J Inf Process Syst, Vol.13, No.1, pp.174~183, February 2017
https://doi.org/10.3745/JIPS.02.0012

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

A Contour Descriptors-Based Generalized Scheme for Handwritten Odia Numerals Recognition

Tusar Kanti Mishra*, Banshidhar Majhi*, and Ratnakar Dash*

Abstract

In this paper, we propose a novel feature for recognizing handwritten Odia numerals. By using polygonal approximation, each numeral is segmented into segments of equal pixel counts where the centroid of the character is kept as the origin. Three primitive contour features namely, distance (l), angle (θ), and arc-to-chord ratio (*r*), are extracted from these segments. These features are used in a neural classifier so that the numerals are recognized. Other existing features are also considered for being recognized in the neural classifier, in order to perform a comparative analysis. We carried out a simulation on a large data set and conducted a comparative analysis with other features with respect to recognition accuracy and time requirements. Furthermore, we also applied the feature to the numeral recognition of two other languages—Bangla and English. In general, we observed that our proposed contour features outperform other schemes.

Keywords

Contour Features, Handwritten Character, Neural Classifier, Numeral Recognition, OCR, Odia

1. Introduction

Handwritten optical character recognition (OCR) has numerous practical and commercial applications. Until now, it has been a challenging task to devise a perfect OCR for all languages. This is due to the presence of severe inter-variables existing between characters of different languages. One of the major complexities is the presence of intra-variability between writing styles among users. In India, 22 official languages with distinct characteristics are used indifferent states. Odia is one of the official languages of India that is used by 27 million people in the state of Odisha. There are 12 vowel characters and 25 consonants. Like any other language, Odia has ten numerals. The character set for the Odia language, including numerals, is given in Fig. 1.

Numerous feature extraction and selection methods have been reported in the literature on OCR [1]. The features like curvature [2,3], F-ratio [4], Zernike moments [5], discrete cosine transformation [6], and skeletal convexity [7] have been used for character recognition. In [8], the authors have used Gabor filters for extracting features from a gray-scale character image. Histogram features are extracted from the real part of the Gabor filter output which shows a satisfactory recognition performance of Chinese characters. A Bayesian decision-based neural network (BDNN) for a multi-linguistic handwritten

Manuscript received October 31, 2013; first revision March 14, 2014; accepted June 20, 2014; onlinefirst December 31, 2014.

Corresponding Author: Tusar Kanti Mishra (tusar.k.mishra@gmail.com)

* Dept. of Computer Science and Engineering, National Institute of Technology Rourkela, Rourkela, Odisha 769001, India (tusar.k.mishra@gmail.com, bmajhi@nitrkl.ac.in, ratnakar.dash@gmail.com)

character recognition scheme has been proposed in [9]. It takes into account the four directional strokes of a character. The prototype system demonstrates the successful utilization of the self-growing probabilistic decision-based neural network (SPDNN) for recognizing Chinese handwritten characters. In [10], character recognition for the Hindi language using an artificial neural network where they considered the histogram based on projection and pixel values in their features part has been proposed. A commercially viable Telugu character recognizer has been reported in [11]. Wavelet multi-resolution analysis and associative memory models have been used for feature extraction and recognition purposes, respectively. For accomplishing the learning task, a Hopfield-based dynamic neural network has been used. Printed Odia characters have been successfully recognized using a combination of stroke and run-number based features [12]. These features are obtained by using the concept of water overflow in a reservoir. However, no fine-tuning of their work has been carried out. The hidden Markov model (HMM) has been used for recognizing handwritten Odia numerals [13]. HMM is generated out of the shape primitives of an individual numeral and is referred to as a template for establishing a match for probe numerals. For this matching task, its class conditional probability is found by comparing it to each HMM.

It has been observed that most of the OCR schemes reported in other studies perform well for a particular language. However, these schemes then perform poorly for other languages. Hence, there exists a potential need to devise a general scheme for the recognition of the handwritten characters of multiple languages. Furthermore, for Odia handwritten character recognition, a fewer number of schemes are available in the literature on character recognition. In this paper, we propose a contour-based feature neural classifier (CFNC), which is a simple and effective scheme for Odia handwritten numeral recognition. Simulations have been carried out on three different languages namely, Odia, Bangla, and English handwritten numerals, in order to validate the suggested scheme.

The paper is organized as follows: Section 2 deals with the proposed scheme, which includes feature extraction, selection, and classification phases. Section 3 describes the simulation results that were performed on numerals from three different languages. Our performance comparison with two other existing schemes is given in Section 4. Finally, we provide our concluding remarks in Section 5.

Fig. 1. Character set for the Odia language.

2. Proposed Scheme

The proposed scheme, CFNC, involves preprocessing, feature extraction, and recognition phases and its overall block diagram is shown in Fig. 2. Each character is represented as a 30×30 (pixel) image. In the preprocessing phase, several tasks, such as noise removal, standardization, and normalization [14-16], are performed using standard schemes on the image prior to feature extraction. Our scheme focuses mostly on feature extraction and recognition. The character is represented by a one dimensional contour descriptor T= (t_1, t_2 . . ., t_m), which is an ordered set of m real valued variables that are taken clockwise from the contour [17, 18]. Polygonal approximation is applied to the contour descriptor to generate three basic feature vectors to represent the numeral image. Subsequently, these feature sets are used to classify the numeral using the back propagation neural network (BPNN). The feature extraction and recognition steps are discussed below sequentially.

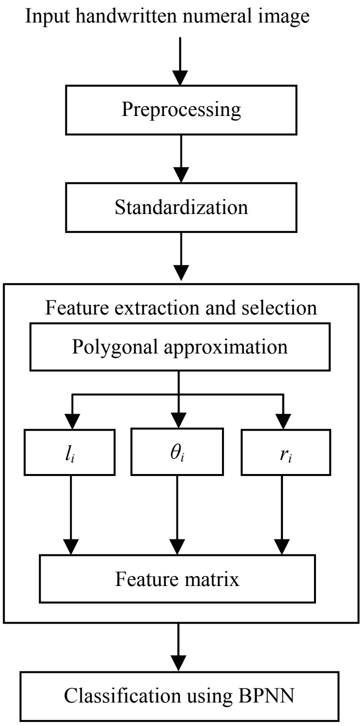

Fig. 2. Block diagram of the proposed scheme. BPNN=back propagation neural network.

2.1 Feature Extraction

The overall process of feature extraction has been described in Algorithm 1. The numeral images are subjected to pre-processing, such as binarization and a thinning operation, prior to the contour descriptor being extracted. Polygonal approximation has been conducted on the pre-processed image. By using polygonal approximation, each numeral is segmented into S different segments of equal pixel counts where the centroid of the character is kept as the origin. To help illustrate and clarify our point, we have shown the segments of the Odia number '3' in Fig. 3. Three primitive features, distance (l_i), angle (θ_i), and ratio (r_i), are extracted from these segments. The distance feature l_i represents the distance between the centroid and the starting point (p_i) of the i^{th} segment. The angle feature, θ_i, is the angle between l_i and chord v_i, which lies between the i^{th} and $(i + 1)^{th}$ segment, and the ratio, r_i, represents the ratio between the chord v_i and arc S_i. Thus, we have a $1 \times S$ dimension of each primary feature and if we arrange the three different features in rows, the feature vector f_v of dimension $3 \times S$ for each numeral is generated, which is given as:

$$f_v = \begin{pmatrix} l_1 & l_2 & \cdots l_i & l_{i+1} & l_s \\ \theta_1 & \theta_2 & \cdots \theta_i & \theta_{i+1} & \theta_s \\ r_1 & r_2 & \cdots r_i & r_{i+1} & r_s \end{pmatrix} \tag{1}$$

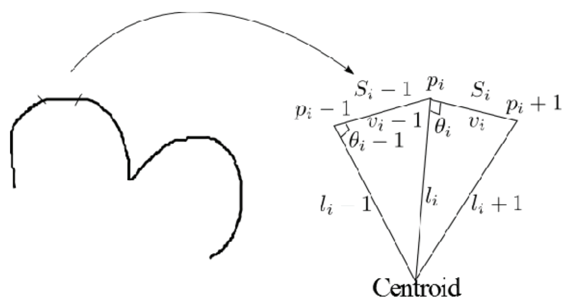

Fig. 3. Polygonal approximation of the Odia numeral '3'.

Algorithm 1. Feature_Extraction_and_Selection

Step 1. Initialize feature vector, $f_v \leftarrow$ NULL
Step 2. For each character image c_i from dataset D do
Step 3. Set counter, *cnt* $\leftarrow$1
Step 4. While *cnt* ≤ *S* ; where *S* is the number of segments
Step 5. Extract features l_{cnt}, θ_{cnt} and r_{cnt} per segment as:
 (a) $l_{cnt} \leftarrow$ length measure from centre of c_i to the segment
 (b) $\theta_{cnt} \leftarrow$ angle between corresponding chord and l_{cnt} at the current segment
 (c) $r_{cnt} \leftarrow$ ratio between current segment chord and corresponding arc
Step 6. Create column vector,

$$cl_{cnt} = \begin{pmatrix} l_{cnt} \\ \theta_{cnt} \\ r_{cnt} \end{pmatrix} \tag{2}$$

Step 7. $f_v \leftarrow$concatenate (f_v, cl_{cnt})
Step 8. Increment counter, *cnt* $\leftarrow$*cnt* + 1
Step 9. End While
Step 10. End For

These feature descriptors are unique to a particular character. For choosing the starting pixel in the profile, one can choose the farthest or nearest pixel from the centroid. If more than one point exists (e.g., with a circular character), then any of these two points can be selected. In our case, the leftmost isolated point on the contour has been chosen as the starting point for segmentation. The algorithm for feature point selection has a time complexity of $O(n_c)$ where n_c is the number of contour points. The selection of the number of segments is a heuristic choice, and it should be selected in such a manner where the average smoothness factor, r (ratio between chord and arc), should be above 75% to keep the quality of the perfect polygonal approximation contour description.

2.2. Recognition by Classification Using BPNN

The neural network has been successfully utilized for pattern classification and recognition [19-21]. For the purpose of recognition in our case, we used a feed forward neural structure for classifying the numerals. A set of feature vectors collected from numerals, along with a corresponding numeral as the target, constitute the training patterns. For training, the back propagation, which is based on conjugate gradient algorithm (CGA), has been used and the update equations are defined as:

$$E_{qw}(y) = E(w) + E^{'}(w)^T y + \frac{1}{2} y^T E^{''}(w) y \quad (3)$$

$$E'_{qw}(y) = E''(w)y + E'(w) \quad (4)$$

Where, $E(w)$ is the error function, E_{qw} andis the quadratic approximation to in a neighborhood of point 'w'. The set of steps involved in carrying out the conjugate gradient (CG) is listed in Algorithm 2. Using a step size scaling mechanism makes it possible to avoid a time-consuming line search in every iterations of the training and it achieves a faster convergence.

Algorithm 2. Conjugate_Gradient_Algorithm (CGA)

Step 1. Chose initial weight vector, w_1

Step 2. Set $p_1 = r_1 = -E(w_1)$, $k = 1$

Step 3. Calculate second order information: $E^H(w_k)p_k$ and $d_k = p'_k(s_k)$

Step 4. Calculate step size $\mu_k = p'_k r_k$ and $\alpha_k = {}^{\mu_k}/_{d_k}$

Step 5. Update weight vector $w_{k+1} = w_k + \alpha_k p_k$ and $r_{k+1} = E'(w_{k+1})$

Step 6. If $k \bmod m = 0$ then restart algorithm: $p_{k+1} = r_{k+1}$, where m is the number of weights

Step 7. Else create new conjugate direction: $\beta_k = \frac{r_{k+1}^2 - r'_{k+1} r_k}{\mu_k}$

Step 8. $p_{k+1} = r_{k+1} + \beta_k p_k$

Step 9. If the steepest descent direction $r_k \neq 0$ then set $k = k + 1$ and go to step-2

Step 10. Else return w_{k+1} as desired minimum and terminate

3. Experimental Evaluation

To validate our proposed CFNC scheme, simulations were carried out with handwritten numerals from the Odia, Bangla, and English languages. The overall simulation was divided into two different experiments to study the various aspects of the CFNC scheme. The experiments are discussed below in detail.

3.1. Experiment 1: Study of Training Convergence Characteristics

Handwritten Odia numerals were collected from various individuals at different times and a dataset size of 3,000 [22] was created. Each numeral has been represented as a 30×30 (pixel) image. The feature vector of f_v, for each numeral was extracted by using Algorithm 1. A total of 34 segments were chosen for each numeral and hence, the feature vector is a 3×34 matrix of distance (l_i), angle (θ_i), and ratio (r_i) as row vectors, respectively. For experimental evaluation, a set of 200 samples was randomly selected and the training patterns$(f_v)_j: t_j,\ j = 1,2, \ldots 200)$ are were used for training. The neural network structure with the node specification of 102-30-10 is shown in Fig. 4. To enhance the reliability and faster convergence of back propagation training, a conjugate gradient was used. The convergence characteristic is shown in Fig. 5.

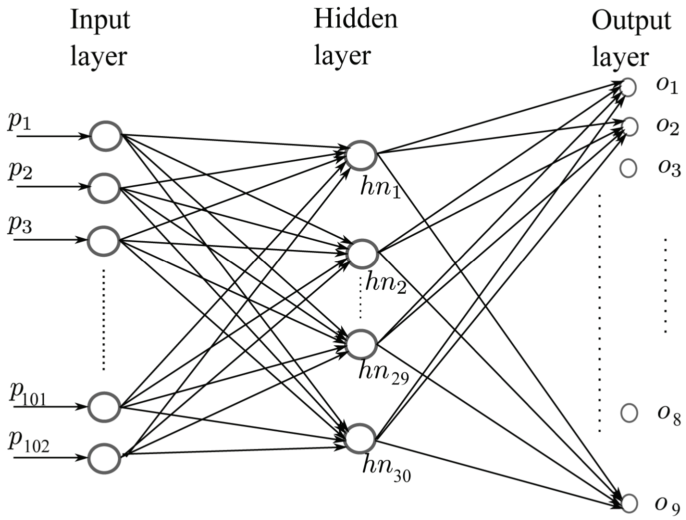

Fig. 4. Structure of the neural classifier.

Fig. 5. Convergence characteristics of back propagation neural network (BPPN) training using Odia numerals. CG=conjugate gradient.

3.2. Experiment 2: Performance Analysis on Odia Numerals

Two hundred handwritten numerals were randomly selected from the dataset for each numeral that was not used during training of the classifier. This was done in order to evaluate the performance of the proposed feature along with the neural classifier. An accuracy comparison of the proposed scheme was carried out with state-of-the-art approaches like Zernike moments, the curvature feature, and discrete cosine transformation (DCT) features, as shown in Table 1. It has been observed that for all numerals, the proposed CFNC scheme outperforms other schemes.

To show the impact of illumination variance on numerals, samples of unequal illuminations were considered and the output results for the Odia number '2' are shown in Fig. 6. It was observed that even if there is a change in illumination, the handwritten numeral '2' has been correctly recognized. However, a misclassification had also been observed. For example, the numeral '6' was recognized as '7,' due the similarity between the two numerals, as shown in Fig. 7.

The experimental evaluation was then extended to the numerals of two other languages, Bangla and English, for which corresponding data samples were collected from [22] and [23], respectively. Handwritten numerals from both of the languages were used with different features and two neural classifiers were devised for recognizing the numerals in these two languages. The overall comparative analysis for Odia, Bangla, and English is shown in Table 2. Furthermore, the time requirement for classifying the test dataset was also computed for all of the features.

Table 1. Performance comparison for handwritten Odia numerals on 200 samples

Digit/class	Scheme	Accuracy (%)
୦	CFNC	98.5
	Zernike	91
	DCT	94
	Curvature	91
୧	CFNC	96
	Zernike	92
	DCT	93.5
	Curvature	90
୨	CFNC	97.5
	Zernike	92.5
	DCT	95.5
	Curvature	90
୩	CFNC	97.5
	Zernike	95
	DCT	95
	Curvature	94
୪	CFNC	98
	Zernike	96.5
	DCT	97.5
	Curvature	92.5
୫	CFNC	98.5
	Zernike	95.5
	DCT	96.5
	Curvature	95
୬	CFNC	89
	Zernike	88
	DCT	87
	Curvature	89.5
୭	CFNC	93
	Zernike	92.5
	DCT	94
	Curvature	94
୮	CFNC	97
	Zernike	96
	DCT	97.5
	Curvature	92
୯	CFNC	98
	Zernike	96
	DCT	97.5
	Curvature	92

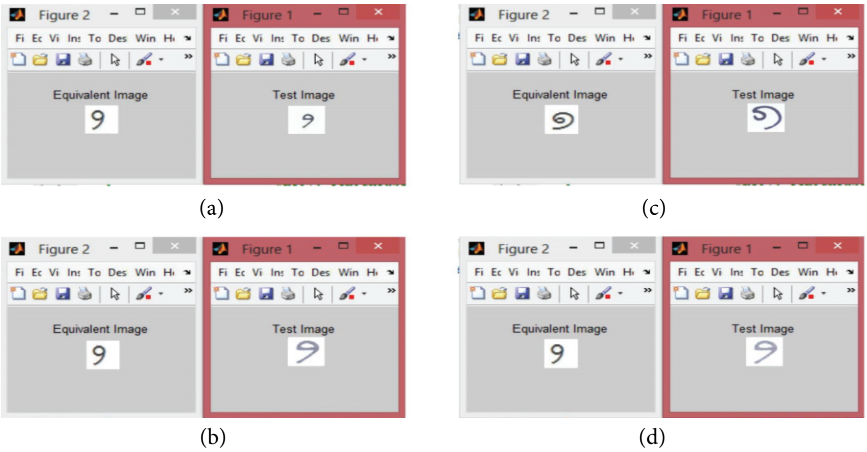

Fig. 6. Impact of illumination variation on numerals.

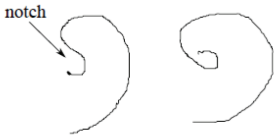

Fig. 7. Similarity between the '6' and '7' of the Odia language.

Table 2. Overall performance comparison

Language/ feature	Training samples	Testing samples	DCT		Zernike moments		Curvature feature		Contour descriptor	
			Time (sec)	Accuracy (%)	Time (sec)	Accuracy (%)	Time (sec)	Accuracy (%)	Time (sec)	Accuracy (%)
Odia	1,000	2,000	1,640	92	275	93.5	412	94.8	70	96.25
Bangla	500	1500	1,320	87.66	244	92	405	93.3	58	94.22
English	1,000	2,000	1,628	91.5	270	92.2	410	94	65	95

It has been observed that the proposed contour feature has outperformed three state-of-the-art features (Table 2) with respect to time requirement and accuracy. Hence, our proposed feature can be considered as an alternative for the numeral recognition of any language. The overall accuracy rates for recognizing Odia numerals when using CFNC, Zernike moments, and the curvature feature were found to be 96.25%, 93.5%, and 94.8%, respectively.

5. Conclusion

In this paper, we propose a novel feature for recognizing the numerals of any language. In particular, more focus was placed on recognizing Odia numerals. Three primary features of length, angle, and ratio, were extracted from segments that had been derived from the contour of a numeral. The contour was further decided based on the polygonal approximation on a numeral image. These features were then used in a neural classifier to recognize the numerals. Other existing features were also considered for recognition in the neural classifier, in order to perform a comparative analysis. In general, it has been observed from our exhaustive simulation study that the proposed feature outperforms other features with respect to recognition accuracy and time requirements. Furthermore, the feature was also applied to the numeral recognition of two other languages—Bangla and English. Our proposed feature also demonstrated a similar performance with these two languages. Hence, the proposed feature can be considered as a generalized alternative for the numeral recognition of any language. The scheme can also be extended for conducting the recognition of any character.

References

[1] U. Pal, R. Jayadevan, and N. Sharma, "Handwriting recognition in Indian regional scripts: a survey of offline techniques," *ACM Transactions on Asian Language Information Processing*, vol. 11, no. 1, pp. 1-35, 2012.

[2] B. Gatos, N. Papamarkos, and C. Chamzas, "Using curvature features in a multiclassifier OCR system," *Engineering Applications of Artificial Intelligence*, vol. 10, no. 2, pp. 213-224, 1997.

[3] U. Pal, T. Wakabayashi, and F. Kimura, "A system for off-line Oriya handwritten character recognition using curvature feature," in *Proceedings of the 10th International Conference on Information Technology(ICIT2007)*, Rourkela, India, 2007, pp.227-229.

[4] T. Wakabayashi, U. Pal, F. Kimura, and Y. Miyake, "F-ratio based weighted feature extraction for similar shape character recognition," in *Proceedings of the 10th International Conference on Document Analysis and Recognition (ICDAR2009)*, Barcelona, Spain, 2009, pp. 196-200.

[5] A. Khotanzad and Y. H. Hong, "Invariant image recognition by Zernike moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 489-497, 1990.

[6] T. K. Mishra, B. Majhi, and S. Panda, "A comparative analysis of image transformations for handwritten Odia numeral recognition," in *Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI2013)*, Mysore, India, 2013, pp. 790-793.

[7] S. Bag, P. Bhowmick, and G. Harit, "Recognition of Bengali handwritten characters using skeletal convexity and dynamic programming," in *Proceedings of the 2nd International Conference on Emerging Applications of Information Technology (EAIT2011)*, Kolkata, India, 2011, pp. 265-268.

[8] X. Wang, X. Ding, and C. Liu, "Gabor filters-based feature extraction for character recognition," *Pattern Recognition*, vol. 38, no. 3, pp. 369-379, 2005.

[9] H.C. Fu and Y. Y. Xu, "Multilinguistic handwritten character recognition by Bayesian decision-based neural networks," *IEEE Transactions on Signal Processing*, vol. 46, no. 10, pp. 2781-2789, 1998.

[10] D. Yadav, S. Sanchez-Cuadrado, and J. Morato, "Optical character recognition for Hindi language using a neural-network approach," *Journal of Information Processing Systems*, vol. 9, no. 1, pp. 117-140, 2013.

[11] A. K. Pujari, C. D. Naidu, M. S. Rao, and B. C. Jinaga, "An intelligent character recognizer for Telugu scripts using multiresolution analysis and associative memory," *Image and Vision Computing*, vol. 22, no. 14, pp. 1221-1227, 2004.

[12] B. B. Chaudhuri, U. Pal, and M. Mitra, "Automatic recognition of printed Oriya script," *Sadhana*, vol. 27, no. 1, pp. 23-34, 2002.

[13] T. K. Bhowmik, S. K. Parui, U. Bhattacharya, and B. Shaw, "An HMM based recognition scheme for handwritten Oriya numerals," in *Proceedings of the 9th International Conference on Information Technology (ICIT2006)*, Bhubaneswar, India, 2006, pp. 105-110.

[14] W. Bieniecki, S. Grabowski, and W. Rozenberg, "Image preprocessing for improving OCR accuracy," in *Proceedings of the International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH2007)*, Lviv-Polyana, Ukraine, 2007, pp. 75-80.

[15] C. C. Fung and R. Chamchong, "A review of evaluation of optimal binarization technique for character segmentation in historical manuscripts," in *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (WKDD2010)*, Phuket, Thailand, 2010, pp. 236-240.
[16] H. P. Le and G. Lee, "Noise removal from binarized text images," in *Proceedings of the 2nd International Conference on Computer and Automation Engineering (ICCAE2010)*, Singapore, 2010, pp. 586-589.
[17] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: experimental comparison of representations and distance measures," *Proceedings of the VLDB Endowment*, vol. 1, no. 2, pp. 1542-1552, 2008.
[18] C. Paul, H. Brent, and A. Niall, "An unsupervised algorithm for segmenting categorical timeseries into episodes," in *Working Notes of the 2002 ESF Exploratory Workshop on Pattern Detection and Discovery in Data Mining*, 2002.
[19] M. F. Moller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, vol. 6, no. 4, pp. 525-533, 1993.
[20] J. V. Stone and R. Lister, "On the relative time complexities of standard and conjugate gradient back-propagation," in *Proceedings of the IEEE International Conference on Neural Networks*, Orlando, FL, 1994, pp. 84-87.
[21] H. B. Kim, S. H. Jung, T. G. Kim, and K. H. Park, "Fast learning method for back-propagation neural network by evolutionary adaptation of learning rates," *Neurocomputing*, vol. 11, no. 1, pp. 101-106, 1996.
[22] www.isical.ac.in/~ujjwal/download/database.html.
[23] Y. LeCun, C. Cortes, and C. J. C. Burges, "The MNIST database of handwritten digits," http://yann.lecun.com/exdb/mnist.

Tusar Kanti Mishra http://orcid.org/0000-0002-4665-6625

He received his B.Tech degree with honours in Computer Science and Engineering from Biju Patnaik University of Technology, Odisha, India in 2004. He received his M.Tech degree in 2011 and is currently a Ph.D. research scholar in the Department of Computer Science and Engineering at the National Institute of Technology Rourkela, India. His research interests are in the areas of computer vision, image processing and pattern recognition.

Banshidhar Majhi http://orcid.org/0000-0002-2843-1908

He is presently working as a professor in the Department of Computer Science and Engineering, NIT Rourkela. He has 24 years of teaching and research experience and 3 years of industry experience. His research interests include image processing, computer vision, cryptographic protocols and iris biometrics. He has guided 10 doctora lworks and published 45 articled in referred journals.

Ratnakar Dash

He earned his B.Tech degree from National Institute of Science and Technology, India, in 2002 and M.Tech degree from University College of Engineering, India in 2005. He is currently pursuing his Ph.D. degree in image processing at National Institute of Technology Rourkela, India. His research interest also includes digital signal processing and communication.