

## Multiclass Recognition of Offline Handwritten Devanagari Characters using CNN

**Mamta Bisht**

Department of Electronics and Communication Engineering,  
Jaypee Institute of Information Technology, Noida, India.  
*Corresponding author: bishmamta29@gmail.com*

**Richa Gupta**

Department of Electronics and Communication Engineering,  
Jaypee Institute of Information Technology, Noida, India.  
E-mail: richa.gupta@jiit.ac.in

(Received May 16, 2020; Accepted July 17, 2020)

### Abstract

The handwriting style of every writer consists of variations, skewness and slanting nature and therefore, it is a stimulating task to recognise these handwritten documents. This article presents a study on various methods available in literature for Devanagari handwritten character recognition and performs its implementation using Convolutional neural network (CNN). Available methods are studied on different parameters and a tabular comparison is also presented which concludes superiority of CNN model in character recognition task. The proposed CNN model results in well acceptable accuracy using dropout and stochastic gradient descent (SGD) optimizer.

**Keywords-** Optical character recognition (OCR), Devanagari script, Numeral recognition, Character recognition, Convolutional neural network.

### 1. Introduction

The recognition of scanned documents in term of characters/words is one of the major applications of pattern recognition known as optical character recognition (OCR). The scanned documents either printed or handwritten, involves pre-processing, segmentation, feature extraction followed by classification and post-processing steps in OCR. For scanned printed documents various tools are available in market with high recognition accuracy (Breuel, 2008). Some tools also perform well on neatly handwritten documents. But the precision of recognition system requires more accurate performance (Adak, 2019). Therefore, it is an active research area to improve results of recognition system. The various application of this research is in the field of handwritten notes/forms digitization, bank cheque processing, postal code recognition etc. In India, Devanagari is one of the most popular scripts in central and Northern part. In rural areas, most of the documentation work is done in Devanagari script. The national language of is also scripted in Devanagari.

This paper focuses on the existing work introduced in literature for Devanagari handwritten character recognition. This script consists total 49 basic characters which comprises 13 vowels (swar) and 33 consonants (vyanjan) and 3 compound characters as mentioned in Table 1. Every character consists a horizontal line in its top, called as headline (or shirorekha) which added when the characters combine. The way of writing of this script is from left to right direction on paper. One of the fundamental characteristics of this script is that when the consonant is trailed by vowel, the shape of consonant is modified with the addition of modifier (or matra) in left, right,

top or bottom position corresponding to that vowel. Sometimes, orthographic shaped character is also formed when consonant or vowel gets added with another consonant known as conjuncts (or Sanyuktakshar). The examples are shown in Table 1.

Table 1. Devanagari characters

Swar (Vowels)	अ आ इ ई उ ऊ ए ऐ ओ औ अं अः ऋ
Vyanjan (Consonants)	क ख ग घ ङ च छ ज झ ञ ट ठ ड ढ ण त थ द ध न प फ ब भ म य र ल व श ष स ह
Compound Characters	क्+ष= क्ष, त्+र= त्र, ज्+ञ=ज्ञ
Modified Characters (for consonant क (Ka))	का कि की कु कू के कै को कौ कं कः
Conjuncts (Few Examples)	क्क क्ख क्क त्र
Numerals	० १ २ ३ ४ ५ ६ ७ ८ ९

The major contributions of this article are as: (1) comparative study of offline Devanagari handwriting recognition in terms of numeral and characters. (2) Propose CNN architecture for Devanagari Handwritten Character Dataset (DHCD) recognition with dropout and SGD optimizer. (3) Classification using Random forest classifier on same dataset and validate CNN performance for multiclass classification. (4) Validate proposed CNN model on other dataset, MNIST.

The rest of the paper is arranged in five sections. Section 2 presents the challenges of Devanagari handwriting recognition system. Section 3 discusses the different existing recognition schemes for offline handwritten Devanagari script at numeral and character level. Section 4 and Section 5 presents the experimental setup and results for Devanagari character recognition using CNN model. Section 6 concludes the article and presents some future aspects.

## 2. Challenges in Devanagari Handwriting Recognition

Lots of challenges are identified for offline Devanagari handwritten documents recognition. The variations in the writing style of every writer, skewness and slanting nature etc., causes difficulty in the segmentation of document from text line to word and then word to character. This script has large number of character sets due to modifiers as compared to non- Indic script like Latin which make recognition system more challenging. Poorly scanned document or image captured from low resolution camera, spots, broken strokes, blurring etc. added noise to the document. The historical documents recognition is also very challenging task because of the availability of low-quality manuscript, missing of standard character and unknown font size etc. Presence of many confusing strokes and compound/conjunct character added more challenges to develop a Devanagari handwriting recognition system.

## 3. Recognition of Offline Handwritten Devanagari Script

Many Indic languages are written in Devanagari script like Hindi, Marathi, Sanskrit, Nepali etc. Indic script consists more variations in shape of characters as compared to non- Indic scripts like Latin, Chinese, Korean and Japanese etc. which introduces more challenges to build a handwriting recognition system for Indic scripts (Bharath and Madhvanath, 2009). Kumar et al. (2018) discussed key issues for character and numeral identification in Indic and non-Indic scripts. A review on different online and offline character recognition including Indic and non-Indic scripts are presented by Kaur et al. (2019). Singh et al. (2012) have discussed a review related to potential aspects of OCR in various fields. A review of methodologies applied in the

Indian language scripts recognition is described by Pal and Chaudhuri (2004) and Pal et al. (2012). Prasad (2014) also discussed an in-depth literature survey of Indic script recognition systems. Nowadays, convolution neural network (CNN) shows its importance in various fields and Indic script recognition is one of these (Mehrotra et al., 2013; Acharya et al., 2015; Maitra et al., 2015; Singh et al., 2016). The details of numeral and character recognition are presented in following subsections.

### **3.1 Devanagari Numeral Recognition**

In literature many feature extraction and classification techniques are used. Density, moment of right, left, upper and lower profile and descriptive component etc. are the main part of statistical features used by Bajaj et al. (2002). Elnagar and Harous (2003) worked on structural (or shape) features i.e., strokes and cavity features of thinned cursive handwritten Hindi numerals. Ramteke and Mehrotra (2006) have been worked on features which are based on moment, image partition, principle component analysis (PCA), correlation coefficient and perturbed moments. Garain et al. (2006) explores the strength of a clonal selection algorithm for 10-class classification problem of handwritten numerals. Sharma et al. (2006) divided the numeral (also applied for character) into blocks and then calculated the chain code histogram in each block. They used quadratic classifier for classification of obtained chain code features. Hanmandlu and Murthy (2007) represent the numerals in the form of modifying exponential relationship functions processed to the fuzzy sets. Box approach is used to obtain normalized distance features and then fitted to fuzzy set classifier. Hanmandlu et al. (2007b) also worked on box approach extracted features of handwritten Hindi numerals followed by bacterial foraging method. Patil and Sontakke (2007) explored general fuzzy neural network classifier for structural features of Devanagari numerals. Pal et al. (2007a) extracted directional information of the contour points of the handwritten numerals and then discussed a Modified Quadratic Discriminant Function (MQDF) for classification. Pal et al. (2009a) worked on postal automation by Indian multi-script full pin code string recognition using gradient features and MQDF classifier. Basu et al. (2010) presented a quad-tree-based longest run (QTLR) feature extraction scheme from the numeric digit designs followed by SVM classifier. Rajput and Mali (2010) used Fourier Descriptors to identify the figure of isolated Marathi handwritten numerals as features and then apply those features into three unlike classifiers namely, nearest neighbourhood (NN), K-nearest neighbourhood (KNN) and SVM. Acharya et al. (2015) presented Deep Convolutional Neural Network (CNN) model for Devanagari handwritten characters including numerals. Khanduja et al. (2016) described the statistical and structural features extraction scheme using MLP classifier for numeral and character recognition in Devanagari script.

### **3.2 Devanagari Character Recognition**

Arora et al. (2007) used structural properties of character for recognition purpose. The main focus of this article was headline (or shirorekha), spine and intersection properties of characters. The structural features are then treated by feed forward neural network for categorization. Hanmandlu et al. (2007a) defined feature extraction process using vector distance method and for classification of characters, they selected fuzzy set classifier. Pal et al. (2007b) discussed gradient filter and gaussian filter for feature extraction process. They choose quadratic classifier in their work for further classification. Bhattacharya and Chaudhuri (2008) presented a multistage cascaded scheme which focuses on wavelet based multi-resolution features and multilayer perceptron classifiers. Deshpande et al. (2008) acquired features from chain code rule and then regular expressions, minimum edit distance method (MED) etc. are used for character recognition scheme. Pal et al. (2008) described two set of features, first is directional information and second

is curvature-based property. The first is acquired from the arc tangent of the gradient and second is guided by gradient information. They used support vector machines (SVM) and modified quadratic discriminant function (MQDF) together for classification task. Arora et al. (2009) described different feature extraction and recognition algorithms in their work. In preliminary recognition step, the extracted features are based on chain code histogram, four side views, shadow etc. and then forwarded to MLP classifier. In final step, weighted majority method is used to combine the results of all MLP's. Kumar (2009) discussed five feature extraction approaches on Devanagari handwritten dataset and reported good performance of SVM classifier of gradient features. Mane and Ragma (2009) described elastic image matching (EM) procedure correlated to eigen-deformation. The calculation of deformation in offline handwritten character comprises category-dependent tendencies (also known as eigen-deformations) is a measure of feature set which is further examined for elastic matching-based character recognition. Pal et al. (2009b) discussed twelve different classifiers and four sets of feature obtained from curvature and gradient information of image. They reported that the best recognition results are achieved by using Mirror Image Learning (MIL) classifier. Arora et al. (2010) proposed character recognition in double stages, first using neural network (NN) and second using minimum edit distance (MED). In the first stage two MLPs are used, one is applied on shadow features of character and other on chain code histogram features. Top three combined results of MLPs are considered using weighted majority rule to compute relative difference values. These relative differences are used in second stage to divide the character in two sets as distinct shapes and disordered characters (or alike shapes) and classified by using MLP and MED method respectively.

The comparative analysis of methods discussed above for offline Devanagari handwritten document recognition is tabulated in Table 2 with description of various parameters.

Below table summarizes various feature extraction and classification methods along with the accuracies obtained on numeral/character recognition process for Devanagari script. The feature extraction algorithm basically finds the relevant details about the input text image and map it to the corresponding label in training process like shape, position etc. Variable length dataset has been used by researchers. In this work, a deep neural network is proposed consisting four convolutional layers; mix with max-pool and nonlinearity 'RELU'. Dropout is added with regular CNN architecture in order to avoid over fitting. We selected SGD optimizer to find the values of parameters and reduce the cost of function.

Table 2. Comparative study of recognition results in Devanagari script

Authors	Numeral/ Character (Dataset size)	Method		Accuracy (%)
		Feature	Classifier	
Bajaj et al. (2002)	Numeral (2460)	Statistical	Neural network	89.68
Elnagar and Harous (2003)	Numeral (1500)	Structural	Matching synthetic representation	96
Ramteke and Mehrotra (2006)	Numeral (2000)	Moment invariants	Gaussian distribution	92
Garain et al. (2006)	Numeral (12000)	Binary image	Clonal selection	96
Sharma et al. (2006)	Numeral (22556) Character (11270)	Chain code	Quadratic	98.86
Hanmandlu and Murthy (2007)	Numeral (35000)	Box approach	Fuzzy Model	95
Hanmandlu et al. (2007b)	Numeral (3500)	Box approach	Bacterial foraging	96
Patil and Sontakke (2007)	Numeral (2000)	Structural	General fuzzy neural network	99.50
Pal et al. (2007a)	Numeral (22556)	Gradient	MQDF	99.56

Table 2 continued...

Arora et al. (2007)	Character (50000)	Structural	Feed forward neural network	89.12
Hanmandlu et al. (2007a)	Character (4750)	Vector distance	Fuzzy sets	90.65
Pal et al. (2007b)	Character (36172)	Gradient and Gaussian filter	Quadratic	94.24
Bhattacharya and Chaudhuri (2008)	Numeral (22556)	Wavelet	MLP	99.27
Deshpande et al. (2008)	Character (5000)	Chain code	Regular expression (RE) and MED	82
Pal et al. (2008)	Character (36172)	Gradient	SVM & MQDF	95.13
Pal et al. (2009a)	Numeral (23340)	Gradient	MQDF	98.41
Arora et al. (2009)	Character (1500)	Combined	MLP	89.58
Kumar (2009)	Character (25000)	Gradient	SVM	94.1
Mane and Ragha (2009)	Character (3600)	Eigen deformation	Elastic Matching	94.91
Pal et al. (2009b)	Character (36172)	Gradient	Mirror Image Learning (MIL)	95.19
Basu et al. (2010)	Numeral (3000)	Quad-tree-based longest run feature (QTLR)	SVM	97
Rajput and Mali (2010)	Numeral (13000)	Fourier descriptor	SVM	97.85
Arora et al. (2010)	Character (7154)	Shadow and chain-code-histogram	MLP & MED	90.74
Acharya et al. (2015)	Character and Numeral (92000)	CNN	CNN classifier	98.47
Khanduja et al. (2016)	Character and Numeral (23000)	Local and Global features, Hybrid features	MLP classifier	91.4 95.5

#### 4. Experimental Setup

The experiments are done on DHCD which consists of 78,200 samples for training and 13,800 samples for testing. Many experiments are performed by varying parameters using sequential model of keras and choosing the one with best validation performance. The proposed CNN model configuration is presented in Figure 1 and summarized in Table 3.

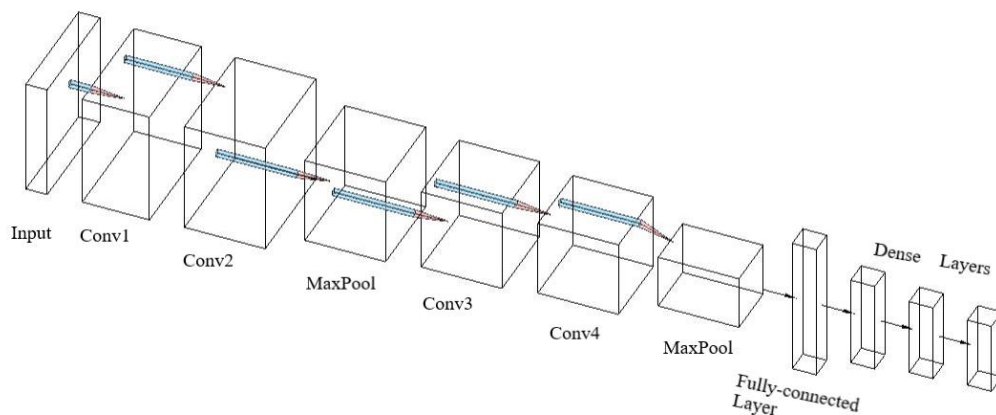


Figure 1. Architecture of proposed CNN model

The sample image (matrix with pixel values) is entered into first convolution layer. From the top left, the kernel (here, kernel size is 3x3) moves along the image by unit 1 as depending on stride

value. The role of kernel is to multiply its values with the pixel values of image and then these multiplications are added to get one value. Similarly, all the operations are done after passing kernel across all positions in image and at last a matrix is obtained, having size smaller than input matrix. The network comprises four convolutional layers mixed with nonlinearity as Rectified linear unit (RELU) and max pooling layers. In the network the output of first convolutional layer becomes the input of second layer and this happens for further layers. The property of activation function, RELU which is added after each convolutional layer is as:

$$f(x) = 0 \quad \text{if } x < 0$$

$$f(x) = x \quad \text{if } x \geq 0.$$

The pooling layer performs down sampling operation on width and height of image in order to reduce the image volume. If some features (e.g. boundaries, line, curve etc.) have been identified in previous layer, then this layer compress image to less detailed values. In deep learning model multiple connections and trainable parameters exists in number of non-linear hidden layers which make the model inclined to over fitting. Therefore, dropout of 0.2 is used to prevent model from overfitting phenomenon. Flatten performs the input role for further fully connected layers. Two fully connected layers are added with the output shape 128, 64 and activation as RELU function. The last fully connected layer is attached with dimension equal to number of output classes (here, 46) with softmax activation function. The softmax function is also known as normalized exponential function which takes k real numbers vector as input and normalizes it to k probabilities proportional to the exponentials of input numbers. It maps the non-normalized output of CNN model to a probability distribution over predicted output classes.

Table 3. Network configuration of CNN model

Type	Configuration
Input	32x32, 3 channel
Convolution (Conv1)	Filters: 32, kernel size: 3x3, stride: 1, Output size: 32x30x30
Convolution (Conv2)	Filters: 64, kernel size: 3x3, stride: 1, Output size: 64x28x28
Max pooling	Pool size: 2x2, stride: 2x2, pad: same, Output size: 64x14x14
Convolution (Conv3)	Filters:64, kernel size: 3x3, stride: 1, Output size:64x12x12
Convolution (Conv4)	Filters:64, kernel size: 3x3, stride: 1, Output size: 64x10x10
Max pooling	Pool size: 2x2, stride: 2x2, pad: same, Output size: 64x5x5
Dropout	0.2
Flatten	1600
Dense	128
Dense	64
Softmax	46

The model architecture is implemented on DHCD for multiclass classification problem. For the compilation of model, stochastic gradient descent (SGD) optimizer is selected for the standard back-propagation and feed-forward network. The categorical cross-entropy loss function (1) and accuracy metrics is used to check the performance of model. The stochastic gradient descent is an iterative optimization algorithm to reduce the loss function which measures the difference

between the actual output and predicted output from the model and updates the weights ( $w$ ) for the neurons for all iteration as shown below:

$$w = w + \lambda \frac{\partial l}{\partial w}$$

where,  $\lambda$  is learning rate and  $\partial l / \partial w$  is the gradient.

The network is trained for 50 epochs with mini-batch size 32. The advantage of CNN model is that it extracts features automatically from sample images which are different from conventional handcrafted features extraction method and it also predict classification probabilities of images to total number of output classes using softmax activation function. CNN is superior for image classification task than traditional machine learning methods for large database.

### 5. Results

Three cases of dataset distribution are presented in Table 4 with validation and test accuracies. It is observed that Case 2 obtained highest validation and test accuracies.

Table 4. Results on DHCD

Cases	Samples			Accuracy	
	Train	Val	Test	Val	Test
Case 1	78200	11040	2760	98.85	98.91
Case 2	78200	6900	6900	99.01	98.93
Case 3	62560	15640	13800	98.81	98.80

For Case 2, the training and validation accuracy and loss curves are shown in Figure 2.

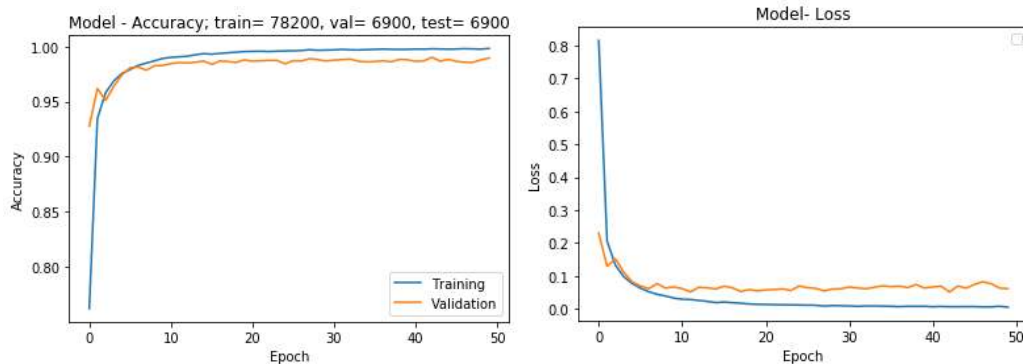


Figure 2. Training and loss curves for best model

To check the superiority of CNN for image classification we check another method. We extract features from above proposed pretrained CNN model at second last layer and train a Random forest (RF) classifier supervisely. Then check the performance of trained RF classifier for test samples and result is presented in Table 5. The RF classifier performance is less than CNN which

proves CNN superior classifier.

Table 5. Results using CNN features and random forest classifier

Samples	Features extracted by Pre-trained CNN model	Test accuracy by Random forest classifier
Train-78200	782400x64	96.89 %.
Test-13800	13800x64	

The validation of proposed CNN model is also validated on MNIST dataset and a well acceptable accuracy of 99.28% is obtained. Result of proposed CNN model is compared with the CNN model presented by Acharya et al. (2015) on the same dataset (DHCD). Results are tabulated in Table 6 and Table 7.

Table 6. Performance of model for MNIST dataset

Samples (MNIST)	Test Accuracy in 12 epochs
Train-60000 Test-10000	99.28%

Table 7. Comparison of results

Model	Test accuracy
Acrarya et al. (2015)	98.47
Proposed model	98.93

## 6. Conclusion

It is observed that various methods have been developed in literature for Devanagari handwritten character recognition and out of all CNN perform best for both character and numeral recognition together for large dataset. This article presents a well performing CNN model for multiclass classification task on Devanagari handwritten character dataset (DHCD). The discussed work is based on fair quality dataset. For degraded documents the performance of recognition system requires further improvement. For this, new most relevant feature extraction and classification method can be explored.

### Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of this work.

### Acknowledgments

The authors are grateful to the editor and reviewers for their helpful suggestions.



## Reference

- Acharya, S., Pant, A.K., & Gyawali, P.K. (2015). Deep learning based large scale handwritten Devanagari character recognition. In *2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)* (pp. 1–6). IEEE. Kathmandu, Nepal.
- Adak, C. (2019). A study on automated handwriting understanding (PhD Thesis). Retrieved from <https://opus.lib.uts.edu.au/bitstream/10453/134139/1/01front.pdf>.
- Arora, S., Bhattacharjee, D., Nasipuri, M., & Malik, L. (2007). A two stage classification approach for handwritten Devnagari characters. In *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)* (Vol. 2, pp.399–403). Sivakasi, Tamil Nadu, India.
- Arora, S., Bhattacharjee, D., Nasipuri, M., Basu, D.K., & Kundu, M. (2010). Recognition of non-compound handwritten Devnagari characters using a combination of MLP and minimum edit distance. *International Journal of Computer Science and Security*, 4(1), 107-120.
- Arora, S., Bhattacharjee, D., Nasipuri, M., Basu, D.K., Kundu, M., & Malik, L. (2009). Study of different features on handwritten Devnagari character. In *2009 Second International Conference on Emerging Trends in Engineering & Technology* (pp.929–933). Nagpur, India.
- Bajaj, R., Dey, L., & Chaudhury, S. (2002). Devnagari numeral recognition by combining decision of multiple connectionist classifiers. *Sadhana*, 27(1), 59–72.
- Basu, S., Das, N., Sarkar, R., Kundu, M., Nasipuri, M., & Basu, D.K. (2010). A novel framework for automatic sorting of postal documents with multi-script address blocks. *Pattern Recognition*, 43(10), 3507–3521.
- Bharath, A., & Madhvanath, S. (2009). Online handwriting recognition for Indic scripts. In *Guide to OCR for Indic scripts* (pp. 209–234). Springer, London.
- Bhattacharya, U., & Chaudhuri, B.B. (2008). Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3), 444–457.
- Breuel, T.M. (2008). The OCRopus open source OCR system. *Document Recognition and Retrieval XV*, 6815, 68150F. International Society for Optics and Photonics. <https://doi.org/10.1117/12.783598>.
- Deshpande, P.S., Malik, L.G., & Arora, S. (2008). Fine classification & recognition of hand written Devnagari characters with regular expressions & minimum edit distance method. *JCP*, 3(5), 11–17.
- Elnagar, A., & Harous, S. (2003). Recognition of handwritten Hindu numerals using structural descriptors. *Journal of Experimental & Theoretical Artificial Intelligence*, 15(3), 299–314.
- Garain, U., Chakraborty, M.P., & Dasgupta, D. (2006). Recognition of handwritten indic script using clonal selection algorithm. In *International Conference on Artificial Immune Systems* (pp. 256–266). Springer, Berlin, Heidelberg.
- Hanmandlu, M., & Murthy, O.R. (2007). Fuzzy model based recognition of handwritten numerals. *Pattern Recognition*, 40(6), 1840–1854.
- Hanmandlu, M., Murthy, O.R., & Madasu, V.K. (2007a). Fuzzy model based recognition of handwritten Hindi characters. In *2007 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications* (pp. 454–461). Glenelg, Australia.
- Hanmandlu, M., Nath, A.V., Mishra, A.C., & Madasu, V.K. (2007b). Fuzzy model based recognition of handwritten Hindi numerals using bacterial foraging. In *2007 6th IEEE/ACIS International Conference on Computer and Information Science* (pp.309–314). Melbourne, Qld., Australia.

- Kaur, S., Bawa, S., & Kumar, R. (2019). A survey of mono-and multi-lingual character recognition using deep and shallow architectures: Indic and non-Indic scripts. *Artificial Intelligence Review*, 53, 1813–1872.
- Khanduja, D., Nain, N., & Panwar, S. (2016). A hybrid feature extraction algorithm for Devanagari script. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 15(1), 1-10.
- Kumar, M., Jindal, M.K., Sharma, R.K., & Jindal, S. R. (2018). Character and numeral recognition for non-Indic and Indic scripts: a survey. *Artificial Intelligence Review*, 52(4), 2235-2261.
- Kumar, S. (2009). Performance comparison of features on Devanagari hand-printed dataset. *International Journal of Recent Trends in Engineering*, 1(2), 33.
- Maitra, D.S., Bhattacharya, U., & Parui, S.K. (2015). CNN based common approach to handwritten character recognition of multiple scripts. In *2015 13th International Conference on Document Analysis and Recognition* (pp.1021–1025). IEEE. Tunis, Tunisia.
- Mane, V., & Ragha, L. (2009). Handwritten character recognition using elastic matching and PCA. In *Proceedings of the International Conference on Advances in Computing, Communication and Control* (pp. 410–415). Mumbai, India.
- Mehrotra, K., Jetley, S., Deshmukh, A., & Belhe, S. (2013). Unconstrained handwritten Devanagari character recognition using convolutional neural networks. In *Proceedings of the 4th International Workshop on Multilingual OCR* (pp.1-5). Washington, D.C., USA.
- Pal, U., Roy, R.K., Roy, K., & Kimura, F. (2009a). Indian multi-script full pin-code string recognition for postal automation. In *2009 10th International Conference on Document Analysis and Recognition* (pp. 456–460). IEEE. Barcelona, Spain.
- Pal, U., & Chaudhuri, B.B. (2004). Indian script character recognition: a survey. *Pattern Recognition*, 37(9), 1887–1899.
- Pal, U., Chanda, S., Wakabayashi, T., & Kimura, F. (2008, August). Accuracy improvement of Devnagari character recognition combining SVM and MQDF. In *Proceedings 11th International Conference on Frontiers in Handwriting Recognition* (pp. 367–372). Montreal, Canada.
- Pal, U., Jayadevan, R., & Sharma, N. (2012). Handwriting recognition in indian regional scripts: A survey of offline techniques. *ACM Transactions on Asian Language Information Processing (TALIP)*, 11(1), 1.
- Pal, U., Sharma, N., Wakabayashi, T., & Kimura, F. (2007a). Handwritten numeral recognition of six popular Indian scripts. In *Ninth International Conference on Document Analysis and Recognition* (Vol. 2, pp.749–753). IEEE. Parana, Brazil.
- Pal, U., Sharma, N., Wakabayashi, T., & Kimura, F. (2007b). Off-line handwritten character recognition of devnagari script. In *Ninth International Conference on Document Analysis and Recognition* (Vol.1, pp.496–500). IEEE. Parana, Brazil.
- Pal, U., Wakabayashi, T., & Kimura, F. (2009b). Comparative study of Devnagari handwritten character recognition using different feature and classifiers. In *2009 10th International Conference on Document Analysis and Recognition* (pp. 1111–1115). IEEE. Barcelona, Spain.
- Patil, P.M., & Sontakke, T.R. (2007). Rotation, scale and translation invariant handwritten Devanagari numeral character recognition using general fuzzy neural network. *Pattern Recognition*, 40(7), 2110–2117.
- Prasad, J.R. (2014). Handwritten character recognition: a review. *International Journal of Computer Science and Network Security*, 3(5), 340–351.

- Rajput, G.G., & Mali, S.M. (2010). Fourier descriptor based isolated Marathi handwritten numeral recognition. *International Journal of Computer Applications*, 3(4), 9–13.
- Ramteke, R.J., & Mehrotra, S.C. (2006). Feature extraction based on moment invariants for handwriting recognition. In *2006 IEEE Conference on Cybernetics and Intelligent Systems* (pp. 1–6). IEEE. Bangkok, Thailand.
- Sharma, N., Pal, U., Kimura, F., & Pal, S. (2006). Recognition of off-line handwritten devnagari characters using quadratic classifier. In: Kalra, P., Peleg, S. (eds) *Computer Vision, Graphics and Image Processing*. Springer, Berlin, Heidelberg, pp. 805–816.
- Singh, A., Bacchuwar, K., & Bhasin, A. (2012). A survey of OCR applications. *International Journal of Machine Learning and Computing*, 2(3), 314.
- Singh, P., Verma, A., & Chaudhari, N.S. (2016). Deep convolutional neural network classifier for handwritten Devanagari character recognition. In: Satapathy, S., Mandal, J., Udgata, S., Bhateja, V. (eds) *Information Systems Design and Intelligent Applications*. Springer, New Delhi, pp. 551–561.

