Chapter 9
Language-Based Classification of Document Images Using Hybrid Texture Features

Umesh D. Dixit and M. S. Shirdhonkar

9.1 Introduction

Rapid development of technology has given rise in number of document images in every language. A huge database of multilingual document images needs an automatic language-based classification system. The language-based classification of documents has following applications:

- Document categorization based on the domain
- Retrieving of documents
- OCR implementation
- Digital libraries
- Text to speech conversion

Figure 9.1 shows taxonomy of language-based/script-based classification of document images. Both global and local analysis can be applied to identify the scripts. Global analysis includes feature extraction at paragraph or block level. Local analysis of document shall be implemented at two levels: word level and line level. In the word level, initially the document image is segmented into words using connected component analysis. Normally the structural features, texture features, or hybrid features are employed. Structural features consist of information about character strokes, orientation, and their sizes. Texture features are the presentation of visual appearance of components and their frequency. Hybrid features include the usage of both structural and texture features. In the word-level implementation,

U. D. Dixit (✉)
Department of Electronics & Communication Engineering, B.L.D.E.A's, V.P. Dr. P.G. Halakatti College of Engineering & Technology, Vijayapur, Karnataka, India

M. S. Shirdhonkar
Department of Computer Science & Engineering, B.L.D.E.A's, V.P. Dr. P.G. Halakatti College of Engineering & Technology, Vijayapur, Karnataka, India

G.R. Sinha (ed.), *Advances in Biometrics*,
https://doi.org/10.1007/978-3-030-30436-2_9

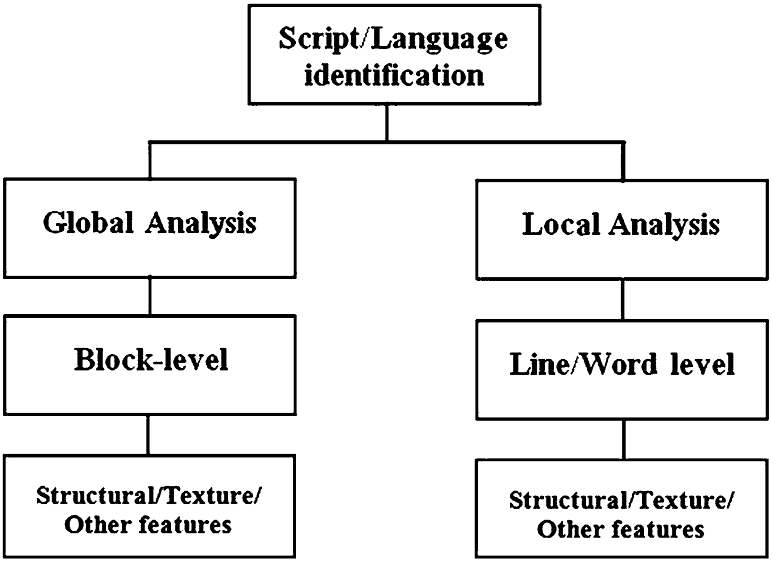


Fig. 9.1 Taxonomy of language/script identification

the extracted features of the connected components are employed for classification of the document images. The line-level implementation includes segmentation of lines using horizontal and vertical profiles of the text, feature extraction, and then classification of the documents based on the language. However in paragraph or block-level implementation entire document image is treated as a block. The features from this block are extracted to train the system in order to carry out language-based document image classification.

As the global analysis of document image is segmentation-free, it has an advantage of faster classification. The local analysis requires an additional preprocessing time for segmentation of the document, but has an advantage of accurate classification. To improve both speed and accuracy of classification, there is a need for development of new feature extraction schemes.

The objective of the work presented in this chapter is to develop a system that classifies printed document images based on the language used. This chapter proposes a segmentation-free technique for language-based classification. The proposed method is evaluated for Kannada, Telugu, Marathi, Hindi, and English documents. Kannada, Telugu, and Marathi are the official languages of Karnataka, Telangana, and Maharashtra states of India. Hindi and English are the national and global languages, which are officially accepted across India. Figure 9.2 shows sample document images of Kannada, Telugu, Marathi, Hindi, and English.

Kannada and Telugu scripts are derived from Brahmi alphabet of ancient India. During the twelfth and fifteenth century, these two scripts are split into separate alphabets. The Kannada language has 16 vowels and 34 consonants with 250 basic, compound, and modified shapes. The Telugu language includes 16 vowels, 3 vowel modifiers, and 41 consonants. Thus it has a total of 60 symbols. Hindi and Marathi

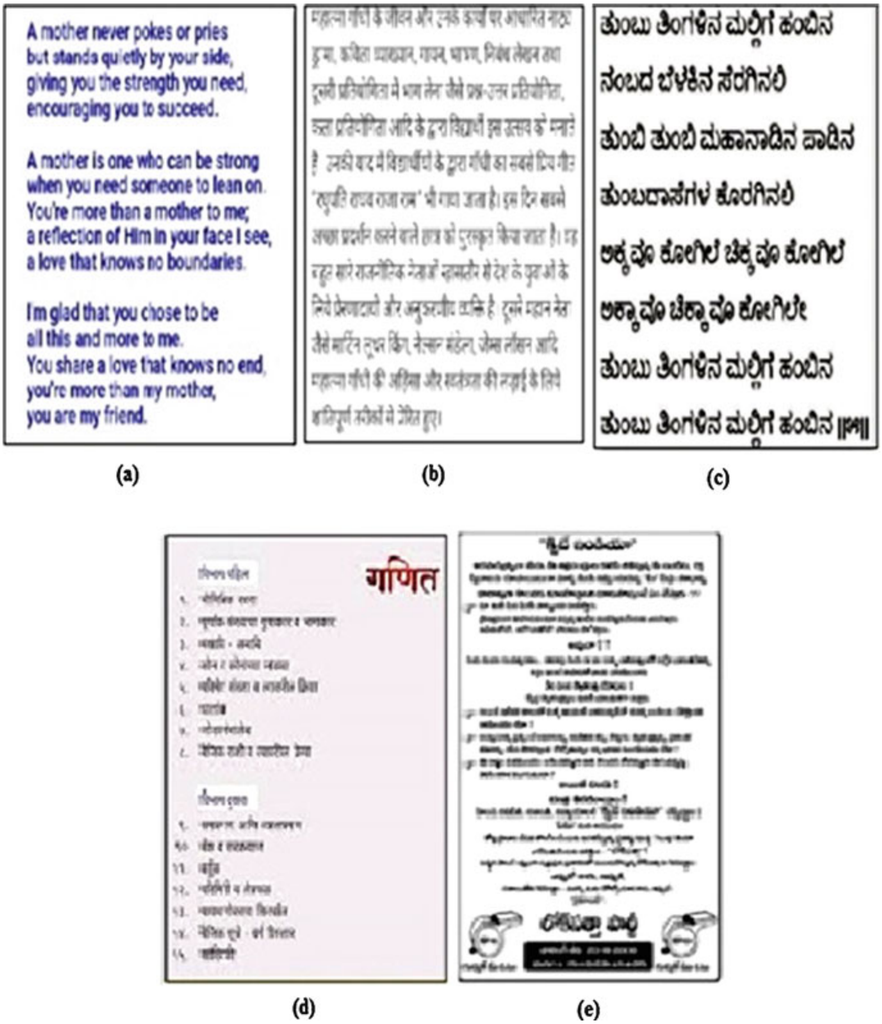


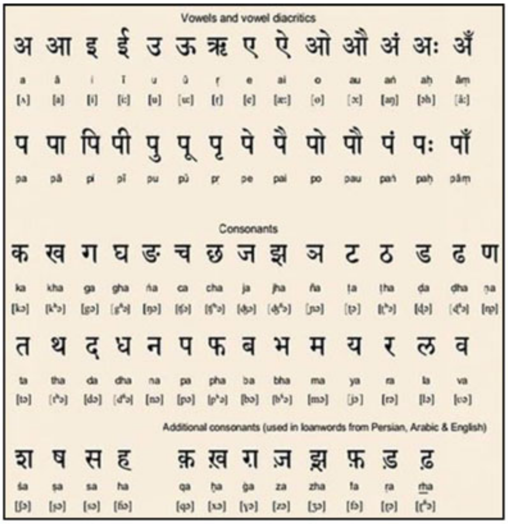
Fig. 9.2 Sample documents of English, Hindi, Kannada, Marathi, and Telugu

are derived from Devanagari script. Both Hindi and Marathi scripts have a horizontal line at the top and connects all the letters. The Hindi script has 12 vowels and 34 consonants, whereas Marathi includes 16 vowels and 36 consonants. English uses Latin-based alphabets with 26 letters. It has 5 vowels and 21 consonants. The important feature of English alphabets is that most of them have vertical and slant strokes. Figure 9.3 shows the vowels and consonants of Kannada, Telugu, Marathi, Hindi, and English languages.

The challenging task in the proposed document classification is due to similarity between Hindi and Marathi scripts as well as Kannada and Telugu scripts. But the texture features formed by the words used in these languages will be distinct. This

Fig. 9.3 Vowels and consonants of Kannada, Telugu, Marathi, and Hindi languages. (**a**) Kannada vowels and consonants. (**b**) Telugu vowels and consonants. (**c**) Marathi vowels and consonants. (**d**) Hindi vowels and consonants. (**e**) English vowels and consonants

Fig. 9.3 (continued)



(d)

(e)

motivated us to propose a suitable texture features for language-based classification of document images. The important contribution of this chapter is proposing the usage of hybrid features employing SWT and HOG to improve classification accuracy. The chapter also presents comparative analysis of proposed hybrid features with (1) rotation invariant local binary pattern, (2) histogram of oriented gradients, and (3) multi-resolution HOG feature. Evaluation of the method is carried out on a database of 1006 document images. The proposed feature extraction scheme with SVM classifier provided better classification accuracy compared with current state-of-art techniques.

The rest of the chapter is organized as follows:

- Section 9.2 briefs the literature review of language-based identification of document images.
- Section 9.3 details about the methodology.
- Section 9.4 presents experimental results and Sect. 9.5 concludes about the work.

9.2 Literature Review

Script/language identification is a subfield of document image analysis. Lot of work has been carried out in document image analysis. A detailed survey on document image analysis is provided in [1]. The following section deals with the work related to script/language identification.

Chaudhury et al. [2] presented script identification system for Indian languages. They used Gabor filter-based features extracted from connected components, with combined classifiers to improve the performance. Kulkarni et al. [3] proposed script identification from multilingual documents using visual clue-based features. Eight different visible features are employed with probabilistic neural network (PNN). Padma and Vijaya [4] employed profile-based features with k-nearest neighbor classifier for script identification from trilingual documents.

Pal and Chaudhuri [5] developed a system for identification of English, Bangla, Arabic, Chinese, and Devanagari script lines from a document. They combined shape-based, statistical-based, and some of the water reservoir-based features in their work. Rajput et al. [6] presented a system for handwritten text identification using DCT and the wavelet features. They processed document at block level and used K-nearest neighbor approach for classification. Mathematical and structure-based features with a series of classifiers have been applied to improve the performance of script identification for Indian document images in [7]. Shirdhonkar and Kokare [8] presented a technique to discriminate printed text and handwritten text using neural network model and SVM.

Pardeshi et al. [9] used multi-resolution spatial features for Indian script identification. They extracted features by applying radon transform, DWT, and DCT on the segmented words of the document images. Tan et al. [10] used word shape analysis to retrieve text from the document images. Wanchoo et al. [11] provided a survey of Devanagari script recognition for Indian postal system. Sahare and Dhok [12] detailed about the algorithms used for recognition of text in their work. They also included comparison of the different schemes used for text recognition. A detailed survey on document image analysis with its applications, challenges, and current state of art is presented by Dixit and Shirdhonkar [13].

Arani et al. [14] used hidden Markov model (HMM) for recognition of handwritten Farsi words. They used multilayer perceptron (MLP) with an input of features obtained from image gradient, contour chain code, and black-white transitions. Bi et al. [15] presented their final version of Chinese handwritten character recognition system using convolutional neural network (CNN) model with GoogLeNet. Djeddi et al. [16] presented a system for writer recognition using multi-script handwritten text, comprising of Greek and English languages. They used run length features with k-NN and SVM classifiers. Dixit and Shirdhonkar [17] proposed the multi-resolution LBP features in their fingerprint-based document image retrieval work. They compared the results of multi-resolution LBP features obtained using DWT and SWT. Roy et al. [18] proposed HMM-based Indic handwritten word recognition system. They used features obtained from zone-wise segmentation of words in their work.

In the literature, we found that most of the text recognition schemes are based on the line level or word level. Only a handful of works were reported text recognition at block level or document level. Segmentation-free script identification at block level improves the classification speed and is helpful to retrieve documents based on the script. The scripts used to form words and sentences, in different languages, will have visually distinct features. We exploited these features for language-based classification. A hybrid feature extraction scheme that combines SWT and HOG is proposed for improved performance. We used SWT to decompose the document image into horizontal, vertical, and diagonal details of the scripts. The decomposition helped in obtaining more precise orientation of gradients to construct the feature vector.

9.3 Proposed Methodology

Figure 9.4 provides architecture of the proposed work. It includes training phase and testing phase. Preprocessing, feature extraction, and classification are the building blocks of the proposed architecture. These blocks are discussed in the subsequent sections.

9.3.1 Preprocessing

In this step the document image is prepared for feature extraction. Initially the document image is converted into grayscale using the equation (9.1):

$$I = 0.2989 \times R + 0.5870 \times G + 0.114 \times B \tag{9.1}$$

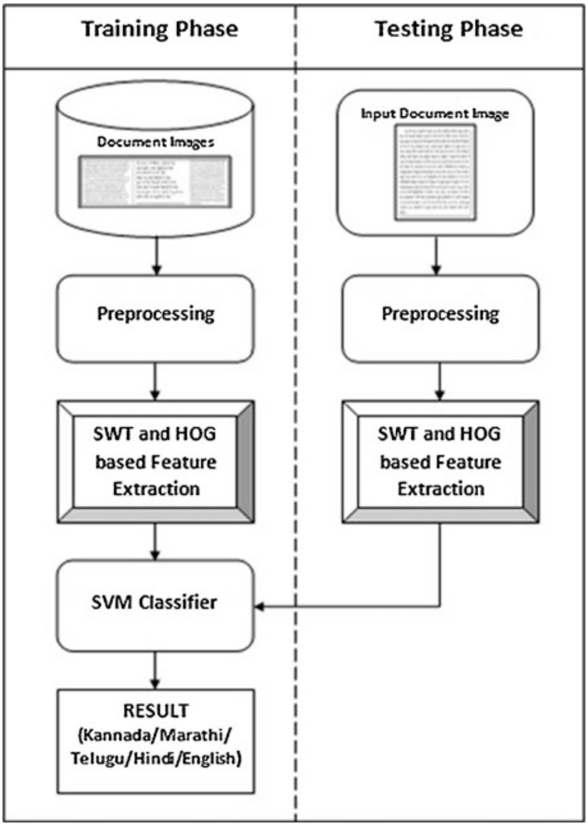


Fig. 9.4 Proposed language-based classification

Fig. 9.5 Unsharp filter mask

1	-1	-1
-1	8	-1
-1	-1	-1

"R," "G," and "B" in the equation (9.1) are the red, green, and blue components of the image. The converted grayscale image is denoted as "I." Low-contrast document images provide inaccurate texture features and lead to poor classification. Hence to increase quality of the input image for improved classification, we convolved the image "I" with a 3×3 filter shown in Fig. 9.5 to perform unsharp masking [19] and then applied a low-pass filter.

Figure 9.6 shows the results of preprocessing steps for a sample document image. It includes input color image, its grayscale version, output of unsharp masking, and the low-pass filtered image. Thus in the preprocessing step, we improve quality of the document image in terms of contrast to obtain more accurate texture features in the next step. The steps used in preprocessing of the document image are listed in Algorithm 9.1.

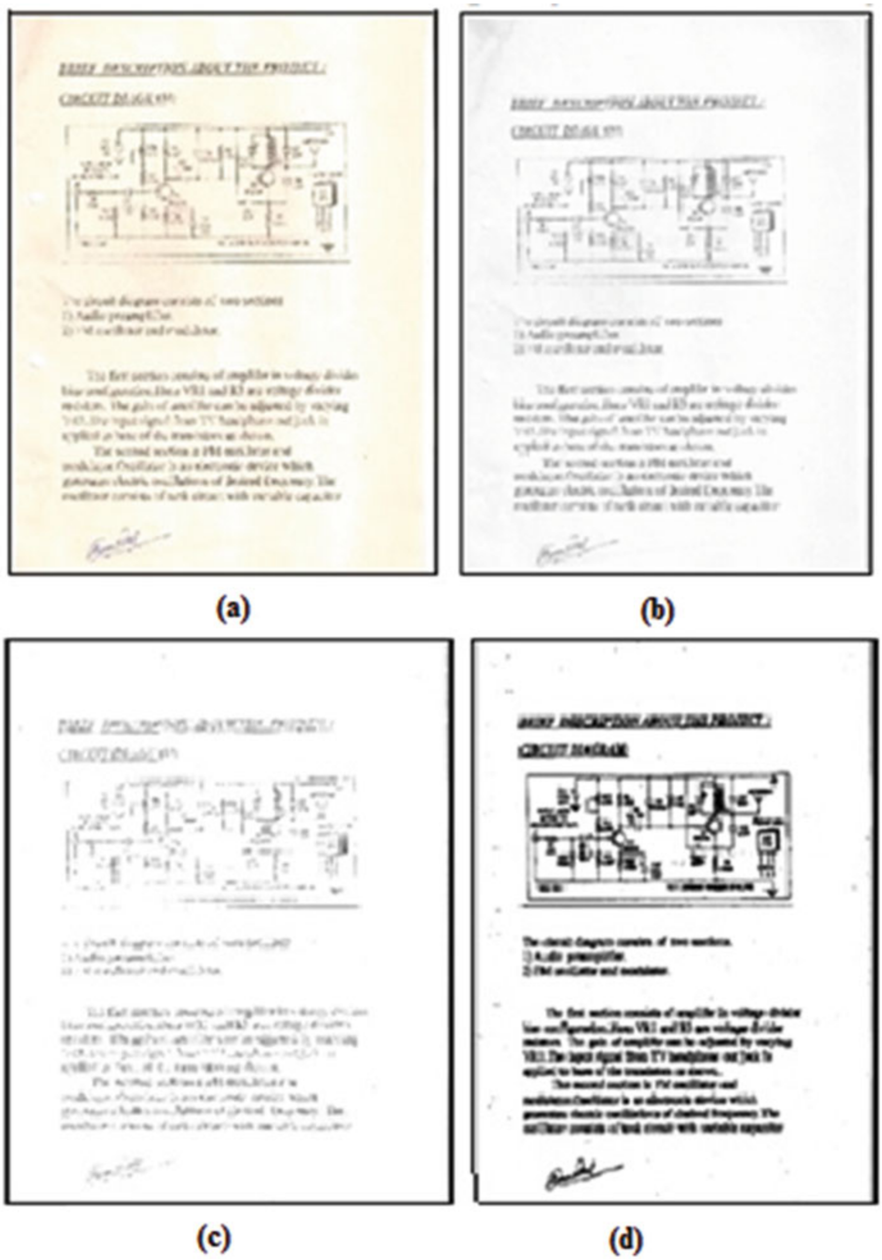


Fig. 9.6 Results of preprocessing steps

Algorithm 9.1 Preprocessing

```
1. Begin
   Input: Document image
   Output: Preprocessed document image D(x,y).
2. Read the input document image
3. if input is color image
      Convert to gray-scale using equation (9.1)
   end if
4. Perform un-sharp masking using mask shown in Fig. 9.5
5. Apply low-pass filter
6. End
```

9.3.2 Proposed Hybrid Texture Features

Figure 9.7 shows the proposed feature extraction scheme employed in this work. We used SWT- and HOG-based hybrid features. The document image is initially decomposed by applying DWT and then HOG features are obtained from each of these decomposed components. This process is explained in the following sections.

9.3.2.1 Stationary Wavelet Transform (SWT)

This section provides the details of DWT, SWT, and the application of SWT in the proposed method of language-based document image classification.

DWT The word wavelet was first introduced by Morlet and Grossman in the design of Morlet wavelet. In 1984, wavelet with new property called orthogonality was proposed by Meyer. The orthogonal property states that the information obtained by one wavelet will be entirely independent of the information captured by another wavelet. An idea of multi-resolution that is a pyramidal algorithm was developed by Stephane Mallat in 1986. The kernel functions used in wavelet transform are obtained by a prototype function referred to as mother wavelet, which is given by equation (9.2):

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}}\Psi\frac{t-b}{a} \tag{9.2}$$

where “*a*” is the scaling factor and “*t* - *b*” is the translation parameter. The term $1/\sqrt{a}$ is used as normalization factor to ensure that all the wavelets carry same energy. Thus the wavelet of a signal is a mother wavelet at scale “*a*,” lagged by *b*. Figure 9.8 shows plot of a mother wavelet.

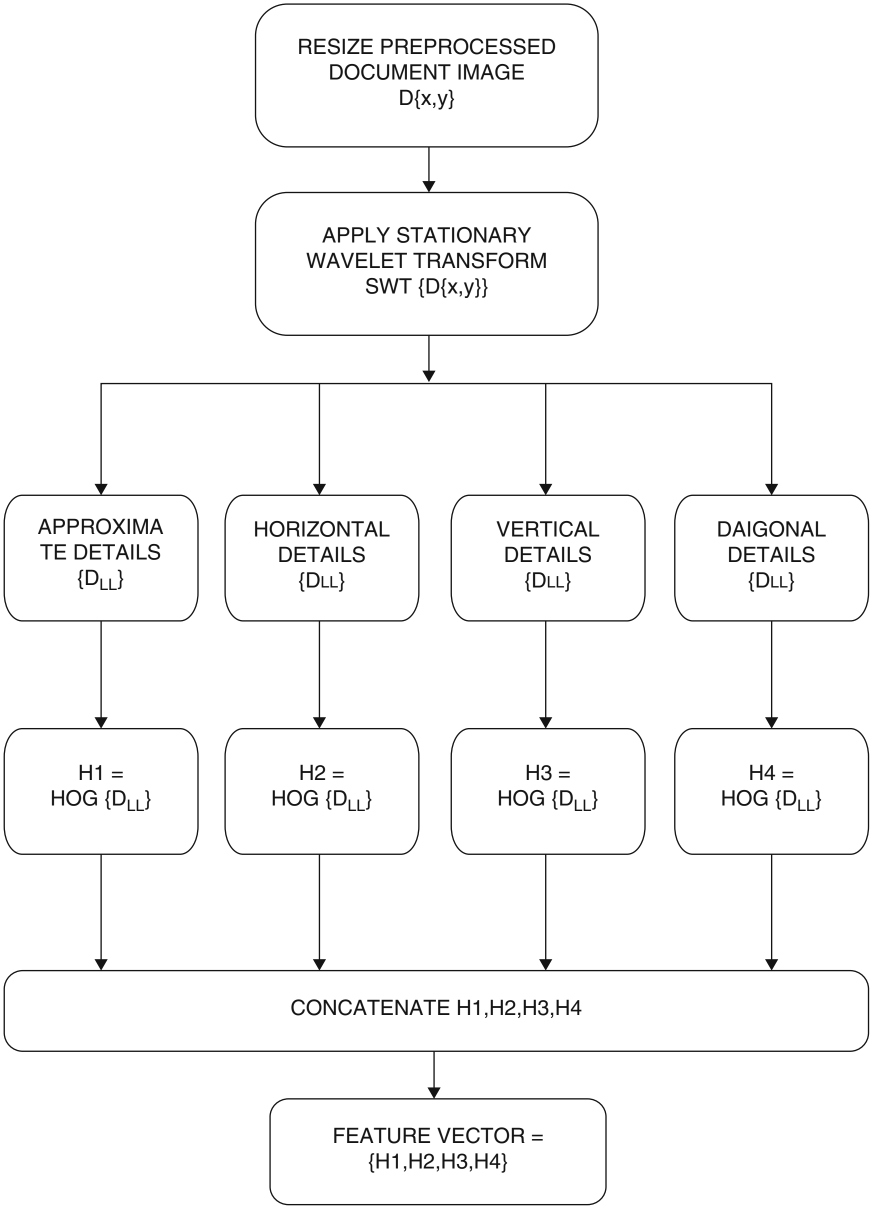


Fig. 9.7 Proposed feature extraction scheme

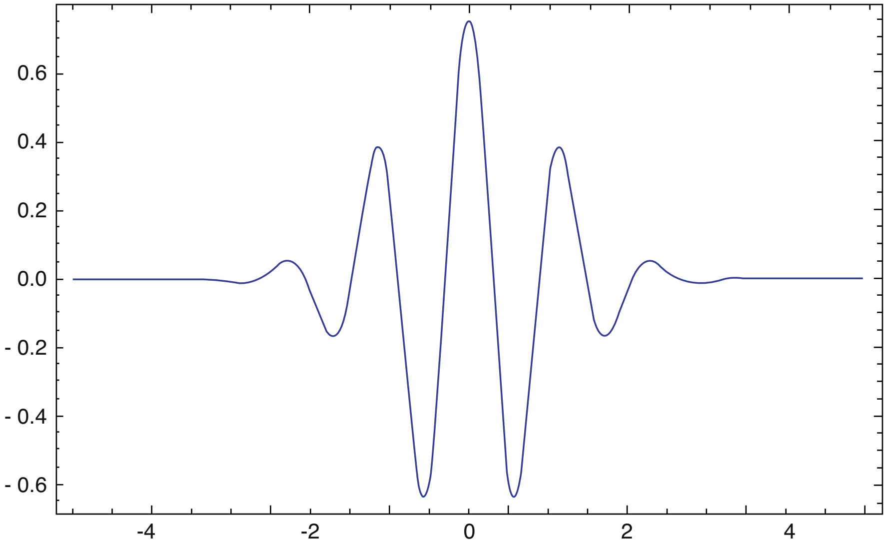


Fig. 9.8 Plot of a mother wavelet

The discrete wavelet transform provides discrete samples of a wavelet transform. DWT of a 2D signal $f(x,y)$ is given by equations (9.3) and (9.4):

$$W_{\varnothing}\,(j_O, m, n) = \frac{1}{\sqrt{M \times N}} \sum\nolimits_{x=0}^{M-1} \sum\nolimits_{y=0}^{N-1} f\,(x,\, y)\,\varnothing_{j\mathrm{o}}, m, n\,(x,\, y) \tag{9.3}$$

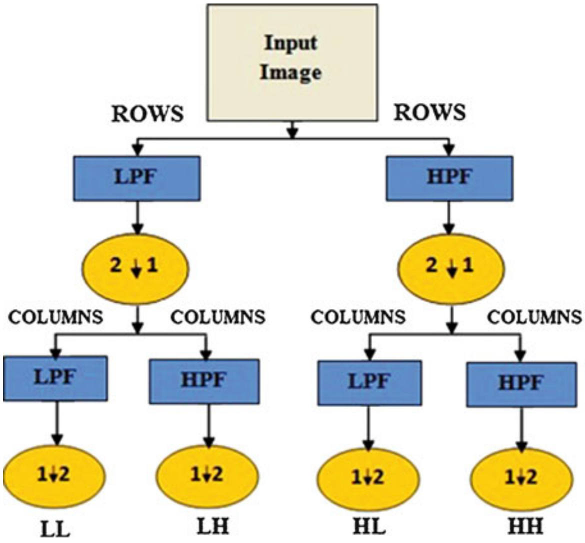
$$W_{\psi}^{\mathrm{i}}(j_O, m, n) = \frac{1}{\sqrt{M \times N}} \sum\nolimits_{x=0}^{M-1} \sum\nolimits_{y=0}^{N-1} f(x,y) \psi^{\mathrm{i}}{}_{\mathrm{jo}}, m, n(x,y) \tag{9.4}$$

where:

- j_{o} is arbitrary starting scale.
- $W_{\varnothing}(j_O, m, n)$ is an approximation of $f(x,y)$ with scale j_{o}.
- $W_{\psi}^{\mathrm{i}}(\mathrm{j}, m, n)$ represents horizontal, vertical, as well as diagonal details with scale $j \geq j_{\mathrm{o}}$.
- M and N are row and column dimensions of the input image.

Figure 9.9 shows the conceptual approach of obtaining two-dimensional DWT, which is obtained using the series of low-pass and high-pass filters [20, 21]. The notations “H” and “G” correspond to low-pass and high-pass filters, obtained by convolving the image with moving average and moving difference masks. The $2\downarrow 1$ and $1\downarrow 2$ indicate down-sampling of columns and rows, respectively. The result of this operation leads to four sub-bands:

Fig. 9.9 DWT decomposition of an image



- LL – Approximate sub-band that contains down-sampled original image
- LH – Horizontal details of an input image
- HL – Vertical details of an input image
- HH – Diagonal details of an input image

The DWT can also be applied at multilevels to obtain more accurate features. Figure 9.10 shows the result of applying 2D DWT on an image with approximate, horizontal, vertical, and diagonal details.

The two-dimensional DWT is found to be useful in many image processing applications, which include:

- Image compression
- Image denoising
- Steganography
- Feature extraction
- Turbulence analysis
- Topographic data analysis
- Financial analysis and many more

Stationary Wavelet Transform (SWT) The discrete wavelet transform lacks with translation invariance property. The SWT is designed to provide translation invariance features and is an improved version of DWT. Translation invariance is achieved by eliminating up-sampling and down-sampling process, followed by up-sampling the filter coefficients with 2^{J-1} factor in J^{th} level of discrete wavelet transform (DWT) algorithm [22–27]. Figure 9.11 shows decomposition of the image using SWT. It can be observed that the up-sampling and down-sampling are removed from the process. The LL, LH, HL, and HH are decomposed components of the input image. In SWT, the number of output samples in every level of decomposition is same as that of input samples.

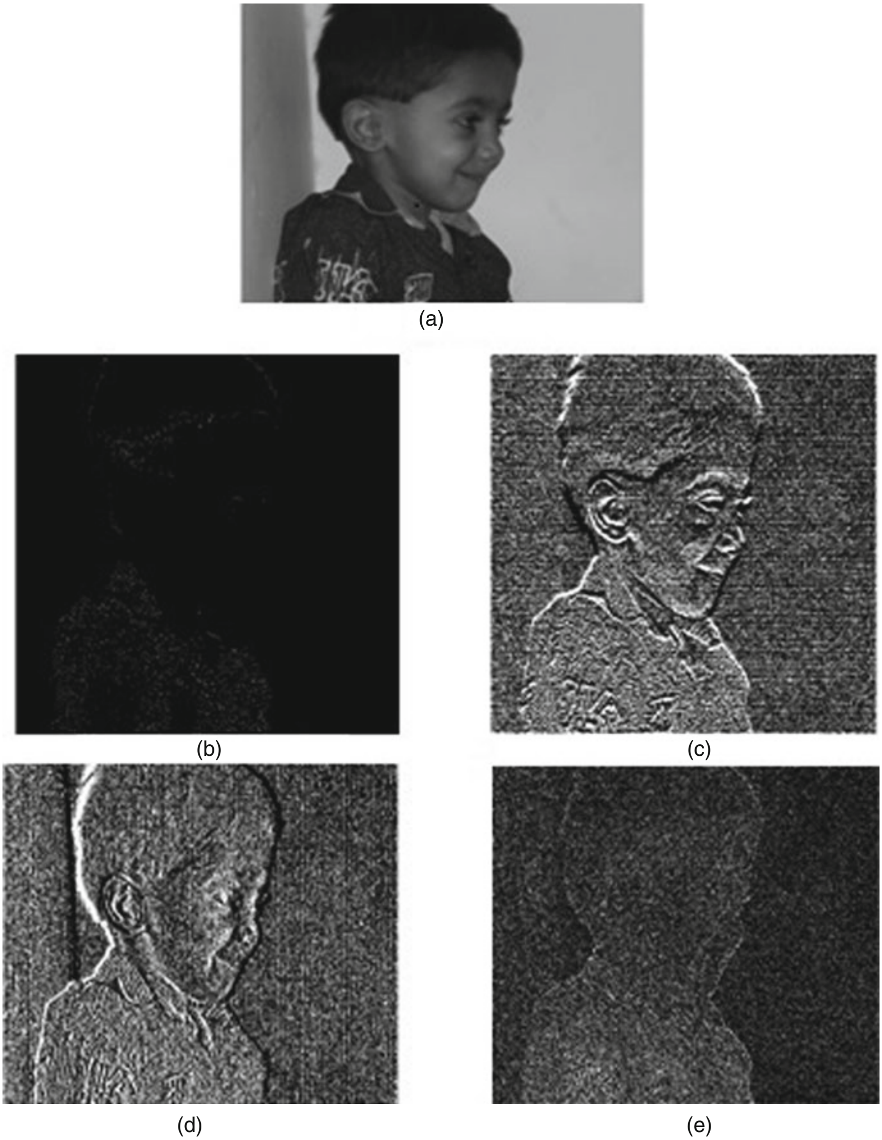


Fig. 9.10 DWT decomposition of an image

In the proposed method, the preprocessed document image is resized to 256 × 256 pixels. Let $D(x,y)$ be a resized document image. Applying SWT on $D(x,y)$ produces four sub-bands of image, namely, D_{LL}, D_{LH}, D_{HL}, and D_{HH}, as given by equation (9.5):

$$\mathrm{SWT}\left\{D\left(x,\,y\right)\right\} = \left\{D_{\mathrm{LL}},\,D_{\mathrm{LH}},\,D_{\mathrm{HL}},\,D_{\mathrm{HH}}\right\} \tag{9.5}$$

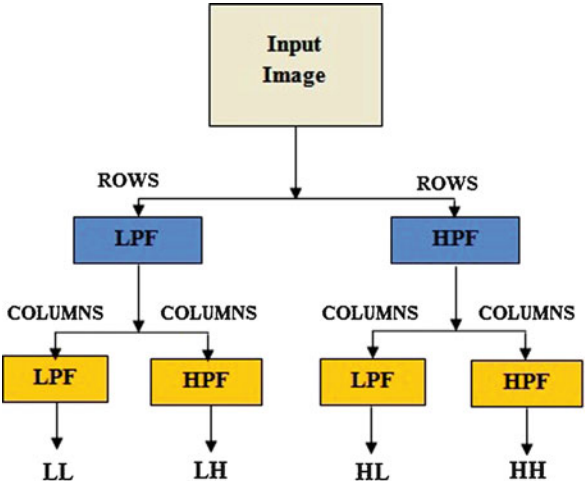


Fig. 9.11 SWT decomposition of the image

where D_{LL} contains approximation coefficients, D_{LH} includes horizontal coefficients, D_{HL} consists of vertical coefficients, and D_{HH} contains diagonal coefficients. Thus the four sub-bands provide multi-resolution version of the input document image with translation invariance features. This decomposition helps in acquiring more precise features in the next step.

9.3.2.2 Histogram of Oriented Gradients (HOG)

HOG was initially proposed by Dalal and Triggs [28] for human face detection. Later HOG and its variants are used for hand detection [29], pedestrian detection [30], fast face recognition, [31] and in many more image recognition applications.

In the proposed work, the four decomposed versions of the image obtained using SWT are divided into blocks of size 2 × 2 cells. This gives a total of 16 cells per image. Deciding size of the cell is an important step during extraction of HOG features. We tested the system with cells of size 128 × 128 and also 64 × 64 pixels. Cells of size 128 × 128 yielded a feature vector of dimension 144 and cells of size 64 × 64 yielded 1296 features. In our experiments we found that 128 × 128 cells with 144 features provided better results in comparison with 64 × 64 cell size.

After dividing the blocks into cells, the gradient and orientation of the pixels in each cell is computed. The gradient of the pixels is first order derivative and it gives finer details of the image. The gradient of a 2D function $f(x,y)$ is a column vector represented using equation (9.6):

$$\begin{bmatrix} \nabla x \\ \nabla y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} H_G(x, y) \\ V_G(x, y) \end{bmatrix} \tag{9.6}$$

where:

- ∇x is gradient along horizontal direction and is represented as $H_G(x,y)$.
- ∇y is gradient along vertical direction and is represented as $V_G(x,y)$.

However, in image processing the equations (9.7) and (9.8) are used to compute the gradient along horizontal and vertical directions of pixels in each cell. The gradient in horizontal direction $H_G(x,y)$ is difference of two successive pixels of a row and the gradient in vertical direction $V_G(x,y)$ is difference of two successive pixels of a column:

$$H_G\,(x, y) = D_i\,(x + 1, y) - D_i\,(x - 1, y) \tag{9.7}$$

$$V_G\,(x, y) = D_i\,(x, y + 1) - D_i\,(x, y - 1) \tag{9.8}$$

The magnitude Mag(x,y) and the direction of gradients $\Theta(x,y)$ are obtained using equations (9.9) and (9.10). The magnitude represents strength of the edge point and the direction gives orientation of the pixel at location (x,y):

$$\mathrm{Mag}\,(x, y) = \sqrt{H_G(x, y)^2 + V_G(x, y)^2} \tag{9.9}$$

$$\Theta\,(x, y) = \tan^{-1}\frac{G_H\,(x, y)}{G_V\,(x, y)} \tag{9.10}$$

In the next step, the orientation of gradient value of the pixels from each cell is represented as a histogram. Histogram is a discrete function that provides information about number of occurrences of a specific data. The histograms obtained for each sub-band of the image are concatenated to form final set of features. Let H_1, H_2, H_3, and H_4 be the histograms of sub-bands D_{LL}, D_{LH}, D_{HL}, and D_{HH}, respectively. These four histograms are concatenated to form a final set of features for language-based classification. Let FV be a final feature vector obtained using equation (9.11):

$$\mathrm{FV} = \{H\,(D_{LL})\;\;U\;\;H\,(D_{LH})\,U\;\;H\,(D_{HL})\,U\;\;H\,(D_{HH})\} \tag{9.11}$$

The range of values used in a histogram is called bins. We used 9 bins to store frequency of gradient values for each cell. As each sub-band comprises of four cells, we get $9 \times 4 = 36$ features per sub-band image. Thus a total of $36 \times 4 = 144$ features per document image are obtained for classification.

Figure 9.12 shows SWT decomposition of the sample document image with horizontal, vertical, and diagonal coefficients. Figure 9.13 depicts the plot of proposed feature values in graphical form. Algorithm 9.2 enlists the steps adopted in proposed feature extraction scheme.

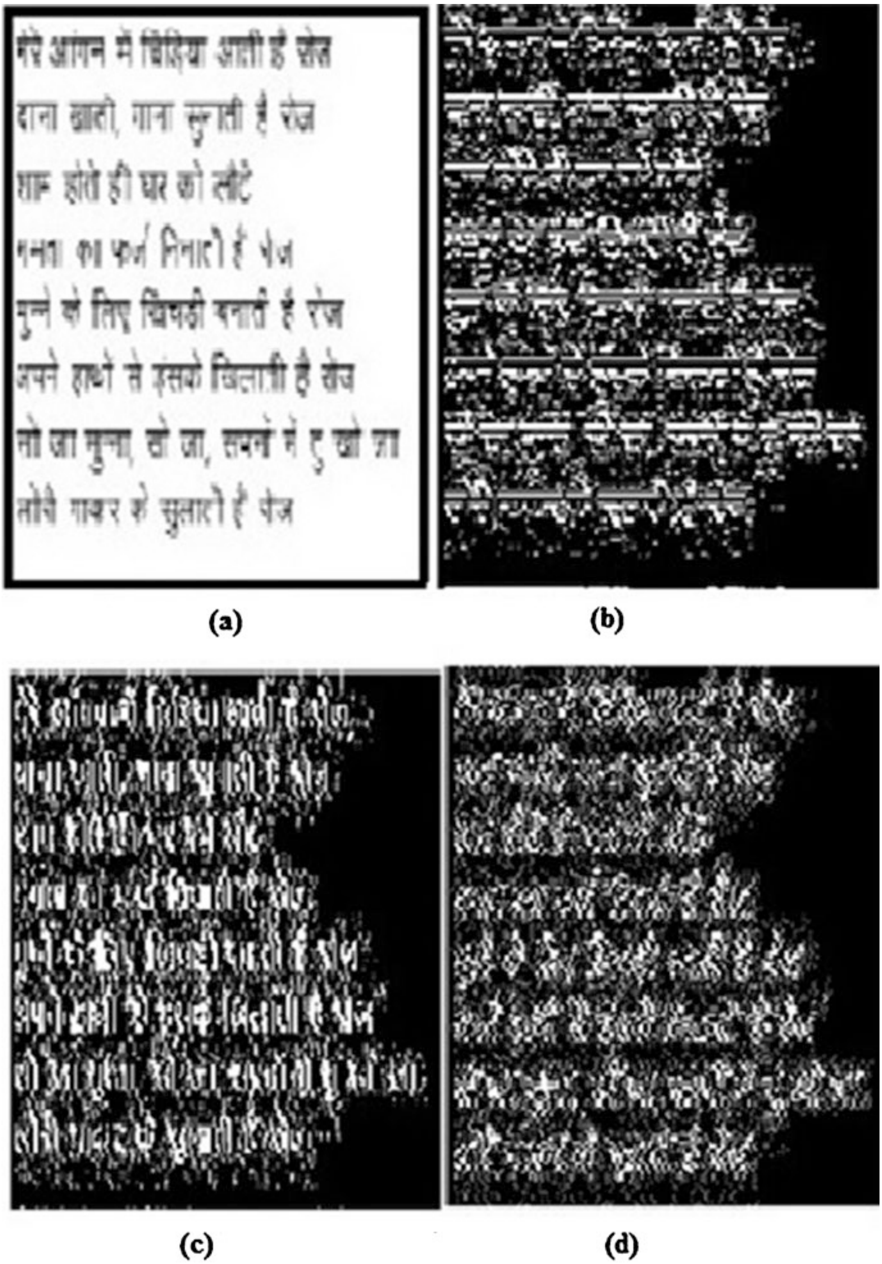


Fig. 9.12 SWT decomposition of sample document

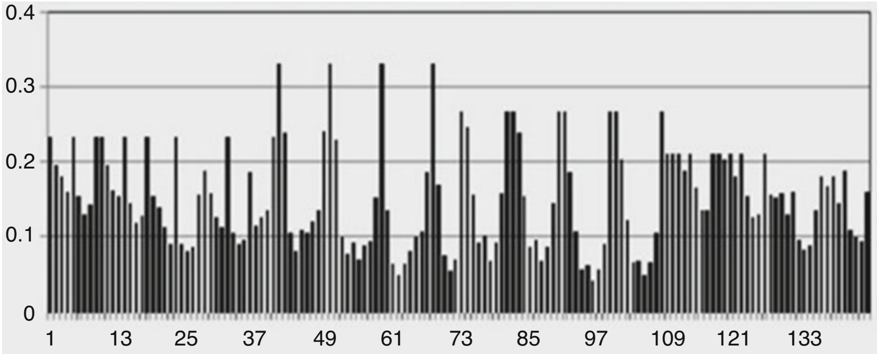


Fig. 9.13 Plot of feature values for sample document image

Algorithm 9.2

Proposed feature extraction scheme

1. **Begin**
   Input: Pre-processed document image D(x,y)
   Output: Feature vector (FV)
2. Resize the document image D(x,y) to 256×256 pixels.
3. Apply stationary wavelet transform on the image D(x,y).
   $[D_{LL}, D_{LH}, D_{HL}, D_{HH}] = SWT\{D(x,y)\}$
4. Extract HOG features from D_{LL}, D_{LH}, D_{HL} and D_{HH}.
   (a) $H_1 = HOG(D_{LL})$
   (b) $H_2 = HOG(D_{HL})$
   (c) $H_3 = HOG(D_{LH})$
   (d) $H_4 = HOG(D_{HH})$
5. Concatenate H1, H2, H3 and H4 to construct feature vector.
6. $FV = \{ H_1 \cup H_2 \cup H_3 \cup H_4 \}$
7. **End**

9.3.3 SVM Classifier

SVM belongs to the supervised machine learning technique and is widely found in the application of image classification [32, 33]. The advantages of SVM are:

- It is effective in high-dimensional space.
- It provides better results with less number of test samples.
- It is more versatile, due to a large number of kernel functions.

As SVM is a supervised learning, it requires training with some known data. Classification of the test data in SVM is obtained using an optimal hyperplane. This line separates the data into two classes. An example of separating circles and squares with a hyperplane is shown in Fig. 9.14.

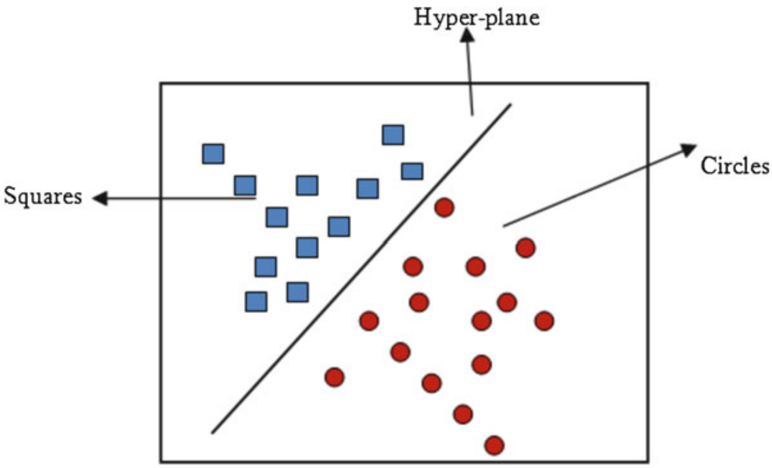


Fig. 9.14 Classification concept using SVM

The equation of hyperplane for classification of data using SVM is given by (9.12):

$$Y = W^{\mathrm{T}}\varnothing(x) + b \tag{9.12}$$

where "W" is the normal vector of hyperplane and "b" is the offset vector. The SVM employs different kernel functions such as polynomial kernel, linear kernel, sigmoid kernel, and Gaussian kernel functions. Thus usage of SVM in image processing includes the following approach:

- Computation of features from known images.
- Train SVM.
- Obtain features from test image.
- Classify using trained SVM.

This work employed linear kernel function for classification. Initially we train the SVM using features obtained from 30% of the total document images of each language. The trained model is then used for testing the samples.

9.4 Experimental Results

To evaluate the proposed method, a data base of 1006 document images of Kannada, Marathi, Telugu, Hindi, and English are considered. The database is built by collecting document images from the textbook, newspapers, and Internet. These document images comprise of printed text, graphics, symbols with various sizes, and resolution.

Table 9.1 shows the details of the database. Documents belonging to each language are considered as different classes. The database has 197 Kannada, 184 Marathi, 198 Telugu, 216 Hindi, and 211 English document images. Figure 9.15 shows the sample document images of the database used for classification.

Table 9.1 Details of the database

Sl. No.	Language	Documents
1	Kannada	197
2	Marathi	184
3	Telugu	198
4	Hindi	216
5	English	211
Total number of document images		**1006**

(a)

Fig. 9.15 Sample document images from the database. (**a**) Kannada document images. (**b**) Telugu document images. (**c**) Marathi document images. (**d**) Hindi document images. (**e**) English document images

(b)

Fig. 9.15 (continued)

The SVM classifier needs to be trained with known set of features before it is used for classification. In the proposed algorithm, from each class 30% of documents are employed to train the SVM model and the remaining are used for testing. The detection rate is used as an evaluation parameter to compare classification performance of each class and it is by equation (9.13):

$$\text{Detection rate} = \frac{\text{Number of document correctly classified}}{\text{Total number of documents}} \tag{9.13}$$

We also used average detection rate to compare overall performance of the methods, which is given by equation (9.14). It is an average of detection rate obtained for all the classes of documents:

$$\text{Average detection rate} = \frac{1}{N}\sum \text{Detection rate} \tag{9.14}$$

(c)

Fig. 9.15 (continued)

Where "*N*" is the total number of classes considered for a domain. In the presented work, as five classes are used, the value of "*N*" is taken as five. For testing, a document image from each class is given as an input and the result of the classifier is noted down. The results of the proposed technique are compared with three feature extraction schemes: rotation invariant LBP features [34], HOG features [28], and multi-resolution HOG features obtained using DWT and HOG. Table 9.2 shows the details of features used for classification with size of the feature vector.

Table 9.3 shows comparison of the results with k-NN classifier and Table 9.4 shows results obtained using SVM. From the tabulated results, it is clear that presented method provides better results with both the classifiers in comparison

(d)

Fig. 9.15 (continued)

with existing methods. Particularly the proposed method with 128 × 128 cells provided better results with a feature vector of size 144, which is smaller feature vector compared to size of other feature vectors. It is found that the classification with SVM is much better compared with k-NN.

Figure 9.16 shows graphical comparison of the results with various feature extraction schemes using k-NN and SVM classifiers. The graphs are plotted for average detection rate versus feature extraction methods.

The observations revealed from comparison of the results are listed below:

- The proposed feature extraction scheme with cell size of 128 × 128 (144 features) performs better compared with existing state-of-art techniques.
- The SVM provides good classification accuracy compared with k-NN classifier irrespective of feature extraction schemes for this particular application.

(e)

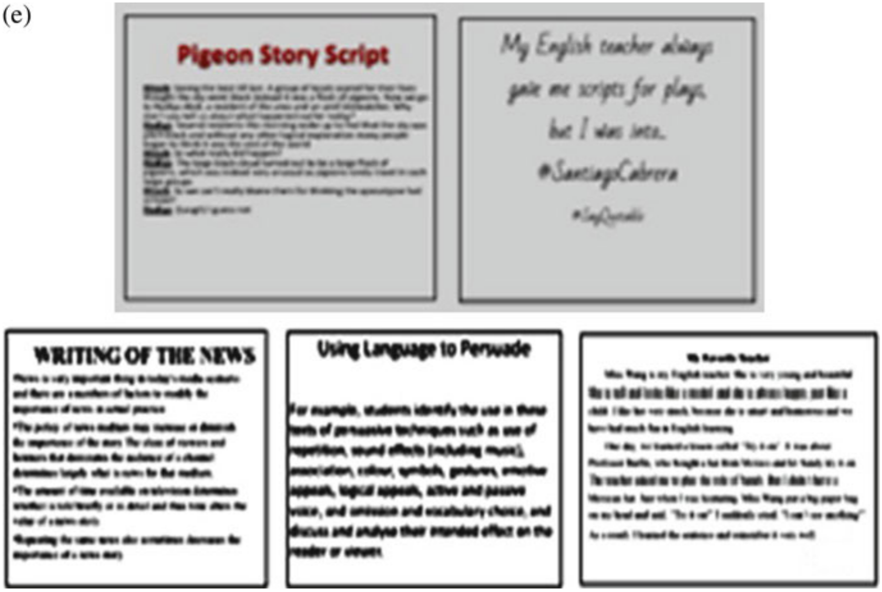


Fig. 9.15 (continued)

Table 9.2 Details of feature extraction schemes

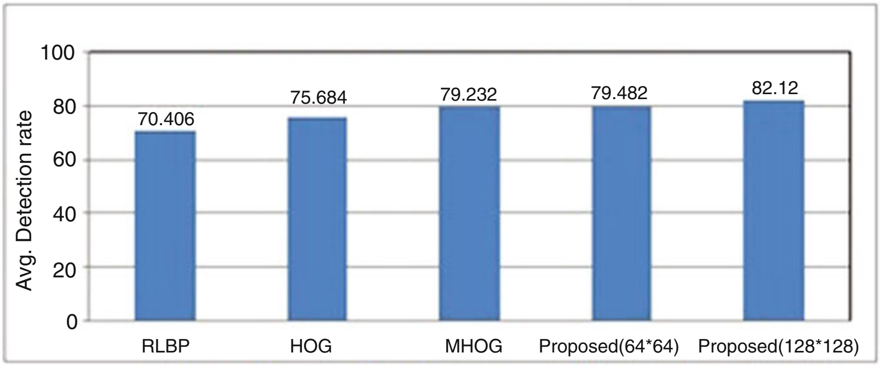
Sl. No.	Feature extraction method	Size of feature vector
1	Rotation invariant LBP	640
2	HOG features	324
3	Multi-resolution HOG features (DWT + HOG)	256
3	Proposed features with 64 × 64 cell size	1296
4	Proposed features with 128 × 128 cell size	144

Table 9.3 Comparison of results with k-NN classifier

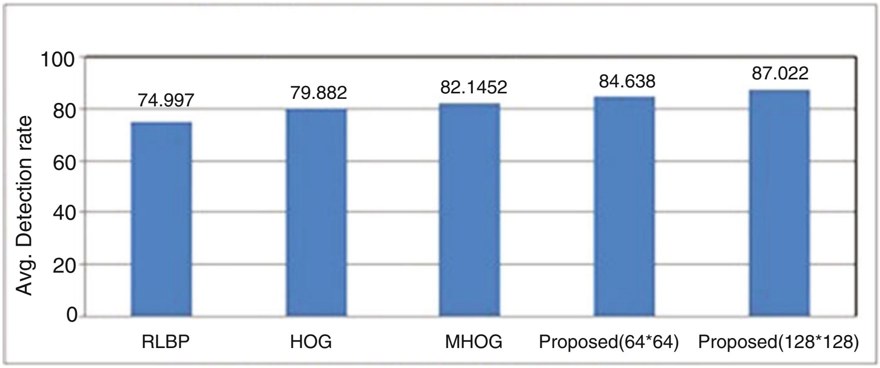
Sl. No.	Language	Detection rate (%)				
		RLBP	HOG	Multi-resolution HOG features	Proposed features with 64 × 64 cell	Proposed features with 128 × 128 cell
1	Kannada	50	57.29	63.25	59.29	61.45
2	Marathi	56.31	77.49	81.43	68.93	81.55
3	Telugu	62.5	66.2	75.73	87.5	79.12
4	Hindi	98.4	94.11	97.97	99.49	100
5	English	84.82	83.33	77.78	82.2	88.48
Avg. detection rate		**70.406**	**75.684**	**79.232**	**79.482**	**82.12**

Table 9.4 Comparison of results with SVM classifier

Sl. No.	Language	Detection rate (%)				
		RLBP	HOG	Multi-resolution HOG features	Proposed method with 64*64 cell	Proposed method with 128*128 cell
1	Kannada	55.2	65.23	67.7	76.04	75
2	Marathi	62.135	73.28	73.78	85.44	87.38
3	Telugu	72.22	86.11	91.66	80.56	80.58
4	Hindi	98.99	95.23	97.46	100	100
5	English	86.44	79.56	80.126	81.15	92.15
Avg. detection rate		**74.997**	**79.882**	**82.1452**	**84.638**	**87.022**



(a)



(b)

Fig. 9.16 Graphical comparison of results

9.5 Conclusion

This work proposed an efficient method for language-based classification of document images using SWT- and HOG-based hybrid texture features. It employs segmentation-free technique for recognition of documents' language. Proposed features are tested using K-NN and SVM classifiers on a database of 1006 document images. These features with SVM classifier provided an average detection rate of 87.02% for five different classes of document images comprising of Kannada, Marathi, Telugu, Hindi, and English language. Proposed feature extraction scheme can be tested on different scripts and also can be used in classification of other images with suitable preprocessing techniques.

References

1. G. Nagy, Twenty years of document image analysis in PAMI. IEEE Trans. Pattern Anal. Mach. Intell. **22**(1), 38–62 (2000)
2. S. Chaudhury, G. Harit, S. Madnani, R.B. Shet, Identification of scripts of Indian languages by combining trainable classifiers, in *ICVGIP 2000* (2000), pp. 20–22
3. A. Kulkarni, P. Upparamani, R. Kadkol, P. Tergundi, Script identification from multilingual text documents. Int. J. Adv. Res. Comput. Commun. Eng. **4**(6), 15–19 (2015)
4. M.C. Padma, P.A. Vijaya, Script identification from trilingual documents using profile based features. Int. J. Comput. Sci. Appl. **7**(4), 16–33 (2010)
5. U. Pal, B.B. Chaudhuri, Automatic identification of english, chinese, arabic, devnagari and bangla script line, in *Proceedings of Sixth International Conference on Document Analysis and Recognition* (2001), pp. 790–794
6. G.G. Rajput, H.B. Anita, Handwritten script recognition using DCT and wavelet features at block level. Int. J. Comput. Appl., Special issue on RTIPPR (3), 158–163 (2010)
7. S.M. Obaidullah, A. Mondal, N. Das, K. Roy, Script identification from printed Indian document images and performance evaluation using different classifiers. Appl. Comput. Intell. Soft Comput. **2014**, 1–12 (2014)
8. M.S. Shirdhonkar, M.B. Kokare (2010). Discrimination between printed and handwritten text in documents, in *IJCA Special Issue on Recent Trends in Image Processing and Pattern Recognition*, pp. 131–134
9. R. Pardeshi, B.B. Chaudhuri, M. Hangarge, K.C. Santosh, Automatic handwritten Indian scripts identification, in *IEEE 14th International Conference on Frontiers in Handwriting Recognition* (September 2014), pp. 375–380
10. C.L. Tan, W. Huang, S.Y. Sung, Z. Yu, Y. Xu, Text retrieval from document images based on word shape analysis. Appl. Intell. **18**(3), 257–270 (2003)
11. A.S. Wanchoo, P. Yadav, A. Anuse, A survey on Devanagari character recognition for Indian postal system automation. Int. J. Appl. Eng. Res. **11**(6), 4529–4536 (2016)
12. P. Sahare, S.B. Dhok, Script identification algorithms: A survey. Int. J. Multimed. Inf. Retr. **6**(3), 211–232 (2017)
13. U.D. Dixit, M.S. Shirdhonkar, A survey on document image analysis and retrieval system. Int. J. Cybern. Informat. **4**(2), 259–270 (2015)
14. S.A.A.A. Arani, E. Kabir, R. Ebrahimpour, Handwritten Farsi word recognition using NN-based fusion of HMM classifiers with different types of features. Int. J. Image Graph. **19**(1), 1–21 (2019)

15. N. Bi, J. Chen, J. Tan, The handwritten Chinese character recognition uses convolutional neural networks with the GoogLeNet. *Intern. J. Pattern Recognit. Artif. Intell.* **33**(11), 1–12 (2019)
16. C. Djeddi, I. Siddiqi, L. Souici-Meslati, A. Ennaji, Text-independent writer recognition using multi-script handwritten texts. Pattern Recognit. Lett. **34**(10), 1196–1202 (2013)
17. U.D. Dixit, M.S. Shirdhonkar, Fingerprint-based document image retrieval. Int. J. Image Graph. **19**(2), 1–17 (2019)
18. P.P. Roy, A.K. Bhunia, A. Das, P. Dey, U. Pal, HMM-based Indic handwritten word recognition using zone segmentation. Pattern Recognit. **60**, 1057–1075 (2016)
19. U.D. Dixit, M.S. Shirdhonkar, Preprocessing framework for document image analysis. Int. J. Adv. Netw. Appl. **10**(4), 3911–3918 (2019)
20. A. Bultheel, Learning to swim in a sea of wavelets. Bull. Belg. Math. Soc. Simon Stevin **2**(1), 1–45 (1995)
21. S.G. Chang, B. Yu, M. Vetterli, Adaptive wavelet thresholding for image denoising and compression. IEEE Trans. Image Process. **9**(9), 1532–1546 (2000)
22. A.N. Akansu, Y. Liu, On-signal decomposition techniques. Opt. Eng. **30**(7), 912–921 (1991)
23. M.J. Shensa, The discrete wavelet transform: wedding the a trous and Mallat algorithms. IEEE Trans. Signal Process. **40**(10), 2464–2482 (1992)
24. M.V. Tazebay, A.N. Akansu, Progressive optimality in hierarchical filter banks, in *Proceedings of 1st International Conference on Image Processing,* vol. 1 (Nov 1994), pp. 825–829
25. M.V. Tazebay, A.N. Akansu, Adaptive subband transforms in time-frequency excisers for DSSS communications systems. IEEE Trans. Signal Process. **43**(11), 2776–2782 (1995)
26. M. Holschneider, R. Kronland-Martinet, J. Morlet, P. Tchamitchian, A real-time algorithm for signal analysis with the help of the wavelet transform, in *Wavelets*, (Springer, Berlin, Heidelberg, 1990), pp. 286–297
27. Y. Zhang, S. Wang, Y. Huo, L. Wu, A. Liu, Feature extraction of brain MRI by stationary wavelet transform and its applications. J. Biol. Syst. **18**, 115–132 (2010)
28. N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in *International Conference on Computer Vision & Pattern Recognition (CVPR'05)*, vol. 1 (2005), pp. 886–893
29. Y. Zhao, Z. Song, X. Wu, Hand detection using multi-resolution HOG features, in *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (Dec 2012), pp. 1715–1720
30. Y. Zhao, Y. Zhang, R. Cheng, D. Wei, G. Li, An enhanced histogram of oriented gradients for pedestrian detection. IEEE Intell. Transp. Syst. Mag. **7**(3), 29–38 (2015)
31. X.Y. Li, Z.X. Lin, Face recognition based on HOG and fast PCA algorithm, in *The Euro-China Conference on Intelligent Data Analysis and Applications*, (Springer, Spain, 2017), pp. 10–21
32. J. Pan, Y. Zhuang, S. Fong, The impact of data normalization on stock market prediction: using SVM and technical indicators, in *International Conference on Soft Computing in Data Science*, (Springer, Malaysia, 2016), pp. 72–88
33. V. Vapnik, *The Nature of Statistical Learning Theory* (Springer Science & Business Media, 2013)
34. Mäenpaa Topi, Matti Pietikäinen. (2005) Texture analysis with local binary patterns, Handbook of Pattern Recognition and Computer Vision*,* (pp. 197–216), Singapore: World Scientific Publishing