of loops and curves. The loops are written frequently in the clockwise order and not case sensitive. Two prominent ways of writing Malayalam scripts exists today. One followed by older generation and the other followed by younger generation. But the latter has become standard form even though usage of the former is still common[4].

The main challenges are the presence of large number of characters, high degree of similarity in character shapes, different writing styles, writing speed, character complexity, poor reliability of extracted stroke features due to variance in handwriting[8].

B. SVM Classifier

We have used SVM for the classification of different features extracted. SVM Classifiers are the best for real world applications. They are powerful classifiers that have proven to be efficient for several pattern recognition tasks such as speech and handwriting recognition, face recognition etc [11][12]. Basically SVM performs binary classification, however several SVM classifiers can be combined to do multiclass classification using one against all or one against one technique. Principle behind SVM is to map the input data on to higher dimensional feature space nonlinearly related to the input space and determine a maximum margin hyper plane separating the two classes of feature space[13][14]. SVM gives unique solution having maximum margin between feature sets and convergence is guaranteed unlike Artificial Neural Networks. The advantage of using SVM over other classifiers is referred in [5],[9],[15].

This paper is organized as follows: Section II is about the data collection of handwritten alphabets. Section III explains the different preprocessing methods used and the Section IV describes the feature extraction. Section V mentions the performance results of the experiments.

II. Data Collection

Experiments in this paper are performed by using the handwritten alphabets drawn with MATLAB GUI to track the raw x,y coordinate vectors. A total of 44 malayalam single strokes are used. Fig. 1 shows the basic characters of vowels (8) and consonants (36) used for the classification.

The dataset consists of character with different speeds, styles and size from various writers. A sample of handwritten alphabets are shown in Fig. 2.

III. Preprocessing

Preprocessing is done to remove the irrelevant information present in the raw x, y data. The main steps used are normalization, duplicate point elimination, smoothing and re-sampling. The x, y coordinates of a character are sequentially recorded as function of time 't'. Normalization is used to scale the raw $x(t), y(t)$ coordinates of an alphabet into {-1 1} range for the size

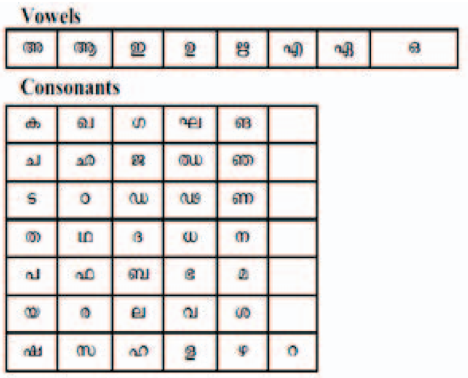

Fig. 1. Basic 44 character set of vowels and consonants used for experiment

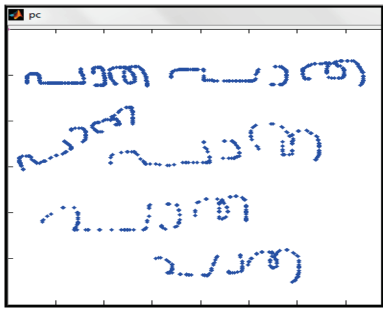

Fig. 2. Different styles(speed,orientation,size) of a Malayalam word "patha"

invariant detection. The x,y coordinates of a character are normalized using the equation shown below.

$$new(x) = \frac{old(x) - min}{(max - min)} * (max_n - min_n) + min_n \quad (1)$$

where $new(x)$ = new x point; $old(x)$ =old x point; $max_n = 1$; $min_n = -1; max$ = maximum value and min = minimum value of the old set of x points. Similarly y coordinates are also normalized. Duplicate point elimination removes points, if same x, y points are repeated in the neighborhood. Moving average smoothing with a window size of $3(N = 3)$ removes abrupt variations in the pen directions. The smoothing equation [16] is shown below.

$$X(t) = \frac{\gamma}{(2N + \alpha)} \quad (2)$$

$$\gamma = x(t-N)+\ldots+x(t-1)+\alpha x(t)+x(t+1)+\ldots+x(t+N) \quad (3)$$

Where $X(t)$ is the smoothed point obtained. Parameter α depicted in Fig. 3 is added to avoid the smoothing of sharp edges which provides important information about pen directions[16]. Similarly y coordinates are also smoothed. In data collection, the points appear equidistant in time but not in space. Hence, the number of points varies depending on the speed of writing and the sampling rate of the digitizer. In order to remove these variations, the coordinate sequence is re-sampled spatially along the trajectory[8]. Here each character is re-sampled into 30 points. Thus each character is represented with a fixed number of points. Equidistant re-sampling is the best to provide good classification results.

IV. Feature Extraction

There is no doubt that computing features is a very important processing step in every online recognition system. *However, neither a standard method for computing features nor a widely accepted feature set currently exists.* We have experimented with a large set of features for their effectiveness in representation of significant characteristics of the strokes. The main features extracted here as follows.

A. Time Domain Features

1) Preprocessed X, Y Coordinate Vectors
2) Polar Coordinates of preprocessed X, Y Coordinate Vectors
3) Writing directional features
4) Writing curvature features
5) Overall length of the stroke
6) Aspect Ratio
7) Moments up to fourth order(Mean, Variance, Skewness, Kurtosis)
8) Curliness
9) Cusp detection
10) Projection histograms in X and Y directions

B. Frequency Domain Features

1) Fourier Transform (FT)
2) Discrete Cosine Transform (DCT)

Since Malayalam is a language having complex script in which alphabets consist multiple loops, cusps and similar character styles, directional and curvature features contributes the major recognition part that cant be avoided.

C. Directional Features

Local writing direction at a point $x(t), y(t)$ is described using sine and cosine of the angle α subtended by the line joining the preceding and succeeding points with the horizontal line[7]. Fig. 3 [4] shows how to get the directional angle of a stroke.

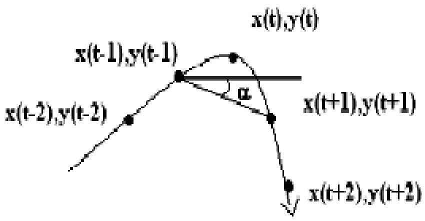

Fig. 3. Writing Directional Feature

The equations are shown below:

$$\Delta y(t) = y(t+1) - y(t-1)$$

$$\Delta x(t) = x(t+1) - x(t-1)$$

$$\sin\alpha(t) = \frac{\Delta y(t)}{\sqrt{\Delta x^2(t) + \Delta y^2(t)}} \quad (4)$$

$$\cos\alpha(t) = \frac{\Delta x(t)}{\sqrt{\Delta x^2(t) + \Delta y^2(t)}} \quad (5)$$

Both $\sin\alpha(t)$ and $\cos\alpha(t)$ gives the direction angle. But if we drop either sine or cosine feature, by experiments it is seen that the recognition accuracy falls down. So we kept both the values of $\sin\alpha$ for y coordinate vectors and $\cos\alpha$ for x coordinate vectors. $\Delta y(t) \& \Delta x(t)$ are needed to find the change in direction of the current point (x, y).

D. Curvature Features

Curvature is the angular difference at a point $x(t), y(t)$ may be represented using the sine and cosine of the angle K. It is the angle by which the preceding direction vector is to be rotated in counter clockwise direction so as to coincide with the succeeding direction vector. It is a measure of angular difference between the preceding and successive direction vectors[7]. The Fig. 4 [4] shows how to get the curvature of a stroke.

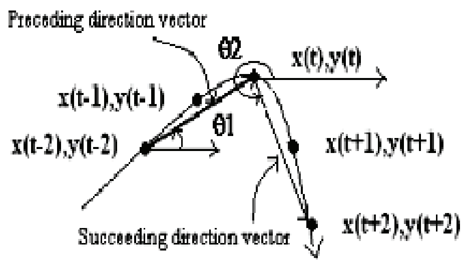

Fig. 4. Writing Curvature Feature

The equations are shown below:

$$\theta_1 = \tan^{-1}\left(\frac{y(t) - y(t-2)}{x(t) - x(t-2)}\right) \quad (6)$$

$$\theta_2 = \tan^{-1}\left(\frac{y(t+2) - y(t)}{x(t+2) - x(t)}\right) \quad (7)$$

if $\theta_1 - \theta_2 > 0$ then K$= 360 - (\theta_1 - \theta_2)$ else K$= \theta_2 - \theta_1$

V. Performance Results

We have chosen 7 samples per character such that the 7 styles are different based on their writing speed, orientation and the size. A total of 308 samples for 44 classes are preprocessed and used for feature extraction. The samples are collected from a group of 20 people. A character is drawn by using hand on touch pad in Laptop computer and is visualized at the same time in MATLAB GUI. Raw x, y coordinates and number of strokes are read into a matrix by using three call back functions (window button down, window button up and window button motion). The first step of algorithm is the normalization. It is really needful when we have various writer's data input with different sizes. All the character's x,y points are scaled into {-1,1}. The normalized stroke "a" is shown in Fig. 5. After normalization, next is the removal of duplicate points. One way, we can fix a small window with $W = 3 or 4$(preferred a small one) in the neighboring range of current point x,y and check for the duplicate points. Another method is based on the distance based removal of insignificant points. If the distance is below some fixed threshold, then the process is done. This method is preferred here. Then the Smoothing and the re-sampling are done. Moving average smoothing with a window size of $3(N = 3)$ removes abrupt variations in the pen directions. While training for SVM, we need to optimize the number of re-sampling points for the best recognition accuracy. This helps writing speed invariant detection. Here the x, y points of each handwritten alphabet is re-sampled into 30 points.

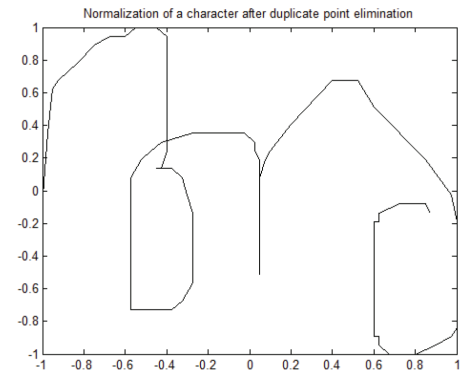

Fig. 5. Normalization of a stroke ”a”

The table I shows the recognition accuracies of a test data of 44 classes for different resampling points.

TABLE I. TESTING DATA RECOGNITION ACCURACY FOR DIFFERENT NUMBER OF RE-SAMPLING POINTS OF A STROKE

Resamping Points	25	30	35	40
Linear Kernel	90.9091%	93.1818 %	93.1818%	93.1818%
Polynomial Kernel	90.9091%	97.7273 %	95.4545%	93.1818%
Gaussian Kernel	90.9091%	93.1818 %	93.1818%	93.1818%
Sigmoid Kernel	86.3636%	93.1818 %	93.1818%	93.1818%

We can go for a high re-sampling rate to get a smooth curve, but it aids no improvement of accuracy and the feature vector dimension get lengthened as well. Re-sampling is carried out by up sampling to get a fine interpolation and then decimation to fixed points. The re-sampled stroke ”a” is shown in Fig. 6.

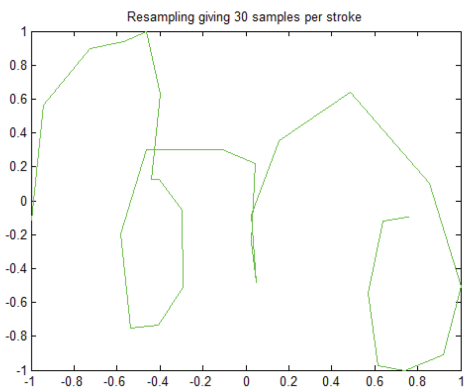

Fig. 6. Re-sampling of a stroke ”a” using 30 points

The SVM One vs One multi-classification is carried out in LIBSVM 3.18 tool. We have analyzed the recognition accuracies of alphabets with 4 kernels - linear, polynomial, gaussian and sigmoid. Parameters like cost, coefficient value, sigma, degree of various kernels are experimentally chosen for the best accuracy. In SVM Torch (another tool), user defined parameter ’C’ is chosen by trial. We have divided the data set as 60 % training 20% testing and 20 % validation. SVM classification is done on validation data to get optimum ’C’ for best classification accuracy. Used that ’C’ value for training dataset for different kernels and obtained the optimum kernel parameters. The comparison results of ’One vs one’ and ’One vs all’ is given in [1]. All the classification results shown in this paper are done by One versus one multi-classification in SVM. Kernel equations are shown below

Linear Kernel : $u' * v$
Polynomial Kernel : $(\gamma * u' * v + a)\hat{}d$
Radial basis function : $\exp(-\gamma * |u - v|\hat{}2)$
Sigmoid Kernel : $\tanh(\gamma * u' * v + a)$

γ is the gamma value depends upon the dimension of feature vector, ”a” is the coefficient value of sigmoid and polynomial kernel and ”d” is the degree of polynomial kernel. It is seen that polynomial kernel gives a good accuracy when the number of re-sampling point is 30. A comparison of re-sampling for recognition ”with smoothing” and ”without smoothing” is shown in table II. It can be seen that ”without smoothing” exhibits better results. Training data of SVM always met 100% accuracy.

TABLE II. TESTING DATA RECOGNITION ACCURACY WITH RE-SAMPLING POINT 30 FOR ”WITH SMOOTHING” AND ”WITHOUT SMOOTHING”

Kernels	With smoothing	Without smoothing
Linear Kernel	90.9091 %	93.1818%
Polynomial Kernel	93.1818 %	97.7273%
Gaussian Kernel	90.9091 %	93.1818%
Sigmoid Kernel	88.6364 %	93.1818%

We tested with different styles of testing data out of which polynomial kernel with degree 3 exhibits a satisfying accuracy. The following table shows the accuracy for different degree values for polynomial kernel. After classification, there is a high rate of miss-classification between similar styles of strokes. The miss-classified letters usually are ”tha” as”na”,”uh” as”la”,”nja” as ”nah”,”oh” as ”da”,”bha as”da”.

TABLE III. COMPARISON OF POLYNOMIAL KERNEL WITH DIFFERENT DEGREE VALUES

Polynomial Kernel	Accuracy
degree 2	95.4545%
degree 3	97.7273%
degree 4	93.1818%
degree 5	93.1818%
degree 6	90.9091%

Major features and their influence on testing data accuracy are shown in tables IV, V, VI and VII.

TABLE IV. CONTRIBUTION OF EACH FEATURE TOWARDS THE RECOGNITION ACCURACY OF HANDWRITTEN ALPHABETS OF 44 CLASSES

Kernels	Preprocessed (x,y) coordinates	Polar coordinates (R&θ)	Moments
Linear Kernel	90.9091 %	72.7273%	47.7273%
Polynomial Kernel	93.1818%	65.9091%	4.54545%
Gaussian Kernel	93.1818 %	68.1818%	45.4545%
Sigmoid Kernel	90.9091 %	54.5455%	38.6364%

Polar coordinates are R&θ of re-sampled points x and y. This features are extracted using MATLAB function

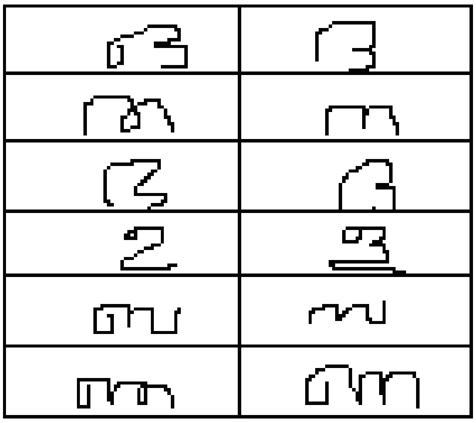

Fig. 7. Examples of common Miss-classifications between the similar alphabets

cart2pol. The first, second, third and fourth order moments are taken. A total of 8 values, 4 values for x and 4 values for y. Aspect at a point of 30 values are calculated each for x and y points and overall stroke length is measured. In order to find Curliness, a total of 24 values are extracted for a character. The values satisfied the equation requirements mentioned. The equations used for these are explained in [16].

TABLE V. CONTRIBUTION OF ASPECT AND STROKE LENGTH FEATURES TOWARDS THE RECOGNITION ACCURACY OF HANDWRITTEN ALPHABETS OF 44 CLASSES

Kernels	Aspect	Aspect&SL	Curliness
Linear Kernel	79.5455 %	81.8182%	40.9091 %
Polynomial Kernel	77.2727%	81.8182%	45.4545%
Gaussian Kernel	84.0909 %	84.0909 %	40.9091%
Sigmoid Kernel	38.6364 %	36.3636%	40.9091%

Directional and Curvature features are really important to recognize a stroke. In case of curvature and directional features, we have used mirroring technique to keep all the x, y points values. But accuracy is getting down. Therefore, first and last points values are excluded to meet the equation. Cusp detection is found using curvature angle α. If the angle is 180 degree then there is a cusp noted. Relaxation is made 5 degrees above and below 180 degree. Count is set taken to identify the cusp points.

The table VI shows the contribution of Directional and Curvature features towards the recognition accuracy of handwritten alphabets of 44 classes.

TABLE VI. CONTRIBUTION OF DIRECTIONAL AND CURVATURE FEATURES TOWARDS THE RECOGNITION ACCURACY OF HANDWRITTEN ALPHABETS OF 44 CLASSES

Kernels	Directional	Curvature
Linear Kernel	88.6364%	56.8182%
Polynomial Kernel	86.3636%	56.8182%
Gaussian Kernel	88.6364 %	59.0909%
Sigmoid Kernel	88.6364%	56.8182%

The tables shows only the features that are having greater impact on recognition of different styles of testing data. The test data is given online directly and also as a test data set from writers. Dependent writers data set almost give 100% accuracy and independent writers give a satisfying range. From experiments it is observed that, the prominent features are preprocessed x, y coordinates, directional and

TABLE VII. CONTRIBUTION OF EACH FEATURE TOWARDS THE RECOGNITION ACCURACY OF HANDWRITTEN ALPHABETS OF 44 CLASSES

Kernels	Histogram	FT	DCT
Linear Kernel	45.4545%	34.0909%	47.7273%
Polynomial Kernel	2.27273%	34.0909%	45.4545%
Gaussian Kernel	45.4545 %	27.2727%	54.5455%
Sigmoid Kernel	38.6364%	20.4545%	47.7273%

curvature features and FFT. Moments upto fourth order, aspect ratio, length of the stroke, histograms etc don't add any improvement of accuracy if we club prominent features as feature vector.

Hence Feature vector for a single stroke = [PX,PY,DC,DS,CC,CS,FT] where as:

1) PX=Preprocessed x coordinate vectors (30 points)
2) PY=Preprocessed y coordinate vectors (30 points)
3) DC=Cosine values of the directional angle (28 values)
4) DC=Sine values of the directional angle(28 values)
5) CC=Cosine values of the curvature angle(26 values)
6) CS=Sine angle of the curvature angle(26 values)
7) FT= Absolute value fourier coefficients for x and y(20 values)

Feature Vector dimension for training data is 308X188. The testing data can be given online to the trained SVM Network. The recognition speed per character is .5 secs noticed by using stopwatch timer("tic-toc") in MATLAB[1]. TIC and TOC functions work together to measure the elapsed time between a real time input character into program logic and recognized label by classifier. The classified label for test samples is matched with the database of Malayalam unicode and matched unicode is displayed into a JAVA frame. Fig. 8 shows the different styles of a stroke "ka" and the recognized result. Fig. 9 shows the identification of a word "janatha". One of the great miss-classifications is between similar character style of "tha" and "na". Here the directional feature will be the same, if the loop of "tha" is not identified. In Fig. 10 third "tha" is miss-classified as "na". This work is implemented in MATLAB R2013a. The processor used is intel i3 with 4GB RAM.

Fig. 8. Recognition of different styles of stroke "ka"

Fig. 9. Recognition of different styles of stroke "janatha"

Fig. 10. Recognition of different styles of stroke "tha" ,One missclassif-cation

VI. Conclusion

This paper proposes an SVM based online Malayalam character recognition. We explained different features and its relevance on online recognition. A total of 308 samples of different styles collected from a group of 20 people are used for training. SVM classifier with polynomial kernel, degree 3 provides good accuracy about 97.7273% in experiments and the classifier needs only less training time compared to Artificial Neural Networks and Hidden Markov Models. Since this is a real time work, we need classifiers with good speed and accuracy. The accuracy is found to be more for "straight hand writing" with a moderate writing speed. Lesser points will be recorded if too fast the writing. Characters having loops and same directionality are prone to miss-classification. Incorporating Malayalam dictionary and more training examples, miss-classification can be minimized . All the strokes experimented here are single strokes. More analysis is needed on double stroke identification and a feature extraction to identify all the strokes of complex script like Malayalam.

References

[1] Joseph, S.M; Hameed, A.,*"Online handwritten malayalam character recognition using LIBSVM in matlab"*, 2014 National Conference on Communication, Signal Processing and Networking (NCCSN).

[2] G. Seni; R.K. Srihari and N. Nasrabadi,*" Large Vocabulary Recognition of On-Line Handwritten Cursive Words"*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 18, no. 7, pp. 757-762,1996.

[3] M. Sreeraj, Sumam mary idicula ,*"On-Line Handwritten Character Recognition using Kohonen Networks"*, World Congress on Nature & Biologically Inspired Computing, 2009(NaBIC 2009).

[4] Sreeraj, M.; Idiculla, S.M., *"K-NN Based On-Line Handwritten Character Recognition System"*, Integrated Intelligent Computing (ICIIC), 2010.

[5] Arora, A., Namboodiri, A.M., *"A Hybrid Model for Recognition of Online Handwriting in Indian Scripts "*, International Conference Frontiers in Handwriting Recognition (ICFHR), 2010.

[6] Chacko, B.P.; Babu, A.P.,*" Online sequential extreme learning machine based handwritten character recognition"*, Students Technology Symposium (TechSym), 2011 IEEE.

[7] Primekumar, K. P. and Sumam Mary Idiculla, *" On-line Malayalam Handwritten Character Recognition using Wavelet transform and SFAM"*, 3rd International Conference on Electronics Computer Technology, 2011.

[8] Indhu T.R.; Bhadran V.K., *"Malayalam online handwriting recognition system: A simplified fuzzy ARTMAP approach"*, India Conference (INDICON), 2012 Annual IEEE.

[9] Primekumar K.P., Idiculla S.M,*" Online Malayalam handwritten character recognition using HMM and SVM"*, IEEE 2013 International Conference on Signal Processing, Image Processing & Pattern Recognition (ICSIPR).

[10] R. Plamondon and S. N. Srihari,*" On-line and offline handwriting recognition: A comprehensive survey"*, IEEE Trans. on Pattern Analysis and Machine Intelligence, 22(1):6384, 2000.

[11] Abdul Rahim Ahmad; Christian Viard-Gaudin; Marzuki Khalid; Rubiyah Y ousof, *"On-line Handwriting Recognition using Support Vector Machine"*, Proceedings of Second International Conference on Artificial Intelligence in Engineering & Technology, August 3-52004, Malaysia.

[12] Barabino N.; Pallavicini M.; Petrolini A; Pontil M. and Verri A, *"Support vector machines vs multi-layer perceptrons in particle identification"*, In Proceedings of the European Symposium on Artificial Neural Networks '99 (D-Facto Press, Belgium), p. 257-262, 1999.

[13] Christopher lCBurges, *"A Tutorial on Support Vector Machines tor Pattern Recognition"*, Knowledge Discovery and Data Mining ,2(2).

[14] Cristianini N. and Shawe-Taylor Ji, *An Introduction to Support Vector Machines and other Kemel-based Learning Methods*,Cambridge University Press, 2000.

[15] Richard O Duda; Peter E Hart and David G Stork,*"Pattern Classification"*, second edition.

[16] S Jaeger; S Manke; J Reichert; A Waibel, *"On-line handwriting recognition: The NPen++ Recognizer,"*, International journal of Document Analysis and Recognition. vol.3, no.3, pp 169-180.