IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-6, June 2018 E-ISSN: 2347-2693

Robust Offline Gurmukhi Handwritten Character Recognition using Multilayer Histogram Oriented Gradient Features

M. K. Mahto[1*], K. Bhatia[2], R. K. Sharma[3]

[1] Department of Computer Science, Gurukula Kangri Vishwavidyalaya, Haridwar, INDIA
[2] Department of Computer Science, Gurukula Kangri Vishwavidyalaya, Haridwar, INDIA
[3] Thapar Institute of Engineering and Technology, Patiala, INDIA

**Corresponding Author: manojkr.bit@gmail.com*

Available online at: www.ijcseonline.org

Accept: 19/Jun/2018, Published: 30/Jun/2018

***Abstract* -** Recognizing offline handwritten characters is a challenging problem and is considered to be more significant than the recognition of on-line handwritten characters. This study is undertaken to resolve the issue of offline handwritten character recognition for Gurmukhi script, one of the prominent scripts in the northern part of India. The Gurmukhi character images are presented using a single layer as well as multi-layer histogram of gradient features. Once the character images are represented using these features, one-against-all classification strategy, implemented through Support Vector Machine and k-Nearest Neighbour classifiers, is employed to recognize these characters. A dataset of 3500 handwritten Gurmukhi characters written by different writers is created and the scope of the Histogram Oriented Gradient (HOG) and Pyramid Histogram Oriented Gradient (PHOG) features is explored for the recognition of offline handwritten Gurmukhi characters. The simulation study reveals character recognition accuracy of 99.1% with SVM classifier for PHOG feature. The technique is robust to inter-class and intra-class variations present in the Gurmukhi script and has significant scope of application to the recognition of other scripts too.

***Keywords* -** Offline Gurmukhi Handwritten Character Recognition, Optical Character Recognition, Histogram Oriented Gradient Feature, Pyramid Histogram Oriented Gradient Feature, Support Vector Machine Classifier, k-NN Classifier

I. INTRODUCTION

Handwriting is an age-old and elegant way of communication and is a distinctive characteristic of an individual. The writing style of a person depends on several factors such as the writing medium, state of mind, environment, mood of the individual, etc. Obtaining high accuracy in offline handwritten character recognition is a challenging task. Several factors like background noise, different writing style of the writers, variations in character size, pen width, pen ink, character spacing, skew and slant, similarity of some characters in shape and size, influence the character recognition rate. Other significant factors, for instance, the absence of header line caused by the writer or segmentation of modifiers and touching characters also become significant in desing an efficient offline handwritten character recognition system. Generally, offline handwritten recognition is the scanned pictures of pre-written textual material on paper and online handwritten identification are the throughout writing activity on a specially designed pen in an electronic device. Recognition of offline handwritten documents has been a vital field for research in broad domain of the pattern recognition. Over the past few years, various laboratories all over the world showed their intense involvement studies on handwriting recognition [1].

Offline handwriting recognition (OHR) embraces the transformation of an image comprising a handwritten text into a format which is understandable by the computer. The text on the image is considered as an unchanging depiction of handwriting [2]. The literature has reported different explanations for recognition of characters in Indian calligraphy. It is necessary to mention that in the context of Indian scripts' character recognition, most of the published works deal with pre-printed papers but there are only rare research papers representing handwritten character recognition.

In recent past, several contemporary surveys of the existing techniques in the field of OHR in Indian regional calligraphy were performed to support the researchers in the subcontinent, and hence a straight attempt is made in this research to discuss its advancement. Handwriting recognition is a growing optical character recognition (OCR) technology that has to be exceptionally robust and adaptive. OHR is a widely researched and a well-known problem which has been studied to a certain extent and utilized to solve problems for a few applications, such as handwriting recognition in personal checks, recognition of hand-printed text in application forms, postal envelope, and parcel address readers [3]. Handwritten character recognition is a standout amongst the most inspiring and appealing territories of pattern recognition [4], [5]. It is characterized as the way towards detecting segments, distinguishing characters and symbols from the scanned image.

In this process, characters from the input images are identified and converted into UNICODE or similar machine-editable text [6], [7], [8]. Character recognition contributes incredibly towards the advance of automation method and enhances the interface amongst the human and machine in several applications.

For last few decades, character recognition is getting additional importance because of the broad domain covering applications. Explicit identification of handwritten typescripts along with their transformation into device editable forms is a crucial and hard research area within the domain of pattern recognition. It builds up the need to protect original manuscripts and records of ancient significance. The steps involved in developing a recognition system are: (i) Image acquisition, (ii) Pre-processing, (iii) Segmentation, (iv) Feature Extraction, (v) Classification, and (vi) Post-processing [9].

This article is structured in seven sections. An exhaustive view of work pertaining to the handwritten character recognition and classification emphasizing on Indian scripts is given in Section 2. Section 3 presents the collection of the dataset used for the study. Section 4 illustrates the proposed work on Offline Gurmukhi Handwritten Character Recognition system. Section 5 deals with the implementation details and results obtained through simulation. Section 6 includes the concluding remarks.

II. RELATED WORK

The inherent inconsistency found in the writing style of an individual is a major challenge in recognition of the handwritten characters. This section presents an extensive literature survey of character recognition methods. In recent past, several methods such as loop aspect ratio [10], pixel distribution and upper lower profile, width, height, area, density, aspect ratio and separator length between two successive connected characters [11], bi-level co-occurrence [12], run-length histogram [13], connected component profiles, analysis [14], Gabor filters [15], word physical sizes, crossing count histogram, baseline profile features, overlapping areas [16] and steerable pyramids transform [17] have been used to determine statistical features. The character recognition assignment has been a testing venture through different methodologies for instance template matching, statistical procedures, Hidden Markov Models (HMMs), Bayesian, Neural Network and so forth. [17] reported a method to recognize the characters of Devanagari script in two steps. In the first step strokes are identified while in a second step the character recognition takes place based on the strokes identified in the preliminary step. [18] have proposed Bangla handwritten character recognition in the multi-stage classifier. Here, the first stage consisted of the extraction of high level attributes and core level classification, and further the extraction of low level features assisted the final ranking in the second stage. [19] presented Generalized Hausdorff Image Comparison (GHIC) system for Devenagari script recognition, which is a form of template matching method for overcoming certain disadvantages of the traditional template matching approach. [20] have proposed to recognize Bangla numerals for sorting the postal documents for the Indian postal automation system using a two-stage Multilayer Perceptron (MLP) based classifier and Modified Quadratic Discriminative Function (MQDF) classification for Bangla handwritten word recognition. They considered contour pixels in a character image and obtained a histogram based features in a directional chain code and demonstrated applicability of their work in the Indian Postal Automation context. [21] have implemented PCA for Tamil online handwritten character recognition. The quadratic interpolation features based classifier is proposed by [22] for offline handwritten Gurmukhi characters and numeral recognition. They emphasized on the contour points of the characters for extracting the features directional chain code information. A number of blocks were created for separating the whole set of characters and then chain code histogram was produced for every block. A quadratic interpolation based classifier employing 64-dimensional attribute along with chain code features have been used for classification. A MLP classifier for Bangla handwritten character recognition is proposed by [23], features are calculated by computing local chain-code histograms of input character images and computed for the contour as the representation of the skeletal input character images. [24] have proposed a string matching algorithm for improving the performance of the recognition system. In [25], authors have proposed MQDF classifier for offline Bangle handwritten compound character recognition system. In this system, the features used were primarily lies on directional information bestowed from the arc tangent of the gradient. [26] presented elastic matching technique in two stages for the recognition of online handwritten Gurmukhi script. For designing handwritten Devanagari OCR, [27] have employed feed-forward MLP in one hidden layer and trained in back propagation algorithm in neural network and Gradient features. [Sharma and Jhajj 2010] have suggested a handwritten Gurmukhi character recognition system using zoning density based features. [28] have employed multi-layer perceptron (MLP) and SVM classifier for Bangla handwritten core and compound character recognition and Confusion matrix has been computed to recognize the results of the MLP classifier. The combination of different features and classifier have been presented by [29] for offline handwritten Gurmukhi character recognition. [30] presented recognition of Devanagari numerals utilizing deep learning of Artificial neural networks in Histogram of Oriented Gradient (HOG) features. [31] proposed Fringe Distance Map, histogram of oriented and shape descriptor to identify nearly 250 classes including middle, upper and lower zone symbols as well as conjunct characters, achieved an accuracy of approximately 98% on 16,000 symbols database using multiple kernel learning-based SVM classifier.

III. DATA COLLECTION

The Gurmukhi script is utilized to compose the Punjabi dialect in the Indian province of Punjab. Gurmukhi script formulated by Guru Nanak Dev, the first Sikh guru amid the sixteenth century and promoted by Guru Angad Dev, the second Sikh guru. The word ‘Gurmukhi’ implies ‘from the mouth of Guru’. Guru Angad Dev expertly improvised, rearranged and recorded the letters of Gurmukhi script. Figure 1 depicts the character set of Gurmukhi script.

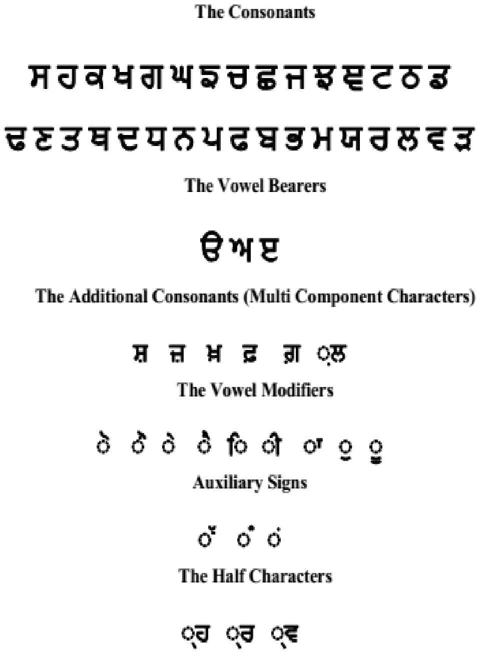
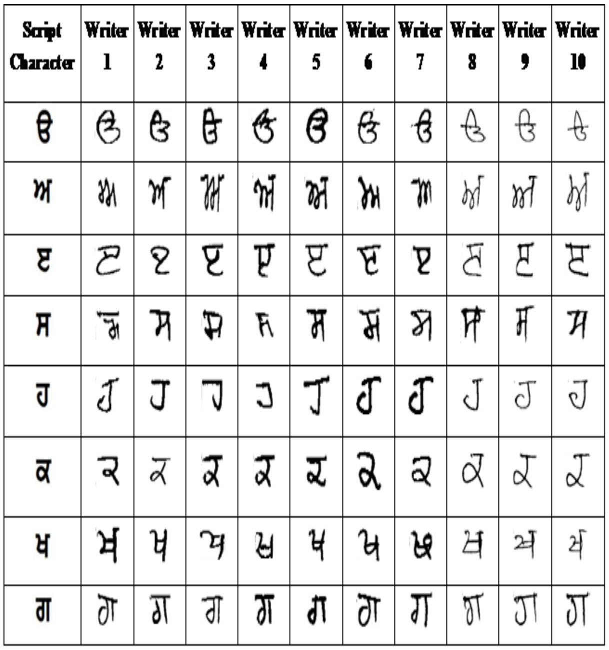


Figure 1. Gurmukhi character set

The Gurmukhi script contains 56 characters in all, including thirty two consonants along with six additional consonants, three vowel bearers, nine vowel modifiers, three auxiliary signs and three half characters [5]. The Gurmukhi script follows the writing style from top to bottom and left to right [6]. The clause of case sensitivity is absent in the Gurmukhi script. Another salient feature of this script is that a horizontal line, known as the headline, connects majority of the characters on their top. The individual characters are connected with one after one through this headline signifying a complete word. A word in this script is segmented into three major zones called as upper, middle and lower zone.

The present work deals with 3,500 samples of discrete offline Gurmukhi handwritten characters that have been collected from 10 distinct writers belonging to different social and economic background. They contributed thirty-two consonants along with three vowel bearers of Gurmukhi characters ten times each on a plain white paper using a pen in different states of mind and mood. There are eight handwritten characters sample of Ten writers (Writer 1, Writer 2… Writer 10) are depicted in Table 1.

Table 1: Sample Dataset



Script Character	Writer 1	Writer 2	Writer 3	Writer 4	Writer 5	Writer 6	Writer 7	Writer 8	Writer 9	Writer 10
ੳ	ੳ	ੳ	ੳ	ੳ	ੳ	ੳ	ੳ	ੳ	ੳ	ੳ
ਅ	ਅ	ਅ	ਅ	ਅ	ਅ	ਅ	ਅ	ਅ	ਅ	ਅ
ੲ	ੲ	ੲ	ੲ	ੲ	ੲ	ੲ	ੲ	ੲ	ੲ	ੲ
ਸ	ਸ	ਸ	ਸ	ਸ	ਸ	ਸ	ਸ	ਸ	ਸ	ਸ
ਹ	ਹ	ਹ	ਹ	ਹ	ਹ	ਹ	ਹ	ਹ	ਹ	ਹ
ਕ	ਕ	ਕ	ਕ	ਕ	ਕ	ਕ	ਕ	ਕ	ਕ	ਕ
ਖ	ਖ	ਖ	ਖ	ਖ	ਖ	ਖ	ਖ	ਖ	ਖ	ਖ
ਗ	ਗ	ਗ	ਗ	ਗ	ਗ	ਗ	ਗ	ਗ	ਗ	ਗ

IV. PROPOSED WORK

An offline handwritten character recognition system has five phases namely image acquisition, preprocessing, segmentation, feature extraction and classification. In the present research for Gurmukhi handwritten character recognition the work is confined to isolated characters only thus eliminating the need of segmentation phase. In view of this, phases of proposed method are shown in Figure 2.

1. *Image Acquisition*

This is the preliminary phase of an off-line handwritten recognition system. This study is undertaken by investigating the samples of handwritten Gurmukhi characters that are collected from 10 different writers. Each writer has contributed ten samples of each character on a plain white A4 sized paper. In this proposed system the total amount of 3500 specimen sample images (100 specimen sample images for each character) that has been applied in recognition system. These specimen samples were generated from HP scan-jet scanner at 300 dpi in bitmap format as given in Table 1.

2. *Preprocessing*

The preprocessing stage is the process of extraction of character from an image or documents and also include preparing a raw data to make a usable data like: noise reduction, degraded image restoration, filtering, etc. The performance of handwritten character recognition relies on character shape normalisation for regulating the size, position and shape of character images. Further, normalization also helps in reducing the variation in shapes of images belonging to same class. The restoration of deformed characters is also considered as a vital task in preprocessing.

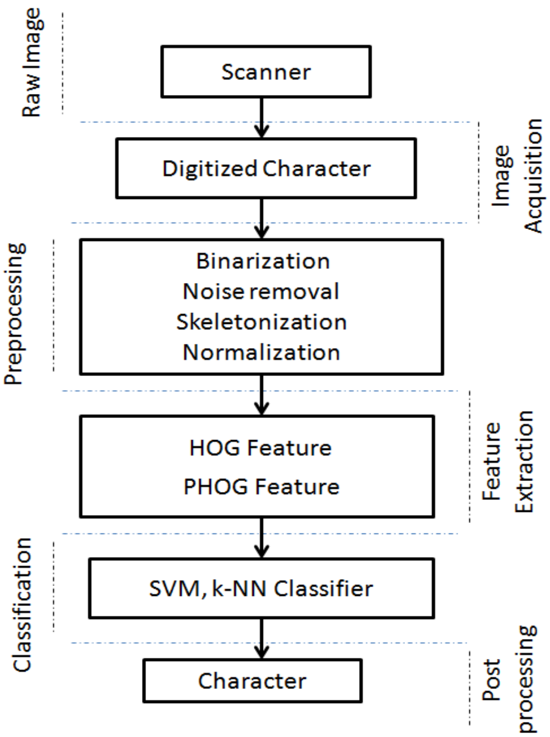
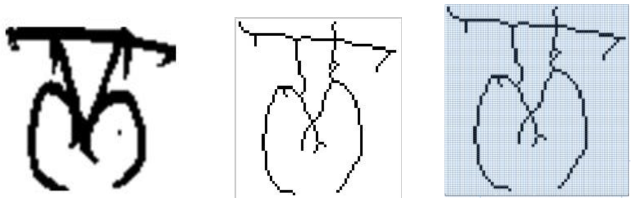


Figure 2: Phases of proposed method

In proposed system, preprocessing steps given underneath are followed:

- The scanned image is transformed to a grayscale image.
- Then, the grayscale image so achieved is converted into a binary image [Figure 3a], which in turn is transformed into a skeletonized binary image [Figure 3b]. (The structural shape of a character is known as scrutinizing)
- Next, in the normalization step images with 64×64 size are obtained from the dataset images of step (2) [Figure 3c].



(a) Original Character (b) Skelton Character (c) Normalized character

Figure 3: Pre-processing steps to filter the samples

3. Segmentation

In the segmentation phase the image document is split into classifiable objects – the individual modifier(s) or character(s). This is the third phase of the complete process. Segmentation usually involves line, word, character segmentation. The horizontal segmentation is separation of the lower and upper modifier(s) of the input documents. As mention earlier, the present character recognition work is limited to isolated characters only thus eliminating the need of segmentation phase.

4. Feature Extraction

In this phase, each and every character is portrayed as a group of feature vector which is the identification of the character. The fundamental role of the feature extraction phase is to identify a group of features so that the rate of character recognition is maximized. Histogram of Oriented Gradients (HOG) [32] and Pyramid Histogram of Oriented Gradients (PHOG) [34] feature extraction techniques are explored in this work.

The count of intensity values available in an image is known as histogram of a digital image representation. In projection histogram, foreground and/ or background pixels in each column and row, present in a specified direction, are computed to separate two or more resembling character(s) such as ਙ (yanza) and ਵ (vava) [9].

A. Histogram Oriented Gradient

The purpose behind HOG feature [32] is to represent an image by a localized distribution of intensity gradients. Thus, HOG features suit well for the resolution of handwriting recognition, as the intensity gradients in an image of a handwritten character will be due to the strokes. For the implementation of HOG, an image is sliced into smaller linked sections known as cells. For every cell, a histogram of oriented gradients is formed by compiling the gradients of each pixel in a meticulous cell; these angles are utilized to vote in an edge orientation bin. As shown in Figure 4, initially, the gradient magnitude is calculated and directions at every pixel were determined, and will consistently divide each ROI into 3 × 3 cells. In addition, the gradient orientations are quantized into six bins. For each cell, the investigators evaluated the gradient orientation histogram based on the gradient magnitude. The handwritten character image is sliced into tiny regions to compute the HOG descriptor [33]. In the present study, the horizontal and vertical gradient components G_x and G_y, respectively, are computed for each pixel (x, y) of the handwritten character image by using a simple kernel [-1, 0, +1] which acts as a gradient detector.

Let $f(x, y)$ be the value of intensity at coordinates (x, y) of the given image, then the derivatives in x-direction and y-direction are computed as follows:

$$G_x f(x,y) = f(x+1,y) - f(x,y) \qquad (1)$$

$$G_y f(x,y) = f(x,y+1) - f(x,y) \qquad (2)$$

From the calculated derivatives, the magnitude M and the orientation θ of gradient can be easily deduced by relations as given under:

$$M(x,y) = \sqrt{G_x^2 + G_y^2} \qquad (3)$$

$$\theta(x,y) = \tan^{-1}\frac{G_y}{G_x} \qquad (4)$$

Further, the image gradient orientation histograms contained by each block are weighted into a particular direction of n bin. The L1-norm is used for normalization of the HOG descriptor obtained by conglomeration of all the blocks in the input character image. The non-overlapping rectangular HOG (R-HOG) is used in the present.

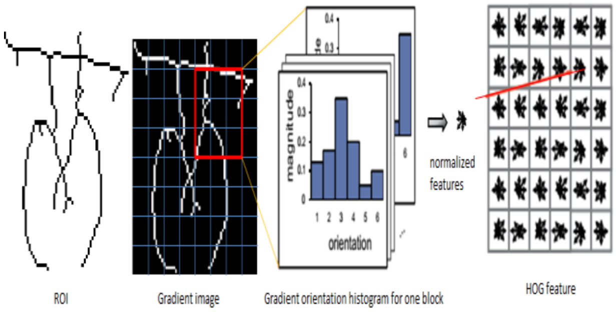


Figure 4: Calculation of HOG features

Compared to all cells in the base layer of the pyramid, cells in the top layer contain fewer pixels which cause advanced description. To amplify the calculation speed of HOG vectors, an essential image is applied to calculate the gradient orientation histograms as in [34]. For each bin, an integral image of gradient magnitude is calculated.

B. *Pyramid Histogram Oriented Gradient*

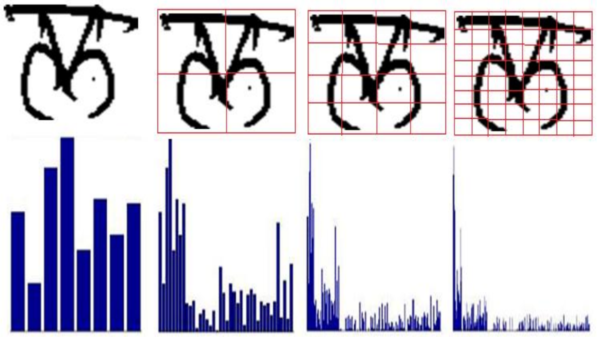
Given an image, the Pyramid Histogram Oriented Gradient (PHOG) is computed as follows:

- The image is divided into pyramids of L labels,
- At each label of pyramid, image is further divided into 4^l, where l is the label number, blocks similar to [35].
- For each block histogram of oriented gradient features of bin size n is computed, and all the features computed at different blocks and levels are concatenated to obtain final descriptor.

Mathematically, the dimensionality D of the PHOG feature vector, for L pyramids and n bin size, is given by:

$$D = n \times \sum_{l=0}^{L} 4^l \qquad (5)$$

In this experiments, value of n is taken to be 8 whereas L is set to value 3. This results in a PHOG descriptor of length 680. It is worthwhile to mention that PHOG has shown compelling results on various computer vision tasks, including retrieval of historical Chinese architectural image [36], object classification [35], and smile identification [37].



(a) Level=0 (b) Level=1 (c) Level=2 (d) Level=3

Figure 5: A schematic illustration of pyramid histograms of gradient orientation (PHOG) at each resolution of level 0 to 3

5. *Classification*

Similar to feature sets, classifiers also play a significant role in OCR systems. In the present work, the efficacy of two classifiers - Support Vector Machine (SVM) and k-Nearest Neighbour (k-NN) classifiers is explored.

A. *Support Vector Machine*

Support Vector Machine has been proved a successful statistical learning tool with a wide range of applications in data mining and pattern classification tasks [38], [39]. Support vector finds an optimal hyper-plane by maximizing the margin of separation between the positive and negative examples. In this experiment, the dataset consists of 35 Gurmukhi alphabetic character classes. The one-vs-all binary SVMs are trained for each class of characters vs rest. Thus, when SVM classifier is trained for a particular (character) class, say ੳ (Urha) then training samples in this particular class are considered as positive samples whereas those in the rest 34 classes are considered as negative samples. When this binary SVM classifier is fed a new test (character) sample, then it becomes necessary to obtain a class likelihood that is associated with this character class. By testing against all 35 SVM classifiers, the class likelihood related to each of these 35-character classes are determined, and the highest class likelihood is taken as the character likelihood.

In this experiment, libSVM [40] implementation for learning binary SVM classifiers is used. The parameters of the SVM (e.g., C and degree of polynomial in case of polynomial kernel) are fixed using cross-validation on independent validation sets.

B. k-Nearest Neighbour

The k-NN is a non-parametric distance-based classifier. Given a test sample, the distance between all the training samples and a test sample is computed. Once these distances are computed, the k nearest samples are considered for classification decision. The majority class of these k samples is assigned to the test sample. Despite a simple algorithm the k-NN works reasonably well for many classification task. There are many variants of k-NN in literature, e.g. in [41], author use it for gradient, structural and concavity features and minimum distance classifier (MCD) is proposed in [42]. We use k-NN as one of our classifier in our experiments, and evaluate our method with different values of the parameter k.

C. s-Fold Cross Validation

The standard cross validation scheme, namely s-fold cross-validation is used in this experiment. It randomly splits N data samples *X* into *s* mutually exclusive subsets (also known as folds), say, X_1, X_2, ..., X_s having an approximate equal cardinality [43]. The one out of s subsets is used as the test set, and the training is realized with rest (*s*−1) subsets. The cross validation is performed *s* number of times, selecting each subset out of s subsets exactly once as the test set. In other words, s-fold cross-validation works as follows:

- The samples of the dataset X are partitioned into s subsets of roughly equal cardinality. The validation set X_{test} is governed by the elements of the s^{th} set. The rest sets form a new learning dataset X_{train}.
- X_{train} is used to get the training of the model G and the error $E_s(G)$ is obtained as:

$$E_s(G) = \frac{\sum_{i=1}^{N/S}(G(x_i^{test}) - y_i^{test})^2}{N/S}, \qquad (6)$$

- Steps 1 and 2 are repeated for varying s (from 1 to s), and average error is computed according to (6).

When S=N, the cross-validation scheme is known as Leave-One-Out.

In the proposed system, the SVM classifier, with dissimilar kernels, has been explored to achieve varied accuracy. Some other parameters also influence the performance of SVM of which the cost (c) and gamma (g) are significant enough to be considered. Several tests were conducted for the system and the best results are found to be for c = 0, and g = 1. Furthermore, different libSVM kernels have been evaluated, and the highest accuracy was obtained by using the SVM linear kernel when 6-fold cross validation strategy with PHOG feature is employed. During all the experiments, the HOG and PHOG feature extractions are used. In all our experiments, three example-values of s (= 4, 5 and 6) for the number of folds have been considered.

V. IMPLEMENTATION AND EXPERIMENTAL RESULTS

The present section deals with implementation details and results obtained through a simulation study.

5.1 Recognition accuracy in Experiments with HOG feature

1. Using 4-fold cross validation.

This subsection describes the classifier recognition accuracy of *s*-fold cross validation when *s* is taken as 4. The dataset is here partitioned into four equal size sets T1, T2, T3 and T4. In this experiment, the investigators have named the four partitioning strategies as a, b, c and d. When HOG features are used in SVM classifier with Polynomial, Linear, and Sigmoid kernels; and in k-NN classifier with k (= 1, 3, 5 and 7), the results show that SVM classifier produces the best accuracy of 94.9% in the 4-fold cross validation when polynomial/ sigmoid kernel is used. With k-NN classifier, the best accuracy achieved is 93.1%, when k = 3 and k = 7. These results are given in Table 2 and also are presented graphically in Figure 6.

Table 2. Recognition accuracy with 4-fold cross validation using HOG features

Dataset Strategy			**SVM**			**k-NN**			
Strategy	*Training data*	*Testing data*	*Polynomial*	*Linear*	*Sigmoid*	*k=1*	*k=3*	*k=5*	*k=7*
a	T2+T3+T4	T1	93.3	92.3	93.3	89.1	91.4	91.3	91.9
b	T1+T3+T4	T2	95.0	94.6	95.0	92.2	92.5	92.2	92.5
c	T1+T2+T4	T3	95.7	95.4	95.7	93.5	93.9	93.8	93.7
d	T1+T2+T3	T4	95.4	96.3	95.4	93.1	94.5	94.6	94.4
4-fold CV			**94.9**	**94.7**	**94.9**	**92.0**	**93.1**	**93.0**	**93.1**

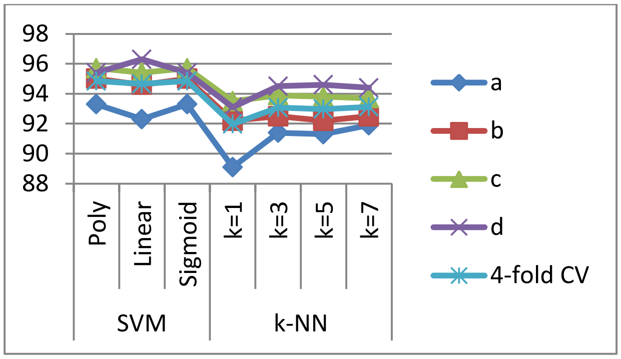


Figure 6: Recognition accuracy with 4-fold cross validation using HOG features for SVM and k-NN

2. *Using 5-fold cross validation*

This subsection deals with classification of Gurmukhi characters when 5-fold cross validation is employed. The entire dataset is segregated into 5 equal sizes: T1, T2, T3, T4 and T5. As such, the investigation is carried out by adopting five dataset segregating strategies: a, b, c, d and e for training case and testing case. From the experimentation, it has been found that while using the SVM classifier, an accuracy of 95.2% is achieved with polynomial and sigmoid kernels. In k-NN classifier, the best performance of 93.2% is achieved when the value of k is taken as 3 and 7.

Table 3. Recognition accuracy with 5-fold cross validation using HOG features

Dataset Strategy			**SVM**			**k-NN**			
Strategy	*Training data*	*Testing data*	*Polynomial*	*Linear*	*Sigmoid*	*k=1*	*k=3*	*k=5*	*k=7*
a	T2+T3+T4+T5	T1	93.6	92.1	93.6	89.9	91.1	91.7	91.6
b	T1+T3+T4+T5	T2	95.0	93.9	95.0	90.9	92.7	92.4	92.4
c	T1+T2+T4+T5	T3	96.4	97.0	96.4	92.4	93.7	93.7	93.0
d	T1+T2+T3+T5	T4	96.4	95.9	96.4	94.6	94.6	95.0	94.3
e	T1+T2+T3+T4	T5	95.9	96.4	95.9	93.3	94.3	95.1	94.9
5-fold CV			**95.2**	**94.9**	**95.2**	**91.6**	**93.0**	**93.2**	**93.0**

The outcomes of these experimental setups are depicted in Table 3 and also in Figure 7.

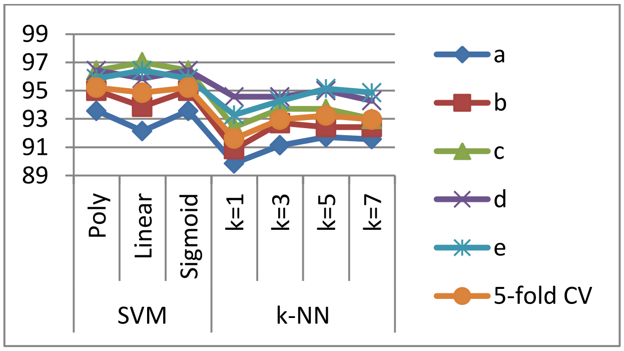


Figure 7: Recognition accuracy with 5-fold cross validation using HOG features for SVM and k-NN

3. *Using 6-fold cross validation*

This subsection presents and discusses the experimentation using 6-fold cross validation of the dataset. The whole dataset is partitioned into six equal size data sets: T1, T2, T3, T4, T5 and T6. In this experiment, investigators have applied six dataset partitioning strategies as: a, b, c, d, e and f. In this experiment, found that an accuracy of 95.1% is achieved when SVM classifier with polynomial and sigmoid kernels is used. Here, in k-NN classifier achieved an accuracy of 93.0%, when the value of k is taken as 5.

Table 4. Recognition accuracy with 6-fold cross validation using HOG features

Dataset Strategy			**SVM**			**k-NN**			
Strategy	*Traing data*	*Testing data*	*Polynomial*	*Linear*	*Sigmoid*	*k=1*	*k=3*	*k=5*	*k=7*
a	T2+T3+T4+T5+T6	T1	93.2	92.1	93.3	89.7	90.3	91.8	91.4
b	T1+T3+T4+T5+T6	T2	95.5	94.1	95.5	90.3	92.4	92.6	91.9
c	T1+T2+T4+T5+T6	T3	95.8	95.8	95.8	92.8	93.1	92.9	93.4
d	T1+T2+T3+T5+T6	T4	95.8	96.3	95.8	92.8	94.5	95.1	93.9
e	T1+T2+T3+T4+T6	T5	96.5	96.1	96.5	94.5	93.6	94.5	94.5
f	T1+T2+T3+T4+T5	T6	95.8	96.3	95.8	93.1	94.5	94.6	95
6-fold CV			**95.1**	**94.6**	**95.1**	**91.5**	**92.6**	**93.0**	**92.9**

The results of these investigational outfits are depicted in Table 4 and also depicted graphically in Figure 8.

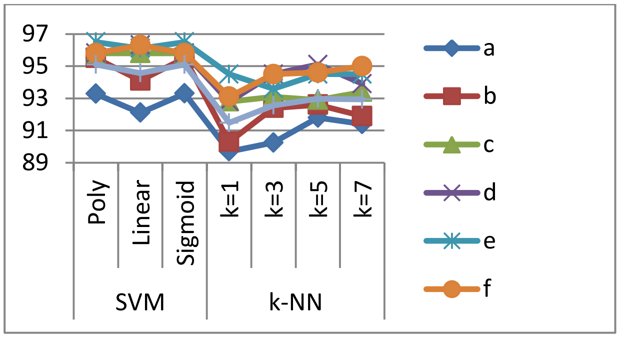


Figure 8: Recognition accuracy with 6-fold cross validation using HOG features for SVM and k-NN

4. *Average recognition accuracy for 4-fold, 5-fold and 6-fold cross validation strategies with HOG features*

In this subsection, classifications strategies have been compared for different values of folds considered in this work. The results are given in Table 5 and also graphically represented in Figure 9. It has been seen that the best performance (95.2%) in this experiments is achieved by SVM classifier when we use polynomial kernel or sigmoid kernel and also employ 5-fold cross-validation strategy. When the k-NN classifier is considered, the best performance 93.2% is achieved, when the value of k is 5 and again 5-fold cross validation strategy is employed.

Table 5. Average recognition accuracies for 4-fold, 5-fold and 6-fold cross validation strategies using HOG features

s-fold CV	**SVM**			**k-NN**			
	Polynomial	*Linear*	*Sigmoid*	*k=1*	*k=3*	*k=5*	*k=7*
4-fold	94.9	94.7	94.9	92.0	93.1	93.0	93.1
5-fold	95.2	94.9	95.2	91.6	93.0	93.2	93.0
6-fold	95.1	94.6	95.1	91.5	92.6	93.0	92.9

One can also infer from Table 5 that other strategies also are in competition with these best performing strategies. SVM, however, performed consistently better than k-NN when we consider the recognition of Gurmukhi characters.

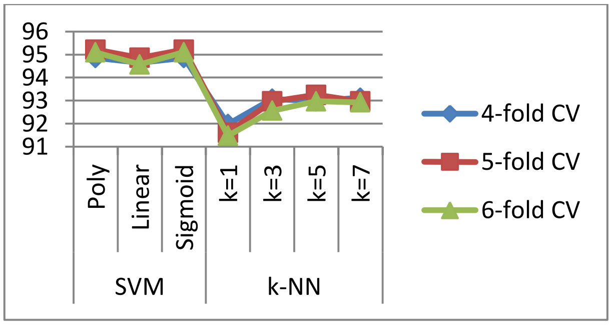


Figure 9. Average recognition accuracies using HOG features for 4-fold, 5-fold and 6-fold cross validation strategies

5.2 *Recognition Accuracy in Experiments with PHOG Features*

1. *Using 4-fold cross validation*

This subsection describes the classifier recognition accuracies when 4-fold cross validation is employed. Here, the dataset partitioning strategy employed is same as considered in section 6.1.1. When PHOG feature is used in SVM and k-NN classifiers, simulation results obtained reveal that the 98.9% is the best accuracy achieved for SVM classifier (with linear kernel) and 97.9% is for k-NN classifier (with k = 3).

Table 6. Recognition accuracy with 4-fold cross validation using PHOG features.

Strategy	**SVM**			**k-NN**			
	Polynomial	*Linear*	*Sigmoid*	*k=1*	*k=3*	*k=5*	*k=7*
a	94.2	98.3	94.2	96.9	97.4	96.8	96.7
b	97.1	98.5	97.1	97.7	98.3	97.9	97.9
c	96.8	99.3	96.8	98.3	98.1	97.7	97.8
d	97.4	99.4	97.4	98.3	98.1	97.8	98.2
4-fold	**96.4**	**98.9**	**96.4**	**97.8**	**97.9**	**97.6**	**97.7**

These results are tabulated in Table 6, and graphically presented in Figure 10.

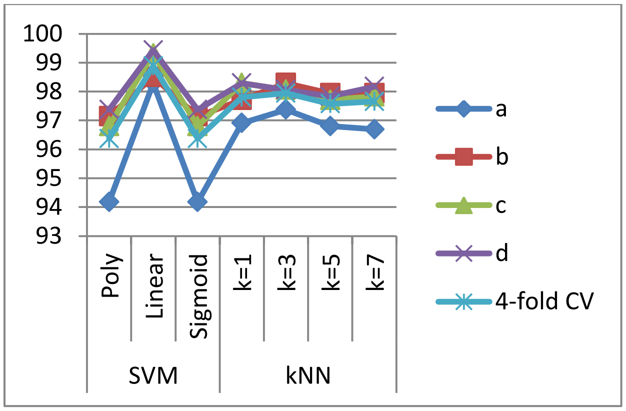


Figure 10. Recognition accuracy with 4-fold cross validation using PHOG features for SVM and k-NN.

2. *Using 5-fold cross validation*

This subsection includes the classification accuracy of Gurmukhi handwritten character recognition process when 5-fold cross validation is considered. Again, the entire dataset is partitioned as in section 6.1.2. From the experiments it has been found that while using the SVM classifier, an accuracy of 99.0% is achieved with linear kernel whereas k-NN classifier produced 97.7% accuracy.

Table 7. Recognition accuracy with 5-fold cross validation using PHOG features.

Strategy	SVM			k-NN			
	Polynomial	*Linear*	*Sigmoid*	*k=1*	*k=3*	*k=5*	*k=7*
a	94.1	98.9	94.1	97.0	97.4	97.0	96.4
b	96.7	98.4	96.7	96.4	97.1	97.0	97.4
c	97.0	99.3	97.0	98.4	98.3	98.0	97.9
d	97.6	99.4	97.6	98.3	98.1	98.1	97.7
e	97.7	99.4	97.7	98.4	97.9	98.0	98.3
5-fold	**96.4**	**99.0**	**96.4**	**97.6**	**97.7**	**97.5**	**97.5**

The outcomes of these experiments are depicted in Table 7 and also graphically depicted in Figure 11.

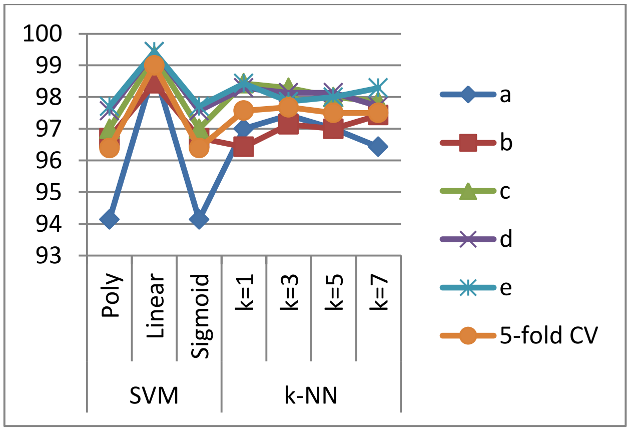


Figure 11. Recognition accuracy with 5-fold cross validation using PHOG features for SVM and k-NN Classifier.

3. *Using 6-fold cross validation*

This subsection presents and discusses the results on recognition accuracy when 6-fold cross validation of dataset is considered. Here, the applied dataset strategy is the same as used in section 6.1.3. In this experiment, SVM with linear kernel achieved an accuracy of 99.1% and k-NN (with k = 3) achieved an accuracy of 98.0%.

Table 8. Recognition accuracy with 6-fold cross validation using PHOG feature

Strategy	SVM			k-NN			
	Polynomial	*Linear*	*Sigmoid*	*k=1*	*k=3*	*k=5*	*k=7*
a	94.0	98.7	94.0	96.6	97.1	96.8	96.1
b	96.6	98.5	96.6	96.6	97.5	97.1	97.5
c	97.7	99.2	97.7	98.3	99.2	99.0	99.0
d	96.8	99.3	96.8	97.8	98.2	97.7	97.5
e	97.8	99.5	97.8	98.0	98.5	98.2	98.0
f	97.8	99.5	97.8	98.3	97.7	97.7	98.2
6-fold	**96.8**	**99.1**	**96.8**	**97.6**	**98.0**	**97.7**	**97.7**

The results of these experiments are depicted in Table 8 and graphically depicted in Figure 12.

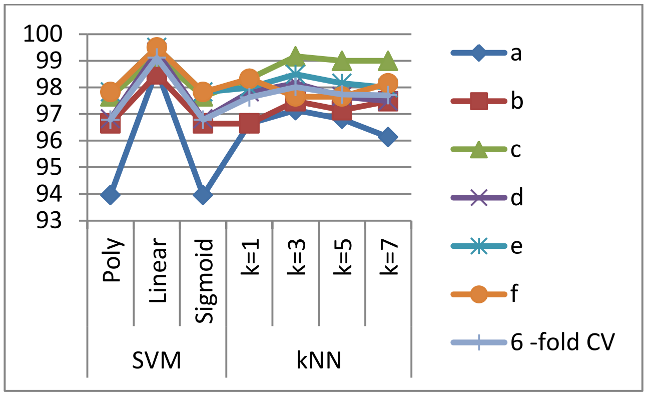


Figure 12. Recognition accuracy with 6-fold cross validation using PHOG feature for SVM and k-NN classifiers.

4. *Average recognition accuracy for 4-fold, 5-fold and 6-fold cross validation strategies with PHOG features.*

In this subsection, a comparison of s-fold cross validation for the two classifier has been carried out. This comparision is illustrated in Table 9. It is evident that the best accuracy of 99.1% is achieved by SVM classifier with linear kernel is used in 6-fold cross validation startegy. In k-NN classifier, the best performance 98.0% is achieved when the value of k is 3, again when 6-fold cross validation strategy is employed.

Table 9. Average recognition accuracies for 4-fold, 5-fold and 6-fold cross validation strategies using PHOG features

s-fold CV	SVM			k-NN			
	Polynomial	*Linear*	*Sigmoid*	*k=1*	*k=3*	*k=5*	*k=7*
4-fold	96.4	98.9	96.4	97.8	97.9	97.6	97.7
5-fold	96.4	99.0	96.4	97.6	97.7	97.5	97.5
6-fold	96.8	**99.1**	96.8	97.6	**98.0**	97.7	97.7

This is also worth mentioning that SVM outpaces consistently the k-NN for the recognition of Gurmukhi Characters.

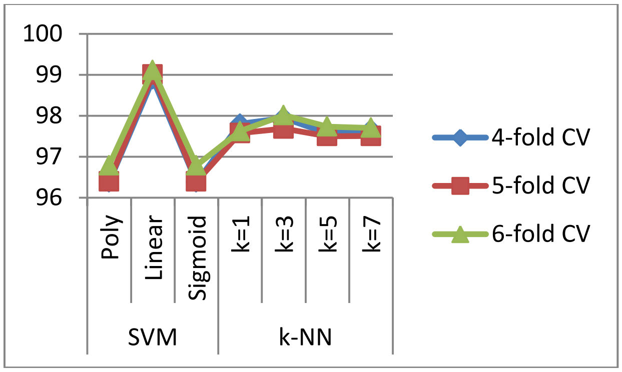


Figure 13. Average recognition accuracies using PHOG features since 4-fold, 5-fold and 6-fold cross validation strategies.

VI. CONCLUSION

In this research work, the significance of HOG and PHOG features for feature extraction of Gurmukhi handwritten characters is investigated. For this purpose, two classifiers, SVM and k-NN, are explored to develop the Gurmukhi Handwritten Character Recognition system. A dataset of 3500 handwritten Gurmukhi characters written by 10 writers is used to develop the system. The character images are considered for both training and testing phases of implementation. The performance evaluation of the SVM and k-NN classifier is also performed in the study. Three kernels are considered to test the efficacy of SVM classifier whereas different values of k are used to examine the efficiency of the k-NN classifier. Three cross validation strategies 4-fold, 5-fold and 6-fold are used for HOG and PHOG features. The experimental study is carried out using MATLAB R2010a. The investigational outcomes reveal that PHOG features exhibit better performance as compared to HOG in all cases. An accuracy of 98.0% in k-NN classifier and overall an accuracy of 99.1% in SVM classifier with linear kernel for PHOG features is achieved in this work. The proposed technique is robust and has significant scope of application to the recognition of other scripts too.

REFERENCES

[1] M. Kumar, M. K. Jindal, and R. K. Sharma, *“Classification of characters and grading writers in offline handwritten Gurmukhi script,”* Proc. Int. Conf. Image Inf. Process., ICIIP 2011.

[2] U. Pal, R. Jayadevan, and N. Sharma, *“Handwriting Recognition in Indian Regional Scripts,”* ACM Trans. Asian Lang. Inf. Process., 2012.

[3] E. Kavallieratou and S. Stamatatos, *“Discrimination of machine-printed from handwritten text using simple structural characteristics,”* Proc. 17th Int. Conf. Pattern Recognition, Vol.1, p. 437–440, ICPR 2004.

[4] A. K. Jain, *“Data clustering: 50 years beyond K-means,”* Pattern Recognit. Lett., vol. 31, no. 8, pp. 651–666, 2010.

[5] M. Kumar, M. K. Jindal, and R. K. Sharma, *“Review on OCR for handwritten indian scripts character recognition,”* Commun. Comput. Inf. Sci., vol. 205 CCIS, pp. 268–276, 2011.

[6] K. Singh Siddharth, M. Jangid, R. Dhir, and R. Rani, *“Handwritten Gurmukhi Character Recognition Using Statistical and Background Directional Distribution Features.”* International Journal on Computer Science and Engineering (IJCSE) 3, no. 06, 2011.

[7] D. Sharma and P. Jhajj, *“Recognition of Isolated Handwritten Characters of Gurumukhi Script using Neocognitron,”* Int. J. Comput. Appl., vol. 4, no. 8, pp. 9–17, 2010.

[8] Due Trier, A. K. Jain, and T. Taxt, *“Feature extraction methods for character recognition-A survey,”* Pattern Recognit., vol. 29, no. 4, pp. 641–662, 1996.

[9] M. K. Mahto, K. Bhatia, and R. K. Sharma, *“Combined Horizontal and Vertical Projection Feature Extraction Technique for Gurmukhi Handwritten Character Recognition,”* In International Conference on Advances in Computer Engineering and Applications (ICACEA), pp. 59–65, 2015.

[10] C. L. Liu and C. Y. Suen, *“A new benchmark on the recognition of handwritten Bangla and Farsi numeral characters,”* Pattern Recognit., vol. 42, no. 12, pp. 3287–3295, 2009.

[11] H. Ma and D. Doermann, *“Word Level Script Identification for Scanned Document Images,”* SPIE Conf. Doc. Recognit. Retr., pp. 124–135, 2004.

[12] Zheng, Y., Liu, C. and Ding, X., "Single-character type identification." In Document Recognition and Retrieval IX, Vol. 4670, pp. 49-57, December 2001.

[13] Zhou, L., Lu, Y., & Tan, C. L. *"Bangla/English script identification based on analysis of connected component profiles."* In International Workshop on Document Analysis Systems, pp. 243-254, 2006.

[14] S. Haboubi, S. Maddouri, N. Ellouze, *“Diff' erenciation de documents textes Arabe et Latin par filtre de Gabor”*, 2007.

[15] S. Mozaffari and P. Bahar, *“Farsi/arabic handwritten from machine-printed words discrimination,”* In Proceedings - International Workshop on Frontiers in Handwriting Recognition, IWFHR, 2012.

[16] G. Muhammad, M. H. Al-Hammadi, M. Hussain, and G. Bebis, *“Image forgery detection using steerable pyramid transform and local binary pattern,”* Mach. Vis. Appl., vol. 25, no. 4, pp. 985–995, 2014.

[17] S. D. Connell and A. K. Jain, *“Recognition of Unconstrained On-Line Devanagari Characters,”* ICPR, pp. 1–4, 2000.

[18] A. F. R. Rahman, R. Rahman, and M. C. Fairhurst, *“Recognition of handwritten Bengali characters: A novel multistage approach,”* Pattern Recognit., *35*(5), 997-1006, 2002.

[19] H. Ma and D. Doermann, *“Adaptive Hindi OCR Using Generalized Hausdorff Image Comparison”,* ACM Transactions on Asian Language Information Processing (TALIP), 2(3), 193-218, 2003.

[20] U. Pal, K. Roy, and F. Kimura, *“A Lexicon-Driven Handwritten City-Name Recognition Scheme for Indian Postal Automation,”* IEICE transactions on information and systems, no. 5, pp. 1146–1158, 2009.

[21] V. Deepu, S. Madhvanath, and A. G. Ramakrishnan, *“Principal component analysis for online handwritten character recognition,”* Pattern Recognit. (ICPR), 17th Int. Conf., vol. 2, no. 2, p. 327–330 Vol.2, 2004.

[22] N. Sharma and U. Pal, *“Recognition of off-line handwritten devnagari characters using quadratic classifier,”* Comput. Vision, Graph. …, pp. 805–816, 2006.

[23] U. Bhattacharya, M. Shridhar, and S. K. Parui, *“On Recognition of Handwritten,”* Comput. Vision, Graph., pp. 817–828, 2006.

[24] M. A. Rahman and A. El Saddik, *“Modified syntactic method to recognize Bengali handwritten characters,”* IEEE Trans. Instrum. Meas., vol. 56, no. 6, pp. 2623–2632, 2007.

[25] U. Pal, T. Wakabayashi, and F. Kimura, *“Handwritten bangla compound character recognition using gradient feature,”* Proc. - 10th Int. Conf. Inf. Technol. ICIT 2007,

pp. 208–213, 2007.

[26] A. Sharma, R. Kumar, and R. K. Sharma, *"Online Handwritten Gurmukhi Character Recognition Using Elastic Matching."* Image and Signal Processing, 2008. CISP'08. Congress on. Vol. 2. IEEE, 2008.

[27] D. Singh, M. Dutta, and S. H. Singh, *"Neural Network Based Handwritten Hindi Character Recognition System."* In Proceedings of the 2nd Bangalore Annual Compute Conference 2009.

[28] N. Das, B. Das, R. Sarkar, S. Basu, M. Kundu, and M. Nasipuri, *"Handwritten Bangla Basic and Compound character recognition using MLP and SVM classifier,"* J. Comput., vol. 2, no. 2, pp. 2151–9617, 2010.

[29] M. Kumar, *"Offline handwritten Gurmukhi character recognition : Study of different feature-classifier combinations,"* In Proceeding of the workshop on Document Analysis and Recognition, pp. 94-99, ACM, 2012.

[30] S. Iamsa-at and P. Horata, *"Handwritten Character Recognition Using Histograms of Oriented Gradient Features in Deep Learning of Artificial Neural Network,"* Int. Conf. IT Converg. Secur. 2013., no. 1, pp. 1–5, 2013.

[31] E. Hassan, *"Exploiting multimedia content: A machine learning based approach,"* Electron. Lett. Comput. Vis. Image Anal., vol. 13, no. 2, p. 69, 2014.

[32] N. Dalal, B. Triggs, N. Dalal, B. Triggs, O. Gradients, and D. Cordelia, *"Histograms of Oriented Gradients for Human Detection*", Computer Vision and Pattern Recognition, (CVPR), IEEE Computer Society Conference on. Vol. 1. 2005.

[33] J. Arróspide, L. Salgado, and M. Camplani, *"Image-based on-road vehicle detection using cost-effective Histograms of Oriented Gradients,"* J. Vis. Commun. Image Represent., 2013.

[34] Q. Zhu, S. Avidan, M. C. Yeh, and K. T. Cheng, *"Fast Human Detection Using a Cascade of Histograms of Oriented Gradients,"* IEEE Conf. Comput. Vis. Pattern Recognit., vol. 2, pp. 1491–1498, 2006.

[35] A. Bosch, A. Zisserman, and X. Munoz, *"Representing shape with a spatial pyramid kernel,"* in Proceedings of the 6th ACM international conference on Image and video retrieval - CIVR '07, 2007.

[36] D. Salvi, J. Zhou, J. Waggoner, and S. Wang, *"Handwritten Text Segmentation using Average Longest Path Algorithm."* In Applications of Computer Vision (WACV), 2013 IEEE Workshop on, pp. 505-512. IEEE, 2013

[37] Y. Bai, L. Guo, L. Jin, and Q. Huang, *"A novel feature extraction method using pyramid histogram of orientation gradients for smile recognition,"* in Proceedings - International Conference on Image Processing, ICIP, 2009.

[38] C. Cortes and V. Vapnik, *"Support-Vector Networks,"* Mach. Learn., vol. 20, no. 3, pp. 273–297, 1995.

[39] V. N. Vapnik, *"An overview of statistical learning theory.,"* IEEE Trans. Neural Netw., vol. 10, no. 5, pp. 988–99, 1999.

[40] C.-C. Chang and C.-J. Lin, *"Libsvm,"* ACM Trans. Intell. Syst. Technol., vol. 2, no. 3, pp. 1–27, 2011.

[41] B. B. Chaudhuri and A. Majumdar, *"Curvelet-based multi SVM recognizer for offline handwritten bangla: A major Indian script,"* in Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, 2007.

[42] A. Dhurandhar, K. Shankarnarayanan, and R. Jawale, *"Robust pattern recognition scheme for devanagari script,"* in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2005.

[43] R. Kohavi, *"A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection,"* Appear. Int. Jt. Conf. Articial Intell., pp. 1–7, 1995.

Authors Profile

Mr. Manoj Kumar Mahto received his Master degree in Computer Science from Birla Institute of Technology, Mesra, Ranchi, India in 2008. He started his carrier as an Assistant Professor in an Engineering institute in MDU, Rohtak, India. He is currently pursuing research work for Ph.D. degree in Computer Science from Gurukula Kangri Vishwavidyalaya, Haridwar, India. His research interests include Machine Learning and Character Recognition.

Dr. Karamjit Bhatia received M.Phil. degree in Computer Applications in 1988 from University of Roorkee (now IIT Roorkee), Roorkee, India and received Ph.D. degree in Computer Science in 2001 from Gurukula Kangri Vishwavidyalaya, Haridwar, India. Currently he is working as a Professor at Department of Computer Science, Gurukula Kangri Vishwavidyalaya, Haridwar, India. His research interests include Distributed Systems, Image Processing and Mobile Ad-hoc Networks. He has published around 40 research papers in International Journals of repute and National/ International conference proceedings. Six Ph.D. degrees have been awarded under his supervision.

Dr. Rajendra Kumar Sharma received his Ph.D. degree in Mathematics from the University of Roorkee (Now, IIT Roorkee), India in 1993. He is currently working as Professor at Thapar Institute of Engineering & Technology, Patiala, India, where he teaches the courses relating to machine learning and statistical models. He has been involved in the organization of a number of conferences and other courses at TIET Patiala. His main research interests are Statistical Models in Computer Science, ML and NLP.