A New Method for Identification of Partially Similar Indian Scripts

Rajiv Kapoor *rajivkapoor@dce.edu*
Department of Electronics and Communication Engg.
Delhi Technological University
Delhi, India

Amit Dhamija *dhamija.amit@hotmail.com*
Department of Electrical Engg.
YMCA University of Science and Technology
Faridabad, Haryana, India

Abstract

In this paper, the texture symmetry/non-symmetry factor has been exploited to identify the Indian scripts. Biwavelants have been proposed to obtain the script texture using third order cumulant and bispectra. As the Indian scripts are partially similar to each other, in order to identify them, the samples must include more number of dissimilar characters. The features of individual lines are added repeatedly to enhance the dissimilarity until it reaches to a saturation level which in turn is used to compute a confidence factor i.e. amount of confidence attained in identifying a particular script sample. This variation in confidence factor also gives an estimate of the optimum sample size (number of lines) required for expected results. Cumulants are sensitive to the script curvatures and therefore are most suitable for the partially similar Indian scripts. The double discrete Fourier transform of third order cumulant gives bispectra which estimates the factor of symmetry/non-symmetry in terms of the quadratically coupled frequencies. The envelope of bispectra (biwavelant) obtained using wavelet (db8) provides an accurate behavior of the script texture; which along with Newton-Raphson technique is used to classify the Indian scripts. Various classifiers have been tested for script identification and out of them SVM gives the best results. The method successfully identified the 8 Indian scripts like Devanagari, Urdu, Gujarati, Telugu, Assamese, Gurmukhi, Kannada, and Bangla with desired accuracy.

Keywords: Indian Scripts, Cumulant, Bispectra and Support Vector Machine (SVM)

1. INTRODUCTION

Script identification is a key part of automatic processing of document images. A document script must be known in order to choose an appropriate OCR algorithm. Further processing like indexing or translation of scripts depends on identifying the language used in a document and here again script identification is crucial. Now-a-days documents are stored digitally so as to have quicker access and to save them from any kind of environmental effect. Most of the states in India have their own language of communication and independent scripts. Thus, many official documents are written in regional scripts. Identification of these regional scripts is one of the challenging tasks faced by the designer of an OCR system. Script identification makes the task of analysis and recognition of the text easier by suitably selecting the modalities of OCR. What makes recognition of Indian scripts daunting is their undistinguishable closeness. A number of attempts have already been made to isolate and identify the scripts of the texts in the case of multi-script Indian documents. Patil and Subbareddy [1] developed a system having a feature extractor and a modular neural network. They dilated the documents using 3 x 3 masks in horizontal, vertical, right diagonal, and left diagonal directions. Average pixel distribution was found in these resulting images. A combination of separately trained feed forward neural network was utilized as classifiers for each script. Hochberg [2] approach was to discover frequent character shapes in each script and then look for same instances in new documents. Some

identification techniques have also used the directional features, however to a meager amount. Dhanaya, Ramakrishnan and Pati [3] used basically two features of the scripts like Roman and Tamil. First was Spatial Spread Features like Zonal pixel concentration and character density. Directional Features were detected by using Gabor filter responses. It was concluded that Tamil script has more horizontal lines and strokes while English has more slant strokes. They used Gabor filters to effectively capture the concentration of energies in various directions. Chaudhuri and Pal [4] used skew angle detection for scanned documents containing popular Indian scripts (Devanagari and Bangla). Most characters in these scripts have horizontal line at the top called headlines (Shirorekha). Chaudhuri and Sheth [5] proposed a Gabor filter-based feature extraction scheme for the connected components. Pal and Chaudhuri [6] proposed an automatic technique of separating the text lines using script characteristics and shape-features. Spitz [7] developed techniques for distinguishing the script into two broad classes: Han-based and Latin-based. This classification was based on the spatial relationships of features related to the upward concavities in character structures. Language identification within the Han script class (Chinese, Japanese, and Korean) was performed by analysis of the distribution of optical density in the text images. Tan [8] extracted rotation invariant texture features and then used such features in script identification from document images. Rotation invariant texture features are computed based on the popular multi-channel Gabor filtering technique. Hochberg [9] used features of connected components to classify six different scripts (Arabic, Chinese, Cyrillic, Devanagari, Japanese, and Roman). Srinivasan, Ramakrishnan and Budhlakoti [10] proposed the spatial entropy obtained after decomposing the characters from the document image. The method is not adaptive to the writing styles and moreover after decomposing the characters, the spatial entropy will be definable under so many constraints which have not been discussed. Veena and Sinha [11] proposed a technique using smallest segments of the Devanagari structures to define the Devanagari characters. The method is very time consuming and detection is an issue. Sameer and Lalitha [12] suggested a preliminary technique based upon multiple classifiers like k-means classifier and Minimum Hamming Distance classifier. Anup and Anil [13] could extract temporal information due to online detection recognition and a set of features like Horizontal Inter-stroke Direction for capturing the writing direction like in the case of Arabic which is written from left to right, detection of Shirorekha for Devanagari, average stroke length, number of strokes per unit length, aspect ratio and few more like VD and VID. In these scripts, specific features could work because the scripts chosen for analysis are not related to each other and therefore the task is easier. Second kind of feature is heuristic and depends highly upon the writing style and hence will not work for all Indian scripts because they are highly related. Andrew, Wageeh and Sridharan [14] considered all scripts as texture of their own kind. Yes, this is true but the use of clustering techniques and the wavelet decomposition helps more in case of grey level images as compared to binary. Scripts which are closely related will have similar structures and texture. Texture of the scripts is formed by symmetrical spread of the structural features like horizontal lines, vertical lines and curves. This texture is of binary levels and not like grey ones as in wooden texture. Therefore the kind of features considered by Andrew, Wageeh and Sridhar do not give the high identification accuracy in case of scripts having structural and textural similarity. Morphological reconstruction [15] based upon the continuous erosion and opening was carried out in 4 directions and the average pixel distribution was found as the feature point. MLP [16] has also been used as classifier with the fuzzy-features from the Hough transform. In [17], support vector machine (SVM) based hierarchical classification scheme has been used for the recognition of handwritten Bangla characters. SVM classifier is found to outperform the other classifiers like multilayer perceptron and radial basis function network. [18] elaborates various noises that affect the performance of a script recognition system and the techniques to counter them.

What makes recognition of Indian scripts difficult is their similarity. But, since they are partially similar, their inherent dissimilarity should be enhanced in order to make them completely distinguishable. The features of individual lines are added repeatedly to enhance the dissimilarity until it reaches to a saturation level. As cumulants are sensitive to the script curvatures, they are completely suitable for identifying the Indian Scripts. This paper discusses the use of symmetry/non-symmetry factor of the script texture for identifying the partially similar Indian scripts. Biwavelants have been proposed to obtain the script texture using the third order

cumulant and the bispectra. The double discrete Fourier transform of third order cumulant gives bispectra which estimates the factor of symmetry/non-symmetry in terms of the quadratically coupled frequencies. The envelope of bispectra (biwavelant) provides an accurate behavior of the script texture which along with Newton-Raphson technique is used to classify the Indian scripts. The paper shows that for the proposed feature extraction technique, SVM gives the best classification results and can successfully identify 8 Indian scripts like Devanagari, Urdu, Gujarati, Telugu, Assamese, Gurmukhi, Kannada and Bangla.

The paper is organized as following: Section 2 describes the sample collection and pre-processing of scripts. Section 3 defines the higher order cumulants and the corresponding poly-spectra estimation. It also shows the results obtained by computing the 3rd order cumulant for different script samples. Section 4 describes the optimum parameter selection for the estimation of bispectra. Section 5 introduces biwavelants and shows the corresponding results obtained for the different Indian scripts. Section 6 describes the pre-classification stage using Newton-Raphson Technique. Section 7 elaborates different classifiers used for the proposed feature extraction technique and section 8 gives a comparison of the results obtained with different classifiers. Section 9 concludes the paper.

2. SAMPLE COLLECTION & PRE-PROCESSING

One third of the training and test data set used in this paper was collected from the news papers (*) available online, an equal amount of data was obtained by preparing the documents using different Indian fonts (#) and the last type of data comprised of the handwritten documents (^). The handwritten data was collected on a normal white paper. The documents were written using a blue ball pen. The documents were scanned offline on a canon scanner with 600 dpi resolution. The contents were not fixed and the choice was left to the writer. The total statistics of the sample collected has been mentioned in table (1) below.

Sr. No.	Script	Number of Pages	Number of Lines	Number of Words	Number of Writers
1	Devanagari	5*+10#+5^	210*+351#+100^	7462	5
2	Gujarati	5*+10#+4^	203*+336#+80^	6979	4
3	Gurmukhi	5*+10#+5^	200*+325#+90^	7000	5
4	Telugu	5*+10#+4^	215*+356#+85^	7475	4
5	Kannada	5*+10#+3^	215*+356#+60^	7225	3
6	Bangla	5*+10#+5^	198*+320#+95^	6920	5
7	Assamese	5*+10#+3^	187*+300#+60^	6305	3
8	Urdu	5*+10#+4^	180*+280#+80^	6300	4

Total number of words = 55666

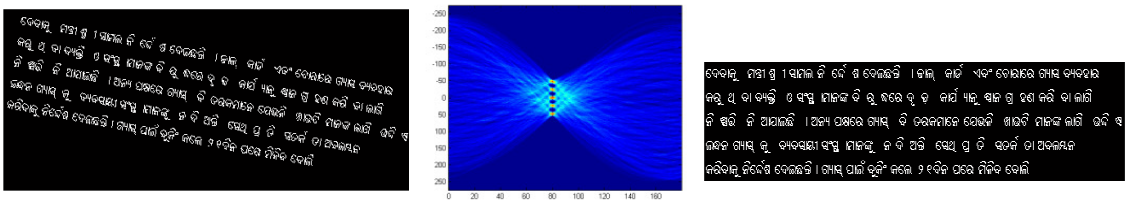
TABLE 1: Number of pages, lines and words collected for each script

The handwritten samples were not just straight lines but had lines and words written irregularly and spread over the whole document. The variety of documents made the task of pre-processing very complex and therefore, the next section is dedicated to the pre-processing.

2.1 Skew Correction

When a document is fed to the optical sensor either mechanically or manually, a few degrees of skew (tilt) is unavoidable. Person scanning the printed data document can also add skew to the text. Hand-written documents written irregularly also contain heavy skew. The lines in sample documents have been written even vertically to each other. To detect the skew angle in the printed documents, we took Radon transform of the whole document image. For an ideal skew free document, peaks corresponding to the horizontal text lines should occur at 90° in Radon image. However the scanned document will actually have peaks at an angle (Ǿ) different from 90° due to the presence of skew. Thus the document is rotated by 90°- Ǿ for skew correction. In figure (1), the Radon transform image of the scanned document showed peaks at Ǿ=80°, thus

document was rotated by 10° anticlockwise to remove the skew error. [19] discusses the method in detail.

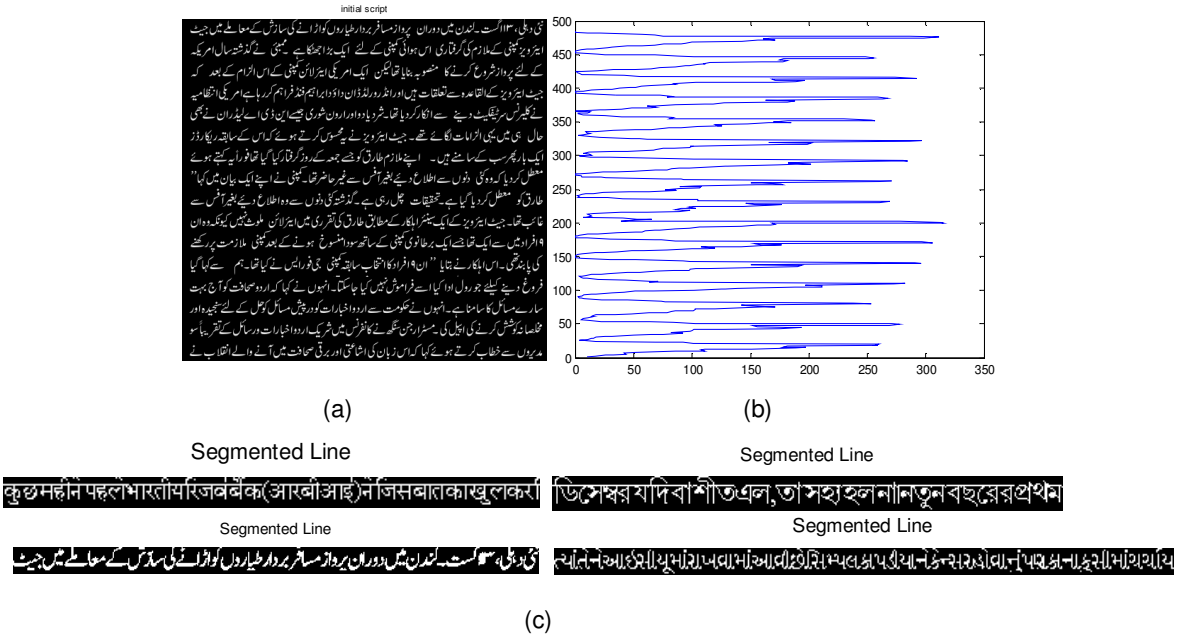


(a) (b) (c)

FIGURE 1: (a) The original scanned document image, (b) Radon transform of the document showing peaks at Ø=80° and (c) The document image after skew correction

2.2 Segmentation

Case 1: In figure (2), the lines were separated using horizontal projection and similarly the words were separated using the vertical projection. The printed documents after skew correction could be segmented completely with 100% accuracy. Separated words were concatenated to each other to remove any space in-between them and finally, the words were joined together to make a bigger line. The length of the line was approximately 14 words. These lines were used as input to the next stage of script recognition process. Space between the words was removed to avoid its effect on the result of the cumulant.



(a) (b)

(c)

FIGURE 2: First technique - line segmentation
(a) Original script, (b) Horizontal projection and (c) Segmented lines

Case 2: In some of the hand-written documents, words could not be separated using the projection technique and hence the morphologically conditioned k-means was used to separate them. The structuring element used was a line of three pixel length at an angle of 90°. Size of the structuring element was decided to make the word look like a cluster. When analyzed, the three pixel length was an optimum choice to make the word of any font and size separable. The documents were initially eroded and then k-means clustering method was applied to get the cluster centroids. The major limitation of using k-means is that it requires the optimal number of

clusters as input i.e. the total number of words in the sample document should be known very precisely before applying the unsupervised clustering techniques. MDL (minimum description length) criteria was used [20] to determine the optimum number of clusters (words) for the individual document. k-NN was used to isolate the words of the hand-written document. This method also works perfectly for the documents having words with some ligature connecting them. Figure (3) shows the document and the segmentation results.

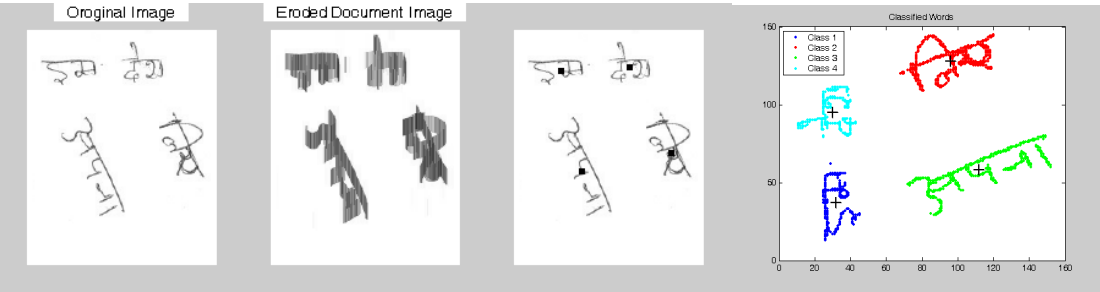


FIGURE 3: Second technique – word segmentation

Space was never allowed to be considered as part of the text for analysis. The script identification is a process which does not consider the space and the carriage return as a part of the text for getting the script features.

3. CUMULANTS

Cumulants are used to extract the inherent features of Indian scripts which are otherwise extremely difficult to extract. Higher order cumulant helps in understanding the multi-dimensional information. Structures are generally specific to the scripts, very complex and some times vary slightly from one script to the other. The paper has successfully attempted to distinguish the Indian Scripts. The first-order cumulant of a stationary process is the mean, $C_{1x} = E\{x(t)\}$. The higher-order cumulants represent central moments and therefore are invariant to the mean shift. Hence, it is convenient to define them under the assumption of zero mean. If the process has a nonzero mean, we subtract the mean and then apply the following definitions to the resulting process. The second, third and fourth-order cumulants of a zero-mean stationary process are defined by equations (1, 2 and 3).

$$C_{2x}(k) = \mathrm{E}\{x^*(n)x(n+k)\} \tag{1}$$

$$C_{3x}(k,l) = \mathrm{E}\{x^*(n)x(n+k)x(n+l)\} \tag{2}$$

$$\begin{aligned} C_{4x}(k,l,m) = {} & \mathrm{E}\{x^*(n)x(n+k)x(n+l)x^*(n+m)\} - C_{2x}(k)C_{2x}(l-m) \\ & - C_{2x}(l)C_{2x}(k-m) - M^*_{2x}(m)M_{2x}(k-l) \end{aligned} \tag{3}$$

where $M_{2x}(m) = E\{x(n)x(n+m)\}$ and equals $C_{2x}(m)$ for a real valued process. The zero-lag cumulants have special names like $C_{2x}(0)$ is the variance and $\sigma_x^2: C_{3x}(0,0)$ and $C_{4x}(0,0,0)$ are usually denoted by γ_{3x} and γ_{4x}. We will refer to the normalized quantities $\gamma_{3x}/\sigma_{2x}^3$ as the skewness and γ^4/σ_{2x}^4 as the kurtosis. These normalized quantities are both shift and scale invariant. If $x(n)$ is symmetrically distributed, its skewness is necessarily zero (but not vice versa); if $x(n)$ is Gaussian distributed, its kurtosis is necessarily zero (but not vice versa). Often the terms skewness and kurtosis are used to refer to the un-normalized quantities, γ_{3x}andγ_{4x}. Equation (4) shows that the cumulants of a stationary real-valued process are symmetric in their arguments.

$$C_{2x}(k) = C_{2x}(-k)$$

$$C_{3x}(k,l) = C_{3x}(l,k) = C_{3x}(-k,l-k)$$
$$C_{4x}(k,l,m) = C_{4x}(l,k,m) = C_{4x}(k,m,l) = C_{4x}(-k,l-k,m-k) \quad (4)$$

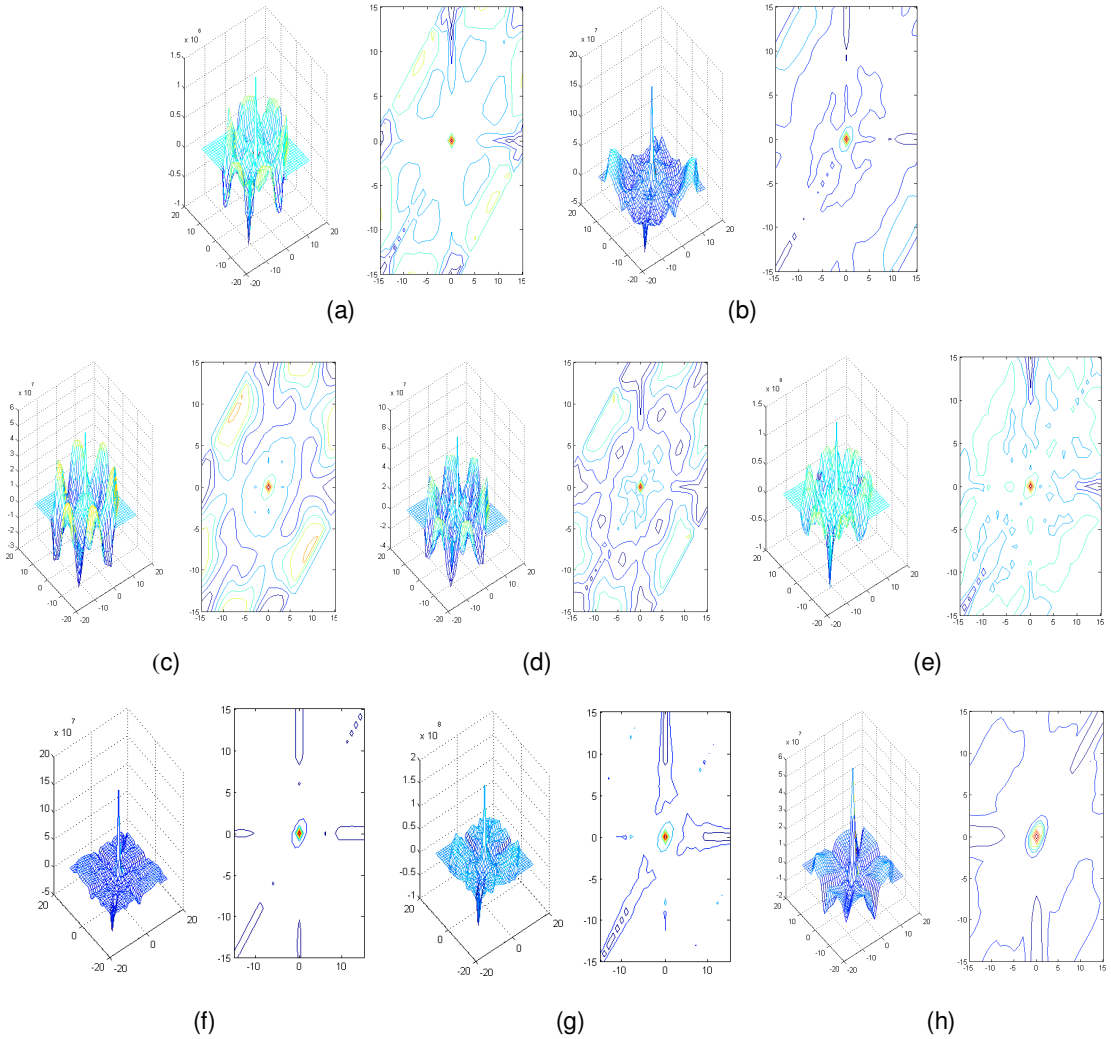


FIGURE 4: 3rd order cumulant computed for different script samples

Figure 4 (a-b) shows the 3rd order cumulant taken for the closely related scripts like Assamese and Bangla. The results demonstrate efficiency of the cumulants to distinguish among the said scripts. Similarly, figure 4(c-h) shows the 3rd order cumulant of the Gujarati, Devanagari, Gurumukhi, Telugu, Kannada and Urdu scripts, respectively. Spectrum of higher order cumulants provides features that are inherent to the script. The k^{th}order poly-spectrum is defined as the FTs of the corresponding cumulant sequence:

$$S_{2x}(f) = \sum_{k=-\infty}^{\infty} C_{2x}(k)\, e^{-j2\pi f\kappa} \quad (5)$$

$$S_{3x}(f_1,f_2) = \sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} C_{3x}(k,l)\, e^{-j2\pi f_1 k} e^{-j2\pi f_2 l} \quad (6)$$

$$S_{4x}(f_1,f_2,f_3) = \sum_{k,l,m=-\infty}^{\infty} C_{4x}(k,l,m)\ e^{-j2\pi(f_1k + f_2l + f_3m)} \quad (7)$$

which are the power spectrum, bi-spectrum and tri-spectrum, respectively. In contrast with the power spectrum which is real-valued, bispectra and tri-spectra are complex valued. For a real-valued process, symmetry properties of cumulants are carried forward to the symmetry properties of corresponding poly-spectra. The power spectrum is symmetric: $S_{2x}(f) = S_{2x}(-f)$.

Equation (8) shows the symmetry properties of the bi-spectrum:

$$S_{3x}(f_1,f_2) = S_{3x}(f_2,f_1) = S_{3x}(f_1, -f_1 - f_2) = S_{3x}(-f_1,-f_2,f_2) = S^*_{3x}(-f_1,-f_2). \quad (8)$$

Equation (9) shows the symmetry properties of the tri-spectrum:

$$S_{4x}(f_1,f_2,f_3) = S_{4x}(f_1,f_3,f_2) = S_{4x}(f_2,f_1,f_3) = S_{4x}(-f_1,f_2-f_1,f_3-f_1) = S^*_{4x}(-f_1,-f_2-f_3) \quad (9)$$

Equation (10) defines the cross-cumulants which are similar to the cross-correlations:

$$C_{xyz}(k,l) = \mathrm{E}\{x^*(n)\,y(n+k)\,z(n+l)\} \quad (10)$$

And equation (11) defined the cross-bi-spectrum:

$$S_{xyz}(f_1,f_2) = \sum_{\kappa=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} C_{xyz}(\kappa,l)\ e^{-j2\pi f_1\kappa}\ e^{-j2\pi f_2 l} \quad (11)$$

Note that the bi-spectrum $S_{2x}(f_1,f_2)$ is a special case of the cross-bi-spectrum obtained when $x = y = z$. The cross-bi-coherence is another useful statistic which is defined in equation (12):

$$bic_{xyz}(f_1,f_2) = \frac{S_{xyz}(f_1,f_2)}{\sqrt{S_{2x}(f_1+f_2)S_{2y}(f_1)S_{2z}(f_2)}} \quad (12)$$

And the cross-bi-spectrum of three processes is defined in equation (13).

$$b_{xyz}(m,n) = \int\int ln\,(S_{xyz}(f_1,f_2))\ e^{j2\pi f_1 m}\ e^{j2\pi f_2 n}\ df_1\ df_2 \quad (13)$$

This equation is well-defined only if $S_{xyz}(f_1,f_2)$ is nonzero everywhere. The 3rd order cumulant and its bispectra effectively measure the symmetry/non-symmetry of the structures belonging to different scripts. The results shown in the next section demonstrate that bispectra can effectively differentiate various Indian scripts.

4. BISPECTRA ESTIMATION

The cross-bispectra is estimated as the FT of third-order cross-cumulant of a sequence given by equation (14):

$$I^N_{xyz}(f) = \sum_{k=-N+1}^{N-1}\sum_{l=-N+1}^{N-1} \ddot{C}_{xyz}(k,l)e^{-j2\pi f_1 k}e^{-j2\pi f_2 l} = \frac{1}{N^2}X^*_N(f_1+f_2)Y_N(f_1)Z_N(f_2) \quad (14)$$

Where $X_N(f)$ is the FT of $\{x(n)\}_{n=0}^{N-1}$. This estimate is known as the cross-biperiodogram but it is not a consistent estimate. As in the case of the power spectrum, the estimate can be made consistent by suitable smoothing. The bi-spectrum and the bi-periodogram are special cases obtained when $x = y = z$. Smoothing can be accomplished by multiplying the third-order cumulant estimates by a lag window function. Let $w(t,s)$ be a 2-D window function whose 2-D FT is bounded and nonnegative with the following assumptions given in equation (15):

$$w(0,0) = 1;$$

$$\iint w^2(t,s)dtds < \infty;$$
$$\iint f_i^2 W(f_1,f_2)df_1df_2 < \infty;$$
$$\iint f_i W(f_1,f_2)df_1df_2 = 0; \quad (15)$$

The window function $w(t,s)$ should also satisfy the symmetry properties of the third-order cumulant. Equation (16) can be used to derive the 2-D lag windows from 1-D lag windows.

$$w(t,s) = w(t)w(s)w(t-s) \quad (16)$$

This satisfies the symmetry conditions of $C_{3x}(m,n)$. Consider the scaled-parameter window $w_M(t,s) = w(t/M, s/M)$ and the smoothed frequency response, given in equation (17).

$$\hat{S}_{xyz}(f_1,f_2) = \sum_{k=-N-1}^{N-1} \sum_{l=-N-1}^{N-1} \hat{C}_{xyz}(k,l)\, w_M(k,l)\, e^{-j2\pi f_1 k} e^{-j2\pi f_2 l} \quad (17)$$

Under the assumption that the cross-bispectrum $\hat{S}_{xyz}(f_1,f_2)$ is sufficiently smooth, the smoothed estimate is known to be consistent with variance given by equation (18).

$$\mathrm{var}(\hat{S}_{xyz}(f_1,f_2)) = \frac{M^2}{N} S_{2x}(f_1+f_2) S_{2y}(f_1) S_{2z}(f_2) \iint w^2(t,s)dtds \quad (18)$$

for $0 < f_1 < f_2 < \pi$. Note that the implied consistency condition is $M \to \infty$ and $M^2/N \to \infty$ as $N \to \infty$ and $\iint w^2(t,s)dtds < \infty$. Equation (17) is used to estimate the bispectra for $x = y = z$. An alternative approach is to perform the smoothing in the frequency domain. As in the case of power spectra, it is possible to segment the data into K records of length L = N/K, compute and average the biperiodogram, and then perform the frequency smoothing using the frequency-domain filter $W_M(f_1,f_2)$ estimated by taking the FT of $w_M(t,s)$. In this case,

$$var\left(\hat{S}_{xyz}(f_1,f_2)\right) = M^2/LK\, S_{2x}(f_1+f_2) S_{2y}(f_1) S_{2z}(f_2) \iint w^2(t,s)dtds \quad (19)$$

for $0 < f_1 < f_2 < \pi$. Windowing is not required in case K is very large. The following sub-section describes the parameter selection and optimization for the estimation of bispectra.

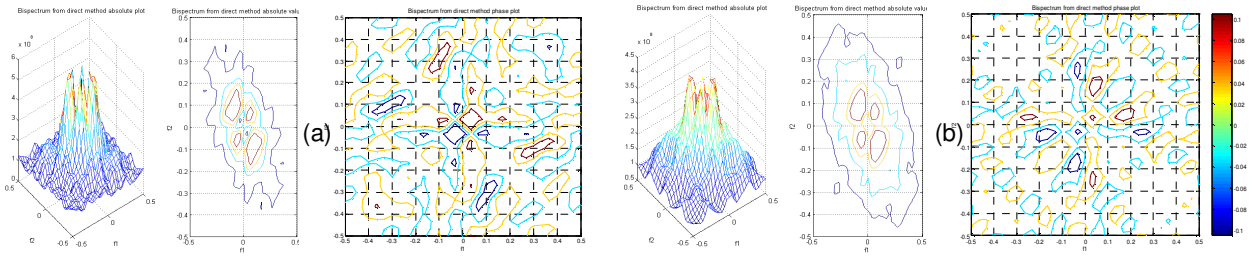
4.1. Parameters Selection and Optimization

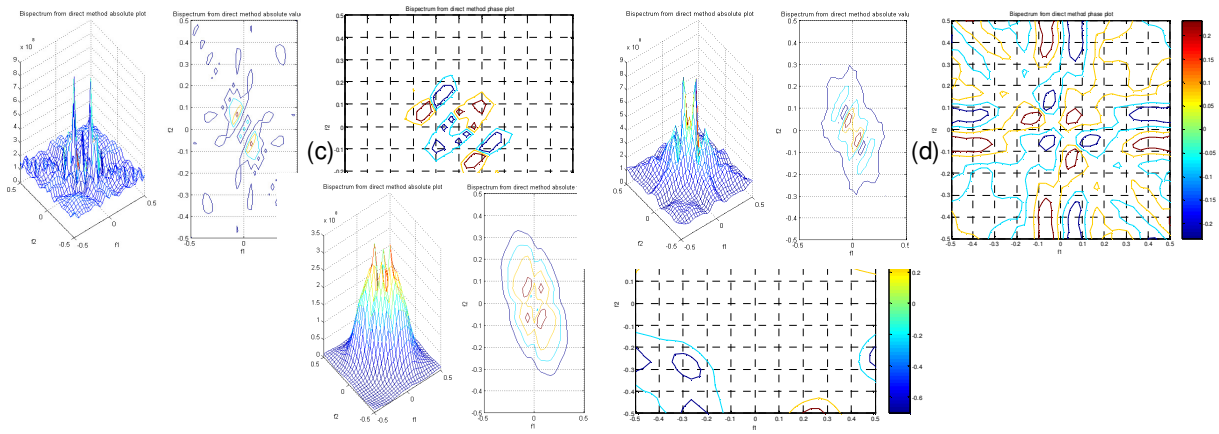
The following table comprises various parameters and their corresponding optimized values required to compute the bispectra of collected script samples. Larger MaxLag gives more number of coupled frequencies and the value lesser than this will make the process of script identification difficult. The value of the MaxLag also depends upon the data size. Here data size means size of the character of a particular script. Maximum value of the MaxLag can be the no. of pixels which describe the height of the character therefore the MaxLag value is proportional to the data size (Height of the character). Hamming window was utilized. The parameters have been optimized for the targeted scripts.

Name of the Parameter	MaxLag	Sample Rate	Window	Scale
Optimized Value	15	5	Hamming	Unbiased

TABLE 2: Optimized parameters for the bispectra estimation of collected samples

The parameters given in table (2) were utilized for the computation of bispectra of various script samples and the results are given below.





(e)

FIGURE 5: (a-e) Bispectra results (magnitude and phase) obtained for Kannada, Telugu, Assamese, Bangla and Urdu scripts, respectively

Figure (5) shows that bispectra can completely distinguish the partially similar Indian scripts.

5. BIWAVELANT

As the Indian scripts are partially similar to each other, in order to identify them, the samples must include more number of dissimilar characters. The features of individual lines are added repeatedly to enhance the dissimilarity until it reaches to a saturation level. The experimental results provided in figure (4) and (5) show that a sample size of 100 lines is sufficient to get the expected results. In order to use bispectra for script identification, the redundant information (high frequency components) is removed keeping only the prominent features (low pass information) which is described further.

5.1 Smoothing Filter vs. Wavelet

Both smoothing filter and wavelet transform can be used to remove the high frequency components from bispectra. But a smoothing filter can't protect the precious details while removing the high pass information, therefore wavelet transform is used. The wavelet transform was introduced in [21], [22], [23] and defined as

$$W_{xy}(b,a) = \frac{1}{\sqrt{a}} \int x(t)\, y(\frac{t-b}{a}) dt \qquad (20)$$

where $x(t)$ is the signal being transformed and $y(t)$ is the 'analyzing wavelet'. $y(t)$ satisfies the admissibility condition $\int |Y(\omega)|^2 \frac{d\omega}{\omega} < \infty$ which is equivalent to $\int Y(t) dt = 0$ i.e. a wavelet has zero mean. The use of the wavelet transform as a multi-resolution analysis tool has been widespread involving many applications such as fractal signal analysis, pitch detection and image compression. However, Frisch [24] and Messer [25] took a different interpretation of the continuous wavelet transform and considered it as a two parameter correlation operation where time and dilation are the correlation parameters i.e. $x(t)$is considered as a received noisy signal with known amplitude, delay and dilated factor. $y(t)$ is the template of the known shape. Therefore using the continuous wavelet transform and an appropriate decision statistic, the detection can be made for a signal buried in Gaussian noise. This interpretation will be later used in the use of wavelants. Two important properties of the cumulants are:

1. The third order cumulant for a Gaussian (or any symmetrically distributed) random process is zero.
2. If a subset of k random variables $\{x_i\}$ is independent of the rest, then the third-order cumulant is zero.

The above formulation exhibits properties closely related to those of cumulants. The higher order wavelant can also be expressed using the Fourier representations of the signals given by equations (21) and (22):

$$W^3_{XYZ}(b_1,a_1;b_2,a_2) = 1/\sqrt[3]{a_1 a_2}\iint S_{3X}(f_1,f_2)Y(a_1 f_1)Z(a_2 f_2)e^{-j(w_1 b_1 + w_2 b_2)}dw_1 dw_2 \quad (21)$$

$$W^3_{XXX}(b_1,a_1;b_2,a_2) = 1/\sqrt[3]{a_1 a_2}\iint S_{3X}(f_1,f_2)X(a_1 f_1)X(a_2 f_2)e^{-j(w_1 b_1 + w_2 b_2)}dw_1 dw_2 \quad (22)$$

The 2-D cross-wavelant for an image can be expressed as:

$$W^3_{xyz}(b_{X1},b_{Y1},a_{X1},a_{Y1};b_{X2},b_{Y2},a_{X2},a_{Y2}) =$$
$$\cdots \frac{1}{\sqrt[3]{a_{X1}a_{X2}a_{Y1}a_{Y2}}}\iint X(t_X,t_Y)Y\left(\frac{t_X-b_{X1}}{a_{X1}},\frac{t_Y-b_{Y1}}{a_{Y1}}\right)Z\left(\frac{t_X-b_{X2}}{a_{X2}},\frac{t_Y-b_{Y2}}{a_{Y2}}\right)dt_X dt_Y$$
(23)

The equations (21) and (22) define the third-order cross and auto wavelants and equation (23) defines the cross-wavelant for 2D images.

5.2 Properties

When the input used for computing the wavelant is translated and/or followed by dilation then the following properties result

If	$x(t),y(t),z(t)$	maps to	$W^3_{XYZ}(b_1,a_1;b_2,a_2)$
Then	$x\left(\frac{t-\tau}{A}\right),y(t),z(t)$	maps to	$W^3_{XYZ}\left(\frac{b_1-\tau}{A},\frac{a_1}{A};\frac{b_2-\tau}{A},\frac{a_2}{A}\right)$
and	$x(t),\frac{1}{\sqrt{A}}y\left(\frac{t-\tau}{A}\right),z(t)$	maps to	$W^3_{XYZ}(b_1+a_1\tau,a_1A;b_2,a_2)$
and	$x(t),y(t),\frac{1}{\sqrt{A}}z\left(\frac{t-\tau}{A}\right)$	maps to	$W^3_{XYZ}(b_1,a_1;b_2+a_2\tau,a_2A)$

(24)

However, if the input is first dilated and then translated, then the results are given by

If	$x(t),y(t),z(t)$	maps to	$W^3_{XYZ}(b_1,a_1;b_2,a_2)$
then	$x(At-\tau),y(t),z(t)$	maps to	$W^3_{XYZ}(Ab_1-\tau,a_1A;Ab_2-\tau,a_2A)$
and	$x(t),\sqrt{A}y(At-\tau),z(t)$	maps to	$W^3_{XYZ}\left(\frac{b1+\tau}{A},\frac{a_1}{A};b_2,a_2\right)$
and	$x(t),y(t),\sqrt{A}z(At-\tau)$	maps to	$W^3_{XYZ}\left(b_1,a_1;\frac{b_2+\tau}{A},\frac{a_2}{A}\right)$

(25)

Before applying the wavelet transform, the 2D bispectra is first converted to a 1D frequency response. The following figure shows a comparison of the results obtained at different levels of approximations (low pass filtering) by applying wavelet transform (db8) on the bispectra results.

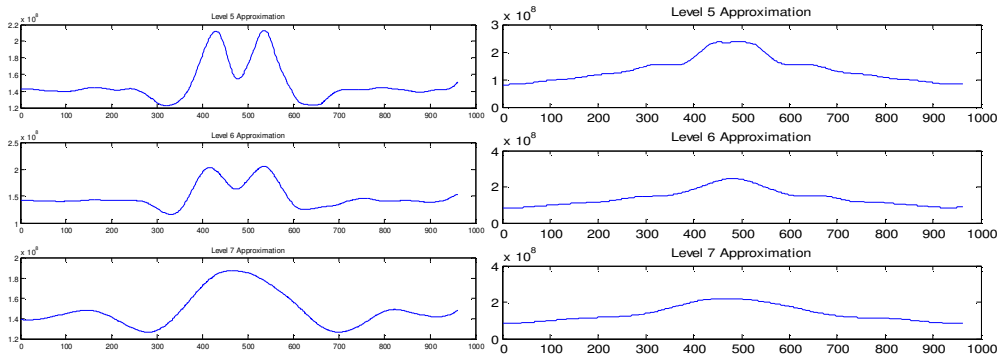


FIGURE 6: Approximation (low pass filtering) results at levels 5, 6 and 7 for (a) Assamese and (b) Bangla scripts

Figure (6) shows that after the 5th level approximation, we start loosing the precious details and therefore, before using bispectra results for identification/classification, they are approximated only up-to the 5th level. Following figure shows the results obtained for various Indian scripts.

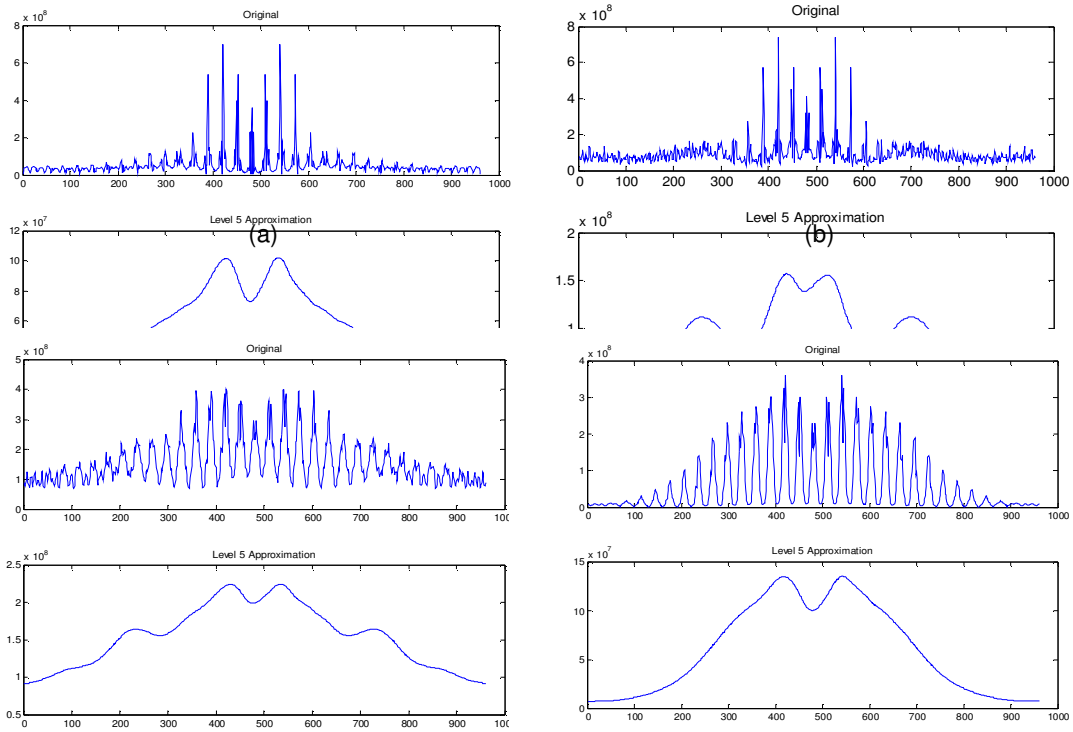


FIGURE 7: Approximated bispectra results for (a) Gujarati, (b) Bangla, (c) Telugu and (d) Urdu scripts

The following figure shows that in addition to the dissimilar results obtained for different scripts, the method gives fairly similar results for same script with different font types and sizes.

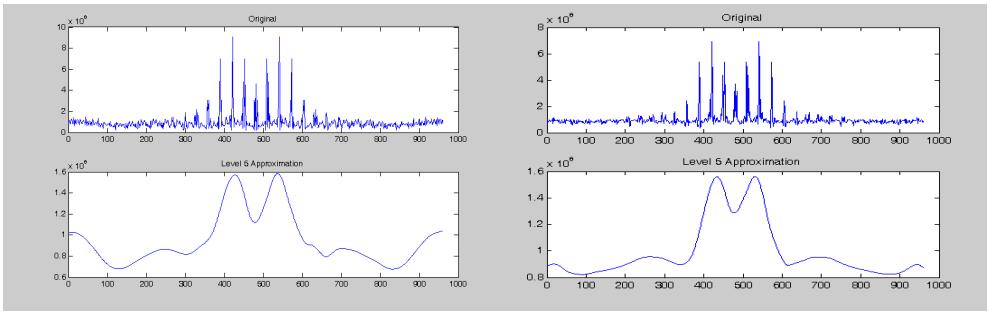


FIGURE 8: Biwavelant results obtained for font size (a) 14 and (b) 16

Figure (8) shows the biwavelant results obtained for Devanagari script for two different font sizes. Figures (7) and (8) illustrate that biwavelant (bispectra + wavelet) gives an envelope of the bispectra which proves to be a convincing feature for script identification.

6. PRE-CLASSIFICATION

6.1. Newton-Raphson Technique

The above results show that a biwavelant envelope can clearly distinguish/identify an Indian script; but using it directly for the classification/identification of a script sample is not suitable because of its high dimensionality. Therefore, Newton-Raphson technique is used to obtain the roots of a biwavelant envelop for each script sample in order to reduce the dimensionality of the feature space. The following table shows that the obtained roots clearly distinguish the Indian scripts.

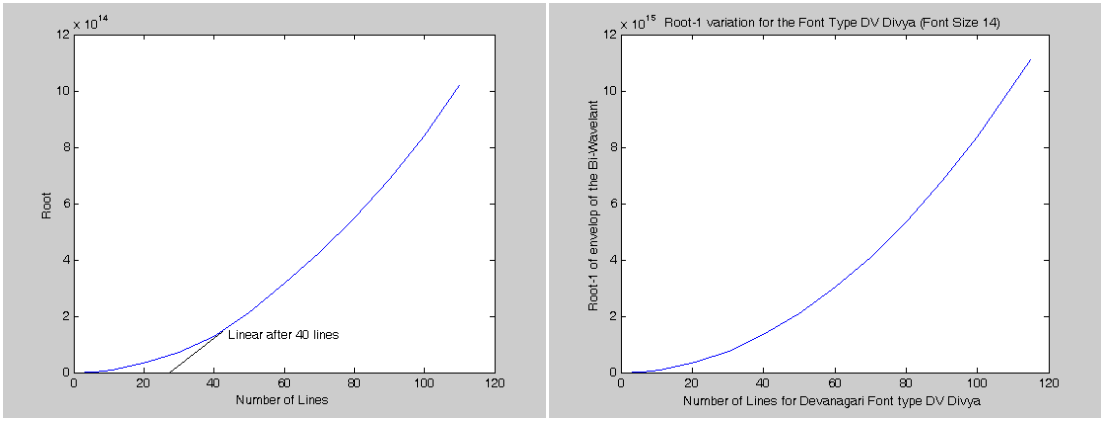
ROOTS TABLE		
Script	**Root 1**	**Root 2**
Urdu	1.36E+16	2.51E+15
Urdu	1.41E+16	2.45E+15
Urdu	1.39E+16	2.47E+15
Telugu	6.11E+15	1.68E+15
Telugu	6.39E+15	1.66E+15
Telugu	6.35E+15	1.68E+15
Bangla	1.74E+16	1.23E+15
Bangla	1.86E+16	1.26E+15
Bangla	1.62E+16	1.10E+15
Kannada	3.44E+13	1.95E+15
Kannada	3.75E+13	2.12E+15
Kannada	3.94E+13	2.11E+15
Guajarati	4.22E+15	5.10E+14
Guajarati	4.16E+15	5.27E+14
Guajarati	4.18E+15	5.16E+14
Gurmukhi	6.73E+15	9.70E+14
Gurmukhi	5.22E+15	6.60E+14
Gurmukhi	8.86E+15	10.3E+14
Assamese	2.03E+16	4.75E+14
Assamese	2.13E+16	4.78E+14
Assamese	1.97E+16	4.84E+14
Devanagari	1.05E+16	5.93E+14
Devanagari	1.06E+16	6.22E+14
Devanagari	1.06E+16	6.02E+14

TABLE 3: Roots obtained for Indian scripts using the Newton-Raphson technique

Script	No. of Lines Used	Root-1	Root-2
Assamese	110	1.70E+16	3.99E+14
Assamese	105	1.55E+16	3.64E+14
Assamese	100	1.42E+16	3.29E+14
Assamese	90	1.14E+16	2.67E+14
Assamese	80	8.97E+15	2.11E+14
Assamese	70	6.75E+15	1.63E+14
Assamese	60	4.85E+15	1.23E+14
Assamese	50	3.37E+15	8.56E+13
Assamese	40	2.07E+15	5.62E+13
Assamese	30	1.14E+15	3.23E+13
Assamese	20	5.16E+14	1.43E+13
Assamese	11	1.58E+14	4.35E+12
Assamese	6	4.69E+13	1.30E+12
Assamese	3	1.22E+13	3.14E+11

TABLE 4: Variation of the roots for Assamese script with the number of lines used in the paragraph of a script sample

This variation of roots is plotted and shown in the following figure (9).



(a) (b)

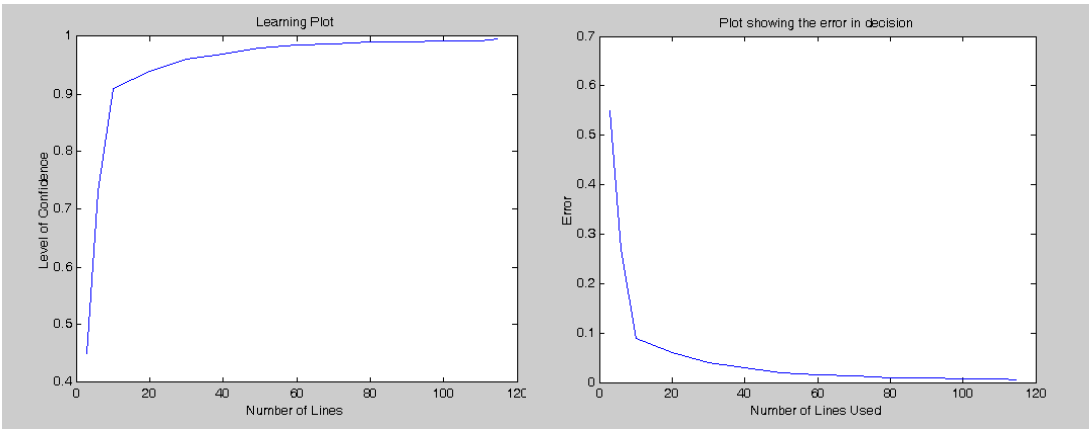
FIGURE 9: Variation of the roots with number of lines used in the paragraph of a script sample for (a) Guajarati, (b) Devanagari scripts

Figure (9) shows that the variation of roots is linear to the sample size (no. of lines) and it holds for all Indian scripts. The small non-linear portion is common to all scripts and hence, the deferential effect gets cancelled. In order to estimate the level of confidence attained in identifying a script sample and the possibility of making an erroneous decision, two parameters, $ConfidenceLevel$ and $ErrorPossibility$ are defined in equation (26) as a function of the number of lines constituting the test sample:

$$ConfidenceLevel = \frac{\sum_{k=1}^{L-1} Cum(k) - Cum(L)}{\sum_{k=1}^{L-1} Cum(k)}$$

$$ErrorPossibility = 1 - ConfidenceLevel \quad (26)$$

where $Cum(k)$ represent the third order cumulant of the k^{th} line and L is number of lines in the script sample. For a particular script sample $4 \leq L \leq 120$. The following figure shows the variation of $ConfidenceLevel$ and $ErrorPossibility$ with the number of lines in the script sample.



(a) (b)

FIGURE 10: Variation of (a) $ConfidenceLevel$ and (b) $ErrorPossibility$ with the number of lines used by the algorithm to identify the script sample

Figure (10) shows that the motive behind using more number of lines for feature extraction is to have a higher confidence level in identifying a script sample.

7. CLASSIFICATION

7.1. k-Nearest Neighbor Classification

The k-means clustering algorithm is a fast, unsupervised, nondeterministic and iterative method for generating a fixed number of disjoint clusters. Each data point is randomly assigned to one of k-initial clusters, such that each cluster has approximately the same number of points. In the subsequent iterations, distance of each point to each of the clusters is calculated using some metric and subsequently moved into the cluster corresponding to the minimum distance. Commonly used metrics are Euclidian distance to the centroid of the clusters or a weighted distance which considers only the closest n-points. The algorithm terminates when no points are moved in a single iteration. As the final result is highly dependent on the initialization of the clusters, the algorithm is often repeated a number of times, with each solution scored according to some evaluation function. The following figure shows the classification results obtained for various Indian scripts using the nearest neighbor classifier.

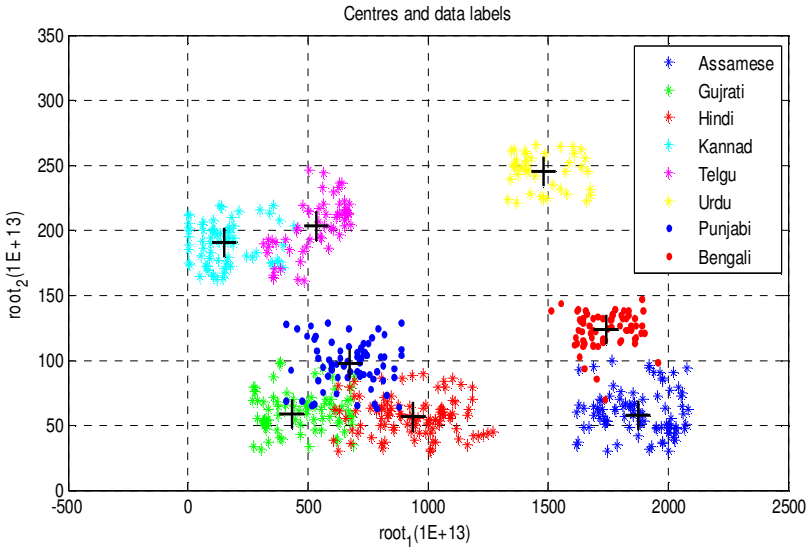


FIGURE 11: Classification results obtained for the Indian scripts

Each point in figure (11) represents the feature vector (root1, root2) corresponding to a script sample. Each sample is classified and associated to one of the eight clusters (scripts) i.e. to a particular script. The clusters are shown with different colors and markers for easy understanding. Centroid of each cluster, represented with + in figure (11), is computed using unsupervised k-means clustering and given in the following table.

CENTROIDS		
Script	**Root 1**	**Root 2**
Urdu	1.48E+16	2.45E+15
Telugu	5.36E+15	2.03E+15
Gujarati	4.31E+15	5.90E+14
Bangla	1.74E+16	1.23E+15
Kannada	1.53E+15	1.90E+15
Assamese	1.87E+16	5.70E+14
Gurmukhi	6.73E+15	9.70E+14
Devanagari	9.39E+15	5.60E+14

TABLE 5: Centroids of the clusters representing individual scripts

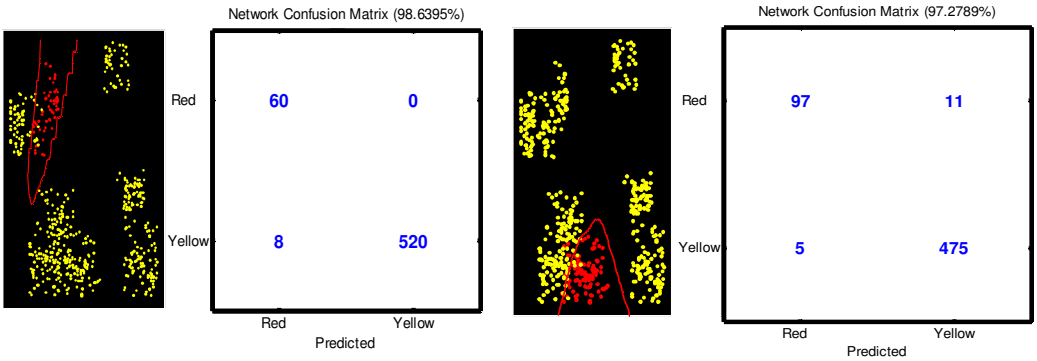
7.2. Multi-Layer Perceptron

The designed MLP network with logistic outputs has been trained with a quasi-Newton optimization algorithm and various other optimized parameters given below in table (6). The multilayer perceptron network takes two dimensional feature vectors as input.

MULTI-LAYER PERCEPTRON NETWORK		
Sr. No.	**Parameter Name**	**Parameter Value**
1	Algorithm Used	quasi-Newton optimization algorithm
2	No. of input neurons	2
3	No. of hidden layer neurons	6
4	No. of output layer neurons	1
5	Rate of weight decay	0.2
6	No. of training cycles	100
7	Activation function for hidden neurons	tanh

TABLE 6: Characteristic parameters of the multilayer network used for the training and classification

The classification results obtained using the above MLP network are shown in the following figure.



(a) (b)

FIGURE 12: Classification results and the corresponding confusion matrix for (a) Telugu, (b) Devanagari scripts

7.3. Support Vector Machine

Finally, the one vs. rest support vector machine was used to classify the partially similar Indian scripts. Various optimized parameters used to design the support vector machine are given below.

Sr. No.	Parameter Name	Parameter Value
1	Classifier Type	One vs. Rest
2	Kernel	Radial Basis Function
3	Scale	0.2
4	C	1000
5	Optimizer	libsvm

TABLE 7: Various parameters used to design the support vector machine

The classification results obtained using the proposed support vector machine is given below in the form of a confusion matrix.

Scripts	Assamese	Bangla	Gujarati	Devanagari	Kannada	Gurmukhi	Telugu	Urdu
Assamese	0.9826	0.0174	0	0	0	0	0	0
Bangla	0.0199	0.9801	0	0	0	0	0	0
Gujarati	0	0	0.9442	0.0421	0	0.0137	0	0
Devanagari	0	0	0.0540	0.9172	0	0.0288	0	0
Kannada	0	0	0	0	0.9899	0	0.0101	0
Gurmukhi	0	0	0.0521	0.0464	0	0.9015	0	0
Telugu	0	0	0	0	0.0118	0	0.9882	0
Urdu	0	0	0	0	0	0	0	1.0000

TABLE 8: Classification results obtained for various scripts and the corresponding confusion matrix

8. SCRIPT SAMPLES & CLASSIFICATION RESULTS

The proposed system was tested for eight Indian scripts using both printed and hand written samples as shown in the following figure.

FIGURE 13: Samples of eight Indian scripts used for training and testing of the proposed algorithm

In addition to the above, the proposed system was also tested on same script samples with different font types and sizes as shown in figure (14).

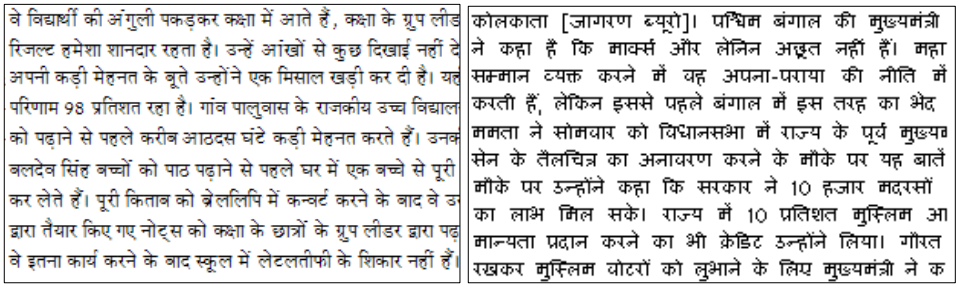


FIGURE 14: Samples of Devanagari script with different font types and sizes

The comparison of identification results obtained with various classifiers mentioned in the previous section is given in table (9).

Sr. No.	Classifier Type	Classification Accuracy (%)
1	Bayes Quadratic	90
2	Decision Layer	91
3	Nearest Neighbour Classifier	87
4	Multi Layer Perceptron	94
5	Support Vector Machine	95

TABLE 9: Classification accuracy with various classifiers

9. CONCLUSION

The method has successfully identified eight Indian scripts and is expected to work for scripts from other nations also. Indian scripts are closely related to each other and as the proposed technique is sensitive to the structural changes in the script, it is able to distinguish them successfully. But, the same sensitivity makes the method vulnerable to noise in the samples, so the document has to be noise free for expected results. However, the pre-processing becomes very complex for the removal of noise from the samples. The features of individual lines were added until they reach to a saturation level. This saturation level in turn helped in determining the confidence level for indentifying a sample. The variation in confidence level with the number of lines in the sample was used to determine an optimum number of lines required in identifying a script. A sample size of 100 lines gives the best result as it considers most of the features in the script. The method works well for both the printed and hand written samples of the scripts, independently. However, it does not work for the sample with a mixture of printed and hand-written lines of a script. Pre-processing of hand-written scripts also adds to the complexity of the method. In such a case of mixed characters, cumulants are not useful as being very sensitive to the curvatures. Indian scripts are partially similar to each other. Because of the partial similarity, we first consolidated on the number of words which enhances the partial dissimilarity and makes it look significant. Then we used the method which is very sensitive to the curvatures and the results were as expected.

REFERENCES

[1] S. B. Patil and N. V. Subbareddy, “Neural network based system for script identification in Indian documents”, *SADHNA,* vol. 27, pp. 82-97, Feb. 2002.

[2] J. Hochberg, P. Kelly, T. Thomas and L. Kerns, “Automatic script identification from document images using cluster-based templates”, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 19 (2), pp.176-181, Feb. 1997.

[3] D. Dhanaya, A. G. Ramakrishnan and P. B. Pati, “Script identification in printed bilingual documents”, *SADHNA,* vol. 27, pp. 73-82, Feb. 2002.

[4] B. B. Chaudhuri and U. Pal, "Skew angle detection of digitized Indian script documents", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 19 (2), pp.182-186, Feb. 1997.

[5] S. Chaudhuri and R. Sheth, "Trainable script identification strategies for Indian languages", *International Conference on Document Analysis and Recognition*, pp. 657-660, Sep. 1999.

[6] U. Pal and B. B. Chaudhuri, "Script line separation from Indian multi-script documents", *International Conference on Document Analysis and Recognition*, pp. 406-409, Sep. 1999.

[7] A. Spitz, "Determination of the script and language content of document images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.19 (3), pp. 235-245, 1997.

[8] T. N. Tan, "Rotation invariant texture features and their use in automatic script identification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20 (7), pp. 751-756, Jul. 1998.

[9] J. Hochberg, K. Bowers, M. Cannon and P. Kelly, "Script and language identification for handwritten document images", *International Journal on Document Analysis and Recognition*, vol. 2, pp. 45-52, Feb. 1999.

[10] S. H. Srinivasan, K. R. Ramakrishnan and S. Budhlakoti, "Character decomposition", *Indian Conference on Vision Graphics and Image Processing* at ISRO Ahmedabad, 2002.

[11] V. Bansal and R. M. K. Sinha, "On how to describe shapes of Devanagari characters and use them for recognition", *International Conference on Document Analysis and Recognition*, pp. 410-413, Sep. 1999.

[12] S. Antani and L. Agnihotri, "Gujarati character recognition", *International Conference on Document Analysis and Recognition*, pp. 418-421, Sep. 1999.

[13] A. M. Namboodiri and A. K. Jain, "Online handwritten script recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26 (1), pp. 124-130, Jan. 2004.

[14] A. Busch, W. W. Boles and S. Sridharan, "Texture for script identification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27 (11), pp. 1720-1732, Nov. 2005.

[15] B. V. Dhandhra, P. Nagabhushan, M. Hangarge, R. Hegadi and V. S. Malemath, "Script identification based on morphological reconstruction in document images", *International Conference on Pattern Recognition*, vol. 2, pp. 950-953, 2006.

[16] S. Sural and P. K. Das, "Recognition of an Indian script using MLP and Fuzzy features", *International Journal on Document Analysis and Recognition,* pp. 1120-1124, 2001.

[17] T. K. Bhowmik, P. Ghanty, A. Roy and S. K. Parui, "SVM based hierarchical architectures for handwritten Bangla character recognition", *International Journal on Document Analysis and Recognition*, vol. 12(2), pp. 97-108, July, 2009.

[18] D. Lopresti, S. Roy, K. Schulz and L. V. Subramaniam, "Special issue on noisy text analytics", *International Journal on Document Analysis and Recognition*, vol. 12(3), Sept. 2009.

[19] R. Kapoor, D. Bagai and T. S. Kamal, "A new technique for skew detection", *Pattern Recognition Letters, Elsevier Science Direct*, vol. 25(11), pp. 1215-1229, 2004.

[20] I. O. Kyrgyzov, H. Maitre and M. Campedel, “Kernel mdl to determine the number of clusters”, *International Conference on Machine Learning and Data Mining,* Jul. 2007.

[21] J. M. Combes, A. Grossman and P. Tchamitchan, “Wavelets, time-frequency methods and phase space”, Springer-Verlag, 1989.

[22] A. Grossman and R. Kronland-Martinet, “Time and scale representation obtained through

[23] continuous wavelet transform”, Signal Processing IV, Theories and Applications, vol. Elsevier Science Pub. B. V. 1988, pp. 475-482.

[24] S. G. Mallat, “A theory for multiresolution signal decomposition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11 (7), Jul. 1989.

[25] M. Frisch and H. Messer, “The use of the wavelet transform in the detection of an unknown transient signal”, *IEEE Transaction on Information Theory*, *Special Issue on Wavelet Transforms and Multiresolution Signal Analysis*, vol. 38(2), pp. 892-897, Mar. 1992.

[26] M. Frisch and H. Messer, “Detection of a transient signal of unknown scaling and arrival time using the discrete wavelet transform”, *International Conference on Acoustics, Speech and Signal Processing,* vol. 2, pp. 1313-1316, Apr. 1991.