Multilinguals Script Identification using LBP

Naveen Kumar K.P.[1] Mahesha D.M ,[2] Bhavya D.N,[3] Sharath Kumar Y. H [4]

[1,2,3] Department of Studies and Research in Computer Science, Karnataka State Open University, Mysore-570006, India.

[4] Maharaja Institute of Technology Mysore, Department of ISE, MIT Mysore, India.

Abstract

In the context of our country, script recognition is a complex task because of more number of prevailing scripts and that many of the official documents are multiscript in nature, English is being commonly used along with regional script. Hence, it is pre-requisite to identify the script in the document and then feed the document to respective OCR for further processing.

1. Introduction

Script identification refers to the automatic identification of the language or cultural background of a text through computer technology. It emerged to deal with linguistic diversity and cultural differences in the process of globalization and to provide basic support for text translation, information extraction, and cultural exchange. With the acceleration of globalization and the development of Internet technology, more and more information and data involve multiple languages and cultural backgrounds. However, manual processing and management of this cross-language and cultural information can no longer meet the needs. Computer scientists have been studying how to use computer technology for script identification, improve the efficiency and accuracy of information processing, and better understand and manage cross-language and cultural information. Script identification has a wide range of practical applications, such as online archiving of multilingual scene images, product image search [1], multilingual machine translation [2], scene image understanding [3], etc. Script identification technology plays a vital role in the Optical Character Recognition (OCR) recognition process [4]. Its greatest significance is that it can help to identify the corresponding language, and further select the best text recognition [5,6,7] model to realize multilingual automatic processing. Script identification can help people better process text information involving multiple languages and improve the efficiency and accuracy of information processing. Script identification in natural scenes refers to the identification of different language types such as the language or dialect to which the text belongs in the real world, so as to perform related processing and analysis. Different script categories include

Arabic, Hindi, Bengali, and Telugu, among others. As shown in **Figure 1**, different scripting languages have different character structures. Script identification encounters significant challenges when dealing with scene text images, which differ significantly from document print [8] images for the following main reasons:

Figure 1. Sample dataset example.

2. Related Works

The script identification process mainly includes preprocessing, feature extraction, and classification. For machine learning methods, the preprocessing part is also very important, but for deep learning methods, some preprocessing steps can be directly omitted. In recent years, script identification methods have developed rapidly in both machine learning and deep learning. Traditional manual feature extraction methods mainly include Local Binary Pattern (LBP) [9], Histogram of Oriented Gradients (HOG) [10], and Gray Level Co-occurrence Matrix (GLCM) [11], etc. These methods mainly extract the texture, color and other information of the text area for classification and identification. Although these methods have certain effects, because their feature extraction process is based on human experience, they cannot fully and effectively express the feature information of the text, so there are certain limitations in practical applications. Deep learning methods mainly include convolutional neural network (CNN) [12], recurrent neural network (RNN) [13], attention mechanism [14], etc. These methods mainly learn the features of text directly from raw data by end-to-end training on text images, so they have a good generalization ability. In recent years, deep learning methods have been widely used in the field of script identification in natural scenes and achieved good results. Technical research on script identification began in 1990, and there are many traditional machine learning methods. At the end of the 20th century, Spitz et al. [15] proposed for the first time to divide text images into two categories according to Chinese and Latin and successfully identified three languages, Chinese, Japanese, and Korean, by analyzing the optical density distribution. In 2010, Padma [16] used the gray level co-occurrence matrix of wavelet transform coefficient subbands to extract Haralick texture features and used them for language identification of machine-printed text. Ferrer et al. [17] used LBP texture features to describe the stroke direction distribution of text characters in 2013 and used them as features of text images to classify them through SVM linear classifiers. In the same year, Rani et al. [18] used directional frequency-based Gabor features and gradient features based on individual character gradient information to identify multi-font and multi-size characters and used multi-class support vector machines for classification. Shivakumara et al. [19] used gradient angle segmentation to segment text words in document images in 2014 and generated candidate features for language identification. Mukarambi et al. [20] proposed a method based on LBP features in 2017 for language identification of Kannada, Indian, and English text images captured by cameras, and the method showed good identification results. The experimental results of these methods show that script identification technology has made remarkable progress. The application of deep learning methods in language identification is developing rapidly. Gomez et al. [21] used methods such as convolutional features and naive Bayesian classifiers to learn the stroke features of text, paying special attention to the discriminative features of small strokes in the fine-grained framework. Its method achieves state-of-the-art results in the task of text-image-genre identification in natural scenes. At the same time, Mei et al. [22] combined the convolutional neural network and the recurrent neural network by means of feature sequence labeling, and designed a network model that can be trained end-to-end. It improves the semantic information and long-term spatial dependence of extracted text image features, making the model perform well in language identification tasks. Zdenek et al. [23] developed a new method that combines local convolutional triplet features and a visual bag-of-words model, enabling the model to obtain a more descriptive vocabulary and enhance the low-resolution ability of weak feature descriptors. It has shown competitive performance in script identification. Ankan et al. [24] proposed a convolutional recurrent neural network based on the attention mechanism, which fuses the global features and local features of text images through dynamic weighting, and demonstrates the performance of the model in scene text and video text images. Meanwhile, Cheng et al. [25] developed a local patch aggregator-based convolutional neural network model that considers the prediction scores of local patches to learn more discriminative feature information. This makes the model play an important role in the classification of similar texts and has achieved excellent performance. Shi Baoguang et al. [26] introduced the discriminative clustering algorithm into the convolutional neural network, designed the DiscCNN network model, and effectively captured the detailed differences in various languages. The multivariate time series classification method proposed by Karim et al. [27] is composed of Long Short-Term Memory (LSTM) and Fully Convolutional Network (FCN), and the effect of adding an attention mechanism is better. Fujii et al. [28] used Encoder and Summarizer to extract local features, which were fused by an attention mechanism to reflect the importance of different patches.

3. Proposed Method

In this work, we have used LBP for feature extraction and for Classification using SVM and KNN.

3.1 Local Binary Pattern (LBP) operator

Local Binary Pattern were first described in the year 1994 in a paper "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions by Ojala et al.,[11] to characterize the spatial structure of the local image texture. It has since then been found to be a powerful feature for texture classification. Many versions of LBP have been defined in literature. The LBP feature vector, in its simplest form, is described as follows. The Local Binary Patterns operator is computed in a local circular region by taking the difference of the centre pixel with respect to its neighbors

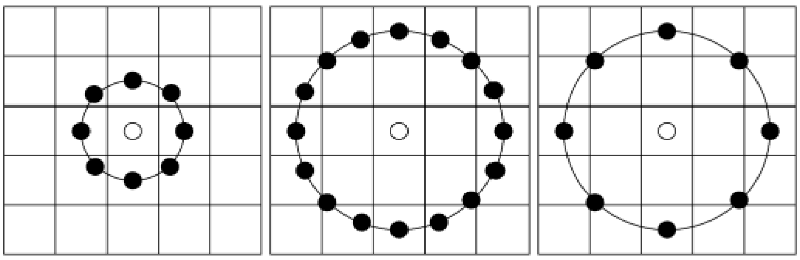

Figure 2: The Circular (8, 1), (16, 2) and (8, 2) neighborhoods. The pixel values are bilinearly interpolated whenever the sampling point is not in the center of a pixel

Now for the central pixel with coordinate (x, y), the coordinates of uniformly spaced circular neighborhood can be obtained as $(x+R\cos(2\pi p/P),\ y-R\sin(2\pi p/P))$ for $p = 0,1,2,\ldots P-1$. In case of non-integer values of the neighboring coordinate a bilinear interpolation is used for estimation of pixel value. Fig.5.1 shows some examples of the LBP operator, where the notation (P, R) denotes a neighborhood of P sampling points on a circle of radius of R. The most fundamental structure from the LBP has been extracted by using the concept of uniform pattern or a local binary pattern in which the binary code contains at most two transitions from 0 to 1 or vice versa. In uniform LBP mapping there is a separate output label for each uniform pattern and all the non-uniform patterns are assigned to a single label. Thus, the number of different output labels for mapping for patterns of P bits is $\mathrm{P}\ (\mathrm{P} - 1) + 3$. For instance, the uniform mapping for neighborhoods of 8 sampling points as shown in Fig.5.2 will produce 59 output labels. Similarly, for neighborhoods of 16 sampling points it would be 243 labels. There are mainly two reasons for omitting the non-uniform patterns. The first reason is the most of the local binary patterns in natural images are in uniformed. The second reason for considering uniform patterns is the statistical robustness. Use of uniform patterns instead of all the possible patterns have often lead to better recognition in many applications, as they are more stable and less prone to noise. Accordingly, the number of possible LBP labels are significantly lowered by considering only uniform patterns leading to the reliable estimation of their distribution with fewer samples. The original rotation invariant LBP is achieved by circularly rotating each bit pattern to the minimum value. For instance, the bit sequences 1000011, 1110000 and 0011100 arising out of different rotations of the same local pattern correspond to the normalized sequence 0000111. Texture of a script is a unique feature that can be used to identify the script type. Since, LBP operator captures statistical based texture features and robust to the change in neighborhood intensity values. LBP can be used effectively for script type identification. In this chapter, we proposed script type identification method by using LBP for feature extraction. Performing connected component analysis on handwritten document images [96], lines are extracted and features are computed using LBP operator. Block diagram of the proposed method is shown in the Fig 3.

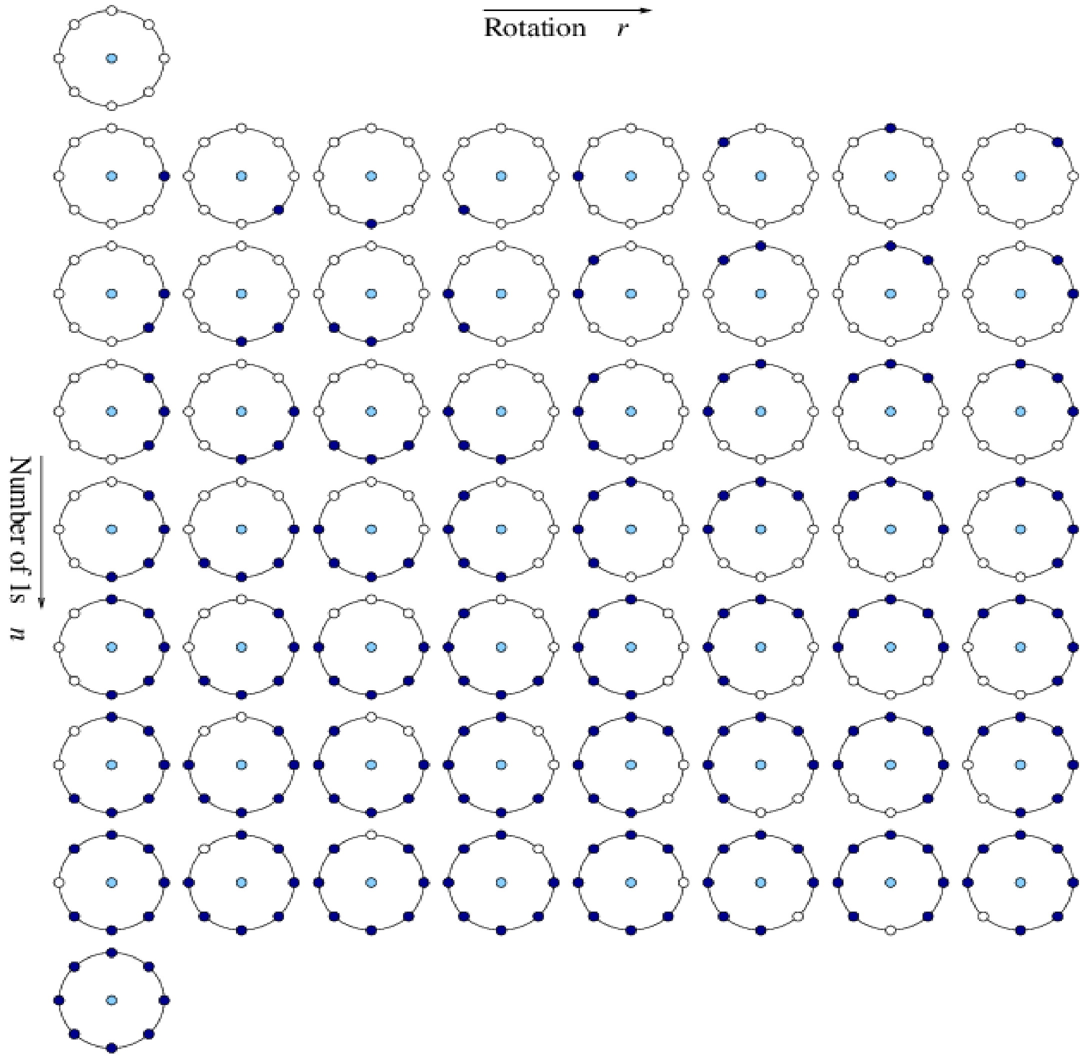

Figure 3: The 58 different uniform patterns in (8, R) neighborhood

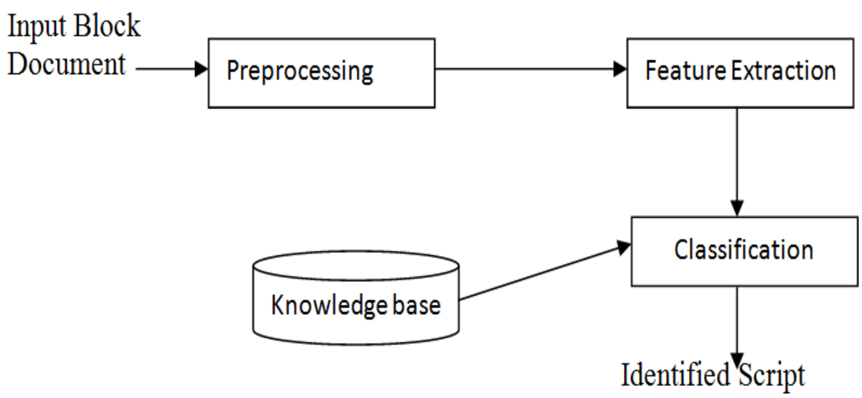

Figure 4: Block diagram of the proposed method

LBP is a gray-scale invariant texture measure, derived from a general definition of texture in a local neighborhood. LBP operator produces a binary code by thresholding a 3x3 neighborhood by the gray value of its center. The histogram of these labels can also be used as a texture descriptor. Most important property is that LBP operator is robust to monotonic gray-scale changes caused and it has the advantage of simple implementation. Many related approaches based on the original implementation have been developed for texture and color texture segmentation. The procedure for computing the binary code is as follows. In its basic form of implementation, LBP operator takes 3 x 3 neighborhood of a pixel and generates a binary 1 if the neighbor of the center pixel has the larger value than the center pixel. The operator generates a binary 0 if the neighbor is less than the center. For 8- neigborhood, an 8-digit binary number is generated which is represented as unsigned integer, making it a compact description. An example of binary code generation is given in figure 5.

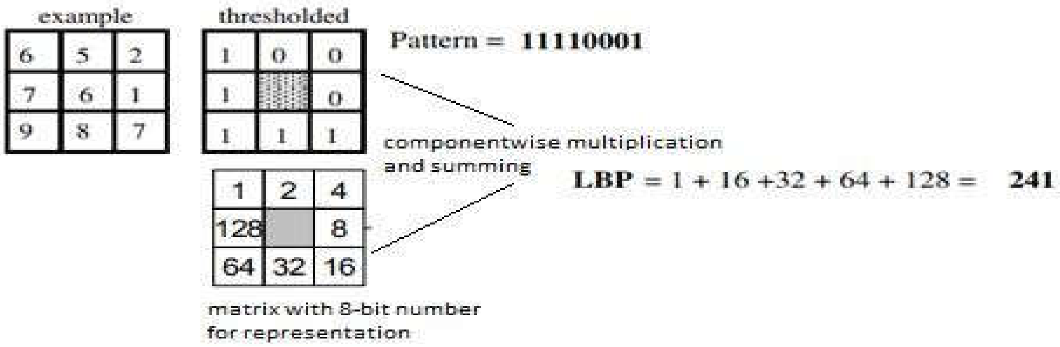

Figure 5 : Calculating the LBP code

3.2 Classification

Classification of the script type is carried out using KNN and SVM classifiers. The KNN is a lazy classifier that compares the test image feature vector with that of feature vector of all the images used for training and labels the test image to be of a specific script type using Euclidean distance measure. The SVM classifier is a supervised learning method. Given a labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples.

3.2.1 SVM Classifier

The SVM is a discriminative model classification technique that mainly relies on two assumptions. First, transforming data into a high-dimensional space may convert complex classification problems (with complex decision surfaces) into simpler problems that can use linear discriminate functions. Second, SVMs are based on using only those training patterns that are near the decision surface assuming they provide the most useful information for classification. In its basic form, SVM obtains the optimal boundary of two sets in a vector space independently on the probabilistic distributions of training vectors in the sets. Basically, Support Vector Machines (SVM) [99] is defined for two-class problem (Figure 5.5) and it finds the optimal hyper-plane which maximizes the distance, the margin, between the nearest examples of both classes, named support vectors (SVs). Given a training database of M data{ xm | m=1, . . ., M},the linear SVM classifier is defined as:

$$F(x) = \Sigma\, \alpha_j\, x_j \,.\, x + b \qquad \text{Eq. 2}$$

where {xj} are the set of support vectors and the parameters j and b have been determined by solving a quadratic problem.

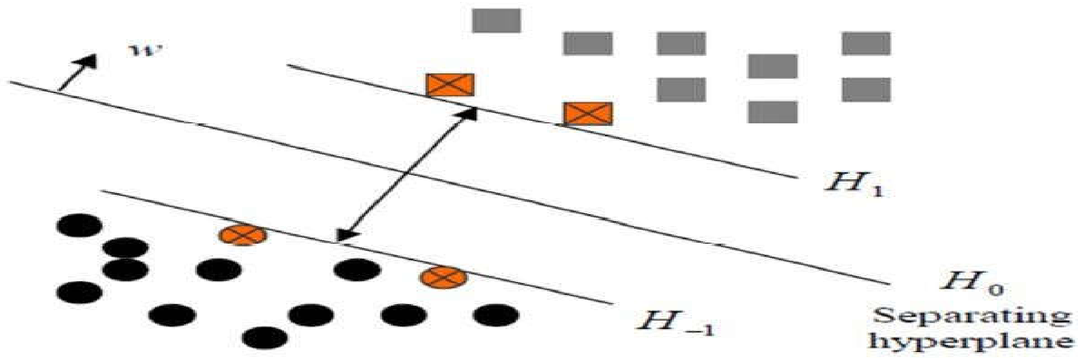

Figure 6: Two class linear classification. The support vectors are indicated with crosses

The linear SVM can be extended to a non-linear classifier by replacing the inner product between the input vector x and the SVs xj, through a kernel function K defined as

$$K(x, y) = \phi(x)\, .\, \phi\, (y) \qquad \text{Eq. 5.3}$$

This kernel function should satisfy the Mercer's Condition. The performance of SVM depends on the kernel used. A number of simple kernels include polynomial SVM, Radial Basis Function (RBF) SVM, and Two-layer neural network SVM. We have chosen RBF (Gaussian) kernel due to its better performance compared to other kernels. A SVM based classifier contracts all the information contained in the training set relevant for classification, into the support vectors. This procedure reduces the size of the training set by identifying its most important points. The SVM is applied to our multiclass script recognition problem by using one-versus-rest type method. More detailed treatment on principles and applications of

SVM can be found elsewhere [100].

3.2.2 Feature Extraction algorithm

Input: pre-processed line level binary image

Output: LBP feature vector

Divide the image into 3x3 cells and for each cell compute the compact representation of LBP operator generated binary code. Compute the histogram of each cell with combination of pixel smaller and greater than the centre pixel value.

Normalize the histogram to neighborhood pattern of (8, R), where R=1.

Feature vector of entire window is obtained by concatenating histogram of each cell having 59 uniforms pattern (out of 256).

Binary image	LBP image

FIGURE 7: Sample Binary and LBP images

3.3 Experimental Results

Experiments are carried out on scripts types viz, Kannada, Hindi, Urdu, Malayalam, Punjabi, Telugu, Tamil, Oriya and English respectively. A total of 230 handwritten documents are considered for experimentation. Preprocessing and segmentation of non-overlapping line is carried out using our earlier work described in chapter 3. Text lines with only 40% of occupancy of text are rejected by considering it as not a complete line. Sample images of text lines rejected are shown in Fig 5.7, 630 text line images are used to test the proposed method. The overall recognition accuracy obtained using K-NN classifier is 92.63% and 94.9%

accuracy is obtained using SVM classifier. Two-fold cross-validation is performed for computing the accuracy of the classification results. The details of the results obtained for bi-script documents are presented in Table 5.1. The results obtained for tri-script classification is shown in Table 5.3. SVM classifier performs better over KNN classifier.

Table 1: Bi-Script Classification Using KNN and SVM classifiers

Sl. No.	Script	KNN	Recognition in %	SVM	Recognition in %
1	Kannada	92.86	95.71	95.71	97.85
	English	98.57		100	
2	Hindi	98.57	98.57	100	99.28
	English	98.57		98.57	
3	Malayalam	92.86	93.57	95.71	90.71
	English	94.29		85.71	
4	Oriya	92.86	90	98.57	95
	English	87.14		91.43	
5	Punjabi	90	90	95.71	92.85
	English	90		90	
6	Tamil	94.29	91.43	95.71	90.71
	English	88.57		85.71	
7	Telugu	95.71	94.96	100	97.85
	English	94.21		95.71	
8	Urdu	91.43	90.71	100	93.57
	English	90		87.14	
Over all Recognition accuracy in %			93.12		94.73

Table 2: Confusion Matrix for Bi-Script Classification using SVM

Script	Kannada	English	Recognition in %
Kannada	67	3	97.85
English	0	70	

Script	Hindi	English	Recognition in %
Hindi	70	0	99.28
English	1	69	

Script	Malayalam	English	Recognition in %

Malayalam	67	3	90.71
English	10	60	

Script	Oriya	English	Recognition in %
Oriya	69	1	95
English	6	64	

Script	Punjabi	English	Recognition in %
Punjabi	67	3	92.85
English	7	63	

Script	Tamil	English	Recognition in %
Tamil	67	3	90.71
English	10	60	

Script	Telugu	English	Recognition in %
Telugu	70	0	97.85
English	3	67	

Script	Urdu	English	Recognition in %
Urdu	70	0	93.57
English	9	61	

Table 3: Tri-Script Classification Using KNN and SVM classifiers

Sl. No.	Script	KNN	Recognition in %	SVM	Recognition in %
1	Kannada	95.71	96.66	92.86	97.14
	Hindi	98.57		100	
	English	95.71		98.57	

2	Malayalam	100	92.86	100	94.76
	Hindi	84.29		94.29	
	English	94.29		90	
3	Oriya	95.71	90.47	97.14	94.29
	Hindi	88.57		94.29	
	English	87.14		91.43	
4	Tamil	98.57	87.62	100	90.01
	Hindi	91.43		88.59	
	English	72.86		81.43	
5	Telugu	97.14	93.81	100	98.57
	Hindi	91.43		100	
	English	92.86		95.71	
6	Urdu	97.14	91.43	100	95.71
	Hindi	88.57		98.57	
	English	88.57		88.57	
Over all Recognition accuracy in %			**92.14**		**95.08**

Table 4: Confusion Matrix for Tri-Script Classification using SVM

Script	Kannada	Hindi	English	Recognition in %
Kannada	65	1	4	97.14
Hindi	0	70	0	
English	0	1	69	

Script	Malayalam	Hindi	English	Recognition in %
Malayalam	70	0	0	94.76
Hindi	0	66	4	
English	1	6	63	

Script	Oriya	Hindi	English	Recognition in %
Oriya	68	0	2	94.29
Hindi	0	66	4	
English	0	6	64	

Script	**Tamil**	**Hindi**	**English**	**Recognition in %**
Tamil	70	0	0	
Hindi	0	62	8	90.01
English	5	8	57	

Script	**Telugu**	**Hindi**	**English**	**Recognition in %**
Telugu	70	0	0	
Hindi	0	70	0	98.57
English	0	3	67	

Script	**Urdu**	**Hindi**	**English**	**Recognition in %**
Urdu	70	0	0	
Hindi	0	69	1	95.71
English	1	7	62	

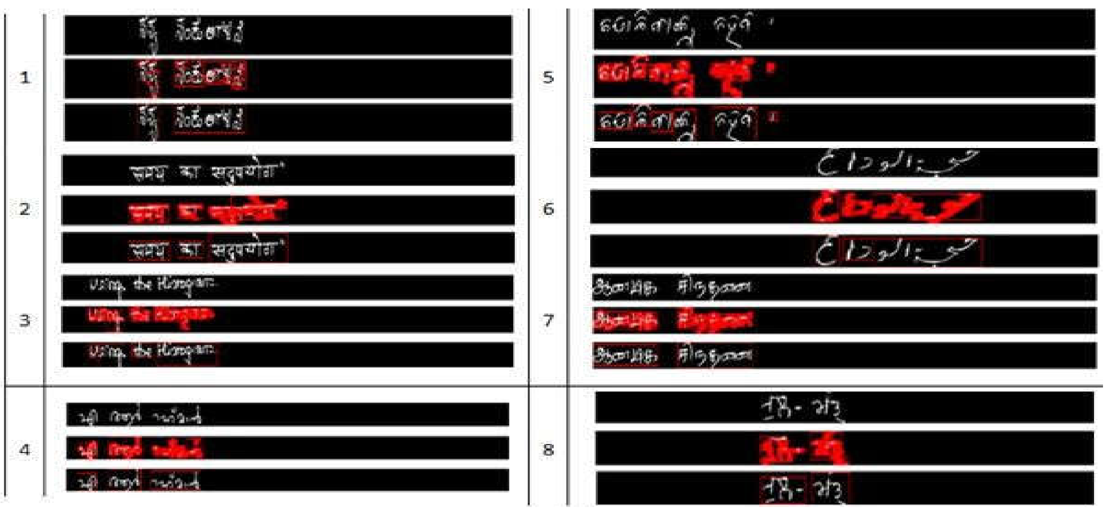

FIGURE 8: Samples images of text lines rejected

4. Summary

Script type identification using LBP, a powerful feature for texture classification, has been presented in this chapter. Experiments are performed on script images at line, word, and block level. Classification results are computed using KNN and SVM classifiers at bi-script and tri-script level. For images of text line, K-NN yielded overall accuracy of 92.63%, and SVM yielded over all accuracy of 94.91%. For word level images, K-NN yielded overall accuracy of 92.28%, and SVM yielded over all accuracy of 95.77%. Finally, for block level images, of 512x512 pixels having 3-lines, 4-lines and more than 4-lines of text, K-NN yielded 98.46 % accuracy and 99.5% accuracy was observed with SVM. It is observed that performance of SVM classifier is better over K-NN classifier in terms of overall recognition of script type identification in all the image types.

References

[1] Banu, J.F.; Muneeshwari, P.; Raja, K.; Suresh, S.; Latchoumi, T.P.; Deepan, S. Ontology based image retrieval by utilizing model annotations and content. In Proceedings of the 2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 27–28 January 2022; pp. 300–305. [**Google Scholar**]

[2] Fan, A.; Bhosale, S.; Schwenk, H.; Ma, Z.; El-Kishky, A.; Goyal, S.; Baines, M.; Celebi, O.; Wenzek, G.; Chaudhary, V.; et al. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.* **2021**, *22*, 4839–4886. [**Google Scholar**]

[3] Sumbul, G.; Charfuelan, M.; Demir, B.; Markl, V. Bigearthnet: A large-scale benchmark archive for remote sensing image understanding. In Proceedings of the IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 5901–5904. [**Google Scholar**]

[4] Islam, N.; Islam, Z.; Noor, N. A survey on optical character recognition system. *arXiv* **2017**, arXiv:1710.05703. [**Google Scholar**]

[5] Qiao, Z.; Zhou, Y.; Yang, D.; Zhou, Y.; Wang, W. Seed: Semantics enhanced encoder-decoder framework for scene text recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 13528–13537. [**Google Scholar**]

[6] Wang, T.; Zhu, Y.; Jin, L.; Luo, C.; Chen, X.; Wu, Y.; Wang, Q.; Cai, M. Decoupled attention network for text recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12216–12224. [**Google Scholar**]

[7] Yu, D.; Li, X.; Zhang, C.; Liu, T.; Han, J.; Liu, J.; Ding, E. Towards accurate scene text recognition with semantic reasoning networks. In Proceedings of the IEEE/CVF Conference

on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 12113–12122. [**Google Scholar**]
[8] Han, X.K.; Aysa, A.; Mamat, H.; Yadikar, N.; Ubul, K. Script Identification of Central Asia Based on Fused Texture Features. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 3675–3680. [**Google Scholar**]
[9] Zeebaree, D.Q.; Haron, H.; Abdulazeez, A.M.; Zebari, D.A. Trainable model based on new uniform LBP feature to identify the risk of the breast cancer. In Proceedings of the 2019 International Conference on Advanced Science and Engineering (ICOASE), Zakho, Iraq, 2–4 April 2019; pp. 106–111. [**Google Scholar**]
[10] Korkmaz, S.A.; Akçiçek, A.; Bínol, H.; Korkmaz, M.F. Recognition of the stomach cancer images with probabilistic HOG feature vector histograms by using HOG features. In Proceedings of the 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia, 14–16 September 2017; pp. 339–342. [**Google Scholar**]
[11] Garg, M.; Dhiman, G. A novel content-based image retrieval approach for classification using GLCM features and texture fused LBP variants. *Neural Comput. Appl.* **2021**, *33*, 1311–1328. [**Google Scholar**] [**CrossRef**]
[12] Kattenborn, T.; Leitloff, J.; Schiefer, F.; Hinz, S. Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 24–49. [**Google Scholar**] [**CrossRef**]
[13] Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Phys. D Nonlinear Phenom.* **2020**, *404*, 132306. [**Google Scholar**] [**CrossRef**] [**Green Version**]
[14] Dai, Z.; Liu, H.; Le, Q.V.; Tan, M. Coatnet: Marrying convolution and attention for all data sizes. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 3965–3977. [**Google Scholar**]
[15] Spitz, A.L. Determination of the script and language content of document images. *IEEE Trans. Pattern Anal. Mach. Intell.* **1997**, *19*, 235–245. [**Google Scholar**] [**CrossRef**] [**Green Version**]
[16] Padma, M.C.; Vijaya, P.A. Global approach for script identification using wavelet packet based features. *Int. J. Signal Process. Image Process. Pattern Recognit.* **2010**, *3*, 29–40. [**Google Scholar**]
[17] Ferrer, M.A.; Morales, A.; Pal, U. Lbp based line-wise script identification. In Proceedings of the 2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, 25–28 August 2013; pp. 369–373. [**Google Scholar**]

[18] Rani, R.; Dhir, R.; Lehal, G.S. Script identification of pre-segmented multi-font characters and digits. In Proceedings of the 2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, 25–28 August 2013; pp. 1150–1154. [**Google Scholar**]

[19] Shivakumara, P.; Sharma, N.; Pal, U.; Blumenstein, M.; Tan, C.L. Gradient-angular-features for word-wise video script identification. In Proceedings of the 2014 22nd International Conference on Pattern Recognition, Stockholm, Sweden, 24–28 August 2014; pp. 3098–3103. [**Google Scholar**]

[20] Mukarambi, G.; Mallapa, S.; Dhandra, B.V. Script identification from camera based Tri-Lingual document. In Proceedings of the 2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS), Chennai, India, 4–5 May 2017; pp. 214–217. [**Google Scholar**]

[21] Gomez, L.; Karatzas, D. A fine-grained approach to scene text script identification. In Proceedings of the 2016 12th IAPR Workshop on Document Analysis Systems (DAS), Santorini, Greece, 11–14 April 2016; pp. 192–197. [**Google Scholar**]

[22] Mei, J.; Dai, L.; Shi, B.; Bai, X. Scene text script identification with convolutional recurrent neural networks. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 4053–4058. [**Google Scholar**]

[23] Zdenek, J.; Nakayama, H. Bag of local convolutional triplets for script identification in scene text. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; Volume 1, pp. 369–375. [**Google Scholar**]

[24] Bhunia, A.K.; Konwer, A.; Bhunia, A.K.; Bhowmick, A.; Roy, P.P.; Pal, U. Script identification in natural scene image and video frames using an attention based convolutional-LSTM network. *Pattern Recognit.* **2019**, *85*, 172–184. [**Google Scholar**] [**CrossRef**] [**Green Version**]

[25] Cheng, C.; Huang, Q.; Bai, X.; Feng, B.; Liu, W. Patch aggregator for scene text script identification. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, NSW, Australia, 20–25 September 2019; pp. 1077–1083. [**Google Scholar**]