

An interactive Tool for Writer Identification based on Offline Text Dependent Approach

Saranya K

Research Scholar
PSGR Krishnammal College for Women
Coimbatore, India

Vijaya MS

Associate Professor
GR Govindarajulu School Of Applied Computer
Technology
Coimbatore, India

Abstract— Writer identification is the process of identifying the writer of the document based on their handwriting. The growth of computational engineering, artificial intelligence and pattern recognition fields owes greatly to one of the highly challenged problem of handwriting identification. This paper proposes the computational intelligence technique to develop discriminative model for writer identification based on handwritten documents. Scanned images of handwritten documents are segmented into words and these words are further segmented into characters for word level and character level writer identification. A set of features are extracted from the segmented words and characters. Feature vectors are trained using support vector machine and obtained 94.27% accuracy for word level, 90.10% for character level. An interactive tool has been developed based on the word level writer identification model.

Keywords- Feature Extraction; Support Vector Machine; Training, Writer Identification.

I. 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). Hence writer identification falls under 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 research proposes text dependent writer identification based on scanned images of English handwriting. The scanned images are segmented into words and these words are further segmented into characters 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 paper includes additional features that were not taken into consideration in [1]. They are the character level features like aspect ratio, loops, junctions and end points. The proposed work is implemented using Support Vector Machine, a supervised learning technique. Additionally an interactive tool has been developed when compared to existing works.

II. PROPOSED WRITER IDENTIFICATION MODEL

The basic 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. Also, two samples of a writer cannot be same. Hence accurate prediction of writer is highly important and challenging task. Hence it is proposed to design and develop a tool for recognizing a writer based on his / her handwriting using pattern recognition technique.

The essential tasks of writer identification are data acquisition, scanning, segmentation, feature extraction, training and writer recognition. The architecture of the proposed system is shown in Fig.1.

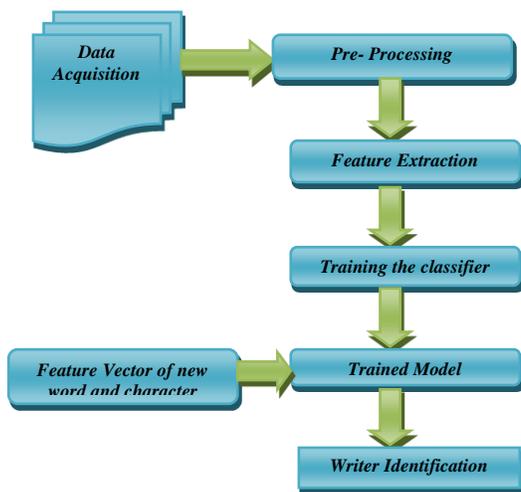


Fig.1 System Architecture

A. Data Acquisition

The data acquisition 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.

B. Pre-Processing

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

Segmentation: The scanned handwritten document containing 100 words of a writer is segmented into words using the edge pixels. These words are further segmented into characters till reaches 100 characters per writer.

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.

C. 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.

1) 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 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.

2) 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.

3) 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.

4) 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.

5) Entropy

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

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

6) 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.

$$M1 = \eta_{20} + \eta_{02},$$

$$M2 = (\eta_{20} - \eta_{02})^2 + (2\eta_{11})^2,$$

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

$$M4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2,$$

$$M5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$M6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}),$$

$$M7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ - (\eta_{30} + 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

7) Length

Length of the word and character 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.

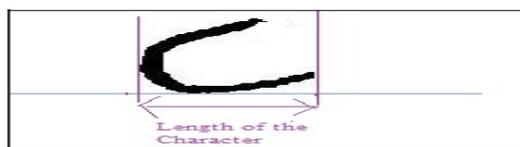


Fig. 2 Length of the character

8) Height

Height of the word and character 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 subtracting from the row number of last pixel to the row number of first pixel.

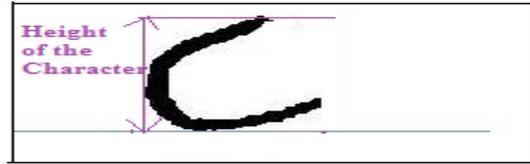


Fig. 3 Height of the character

9) Area

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

10) Height from baseline to upper edge

The height of the text from baseline to upper edge 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.

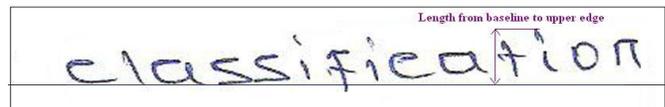


Fig. 4 Length from baseline to upper edge

11) 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.

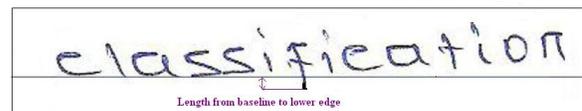


Fig. 5 Length from baseline to lower edge

12) 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.

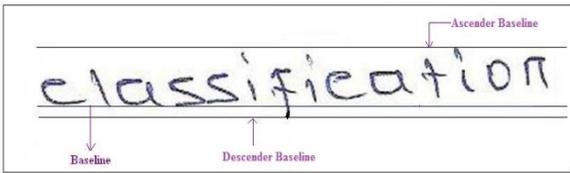


Fig. 6 Ascender and Descender Baseline

13) Aspect Ratio

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

14) 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.

15) 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.

16) 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.

17) 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 $\tan(\mathbf{A})$, which gives us the slope angle.

18) 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 26 features are extracted from a single word image. The same set of 26 features is extracted from each character image. Finally two independent training dataset each consisting of 1000 instances has been developed using MATLAB.

D. 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, (x_i, y_i) where the x_i are the real world data instances and the y_i are the labels indicating which class the instance belongs to. For the two class pattern recognition problem, $y_i = +1$ or $y_i = -1$.

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 $w^T x - \gamma = 0$ 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.

$$\begin{aligned} \text{Minimize} &= \frac{1}{2} \|W\|^2 \\ \text{subject to} & D_{ii} (W^T X_i - \gamma) \geq 1, i = 1, \dots, l. \end{aligned}$$

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 ξ_i that measure the amount of violation of the constraints is introduced and the optimization problem referred to as soft margin is given

$$\begin{aligned} \text{below. Minimize}_{w, \gamma} &= c \sum_{i=1}^l \varepsilon_i + \frac{1}{2} \|W\|^2 \\ \text{subject to} & D_{ii} (W^T X_i - \gamma) + y_i \geq 1, i = 1, \dots, l, \\ & \varepsilon_i \geq 0 \end{aligned}$$

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.

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

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),

$$\begin{aligned} \text{Minimize } L_D(\mathbf{u}) &= \frac{1}{2} \mathbf{u}^T \mathbf{Q} \mathbf{u} - \mathbf{e}^T \mathbf{u} \\ \mathbf{d}^T \mathbf{u} &= 0, \mathbf{0} \leq \mathbf{u} \leq \mathbf{C} \mathbf{e} \end{aligned}$$

Where $\mathbf{Q} = \mathbf{DKD}$ and \mathbf{K} - the Kernel Matrix. The kernel function \mathbf{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

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

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. 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.

III. EXPERIMENT AND RESULTS

Two independent experiments have been carried out, one for word level writer identification and another for character level writer identification. Two training datasets that have been developed for word level and character level are used for implementation. The datasets are normalized using min-max normalization and the normalized datasets is used for learning SVM.

The normalized datasets are trained independently using SVM^{light} for support vector machine with linear, polynomial and RBF kernels with different parameter value for C, where C is the regularization parameter. The parameters d and gamma are associated with polynomial kernel and RBF kernel respectively. The performance of trained models is evaluated using 10-fold cross validation for its predictive accuracy and the learning time. The prediction accuracy is the ratio of number of correctly classified instances in the test dataset and the total number of test cases.

A. Word Level Writer Identification

The regularization parameter C is assigned values 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 for the value C = 5.

The results of word level writer identification model based on SVM with linear kernel are shown in Table I.

TABLE I. RESULTS OF LINEAR SVM FOR WORD LEVEL

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 word level writer identification model based on SVM with polynomial kernel are shown in Table II.

TABLE II. RESULTS OF SVM WITH POLYNOMIAL KERNEL FOR WORD LEVEL

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

The results of word level writer identification model based on SVM with RBF kernel are shown in Table III.

TABLE III. RESULTS OF SVM WITH RBF KERNEL FOR WORD LEVEL

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

B. Character Level Writer Identification

The training dataset that has been developed using character level images is used here for SVM learning. The

parameter settings for character level training are same as word level training.

The results of character level writer identification model based on SVM with linear kernel are shown in Table IV.

TABLE IV. RESULTS OF LINEAR SVM FOR CHARACTER LEVEL

C	Prediction Accuracy (%)	Time Taken (in Sec)
0.5	62.08	0.01
1	60.18	0.02
5	66.35	0.02
10	72.03	0.01
15	63.98	0.03
20	64.45	0.03
25	70.61	0.01
30	72.58	0.04
35	61.13	0.03
40	74.40	0.02
45	67.77	0.05
50	72.98	0.03

The results of character level writer identification model based on SVM with polynomial kernel are shown in Table V.

TABLE V. RESULTS OF SVM WITH POLYNOMIAL KERNEL FOR CHARACTER LEVEL

C	d	Prediction Accuracy (%)	Time Taken (in Sec)
0.5	1	64.58	1.25
	2	68.75	1.95
	3	66.67	2.06
	4	88.54	3.56
1	1	73.43	1.45
	2	71.35	2.09
	3	77.60	3.12
	4	79.16	4.45
5	1	61.45	1.57
	2	63.02	2.20
	3	83.85	5.65
	4	90.10	8.03

The results of character level writer identification model based on SVM with RBF kernel are shown in Table VI.

TABLE VI. RESULTS OF SVM WITH RBF KERNEL FOR CHARACTER LEVEL

C	G	Prediction Accuracy (%)	Time Taken (in Sec)
0.5	0.5	65.10	0.05
	1	76.56	0.06
	1.5	81.52	0.06
	2	82.07	0.04
	2.5	83.03	0.06
	3	84.21	0.05
1	0.5	52.60	0.13
	1	76.56	0.07
	1.5	81.77	0.06
	2	82.60	0.06
	2.5	83.00	0.07
	3	84.37	0.06
5	0.5	52.60	0.12
	1	71.35	0.12
	1.5	79.16	0.12
	2	85.41	0.13
	2.5	82.81	0.06
	3	83.33	0.06

C. Comparative Analysis

The average and comparative performance of SVM's with various kernels for word level and character level writer identification is given in the Table. VII and shown in Fig.7 and Fig.8.

TABLE VII. AVERAGE PERFORMANCE OF THREE MODELS

SVM Kernels	Word Level Writer Identification			Character Level Writer Identification		
	Parameter	Accuracy (%)	Time Taken (in secs)	Parameter	Accuracy (%)	Time Taken (in secs)
Linear	C = 45	77.60	0.01	C = 40	74.40	0.02
RBF	C = 5, g = 2	92.28	0.08	C = 5, g = 2	85.41	0.13
Polynomial	C = 5, d = 3	94.27	3.49	C = 5, d = 4	90.10	8.03

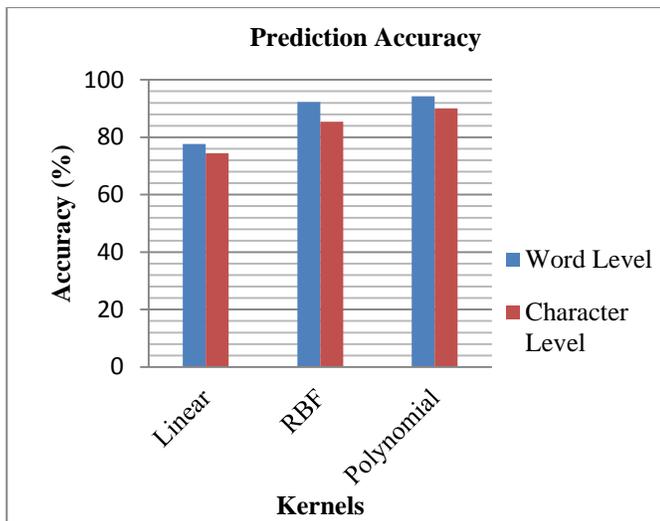


Fig. 7 Prediction Accuracy

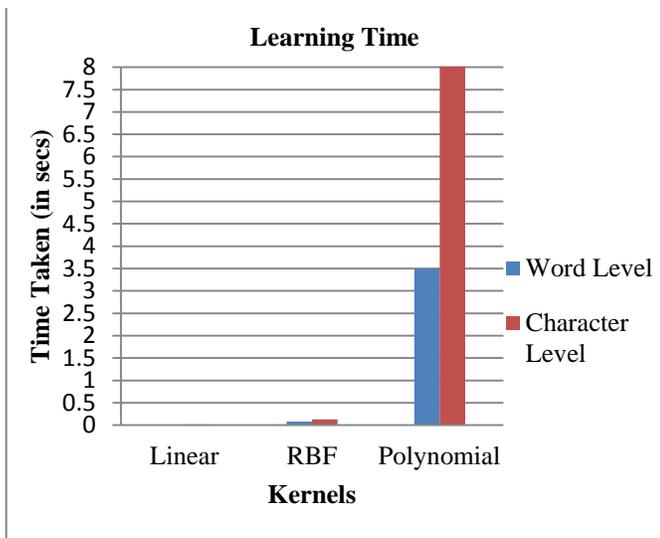


Fig. 8 Learning Time

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.

Also it is found that about 94.27% predictive accuracy for word level writer identification and 90.10% predictive accuracy for character level writer identification are shown by SVM based prediction model. Hence it is found that word level writer identification performs better than character level writer identification. The SVM polynomial based model which is producing higher accuracy is taken into consideration for developing writer identification tool.

IV. WRITER IDENTIFICATION TOOL

An interactive writer identification tool is developed using MATLAB by incorporating the SVM model with GUI. This tool is used to predict the individual when his/her offline

handwriting of a new word is given as input. Screenshots for writer identification tool are shown in Fig. 9 to 12.

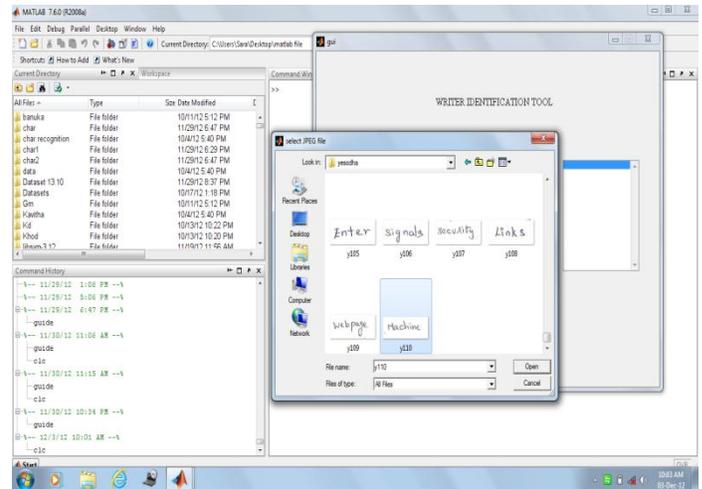


Fig. 9 Choosing the word

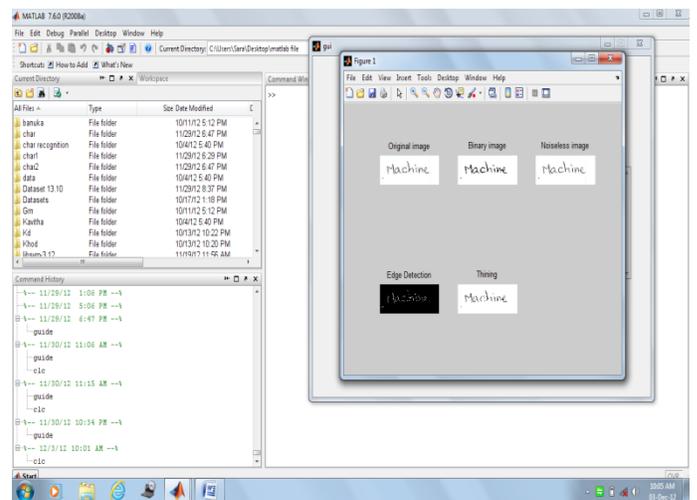


Fig. 10 After Pre-Processing

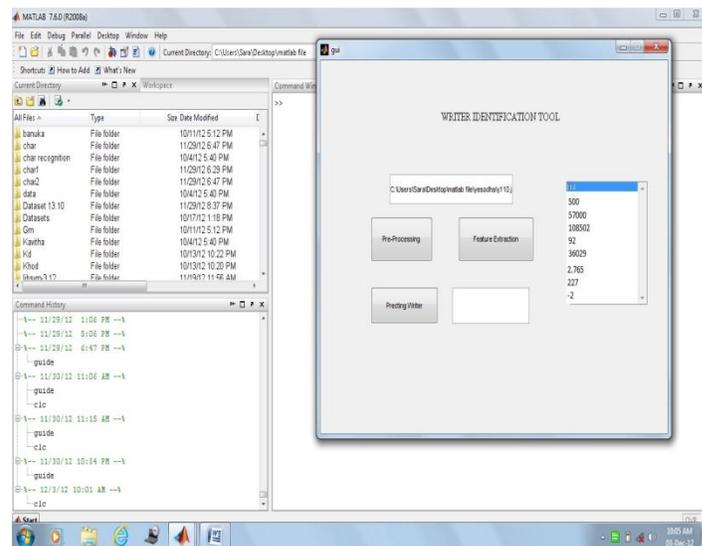


Fig. 11 Feature Extraction

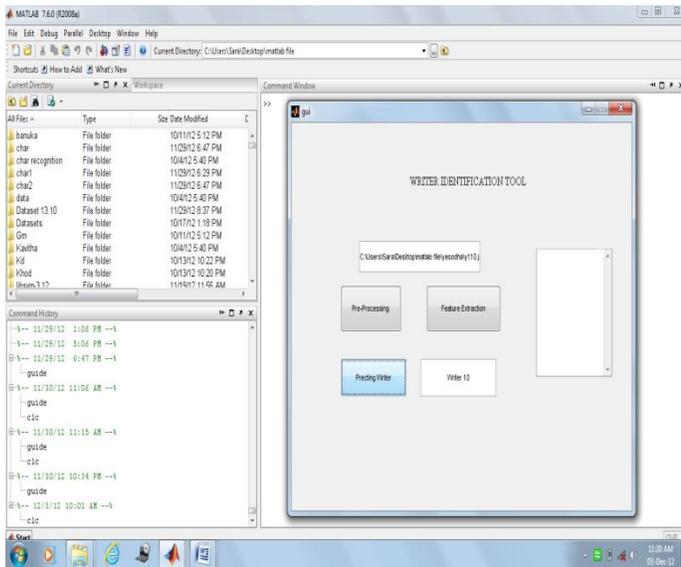


Fig. 12 Predicting the writer

V. CONCLUSION

This paper describes the modeling of writer identification problem as classification task. Two independent dataset has been prepared in order to facilitate training and implementation. The outcome of the experiments indicates that the SVM with polynomial kernel for word level writer identification predicts the writer of the handwritten document more accurately than the other models. Based on this model writer identification tool has been developed to predict the writer. Online text independent approach can be implemented as a future work,

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] 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.
- [3] Andreas Schlapbach and Horst Bunke, "Off-line Writer Identification Using Gaussian Mixture Models", 18th International Conference on Pattern Recognition, 2006
- [4] 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.
- [5] Cha SH and Srihari S, "Writer identification: statistical analysis and dichotomizer," Springer LNCS 1876, pp. 123-132, 2000.
- [6] Crammer K and Singer Y, "On the algorithmic implementation of Multiclass SVMs", JMLR, 2001
- [7] Cristianini N and Shawe-Taylor J, "An Introduction to Support Vector Machines," Cambridge University Press, 2000.
- [8] Helli B and Moghaddam ME, "A text-independent Persian Writer Identification based on Feature Relation Graph", Pattern Recognition, vol. 43, pp. 2199-2209, 2010.
- [9] 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.
- [10] 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.
- [11] Said HES, Peake GS, Tan TN and Baker KD, "Personal identification based on handwriting," Pattern Recognition, vol. 33, pp. 149-160, 2000.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] Sreeraj M, Idicula SM, "Identifying Decisive Features for Distinctive Analysis of Writings in Malayalam," IMACST: vol. 2, No. 1, may 2011.
- [16] Srihari S, Cha S, Arora H, and Lee S, "Individuality of Handwriting," J. Forensic Sciences, vol. 47, no. 4, pp. 1-17, 2002.
- [17] Ubul K, et al., "Research on Uyghur off-line handwriting-based writer identification," 9th International Conference in Signal Processing, pp. 1656-1659, 2008.
- [18] Xin Li, Xianliang Wang and Xiaoqing Ding, "An Off-line Chinese Writer Retrieval System Based on Text-sensitive Writer Identification", 18th International Conference on Pattern Recognition, 2006.
- [19] 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.
- [20] Zois EN and Anastassopoulos V, "Morphological waveform coding for writer identification," Pattern Recognition, vol. 33(3), pp. 385-398, 2000.