Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Science and Engineering

Elixir Comp. Sci. & Engg. 89 (2015) 37132-37136

ISSN: 2229-712X

Text dependent writer identification using support vector machine

Saranya K[1] and Vijaya MS[2]

[1]PSGR Krishnammal College For Women, Coimbatore, India.

[2]GR Govindarajulu School of Applied Computer Technology, Coimbatore, India.

ARTICLE INFO

Article history:

Received: 1 February 2013;

Received in revised form:

18 December 2015;

Accepted: 23 December 2015;

Keywords

Feature Extraction,

Support Vector Machine,

Training,

Writer Identification.

ABSTRACT

Writer identification is the process of identifying the writer of the document based on their handwriting. Recent advances in computational engineering, artificial intelligence, data mining, image processing, pattern recognition and machine learning have shown that it is possible to automate writer identification. This paper proposes a model for text-dependent writer identification based on English handwriting. Features are extracted from scanned images of handwritten words and trained using pattern classification algorithm namely support vector machine. It is observed that accuracy of proposed writer identification model with Polynomial kernel show 94.27% accuracy.

INTRODUCTION

The significance and scope of writer identification is becoming more prominent in these days. Identification of a writer is highly essential in areas like forensic expert decision-making systems, biometric authentication in information and network security, digital rights administration, document analysis systems and also as a strong tool for physiological identification purposes.

In forensic science writer identification is used to authenticate documents such as records, diaries, wills, signatures and also in criminal justice. The digital rights administration system is used to protect the copyrights of electronic media. Two broad categories of biometric modalities are: physiological biometrics that perform person identification based on measuring a physical property of the human body (e.g. fingerprint, face, iris, retinal, hand geometry) and behavioral biometrics that use individual traits of a person's behavior for identification (e.g. voice, gait, signature, handwriting). Therefore writer identification is the category of behavioral biometrics. Handwritten document analysis is applied in fields of information retrieval either textually or graphically [1].

Writer identification mode can be generally classified into two types as online and offline. In online, the writing behavior is directly captured from the writer and converted to a sequence of signals using a transducer device but in offline the handwritten text is used for identification in the form of scanned images. Off-line writer identification is extensively considered as more challenging than on-line because it contains more information about the writing style of a person, such as pressure, speed, angle which is not available in the off-line mode.

Writer identification approaches can be categorized into two types: text-dependent and text-independent methods. In text-dependent methods, a writer has to write the identical text to perform identification but in text independent methods any text may be used to establish the identity of writer [2].

Various approaches and techniques have been proposed so far for writer identification. Writer identification using connected component contours codebook and its probability density function was proposed in [3]. This paper exhibits better identification rates by combining connected-component contours with an independent edge-based orientation and curvature PDF. In [4], eleven macro-features and micro-features have been used for writer identification. Authors in [8] have used a set of features extracted from lines of text correspond to visible characteristics of the writing such as width, slant, height of the three main writing zones and also features based on the fractal behavior of the writing for writer identification. A system for writer identification using textural features derived from the gray-level co-occurrence matrix and Gabor filters has been described in [12]. In the research work [14], Morphological features obtained from transforming the projection of the thinned writing have been computed and used for writer identification. A HMM based approach for writer identification and verification built an individual recognizer for each writer and train it with text lines of writer was proposed in [15]. A system developed in [16] for writer identification and verification takes two pages of handwritten text as input and determines whether the same writer has written those two pages and features like character height, stroke width, writing slant and skew, frequency of loops and blobs have been used.

This paper demonstrates the application of computational intelligence technique to develop discriminative model for writer identification using scanned images of English handwriting. The scanned images are segmented into words on which pre-processing and features extraction tasks are performed.

Features like edge based features, word measurements, moment invariants used in the existing research work are taken into account. Edge based features are computed using edge detected image. Edge based directional distribution and edge hinge distributions are two edge based features. Features such as length of the word, height of the word, height from baseline to upper edge, height from baseline to lower edge, ascender and descender baseline are word measurement features. Moment invariant calculates a set of seven moments for a given image.

This research work makes use of additional features that were not taken into consideration in [1]. They are the character level features like aspect ratio, loops, junctions and end points.

Tele:

E-mail addresses: k.saranyamphil@gmail.com

Aspect ratio is given by ratio of width to height. Loops identify the loop length, area etc. Junctions are defined as a point where two strokes meet or cross each other. End points are those which contain only one pixel in their 8-pixel neighborhood. These are computed by traversing the thinned image. By applying these character level features to text dependent writer identification, yields better results. The proposed model is implemented using Support Vector Machine.

PROPOSED WRITER IDENTIFICATION MODEL

The fundamental property of handwriting is that there exists writer invariant which makes writer identification possible. The writer's invariants reflecting the writing style or writing individuality of handwriting can be defined as the set of similar patterns. It is true that, the existence of writer's comparative invariance does not deny the existence of writer's variance. Also, two samples of a writer cannot be same. Hence accurate prediction of writer is highly important and challenging task. The goal of this research is to build a model based on English handwriting using Supervised learning approach namely Support Vector Machine.

In modeling automatic writer identification, the essential tasks such as data acquistion, scanning, segmentation, feature extraction, training and writer recognition have been carried out. The architecture of the proposed system is shown in Fig.1.

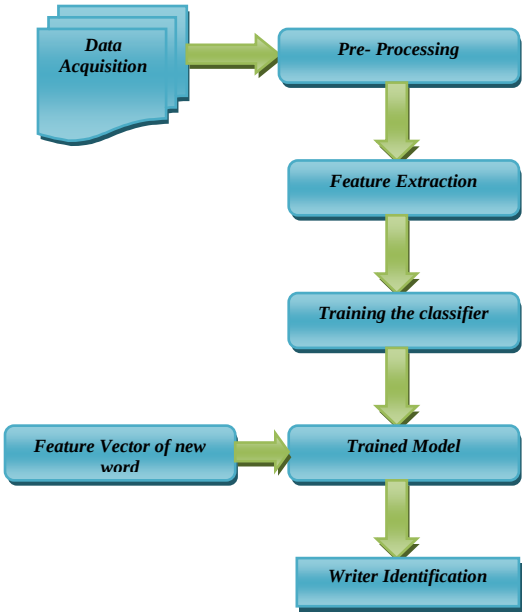

Fig. 1. **System Architecture**

Data Aquistion

The data acquistion is an important task in writer identification. In order to acquire an acceptable data, identical words written using the same pen by different writers have been used. Words that have been collected are not case sensitive and the words are scanned using scanner of resolution 300 dpi. A total of 1000 JPEG text images from 10 writers of different age groups and 100 words per writer are obtained.

Pre-Processing

In pre-processing document segmentation, noise removal, binarization, edge detection and thinning operations are performed.

Document Segmentation: The scanned handwritten document containing 100 words of a writer is into words using the edge pixels.

Noise Removal: Noise in an image is removed using median filtering.

Binarization: This converts gray scale image into binary image using Otu's method.

Edge Detection: Edges in the binary image are detected using sobel method.

Thinning: Morphological operations are used for thinning the binary image.

Feature Extraction

Feature extraction plays a vital role in improving the classification effectiveness and computational efficiency. A set of distinctive features describing the writing style and writer's invariance is extracted to form a feature vector. The features are described below.

Edge Direction Distribution

In edge direction distribution, first the edge of the binary image is detected using Sobel detection method. The edge detected image are labelled using 8- connected pixel neighbourhood. Then the number of rows and columns in an binary image is found using size function. Next, the first black pixel in an image is found and this pixel is considered as center pixel of the square neighbourhood. Then the black edge is checked using logical AND operator in all direction starting from the center pixel and ending in any one of the edge in the square. In order to avoid redundancy the upper two quadrants in the neighbourhood is checked because without on-line information, it is difficult to identify the way the writer travelled along the edge fragment. This will gives us "n" possible angles. Subsequently, the verified angles of each pixel are counted into n-bin histogram which is then normalized to a probability distribution which in turn gives the probability of an edge fragment oriented in the image at the angle measured from the horizontal. Here "n" is taken as 4, 8, 12, and 16.

Edge Hinge Distribution

To capture the curvature of ink trace, which is very distinctive for different writer, edge hinge distribution is needed, which is calculated with the help of local angles along the edges. Edge hinge feature considers two edge fragments emerging from center pixel and, subsequently, joint probability distribution of the orientations of the two fragments of a 'hinge' are calculated. Finally, normalized histogram gives the joint probability distribution for "hinged" edge fragments oriented at the angles 1 and 2. The orientation is counted in 16 directions for a single angle. From the total number of combinations of two angles only non- redundant values are considered and the common ending pixels are eliminated.

Run Length Distribution

Run lengths are determined on binarized image taking into consideration either the black pixels consistent to the ink trace or the white pixels matching to the background. Scanning procedures are of two types: horizontal along the rows of the image and vertical along the column of the image. Next, the probability distribution is interpreted by using the normalized histogram of run lengths. Orthogonal information to the directional features is obtained by using the run lengths.

Auto-Correlation

Auto-correlation function identifies the presence of predictability in writing. By giving the offset value, every row of the image is shifted onto itself. Then the normalized dot product is found between the original row and the shifted row. Auto-correlation function is computed for all rows and the sum is normalized to obtain a zero-lag correlation of 1.

Entropy

Entropy provides the average information of an image such as luminance, contrast and pixel value. It is calculated using the formula:

$$E = H[p(g)] - \sum_{j-1}^{j} p(j)H[p_j(g)]$$

Moment Invariants

Geometric moment invariant is commonly used in pattern recognition. A distinctive set of features calculated for an object must be able to identify the same object with another possible different size and orientation. Moment invariants can be used to recognize object when the object is changed in transformations. Here the following seven moments are computed.

$$M1 = \eta 20 + \eta 02,$$

$$M2 = (\eta 20 - \eta 02)^2 + (2\eta 11)^2,$$

$$M3 = (\eta 30 - 3\eta 12)^2 + (3\eta 21 - \eta 03)^2,$$

$$M4 = (\eta 30 + \eta 12)^2 + (\eta 21 + \eta 03)^2,$$

$$M5 = (\eta 30 - 3\eta 12)(\eta 30 + \eta 12)\left[(\eta 30 + \eta 12)^2 - 3(\eta 21 + \eta 03)^2\right] + (3\eta 21 - \eta 03)(\eta 21 + \eta 03)\left[3(\eta 30 + \eta 12)^2 - (\eta 21 + \eta 03)^2\right],$$

$$M6 = (\eta 20 - \eta 02)\left[(\eta 30 + \eta 12)^2 - (\eta 21 + \eta 03)^2\right] + 4\eta 11(\eta 30 + \eta 12)(\eta 21 + \eta 03),$$

$$M7 = (3\eta 21 - \eta 03)(\eta 30 + \eta 12)\left[(\eta 30 + \eta 12)^2 - 3(\eta 21 + \eta 03)^2\right] - (\eta 30 + 3\eta 12)(\eta 21 + \eta 03)\left[3(\eta 30 + \eta 12)^2 - (\eta 21 + \eta 03)^2\right].$$

Length

Length of the word is found by successively penetrating each column in the binary image to find the first and last pixels in the image and store their column numbers. The length of the image is calculated by subtracting the column number of last pixel to the column number of first pixel.

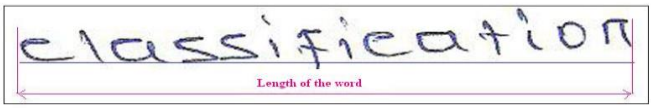

Fig. 2. Length of the word

Height

Height of the word is found by consecutively probing each row in the binary image. The first and last pixels of the image are found and the corresponding row numbers are stored. The height of the image is computed by subracting from the row number of last pixel to the row number of first pixel.

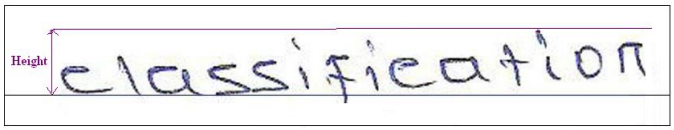

Fig. 3. Height of the word

Area

Area of the word is calculated as the product of height and length.

Height from baseline to upper edge

The height of the text from baseline to upperedge is calculated by first determining the baseline position of the image. This is functioned by casting an array where the index is row number in the image. Then, the number of black pixels in each row is calculated and the results are stored in an array. After completing the entire image, the maximum value of the array is identified and the corresponding row number is stored as the baseline. The length of the image from the baseline to the upper edge is computed by subtracting the row number of first pixel in the image from its row number of the baseline.

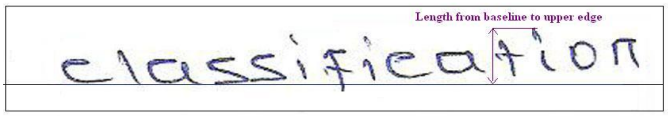

Fig. 4. Length from baseline to upper edge

Height from baseline to lower edge

The height of the binary image from the baseline to the lower edge is determined by calculating the baseline row number, as above. Then, the row number of the last pixel of the image is considered. The height of the image from the baseline to the lower edge is calculated by subtracting the last pixel row number to the baseline row number.

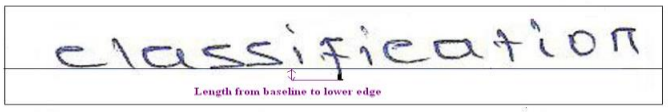

Fig. 5. Length from baseline to lower edge

Ascender and descender baseline

Ascender baseline is the first non-zero value of column and the descender baseline is the last non-zero value of column of the vertical histogram of the line.

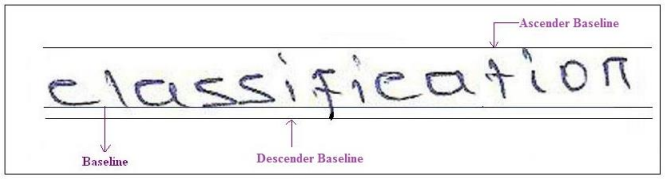

Fig. 6. Ascender and Descender Baseline

Aspect Ratio

Aspect ratio is considered as one of the global features in writer identification. It is calculated as ratio of width to height.

End Points

End-points contain only one pixel in their 8-pixel neighborhood. It is computed using end point function which gives the number of end points in the thinned image.

Junctions

Junctions occur where two strokes meet or cross and are found in the skeleton as points with more than two neighbors. It produces number of junctions, positions of each junction, angle and distance between the junctions of the thinned image.

Loops

The loops of a character are the major distinguishing feature for many writers. The loop function gives loop length, angle of loop, position of the loop, area and average radius of the loop of the edge image.

Slope Angle

The slope of angle **A** is the ratio of height to length. In geometry, it is also referred to as the tangent of the angle **A** and denoted by tan(**A**), which gives us the slope angle.

Slant Angle

It is the angle of the word forms against the baseline. It is estimated on structural features by maxima and minima of the word are detected and targets uniform slant angle estimation.

Thus a total of 25 features are extracted from a single word document image and the training dataset consisting of 1000 feature vectors is developed using MATLAB.

Support Vector Machine

Support vector machine is a training algorithm for learning classification and regression rules from data. SVM is very suitable for working accurately and efficiently with high

dimensionality feature spaces. The machine is presented with a set of training examples, (xi,yi) where the xi are the real world data instances and the yi are the labels indicating which class the instance belongs to. For the two class pattern recognition problem, yi = +1 or yi = -1. A training example (xi,yi) is called positive if yi = +1 and negative otherwise.

SVMs construct a hyperplane that separates two classes and tries to achieve maximum separation between the classes. Separating the classes with a large margin minimizes a bound on the expected generalization error. The simplest model of SVM called Maximal Margin classifier, constructs a linear separator (an optimal hyperplane) given by w T x - γ = 0 between two classes of examples. The free parameters are a vector of weights **w,** which is orthogonal to the hyperplane and a threshold value γ. These parameters are obtained by solving the following optimization problem using Lagrangian duality.

$$\text{Minimize} \quad = \frac{1}{2}\|W\|^2$$

$$\text{subject to} \quad D_{ii}\left(W^T X_i - \gamma\right) \geq 1, i = 1,.....,l.$$

Where D_{ii} corresponds to class labels +1 and –1. The instances with non-null weights are called support vectors. In the presence of outliers and wrongly classified training examples it may be useful to allow some training errors in order to avoid over fitting. A vector of slack variables ξi that measure the amount of violation of the constraints is introduced and the optimization problem referred to as soft margin is given below.

$$\underset{W,\gamma}{Minimize} = c\sum_{i=1}^{l}\varepsilon_i + \frac{1}{2}\|W\|^2$$

$$\text{subject to} \quad \underset{\varepsilon_i \geq 0}{D_{ii}}\left(W^T X_i - \gamma\right) + y_i \geq 1, i = 1,.....,l.$$

In this formulation the contribution to the objective function of margin maximization and training errors can be balanced through the use of regularization parameter c. The following decision rule is used to correctly predict the class of new instance with a minimum error.

$$f(X) = \text{sgn}\left[W^T x - \gamma\right]$$

The advantage of the dual formulation is that it permits an efficient learning of non–linear SVM separators, by introducing kernel functions. Technically, a kernel function calculates a dot product between two vectors that have been (non linearly) mapped into a high dimensional feature space. Since there is no need to perform this mapping explicitly, the training is still feasible although the dimension of the real feature space can be very high or even infinite. The parameters are obtained by solving the following non-linear SVM formulation (in Matrix form),

$$\text{Minimize } L_D(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T\mathbf{Q}\mathbf{u} - \mathbf{e}^T\mathbf{u}$$

$$\mathbf{d}^T\mathbf{u} = 0 \;,\; 0 \leq \boldsymbol{u} \leq \boldsymbol{Ce}$$

Where $\boldsymbol{Q} = \boldsymbol{DKD}$ and $\boldsymbol{K}$ - the Kernel Matrix. The kernel function K (AAT)(polynomial or Gaussian) is used to construct hyperplane in the feature space, which separates two classes linearly, by performing computations in the input space. The decision function is given by

$$f(X) = \text{sgn}\left(K\left(x, x_i^T\right) * u - \gamma\right)$$

where, u - the Lagrangian multipliers.

When the number of class labels is more than two, the binary SVM can be extended to multi class SVM. One of the indirect methods for multiclass SVM is one versus rest method. For each class a binary SVM classifier is constructed, discriminating the data points of that class against the rest. Thus in case of N classes, N binary SVM classifiers are built. During testing, each classifier yields a decision value for the test data point and the classifier with the highest positive decision value assigns its label to the data point. The comparison between the decision values produced by different SVMs is still valid because the training parameters and the dataset remain the same.

EXPERIMENT AND RESULTS

The writer identification model is implemented using SVM^{light}. It is an implementation of Vapnik's Support Vector Machine for the problem of pattern recognition, regression, and for learning a ranking function. The dataset with 1000 records are used for implementation. The features describing the properties of writers are extracted and the size of each feature vector is 25. The class label for each feature vector is assigned from 1 to 10, as the number of writers taken into account is 10. The features are normalized using min-max normalization and the normalized dataset is used for learning SVM.

The dataset is trained with linear, polynomial and RBF kernel with different parameter settings for C regularization parameter. In case of polynomial and RBF kernels, the default settings for d and gamma are used. The performance of the trained models is evaluated using 10-fold cross validation for its predictive accuracy. The prediction accuracy is measured as the ratio of number of correctly classified instances in the test dataset and the total number of test cases. The performances of the linear and nonlinear SVM classifiers are evaluated based on the two criteria, the prediction accuracy and the training time. The values of the regularization parameter C is assigned between 0.5 and 50 for linear kernel. For polynomial and RBF kernels the value for C is assigned as 0.5, 1 and 5, d is assigned from 1 to 4 and g is taken from 0.5 to 5 respectively. It is found that the model performs better and reaches a stable state for the value C = 5**.**

The result of the classification model based on SVM with linear kernel is shown Table I.

TABLE I. RESULTS OF LINEAR SVM

C	Prediction Accuracy (%)	Time Taken (in Secs)
0.5	62.50	0.01
1	65.10	0.01
5	70.83	0.02
10	70.31	0.02
15	75.00	0.01
20	70.65	0.03
25	69.27	0.01
30	68.75	0.04
35	73.95	0.02
40	71.35	0.03
45	77.60	0.01
50	73.43	0.02

The results of the classification model based on SVM with polynomial kernel and with parameters d and C are shown in Table II.

The results of the classification model based on SVM with RBF kernel and with parameters C and g are shown in Table III.

TABLE II. RESULTS OF SVM WITH POLYNOMIAL KERNEL

C	d	Prediction Accuracy (%)	Time Taken (in Secs)
0.5	1	78.12	1.18
	2	93.75	0.77
	3	91.66	2.90
	4	93.85	0.76
1	1	66.66	1.58
	2	72.91	2.35
	3	94.07	2.26
	4	93.22	0.96
5	1	71.87	1.79
	2	78.64	2.02
	3	94.27	3.49
	4	91.69	1.52

TABLE III. RESULTS OF SVM WITH RBF KERNEL

C	g	Prediction Accuracy (%)	Time Taken (in Secs)
0.5	0.5	84.37	0.04
	1	89.58	0.03
	1.5	91.10	0.04
	2	92.18	0.04
	2.5	91.19	0.05
	3	91.05	0.04
1	0.5	84.65	0.06
	1	89.15	0.08
	1.5	91.14	0.06
	2	92.08	0.07
	2.5	91.66	0.06
	3	91.86	0.06
5	0.5	78.12	0.14
	1	89.40	0.12
	1.5	91.14	0.06
	2	92.28	0.08
	2.5	91.66	0.07
	3	91.69	0.06

The average and comparative performance of SVM's is given in the Table. IV and shown in Fig.7 and Fig.8.

TABLE IV. AVERAGE PERFORMANCE OF THREE MODELS

Kernels	Accuracy	Learning time
Linear	77.60	0.01
Polynomial	94.27	3.49
RBF	92.28	0.08

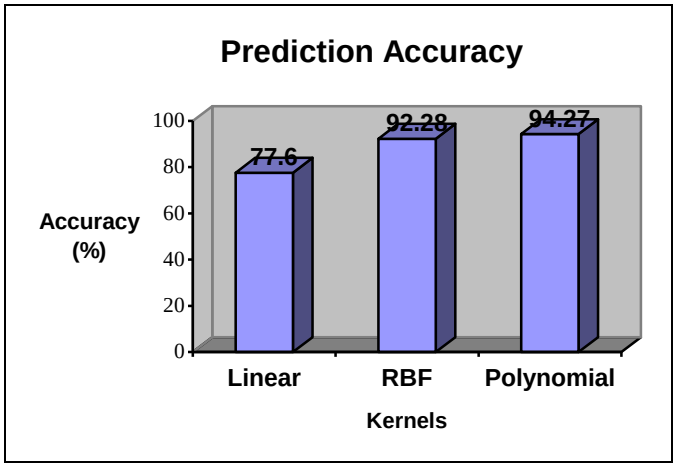

Fig. 7. Prediction Accuracy

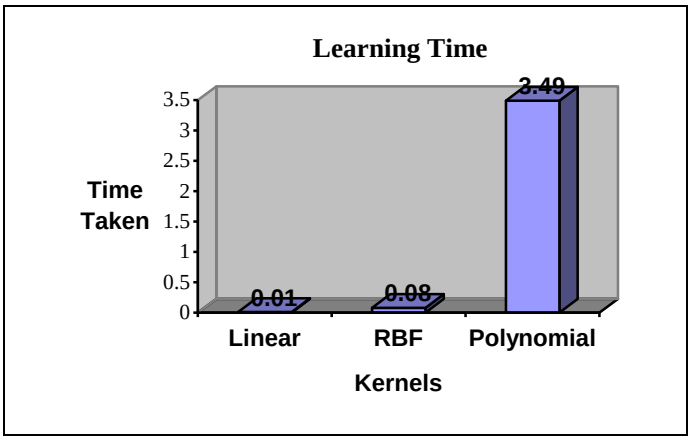

Fig. 8. Learning Time

From the above comparative analysis the predictive accuracy shown by SVM with Polynomial kernel is higher than the linear and RBF kernel. The time taken to build the model using SVM with polynomial kernel is more, than linear and RBF kernel. As far as the writer identification is concerned accuracy plays major role than learning time in identifying the writer. Hence it is concluded that SVM-Polynomial based writer identification model performs well compared to linear and RBF SVM.

CONCLUSION

This paper describes the modeling of writer identification problem as classification task. Support vector machine, a powerful supervised learning algorithm has been used for implementing the model and training dataset with 1000 instances has been prepared in order to facilitate training and implementation. The outcome of the experiments indicates that the SVM with Polynomial kernel predicts the writer of the handwritten document more accurately than the other models. It is desired that more fascinating results will pursue on further evaluation of data.

REFERENCES

[1] Al-Ma'adeed S, Mohammed E, AlKassis D, Al-Muslih F, "Writer identification using edge-based directional probability distribution features for Arabic words," IEEE/ACS International Conference on Computer Systems and Applications, pp.582-590, 2008.

[2] Bulacu M and Schomaker L, "Text-independent writer identification and verification using textural and allographic features," IEEE Trans. on Pattern Analysis and Machine Intelligence, pp.701–717, April 2007.

[3] Schomaker L and Bulacu M, "Automatic writer identification using connected component contours and edge-based features of uppercase western script", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, pp. 787–798, June 2004.

[4] Srihari S, Cha S, Arora H, and Lee S, "Individuality of Handwriting," J. Forensic Sciences, vol. 47, no. 4, pp. 1-17, 2002.

[5] Yan Y, Chen Q, Deng W and Yuan F, "Chinese Handwriting Identification Based on Stable Spectral Feature of Texture Images," International Journal of Intelligent Engineering and Systems, Vol.2, No.1, 2009.

[6] Mallikarjunaswamy BP, Karunakara K, **"**Writer Identification based on offline Handwritten Document Images in Kannada language using Empirical Mode Decomposition method," International Journal of Computer Applications, Vol. 30, No. 6, September 2011.

[7] Shahabi F and Rahmati M, "A New Method for Writer Identification of Handwritten Farsi Documents," 10th International Conference in Document Analysis and Recognition, pp. 426-430, 2009.
[8] Ubul K, et al., "Research on Uyghur off-line handwriting-based writer identification," 9th International Conference in Signal Processing, pp. 1656-1659, 2008.
[9] Helli B and Moghaddam ME, "A text-independent Persian Writer Identification based on Feature Relation Graph", Pattern Recognition, vol. 43, pp. 2199–2209, 2010.
[10] Al-Dmour A, Zitar RA, "Arabic writer identification based on hybrid spectral-statistical measures," Journal of Experimental and Theoretical Artificial Intelligence, vol. 19, no. 4, pp. 307–332, 2007.
[11] Sreeraj M, Idicula SM, "Identifying Decisive Features for Distinctive Analysis of Writings in Malayalam," IMACST: vol. 2, No. 1, may 2011.
[12] Said HES, Peake GS, Tan TN and Baker KD, "Personal identification based on handwriting," Pattern Recognition, vol. 33, pp. 149-160, 2000.
[13] Cha SH and Srihari S, "Writer identification: statistical analysis and dichotomizer," Springer LNCS 1876, pp. 123-132, 2000.
[14] Zois EN and Anastassopoulos V, "Morphological waveform coding for writer identification," Pattern Recognition, vol. 33(3), pp. 385-398, 2000.
[15] Schlapbach A and Bunke H, "Using HMM based recognizers for writer identification and verification," IEEE Proc. Of 9th Int. Workshop on Frontiers in Handwriting Recognition, pp. 167-172, 2004.
[16] Marti UV, Messerli N and Bunke H, "Writer identification using text line based features," IEEE Proc. of 6th Int. conf. on Document Analysis and Recognition, pp. 101-105, 2001.
[17] Cristianini N and Shawe-Taylor J, "An Introduction to SupportVector Machines," Cambridge University Press, 2000.