IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 33, No. 6, December, 2019, pp. 441-446
Journal homepage: http://iieta.org/journals/ria

Histogram of Oriented Gradients Based Off-Line Handwritten Devanagari Characters Recognition Using SVM, K-NN and NN Classifiers

Shalaka P. Deore[1,3*], Albert Pravin[2]

[1] Research Scholar at Sathyabama Institute of Science and Technology, Chennai 600119, India
[2] Department of Computer Science & Engineering, Sathyabama Institute of Science and Technology, Chennai 600119, India
[3] Department of Computer Engineering, M.E.S. College of Engineering, Pune, S. P. Pune University, Maharashtra 411001, India

Corresponding Author Email: shalaka.deore@mescoepune.org

https://doi.org/10.18280/ria.330606

Received: 20 August 2019
Accepted: 12 October 2019

Keywords:
Devanagari character recognition, feature extraction, digitization, histogram of oriented gradients, K-nearest neighbor, neural network, support vector machine

ABSTRACT

This paper presents influence of Histogram of Oriented Gradients (HOG) features on three different popular classifiers: Support Vector Machine (SVM), K-Nearest Neighbor (K-NN) and Neural Network (NN). Devanagari script is one among the foremost standard scripts in India used in many languages. The main objective of our study is to recognize Handwritten Devanagari Characters using HOG features and also examine the efficiency of HOG features on it using multiple cells of HOG descriptor. Here, presented Handwritten Devanagari Character Recognition System (HDCRS) is used to recognize isolated characters by making use of SVM, K-NN and NN well known classifiers. HOG is very effective and dense technique used to extract features from every positions of an image by dividing images into cells. HOG tries to capture shape of an image by using gradients information. The features extracted are classified using above mentioned classifiers. Through this study, we observed that SVM+HOG combination achieved highest accuracy: a recognition rate of 87.38%. The findings of this research may help in digitization of Handwritten Devanagari Documents.

1. INTRODUCTION

In modern communication utilization of electronic files is common and most favored of information sharing. This electronic document saves human efforts and time of rewriting. It is also called as soft copy which is electronic copy of some type of data. It is easy and secure way of storing and sharing documents. But still many applications like mail sorting, digital signature verification, postal cards [1], government documents, historical data processing [2] require handwritten documents to be processed. Consequently, the necessity of having automated tool to recognize handwritten documents accurately is required. Recognizing the printed documents by a computer (referred as OCR) and recognizing the handwritten documents (referred as HCR) are two major groupings of character recognition (CR) process. HCR again divided into two types: online HCR and offline HCR. In Online HCR technique input images are accepted using pen tablet called as digitizer. While writing character using pen tablet its online information is captured such as x, y coordinates, pressure and stroke information etc. In offline HCR method input images are collected by optical scanning of the paper using scanner and converting it into computer format. Due to many reasons listed below it is very difficult to recognize handwritten characters: i) low distraction due to noise while collecting the OCR data compared to HCR, ii) In OCR variation is only limited to selected font type and size whereas in HCR the variation is result of individual's style of writing, iii) Individual's personal situation also causes variation in handwriting style making HCR further complex, iv) Shape similarity and v) complex variation in writing due to number of combinations of different characters like Matras and composite letters.

Generally, recognition of character is classified into two classes: OCR and HCR. In the category of OCR has achieved relatively good performance and now researchers are focusing on HCR systems. While Chinese and English are top two written languages in the world, Hindi stands third widely spoken and written language [3]. Likewise, explore on Devanagari script has attracted ample interest because of its great market potential. Bristow et al. [4] showed that HOG features can be reformulated as affine weighting on the margin of SVM classifier to its quadratic kernel and achieved great success. In paper [5], authors state comparative study on Arabic text using SVM and NN machine learning algorithms. They collected data from news articles for training and testing and reveal that SVM performing better. In paper [6] author used different types of moments as features to recognize online as well as offline Devanagari characters. Here classification is done using NN. Different moment feature combination is explored to improve recognition accuracy of Devanagari characters. Wan et al. [7] proposed hybrid approach (SVM+K-NN classifiers) for categorization of text. In this hybrid approach, training data points are reduced to support vectors using SVM for each class and for calculating distance between testing data to support vectors is done by using KNN. Experiment results of Kamble and Hegadi [8] reveal that proposed hybrid approach has minimum dependency on training parameters. Author explained rectangle HOG feature extraction method and they achieved great results on these features using SVM and NN classifiers. Effective indexing schemas on images based on texture based features are explained in the study of Minu et al. [9], which is used for dimensionality reduction purpose. The approach proposed by author required less computational time due to reduction happened in feature vector. Features extracted by HOG are so efficient to recognize not only Devanagari characters but also Malayalam [14], Bangla [15] and Kannada [16] popular scripts of India. Bannigidad et al. [16] presented the work which shows the importance of HOG feature descriptor to recognize different age-typed historical Kannada documents. Classification done by using two different classifiers K-NN and SVM. SVM achieved best accuracy than K-NN on HOG features. The main research work proposed by Nongmeikapam et al. [17] is to recognize all characters of Manipuri Meetei-Mayek using feed forward NN and SVM with HOG descriptor. Further, a comprehensive survey is done by Puri and Singh [18] along with the concept of Hindi handwritten as well as printed document classification using fuzzy logic and SVM. On-line Devanagari character recognition is presented by Deore and Pravin [19] using 1-D moment descriptor. They explored that in case of on-line character recognition system 1-D moment features are giving good accuracy on handwritten characters which are is not subjective to stroke sequence. Experiments are done by the author proved that 1-D moments can also be used to recognize character images which are not influenced by stroke information.

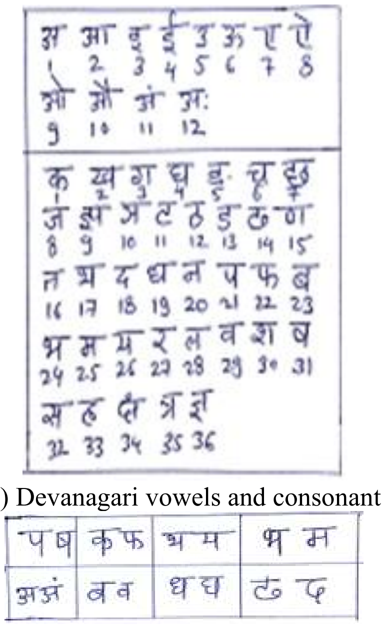
The study reveals that many researches have contributed significant work towards recognition of handwritten/ printed characters in spite of many challenges. The common challenges/problems encounter is difficulty in recognition of similar shape and compound characters, time, memory space and very important reliability. Where the numerical values are involved reliability is very important. They have explored many different feature extraction and classification methods where HOG descriptor giving very good performance in classification. HOG features having some unique features like they are stable with change in radiance and due to operating on local cells they are invariant to geometric and photometric transformations. Motivated by this particular work, our contributions toward research are:

- Presented a system to recognize Isolated Handwritten Devanagari Characters based on HOG features to minimize the challenges arising due to complexities discussed above.
- Analysis of recognition time and accuracy with different HOG cells. Experiments shows efficiency of the system improves as cell size increases.
- With SVM we also explored two more popular classifiers: K-NN and NN to evaluate effectiveness of HOG descriptor.

The flow of the paper is ordered as: Section 2 introduces features of Devanagari Scripts and its importance, Section 3 describes architecture of the proposed system with classifiers, experimental results are discussed in Section 4. Finally, Section 5 presents conclusion and perspective.

2. DEVANAGARI SCRIPT

Devanagari script has total 58 characters, of which 12 are vowels, 36 are consonants and 10 numbers. Over 120 Indian languages practice the Devanagari script in their daily communication written as well as verbal. Devanagari language follows left to right writing approach. Each word consists of one horizontal line on top of the word to differentiate with other word. This horizontal line is called as "Shirorekha" and must be present on top of the word. Due to different handwriting style and similarity between some characters it is difficult to achieve satisfactory recognition rate. In Figure 1 first part (a) is the sample of Devanagari vowels and consonants. Second part (b) shows some similar shape character samples. Figure 1(b) clearly depicts that difference between these characters are very low that is they can differentiate only with the loop present 'य' and 'थ'. For some simple dot is present 'अ' and 'अं'. Sometimes this dot can be treated like noise by the system. So it is complex task to achieve satisfactory recognition rate of similar shape characters.



(a) Devanagari vowels and consonants

(b) Similar shape Devanagari characters

Figure 1. Samples of handwritten Devanagari characters

Devanagari script has many unique properties and contains numerous methods in its writing and reading styles, as stated below:

- No uppercase and lowercase separation between characters.
- It is read in same order as written form left to right and top to bottom.
- It consists of horizontal line on top of the characters but for some characters like 'भ' and 'ध' it is partly not present as shown in Figure 1(b).
- Devanagari characters are structurally divided into three groups based on the kana (vertical line) modifier present in the character. The vertical line present at the end, in between or not present. Some samples are shown in Figure 1(b). These separate sets make classification easy.
- It uses phonetic order.
- In speech, a vowel takes more periods compared to consonants.
- Combination of vowels and consonants are used to form "Barakhadi".
- Combining syllables and characters builds syllabic alphabets in Devanagari script.

3. ARCHITECTURE OF PROPOSED HDCRS

The proposed system works on 48 characters classes which consist of 12 Devanagari vowels and 36 Devanagari consonants shown in Figure 1. Figure 2 depicts proposed architecture of the HDCRS model. Mainly HDCRS is divided into three phases. Firstly, input image is passed to preprocessing stage then in next phase features are extracted and lastly it sends for recognition. These three phases are explained below in detailed.

3.1 Devanagari dataset and pre-processing

Dataset used for any classification system plays important role. Proposed system trained on total 48 Devanagari characters which is consist of 12 vowels and 36 consonant characters [10]. Data set used for the experiment is taken form Kaggle which is standard and publicly available dataset of Devanagari characters. Samples of these dataset are collected from school level students of different age [13]. In this work, dataset comprises of total 10560 images of 48 classes. Experiments are conducted on 220 images of each 48 classes. Both SVM and K-NN are given 170 images of each class for training purpose and 50 images of each class for testing purpose. NN is trained on 160 images of each class, tested on 50 images and validation is done on 10 images of each class. Three preprocessing operations are performed: Size normalization, Binarization and Thinning. All images are resized to 32×32.

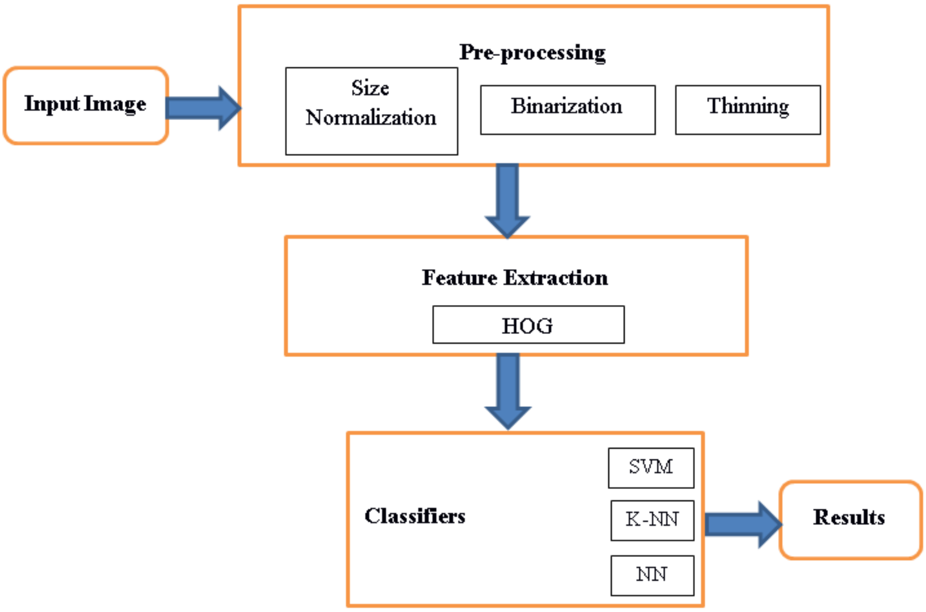


Figure 2. HDCRS architecture

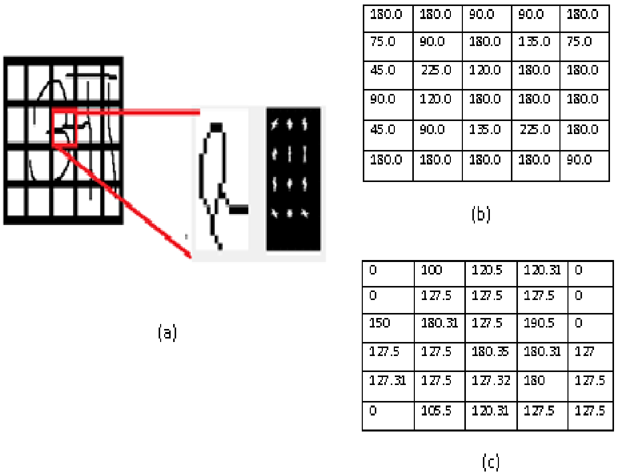
3.2 Feature extraction: Histogram of oriented gradients

The HOG feature extraction technique uses the distribution of directions of gradients as features. Important feature like oriented gradients are extracted using HOG. Gradients of an image are beneficial because the magnitude of gradients is large around edges and corners because there only intensity changes suddenly. In our work, edges and corners information is considered as they specify shape of the image perfectly. Magnitude and direction of gradient is computed by Eq. (1).and Eq. (2). as shown below [11]:

$$g = \sqrt{gx^2 + gy^2} \qquad (1)$$

$$\theta = \arctan\left({}^{gy}/_{gx}\right) \qquad (2)$$

where, *gx* is change in the x-direction and *gy* is change in y-direction. Then total gradient magnitude g is calculated using Eq. (1). Magnitude is high when large change in intensity values specially around edges. Then the orientation θ is calculated. Now gradients and orientation used to calculate histrogram. Experiments are performed on dividing images into 2×2, 4×4 and 8×8 cells and then HOG is calculated on each cell. Figure 3 shows gradient directions and magnitude computed using 8×8 cells on specific patch of image.



180.0	180.0	90.0	90.0	180.0
75.0	90.0	180.0	135.0	75.0
45.0	225.0	120.0	180.0	180.0
90.0	120.0	180.0	180.0	180.0
45.0	90.0	135.0	225.0	180.0
180.0	180.0	180.0	180.0	90.0

0	100	120.5	120.31	0
0	127.5	127.5	127.5	0
150	180.31	127.5	190.5	0
127.5	127.5	180.35	180.31	127
127.31	127.5	127.32	180	127.5
0	105.5	120.31	127.5	127.5

Figure 3. (a) HOG features on one patch of the image. (b) Gradient Direction. (c) Gradient Magnitude

Next step is to calculate histogram of gradients. The histogram contain array of 9 bins related to angles and particular magnitude information will go to respective bin. Like these 324 feature vectors are computed on 32×32 image using 8×8 cell.

3.3 Classification

In classification phase three classifiers SVM, K-NN and NN are implemented to comprehend effect of HOG features. This section presents brief theory of SVM, K-NN and NN classifiers

3.3.1 SVM classifier

Support Vector Machine (SVM) classifier is mainly defined by the presence of a separating hyperplane. That is, given a labelled set of training data the algorithm outputs an optimal hyper plane which classifies the new examples. Hyperplane basically divides data into two classes. The main aim is to discover a plane which having maximum margin. Margin distance maximization gives some support so that new data classified with more certainty. Linear classification is very easy but for non-linear classification different kernel functions are used to map the given input data into higher dimensional space so that separation can be easily possible using lines. A non-linear curve makes data separation so difficult. Support vector machine is extremely preferred by several as it gives significant accuracy with minimum computation power. Mainly SVM comes under binary classification category and also can be used for multiclass classification. Several binary SVMs are required implementing for multiclass problem. Two approaches are useful: One-vs-rest and one-vs-one SVM. The proposed framework utilizes K (K – 1)/2 binary SVM models utilizing the one vs one methodology, where k is number of class labels.

3.3.2 K-NN classifier

K-NN is one of the simplest classifiers uses distance measure to classify the class of new data. In training phase feature vectors divided into number of regions and then according to the similarity present between them are combined to specific region. For similarity calculation distance measure is used. Basically it works well on small data set. K-NN algorithm locates k closest training vector irrespective of labels as shown in Figure 4. When new data Xi is given for classification algorithm it locates for k closest neighbors form new data point using distance function here in example k is 3. Then voting will happened and new data is classified accordingly in testing phase. For our experiment k value is selected as 3 and Euclidean distance function for performing experiment. The Euclidean distance function is stated below:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad (3)$$

where, x is the feature vector of the input data, y stands for training dataset samples and n denotes the size of the feature vector [12].

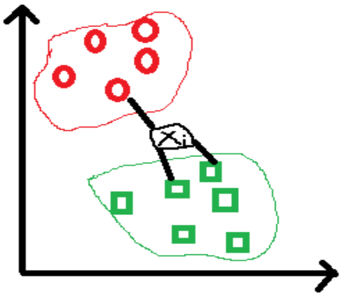


Figure 4. 2-D Feature vectors with 3-NN classifier

3.3.3 NN classifier

Multilayer feedforward neural network is designed using backpropagation method. It comprises 3 layers as input, one hidden and output layers which usage gradient descent to reduce the squared error between observed output and target output. Initially assign small random weights to the network. Propagate input forward through the network. Calculate output for each unit using predefined weights. For each output unit calculate error which is propagated back through the network and weights are updated accordingly. The objective is to minimize these errors and improve efficiency. Network parameters like hidden layer neurons, its activation function and training function can be different and chosen for the optimal functioning of the network. Designated parameters for the proposed system are given in Table 1.

Table 1. Neural network parameters of HDCRS

Neural Network Parameters	Values
Inputs	Training: 160, Testing: 50 and Validation 10 images per class
Number of outputs	Characters in 48 classes
Hidden layers	One
Hidden units	70
Activation function of Hidden layer	Tangent Sigmoid
Number of epochs	107 Iterations
Output layer activation function	Linear
Error Function	Mean Square Error (MSE)

4. RESULTS AND DISCUSSIONS

The Handwritten character dataset used for experiment consist of 220 images of each 48 classes. This result total 10560 Devanagari character images in the dataset. 170 of those 220 imaged are trained and the rest 50 are tested using the SVM and K-NN classifiers. And 160 images are given for training, 50 for testing and 10 for validation purpose to NN classifier. No restriction on size of the database but system works on isolated unconstrained characters. Various experiments are done to check effectiveness of HOG feature extraction method on thinned and not thinned character images with using different classifiers like SVM, K-NN and NN. Also effect of various cell sizes is analyzed. Experiment results are tabulated below. Table 2 shows experiments are done on thinned images as well as without thinned characters.

Table 2. Result of thinning pre-processing operation

Classifiers	Cell Size	Thinned image Accuracy (%)	Without Thinned image Accuracy (%)
SVM + HOG	[8×8]	83.04	87.38
KNN + HOG	[8×8]	78.92	84.33
NN + HOG	[8×8]	81.2	82

Skeletonization (thinning) operation is not better option for HOG feature extraction method which gives less recognition rate as compared to without thinning because gradients of an image play very vital role in feature extraction and magnitude of gradients is large around edges of images compared with flat area. Magnitude of gradients changes with respect to change in intensity values of an image. So when we performed thinning operation on an image it does not consider higher intensity values for calculation. Recognition rate of SVM, KNN and NN it is reduced by 4.34%, 5.16% and 0.8%, respectively.

Table 3. Result of different classifiers with different cell size

Classifiers	Cell Size	Time	Accuracy (%)
SVM + HOG	[2×2]	355sec	85.42
	[4×4]	233sec	86.58
	[8×8]	61sec	87.38
KNN + HOG	[2×2]	296sec	82.54
	[4×4]	89sec	84.08
	[8×8]	34sec	84.33
NN + HOG	[2×2]	200sec	79.6
	[4×4]	90sec	82.4
	[8×8]	36sec	82

[2×2] and [4×4] cell sizes also used for feature extraction. We noticed that with the cell size [8×8] recognition rate improved by 1% approximately as shown in Table 3. Size of the input image is 32×32, on that [8×8] cell working effectively. It also minimizes execution time as shown in Table 3. Linear classifier works well on the dataset comprising more feature dimensions. When HOG applied on images gives a greater number of feature dimensions and so that SVM+HOG combination achieved high recognition rate compared with other classifiers used. Especially for NN classifier database size should be large for better learning.

5. CONCLUSIONS

Recognition of handwritten Indian languages is very tedious and difficult task due to the different styles used by different peoples. Devanagari script is used as base for various Indian languages. The proposed offline handwritten Devanagari character recognition system recognizes isolated characters in Devanagari script. We used 10560 character images for experiment and observed effect of HOG on different classifiers. Our study reveals that:

- HOG gives better recognition rate on direct binary image as compared to the thinned image.
- [8×8] cell size gives better result with respect to time and accuracy as compared to [2×2] and [4×4] cell size.
- SVM+HOG give more accuracy as compared to K-NN+HOG and NN+HOG: a recognition rate of 87.38%.

In order to enhance the performance of the HDCRS model we need to increase the size of the database, work on prominent features and to find out different sources of errors. In future, we are planning to divide image into different zones. Then extracting features form each zone of the image using HOG will get a greater number of prominent features which can help us to achieve higher recognition accuracy. As well as to improve the accuracy of the similar shape characters shown in Figure 1(b) by adding structural features like concavity analysis which can eliminate some confusion among them wants to explore in our future work.

REFERENCES

[1] Pal, U., Roy, R., Kimura, F. (2009). Indian multiscript full pin-code string recognition for postal automation. 10th Int. Conf. Document Analysis and Recognition, Barcelona, Spain, pp. 456-460. http://dx.doi.org/10.1109/ICDAR.2009.171

[2] Romero, V., Serrano, N., Toselli, A.H., Sanchez, J.A., Vidal, E. (2011). Handwritten text recognition for historical documents. Proc. of Language Technologies for Digital Humanities and Cultural Heritage Workshop, Hissar, Bulgaria, pp. 90-96.

[3] Jayadevan, R., Kolhe, S., Patil, P., Pal, U. (2011). Offline recognition of Devanagari script: A survey. IEEE Transactions on Systems, Man, and Cybernetics Part c: Applications and Reviews, 41(6): 782-796. http://dx.doi.org/10.1109/TSMCC.2010.2095841

[4] Bristow, H., Lucey, S. (2014). Why do linear SVMs trained on HOG features perform so well? In: arXiv preprint arXiv:1406.2419.

[5] Hmeidi, I., Hawashin, B., El-Qawasmeh, E. (2008). Performance of KNN and SVM classifiers on full word Arabic articles. Advanced Engineering Informatics, 22(1):106-111. http://dx.doi.org/10.1016/j.aei.2007.12.001

[6] Deore, S., Ragha, L. (2011). Moment based online and offline handwritten character recognition. CiiT International Journal of Biometrics and Bioinformatics, 3(3): 111-115.

[7] Wan, C.H., Lee, L.H., Rajkumar, R., Isa, D. (2012). A hybrid text classification approach with low dependency on parameter by integrating K-nearest neighbor and support vector machine. Expert Systems with Applications, 39(15): 11880-11888. https://doi.org/10.1016/j.eswa.2012.02.068

[8] Kamble, P., Hegadi, R. (2015). Handwritten marathi character recognition using R-HOG feature. Procedia Computer Science, 45: 266-274. http://dx.doi.org/10.1016/j.procs.2015.03.137

[9] Minu, R.I., Nagarajan, G., Jacob, T., Pravin, A. (2018). BIP: A dimensionality reduction for image indexing. ICT Express, 5(3): 187-191.

[10] Anjali, E.P., James, A., Chandran, S. (2018). HOG feature-based recognition for Malayalam handwritten characters. In: Emerging Trends in Engineering, Science and Technology for Society, Energy and Environment, CRC Press, 823-828.

[11] Choudhury, A., Rana, H.S., Bhowmik, T. (2018). Handwritten Bengali numeral recognition using HOG based feature extraction algorithm. 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN), IEEE, pp. 687-690. http://dx.doi.org/10.1109/SPIN.2018.8474215

[12] Bannigidad, P., Gudada, C. (2019). Age-type identification and recognition of historical Kannada handwritten document images using HOG feature descriptors. In: Computing, Communication and Signal Processing, Springer, Singapore, pp. 1001-1010. http://dx.doi.org/10.1007/978-981-13-1513-8_101

[13] Nongmeikapam, K., Kumar, W.K., Meetei, O.N., Tuithung, T. (2019). Increasing the effectiveness of handwritten Manipuri Meetei-Mayek character recognition using multiple-HOG-feature descriptors. Sādhanā, 44(5): 104. https://doi.org/10.1007/s12046-019-1086-0

[14] Puri, S., Singh, S.P. (2018). Hindi text document classification system using SVM and fuzzy – a survey. International Journal of Rough Sets and Data Analysis, 5(4): 1-31. https://doi.org/10.4018/IJRSDA.2018100101

[15] Deore, S.P., Pravin, A. (2019). On-line devanagari handwritten character recognition using moments features. In: Santosh K., Hegadi R. (eds) Recent Trends in Image Processing and Pattern Recognition. RTIP2R 2018. Communications in Computer and Information Science, Springer, Singapore, 1037: 37-48. https://doi.org/10.1007/978-981-13-9187-3_4

[16] Deore, S., Pravin, A. (2017). Ensembling Model of histogram of oriented gradient based handwritten Devanagari character recognition system. Traitement du Signal, 34: 7-20. https://doi.org/10.3166/ts.34.7-20

[17] Complete tutorial on HOG with MATLAB, OpenCV code. Available: http://www.learnopencv.com.

[18] Zanchettin, C., Bezerra, B., Azevedo, W. (2012). A KNN-SVM hybrid model for cursive handwriting recognition. In: The 2012 Int. Joint Conf. on Neural Networks (IJCNN), Brisbane, QLD, pp. 1-8. http://dx.doi.org/10.1109/IJCNN.2012.6252719

[19] Acharya, S., Pant, A.K., Gyawali, P.K. (2016). Deep learning based large scale handwritten Devanagari character recognition. IEEE Proc. of International conference on Software, Knowledge, Information Management and Applications (SKIMA2015), Kathmandu, Nepal, pp. 1-6. http://dx.doi.org/1109/SKIMA.2015.7400041