Recognition of Degraded and Non Degraded Roman Characters Using Different Classifiers

Deval Verma*, Himanshu Agarwal , A. K Aggarwal

Department of Mathematics, Jaypee Institute of Information Technology, Noida (U.P.)-201309, India

Received November 13, 2019; Revised December 20, 2019; Accepted December 25, 2019

Copyright ©2019 by authors, all rights reserved. Authors agree that this article remains permanently open access under the terms of the Creative Commons Attribution License 4.0 International License

Abstract In this paper, we present an efficient and robust technique for the recognition of offline roman characters. The main strategy is to extract statistical and similarity features using a combination of grey level co-occurrence matrix (GLCM) and complementary similarity measure (CSM) method. In this work, the CSM method is used to extract features from binary images and combined with GLCM to boost the accuracy of character recognition. The recognition has been done using four different classifiers i.e. artificial neural network (ANN), Naive Bayes classifier, random forest (RF) and support vector machine (SVM). The standard dataset has been used for experimental work. We have done experiments on the clean and noisy dataset. It achieves the accuracy of 100% for some characters without noise and 94.11% with impulsive noise. A comparison of these four classifiers is recorded with and without a noisy environment. On a clean dataset, the random forest provides the best average recognition accuracy of 84.9% for all characters. On low noise datasets, random forest and artificial neural networks have almost the same recognition accuracy and on high noise datasets, SVM provides the highest recognition accuracy.

Keywords Offline, OCR, CSM, GLCM, SVM, ANN, RF, Classifier

1 Introduction

Optical character recognition (OCR) is an automated process for reading printed or handwritten text. OCR is an electronic conversion of books text, administrative records, office files, marriage records, security records and many other important printed text into machine encrypted text. This machine encoded text takes less memory space as compared to image and helps in formatting, editing and displaying text properly [1, 2]. The feature extraction is one of the most important step for matching or classification. Feature extraction techniques are divided into two groups: linear and nonlinear. The linear extraction techniques are principal component analysis (PCA) [3], independent component analysis (ICA) [4], linear discriminant analysis (LDA) [5]. The techniques belong to nonlinear extraction methods are kernel PCA and contour based features. The extracted features of the character along with the class labels are fed to the classifiers for classification of the character image. Three main approaches for classification are simplest, probabilistic and decision boundary. The simplest approach is based on similarity, the probabilistic approach is based on Bayes rule and decision boundary approach is based on optimizing the error in character images.

The important problems of OCR are related to alphabetical character recognition, degraded document, character recognition in license plate, logo & seal and word spotting. Alphabetical character recognition is the most basic and most important problem of OCR. This paper presents a novel approach for character recognition of offline english alphabetical characters. The main strategy is to extract statistical features using a combination of complementary similarity measure (CSM) and grey level co-variance matrix (GLCM). The CSM method is generally used as a classification method in the area of character recognition. In this work, CSM method is used as a feature extractor and combined with GLCM to boost the accuracy of character recognition.

The work presented in this paper is focused on similarity measures and statistical features of dataset. The features are extracted from the offline alphabetical characters using statistical complementary similarity measure (CSM) and grey level co-variance matrix (GLCM). The recognition has been done using four different classifiers i.e. artificial neural network (ANN), Naive Bayes classifier, random forest (RF) and support vector machine (SVM). Standard dataset has been used for experimental work. The experimental results show that the overall average recognition accuracy achieved is in the range from 63.9% to 84.9% in noiseless environment and 75.82% is recorded in noisy environment.
2 Literature on Optical Character Recognition

2.1 Related Work on Alphabetical Character Recognition

This section reviews the literature related to alphabetical character recognition in various fields. Uttal et al. [6] used conflicted dot patterns to recognize characters by dynamic visual noise (DVN) technique. Increasing the number of dots in the form of geometric structures leads to increase in recognizability.

Kahan et al. [7] presented an algorithm in which they recognized roman alphabetic characters of different font sizes. They have used shape clustering approach and line adjacency graph (LAG) for binary images. 'Blobs' were used as connected components of LAG and thinning was chosen as basic factor for feature extraction. The features extracted using LAG based thinning had number of holes in structure, crossings of strokes, concavity of skeleton structure, vertical direction end points, exact position of holes and rectangular boxes that contained characters. The time taken by LAG traversal based thinning was less as compared to pixel based thinning approach. Khan et al [7] found some confusion groups in following alphabets for example a, e, s, g, h, b, t, f, d, o, i, j. They have used Bayesian classifier, which classified six font sizes. But due to some confusion features, recognition rates were reduced and accuracy was compromised i.e. for unlike fonts, it was 97 % and for likely font, it was 99 %.

Comelli et al. [8] found some degraded pictures of vehicles, which were captured by TV camera. The captured pictures were having some imperfection like geometric distortion, presence of noise and blurring. So, it was very difficult to recognize the license plate number properly. The authors have considered template matching and cross correlation technique to overcome this problem. This work was verified by using RITA software as a license plate recognizer. The recognition rate of alphabetical characters was 97.1 percent.

Fukushima et al. [9] presented their work on handwritten recognition that was based on neocognitron technique. This technique was a deformation invariant technique.

Hirwani et al. [10] presented a character recognition of handwritten alphabets in which they used local binary pattern (LBP) as a feature and used a nearest neighbor classifier for recognition.

Bag et al. [2] presented a recognition of bangla compound characters in which they used skeleton segmentation for feature extraction and used strokes segments to extract shape components. This improves the recognition and efficacy of features.

Heutte et al. [11] presented a combined structural and statistical features based vectors (SSFBV) for character recognition. They divided these SSFBV features into two categories, local features and global features. Global features are combined with invariant moments, projections and profiles. Local features consists of four features, namely intersections with straight lines, holes and concave arcs, extrema, end points and junctions. A statistical classifier based on linear discriminant function was used for classification. The recognition rate reported for proprietary datasets was 97.4 percent.

Li et al. [12] used partition combination method for character recognition. Partition parts was called bases, roots, etc. Further, basic features like mean, standard deviation were extracted and recognition was carried out. Finally, they integrated all results of different patterns and accuracy of 98 % was achieved. It was difficult to find out new techniques which integrate the whole results.

Conell et al. [13] presented an unconstrained online devanagari script recognition. They have used two different classifiers i.e. hidden Markov model (HMM) and neural network (NN) in five different ways. The extracted features which were classified by classifier 1 (HMM) provides directional change in the x axis and y axis. The features are center of character and inclined angle θ in sine and cosine forms. Classifier 2 (HMM) classified orientation of critical point, classifier 3 (NN) classified each 5 × 5 zone stroke directional histogram. Classifier 4 (NN) classified each stroke beginning to end and classifier 5 (NN) used global features for each zone like centroid. In this way, they classified different features by different classifiers and reported the recognition accuracy of 86.5 %.

Likforman et al. [14] presented an enhancement scheme with a total variation and non local means filtering methods. This help in removing background noise from images and improves the character recognition rate.

Bawane et al. [15] presented a recognition of handwritten document and object by applying spiking neural network (SNN), Leaky Integrate Fire (LIF) model and two level Network. They have used SVM for comparison purpose.

Vasudeva et al. [16] presented a technique in which they recognized characters by artificial neural network (ANN) and directions of pixels are taken as features in their work. In this work they have used back propagation neural network for recognition.

Zhang et al. [17] presented a survey for classification. They highlighted some issues of posterior probability estimation, the connection between neural and conventional classifiers. They presented some feature selection algorithms and also examined misclassification.

Albregtsen et al. [18] presented a method behind the GLCM matrices. They have discussed about the texture features which were extracted from GLCM matrices [19].

Chawaki et al. [20] presented a method in which they have discussed texture features which were extracted from grey level run length (GLRL) and grey level co-occurrence matrices. They have reported that the GLRL matrices contain more discriminatory information.

Patel et al. [21] presented a handwritten recognition using neural network.

Dojvicinovic et al. [22] presented a neural network based classification of characters in which they have considered whole image as a feature. Extraction and segmentation of characters were recognized in this work. They applied MSER (Maximally Stable Extremal Regions) as a feature detector and recognized it by a neural network.

Chherawala et al. [23] presented an offline character recognition using bidirectional long short term memory neural network.
2.2 Applications of Roman Character Recognition in Degraded Document

In literature, several methods are reported to address the issue of degraded documents. Sawaki and Hagita [24] presented a strong technique based on complementary similarity measure (CSM) for recognition of character with graphical design and degraded documents. They have used newspaper headlines with graphical designs for recognition in their experimental work and reported the recognition rate of 97.7%.

Hobby and Ho [25] presented a technique to enhance degraded document images to display good quality and recognition accuracy. They have used fax images for experimental work and found that the obtained outline descriptions of the printed symbols were rendered with an arbitrary resolution. Some missing parts of the characters, broken characters or stains of the characters were not effective by enhancement method.

Tonazzini et al. [26] presented a technique to recognize text characters in highly degraded printed documents. They have used wavelet based decomposition and filtering for preprocessing the ancient printed text. They applied Markov random field (MRF) segmentation with blind restoration to small portions called blob and trained with back propagation algorithm used in multilayer neural network (MLP). They have used very highly effective recognition for strong degraded texts with precise segmentation. In their work they used degraded touched characters for segmentation and lastly got a resultant text which was tolerable. For degraded documents several methods are reported in the literature to address this issue.

The method introduced by Likforman et al. [27] has the advantage that they used an extension of one-dimensional hidden Markov models (HMMs) called dynamic Bayesian networks (DBN) for the recognition of degraded old printed characters from historical printed books. They used the pairing of two HMMs for capturing the 2D behavior of character images, the columns of image examined to be vertical HMM and rows of image noticed to be horizontal HMM in combination. These two streams were model interacted by two coupled DBN architectures. Here coupled architectures survived better with highly broken characters than both basic HMMs and discriminative methods like SVMs. The coupled architectures predicted missing parts stains and also provided at least one unstained stream within shift. Both the image column and rows are strongly updated which yield unsubstantial data for the classification decision.

Namane et al. [28] presented a technique for degraded characters in typewritten documents produced by typesetting machine. They used complementary similarity measure (CSM) as feature extractor and multi layered perceptron (MLP) for classification. They used CSM features to lift the MLP which was very useful for rejection. They reported the accuracy of 97.95 percent of recognition rate, 0.09 percent of rejection rate and 1.96 percent of error rate on poor quality typewritten characters on typewritten A4 page documents. In the light of reported accuracy it is conceivable that in combination architecture (CSM-MLP) the decision is only made by the MLP, whereas the rejection is either made by the CSM or the MLP and it decreases the system fullfulness.

Ramesh Babu et al. [29] presented a novel technique for recognizing degraded printed characters based on gradient patterns of a particular character. Experiments were organized on character images which were either digitally written or on degraded old historical documents. This work was reported to be tolerable. This method was found to be sensitive to added stains which modify the gradient pattern automatically and decrease the performance of the work.

Namane et al. [30] presented a method for degraded characters recognition in which it contained only model similarities in order to accept or reject an incoming character by a first classifier similarity measure neural network (SMNN). The second classifier Wiener Takes All (WTA) used all the similarities produced by the first classifier (cases of rejection). Similarity measure neural network (SMNN) was designed here, relative distance like hamming distance was used as a quality measurement and proposed a design of a cascaded combination of SMNN and WTA, or CSM net. Experiments were conducted on isolated printed characters collected from post checks. All the recognition methods that are based on the use of feature extraction should use appropriate classification methods.

Virmani et al. [31] presented a technique in which they used singular value decomposition of GLCM matrix and SVM-based characterization of liver cirrhosis.

2.3 Applications of Roman Character Recognition in License Plate

Kim et al. [32] presented license plate recognizer approach. In his work, they recognized the segmented characters by support vector machine (SVM). The character recognition accuracy reported was 97.2 percent.

Wang et al. [33] presented an automatic license plate recognition. They have used invariant rotated characters and orientation had been taken about major axis. In this way after using rotation free character recognition on 102 different car images they reported the accuracy of 98.5 percent.

Pan et al. [34] presented a homogenized technique of structural and statistical methods for recognition of characters. Bayes method and BP neural network method were used to combined these results and achieved the accuracy of 98 percent for hybrid and 97 percent for BP respectively.

Gatos et al. [35] used binarization and enhancement based technique for degraded documents. They used adaptive thresholding scheme for this process. Otsu thresholding technique was not tolerable for degraded documents due to display local variance. Very large amount of noise was found in non text areas in Niblack’s approach.

Some algorithms contained background noise, thinning and broken of characters. In Kim et al. [32] approach great amount of noise and broken character were presented.

Roy et al. [36] presented a work in which they separated a text and symbols from color graphics by using connected component features and geometrical features [37]. They also worked on Multioriented touching characters in the graphical documents. They used Circular and convex hull ring along
with angular information of the contour pixels of the character to make the feature rotation invariant. They reported the recognition accuracy of 97.6 percent by SVM classifier.

Chacko et al. [38] used a wavelet energy feature and extreme learning transform (elm) classifier for handwritten character recognition.

Soora et al. [39] presented a robust technique for license plate character recognition. Angular Width Feature Vector (AWFV) and Geometrical Shape Feature Vector (GSFV) were used and recognized by edit distance metric. They reported the accuracy of 98.6 percent.

![Figure 1](image1.png)

**Figure 1.** Some images of roman characters used in our experiments

### 3 Classifiers and Their Algorithms

This section describes the algorithms of the various classifiers such as support vector machine (SVM), naive Bayes, random forest (RF), and artificial neural network (ANN) for character recognition. In general, any classifier consists of two phases: training phase and testing phase. During training phase, a model is constructed and during testing phase, results are predicted by using the model.

#### 3.1 Support Vector Machine

Support vector machine (SVM) is a supervised learning algorithm. It constructs a model, that depends on the attributes and labels of the training dataset. The constructed model predicts the labels of the testing dataset. Boser et al. [40] presented a training algorithm that maximizes the margin of decision boundaries. Cortes et al. [41] proposed a method, that plots input vectors to the high dimensional space.

#### 3.2 Mathematical Formulation

The two-class classification problem has the following form:

\[ y(x) = w^T \phi(x) + b, \]

where \( \phi(x) \) is feature-space transformation of the feature vector \( x \) of independent variable of data point, \( w^T \) is the weight vector, and \( b \) is the bias. Unknown class for feature vector \( x \) of independent variable of a data point is classified as sign of \( y(x) \).

The main aim in the model creation is to compute the \( w^T \) and \( b \). Features of independent variable and target value in the set of training dataset are used in model creation. Let \( \{ (x_1, t_1), (x_2, t_2), \cdots (x_N, t_N) \} \) be the set of training data points, where \( x_i \) are feature vectors of the independent variables and \( t_i \in \{ -1, 1 \} \) are the target values. For overlapping class distributions, the SVM model is created by solving the following optimization problem

\[
\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i \tag{1}
\]

subject to \( t_i(w^T \phi(x_i) + b) \geq 1 - \xi_i \) \tag{2}

\( \xi_i \geq 0 \) \tag{3}

\( i = 1, 2, \cdots N \), \( N \) is size of the training data-set. Training data point \( (x_i, t_i) \) is on or inside the correct margin boundary if and only if \( \xi_i = 0 \), and \( \xi_i = |t_i - y(x_i)| \) for other training data points. Training data point \( (x_i, t_i) \) is on the decision boundary if and only if \( \xi_i = 1 \). Training data point \( (x_i, t_i) \) is misclassified if and only if \( \xi_i > 1 \). The model is solved as dual representation of the problem by formulating the Lagrange function. Any arbitrary independent variable is classified by using the trained model as follows:

\[
predicted \text{class}(x) = \text{sign}(y(x)), \tag{4}
\]

where,

\[
y(x) = \sum_{x_m \in S} a(x_m) y(x_m) k(x,x_m) + b \tag{5}
\]

\( S = \{ x_i : t_i y(x_i) = 1 - \epsilon_i : (x_i, t_i) \in \text{training dataset} \} \) is the set of support vectors, \( a_m \) are Lagrange multiplies corresponding to inequality of support vectors, \( k(x,x_m) \) is the radial basis function (RBF) kernel defined as

\[
K(x,x_m) = \exp(-\gamma ||x - x_m||^2), \gamma > 0 \tag{6}
\]

where, \( C > 0 \) and \( \gamma \) are the kernel parameters.

Multiclass SVM “One-against-one” approach is used for multiclass classification [42]. If \( k \) is the number of classes, then \( k(k-1)/2 \) different 2-class SVM classifiers are constructed on all possible pairs of classes. Test points are classified according to the highest number of votes. In our problem, \( k = 26 \), as 26 lower case characters are recognized. The block diagram of the algorithm is presented in the figure 2.
3.3 Artificial Neural Network

Rumelhart et al. [44] proposed the widely used backpropagation algorithm. This algorithm is the core part to develop the multilayer perceptron neural network. The fundamental computing unit of the network is called node or neuron or perceptron. Multilayer feed forward neural network (MFFNN) consists of input layer (first layer), hidden layers (second layer, third layer and so on) and output layer (last layer). Each layer is made of structurally identical neurons. The inputs for neurons of input layer is weighted features of feature vector \( x \) of independent variable of data point. For other layers, output of every neuron (which is a scalar quantity) in one layer is input for every neuron in the next layer. Neurons are connected by weights (which are scalars). In the output layer, each neuron represents a unique class. Therefore, number of nodes in output layer is equal to the number of classes to be identified. An unknown data-point belongs to that class, which is corresponding to the node with highest output value.

The output \( O_{i,1} \) of \( i \)th neuron of input layer (first layer) is given as

\[
O_{i,1} = h_{i,1}\left( \sum_{k=0}^{[x]} w_{(i,1),(k,0)}x_k \right) \tag{7}
\]

where, \( h_{i,1}(\cdot) \) is an activation function, \( w_{(i,1),(k,0)} \) is the connection weight between \( i \)th neuron of first layer and \( k \)th element \( x_k \) of the feature vector \( x \), and \([x]\) is the dimension of \( x \).

The output \( O_{i,j} \) of \( i \)th neuron of \( j \)th layer (other than first layer) is given as

\[
O_{i,j} = h_{i,j}\left( \sum_{k=1}^{N_{j-1}} w_{(i,j),(k,j-1)}O_{k,j-1} \right) \tag{8}
\]

where, \( h_{i,j}(\cdot) \) is an activation function, \( w_{(i,j),(k,j-1)} \) is the connection weight between \( i \)th neuron of \( j \)th layer and \( k \)th neuron of \( (j-1) \)th layer, \( N_{j-1} \) is number of nodes in the \((j-1)\)th layer.

Calculating the connection weights is the step in model creation. Weights are calculated based on the error function \( E_L \) defined as follows:

\[
E_L = \frac{1}{2} \left( \sum_{k=1}^{N_k} (O_{k,L} - r_{k,L})^2 \right) \tag{9}
\]

where, \( r_{k,L} \) is the desired response at \( k \)th neuron of the output layer. Equation (9) provides the error for one data-point. The total error for training data-set is defined as follows:

\[
E_{\text{total}} = \sum_{\text{Training dataset}} E_L \tag{10}
\]

The gradient descent algorithm is used to optimize the error \( E_L \) with respect to connection weights [45]. Connection weights are updated as follows:

\[
\Delta w_{(i,j),(k,j-1)} = \alpha \delta_{i,j} O_{k,j-1} \tag{11}
\]

where, \( \delta \)s for output layer (\( \delta_{i,LS} \)) are computed as follows:

\[
\delta_{i,L} = (r_{i,L} - O_{i,L})h'_i\left( \sum_{k=1}^{N_{j-1}} w_{(i,L),(k,L-1)}O_{k,L-1} \right) \tag{12}
\]

and \( \delta \)s for other layers are computed as follows:

\[
\delta_{i,j} = h'_i \left( \sum_{k=1}^{N_{j-1}} w_{(i,j),(k,j-1)}O_{k,j-1} \right) \sum_{j=1}^{N_{j+1}} w_{(i,j+1),(j,j+1)} \delta_{j+1} \tag{13}
\]

Class of unknown data \( x \) is defined as:

\[
\text{Class} = \text{argmax}_{k} \{O_{k,L}(x)\} \tag{14}
\]
3. Create a trained model \( ANNmodel \) by using the function \( train \) of MATLAB2016 as follows:

\[
ANNmodel = train(NN,X,RLtr).
\]

4. Get the response matrix \( RLpr \) for \( Y \) by using the trained model \( ANNmodel \) as follows

\[
RLpr = ANNmodel(Y).
\]

Note that values of \( RLpr \) belongs to \([0, 1]\).

5. Get the predicted labels from \( RLpr \) by using the function \( vec2ind \) of MATLAB2016 as follows:

\[
Lpr = vec2ind(RLpr)
\]

6. Return the predicted labels \( Lpr \).

\subsection{3.4 Naive Bayes Classifiers}

Bayesian classifier is a supervised learning algorithm, initially proposed by [46]. Later, this classifier is improved by Langley et al. [47] and Zhang et al. [48].

Naive Bayes classifier is based on minimization of total average loss. Let we have a data-set of \( W \) classes. The loss function \( r_j(x) \), when feature vector \( x \) is assigned to class \( \omega_j \) is defined as

\[
r_j(x) = \sum_{k=1}^{W} L_{kj} p \left( \frac{x}{\omega_k} \right) P(\omega_k) \quad j = 1, 2, \cdots W
\]

(15)

The class of unknown feature vector \( x \) is given as

\[
\text{cnb}(x) = \{ \omega_k : r_i(x) < r_j(x) \text{ for all } j, j \neq i \}
\]

(16)

For \( 0 - 1 \) loss condition \([45]\) \((L_{kj} \ 0 \text{ if } k = j, 1 \text{ otherwise } (k \neq j))\), the class of unknown feature vector \( x \) is given as

\[
\text{cnb}_{0,1}(x) = \{ \omega_k : d_i(x) < d_j(x) \text{ for all } j, j \neq i \}
\]

(17)

where,

\[
d_j(x) = \sum_{k=1}^{W} p \left( \frac{x}{\omega_k} \right) P(\omega_k)
\]

(18)

Computation of \( d_j \)s is the step of model creation. The most challenging task in the computation of \( d_j \) is subcomputation of the \( p(.) \). If we assume that \( p(.) \) has Gaussian distribution, then \( d_j \)s are computed as follows:

\[
d_j(x) = \ln P(\omega_j) + x^T C^{-1} m_j - \frac{1}{2} m_j^T C^{-1} m_j
\]

(19)

where,

\[
m_j = \frac{1}{N_j} \sum_{x \in \omega_j} x_j, \ j = 1, 2, \cdots W
\]

(20)

\[
C_j = \frac{1}{N_j} \sum_{x \in \omega_j} xx^T - m_j m_j^T
\]

(21)

\section*{Algorithm:3 Prediction of unknown labels by using the Naive Bayes classifier}

\textbf{Procedure:} NBC\((X, Ltr, Y)\)

\textbf{Input:} \( X \): features set of training dataset, \( Ltr = (L_1, L_2, L_3, \cdots L_m) \): labels of training dataset, \( Y \): features of dataset of character images, whose labels are to be predicted,

\textbf{Output} \( Lpr = (L_1, L_2, L_3, \cdots L_n) \): predicted labels corresponding to \( Y \).
3.5 Random Forest (RF)

In Breiman’s [49] random forest classifier creates a set of decision trees from randomly selected subset of training set. It then aggregates the votes from different decision trees to decide the final class of the test object. Basic parameters to Random Forest Classifier can be total number of trees to be generated and decision tree related parameters like minimum split, split criteria etc. In decision trees, the best split among all variables is chosen to split a node where as in RF, the best subset among random predictors is chosen to split a node. The unknown samples are classified using a weighted or unweighted voting of a set of classifiers in the forest. Here, Bagging technique has been used to create the training dataset by randomly choosing samples. A test sample has been classified by assigning the maximum voted class label from all the classifiers and with the predicted class label, the confidence score corresponding to that label has also been predicted. If a training set \( X = x_1, x_2, \ldots, x_n \) and output are \( L = l_1, l_2, l_3, \ldots, l_n \) bagging repeatedly \( B \) times selects a random sample with replacement of the training set and fits trees to these samples: For \( b = 1, \ldots, B \) Sample, with replacement, \( n \) training examples from \( X, L \); say \( X_b, L_b \). Train a classification or regression tree \( f_b; \)on; \( X_b, L_b \). After training, predictions for unseen samples \( x' \) can be made by averaging the predictions from all the individual regression trees on \( x' \)

\[
f' = \frac{1}{B} \sum_{b=1}^{B} f_b(x')
\]

Algorithm: Prediction of unknown labels by using the random forest classifier.

Procedure: RFC\((X, L_{tr}, Y, method, nlearn)\)

Input: \(X\): features set of training dataset, \(L_{tr} = (L_1, L_2, L_3, \ldots, L_m)\) : labels of training dataset, \(Y\) : features set of training dataset, whose labels are to be predicted.

\(method\) : ensemble-aggregation method.

\(nlearn\) : number of ensemble learning cycles

Output \(L_{pr} = (L_1, L_2, L_3, \ldots, L_n)\) : predicted labels corresponding to \(Y\).

1. Create a RFC model by using the function fitensemble of MATLAB2016 as follows

\[
RFCmodel = \text{fitensemble}(X, L_{tr}, \text{’method’} = \text{’Bag’}, \text{’nlearn’} = 100 \ldots)
\]

Note that we have considered that learning rate is 1, MinParent: 2, MinLeaf: 1, Child Node: 2.

2. Predict the labels of \(Y\) by using the function predict as follows:

\[
L_{pr} = \text{predict}(RFCmodel, Y)
\]

3. Return the predicted labels \(L_{pr}\).

Figure 5. Block diagram of training and identification using RF classifier

4 Feature Extraction

This section explains the theory of feature extraction from the set of roman alphabetical characters. CSM features and GLCM features are extracted.

4.1 CSM Features

The CSM is used as discriminating functions for the recognition phases applied to binary character images [30]. It is based on primarily the similarity measures between two binary images; a model \( y \) stored in the training set, and a sample image \( x \) of the size \( n \). The attribution of a sample image character to a single class among other ones (where each class is represented by one or more model images) is performed by using the highest score of similarity between \( x \) and \( y \). These two image characters are expressed as \( n \)-dimensional binary feature vectors as follows: \( x = [x_1, x_2, \ldots, x_i, \ldots, x_n] \) where \( x_i = 0\text{(white)}\)or \( 1\text{(black)}\).

\( y = [y_1, y_2, \ldots, y_i, \ldots, y_n] \) where \( y_i = 0\text{(white)}\)or \( 1\text{(black)}\).
The complementarity similarity measure of model(x) and sample(y) [28] is calculated by using following formula

\[ S_c(x,y) = \frac{a.e - b.c}{T(n-T)} \] (23)

Where \( a = \sum_{i=1}^{n} x_i y_i \)
\( b = \sum_{i=1}^{n} (1-x_i) y_i \)
\( c = \sum_{i=1}^{n} x_i (1-y_i) \)
\( e = \sum_{i=1}^{n} (1-x_i)(1-y_i) \)
\( T = ||y|| = \sum_{i=1}^{n} y_i \)

In this work all the CSM are kept and used as feature components to form a feature vector. The overview of CSM features is shown in figure 6. In this figure CSM plot of character 'a' with two characters 'a' and 'b' is given. The highest CSM value of character 'a' with sample image 'a' is computed as 1 in figure 6 and for sample image 'b' the CSM values are lies between -0.2 to 0.3.

![Figure 6. CSM values Plot of alphabet 'a' with 34 samples of 'a' and 34 samples of 'b'. Class Size =2, Sample Size =34, Total Sample = 2x34=68](image)

### 4.2 GLCM Features

A statistical method of examining texture that considers the spatial relationship of pixels is the gray-level co-occurrence matrix (GLCM). After creating the GLCMs, using graycomatrics [18, 19]. We can derive several statistics from them using graycomatrics. Given an input image \( M \times N \) neighborhood of an input image containing \( G \) gray levels from 0 to \( G-1 \), gray level co-occurrence matrix \( P \) is defined as \([P = i, j|d = 1, \theta = 0]\) \( \mu \) is mean value of \( P \) and \( \mu_x, \mu_y, \sigma_x, \sigma_y \) are means of \( P_x \) and \( P_y \). Where \( P_x(i) \) is the \( i \)th entry in the marginal-probability matrix obtained by summing the rows of \( P(i,j) \) [18].

\[ P_x(i) = \sum_{j=0}^{G-1} P(i,j) \] (24)
\[ P_y(i) = \sum_{i=0}^{G-1} P(i,j) \] (25)
\[ \mu_x = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} i P(i,j) = \sum_{i=0}^{G-1} i P_x(i) \] (26)
\[ \mu_y = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} j P(i,j) = \sum_{j=0}^{G-1} j P_y(j) \] (27)
\[ \sigma_x^2 = \sum_{i=0}^{G-1} (i-\mu_x)^2 \sum_{j=0}^{G-1} P(i,j) = \sum_{i=0}^{G-1} (P_x(i) - \mu_x(i))^2 \] (28)
\[ \sigma_y^2 = \sum_{j=0}^{G-1} (j-\mu_y)^2 \sum_{i=0}^{G-1} P(i,j) = \sum_{j=0}^{G-1} (P_y(j) - \mu_y(j))^2 \] (29)

1. **Contrast**- Measures the local variations in the gray-level co-occurrence matrix.
\[ Contrast = \sum_{i,j} |i-j|^2 P(i,j) \] (30)

2. **Correlation**- Measures the joint probability occurrence of the specified pixel pairs.
\[ Correlation = \frac{\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} i \times j \times P(i,j) - \mu_i \mu_j}{\sigma_i \sigma_j} \] (31)

3. **Energy**- Provides the sum of squared elements in the GLCM. Also known as uniformity or the angular second moment.
\[ Energy = \sum_{i,j} P(i,j)^2 \] (32)

4. **Homogeneity**- Measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal.
\[ Homogeneity = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{P(i,j)}{1+|i-j|} \] (33)

### 5 Experiments, Results and Analysis

This section discusses the details of the dataset, procedure and five experiments.

#### 5.1 Procedure

A general procedure of the experiments is depicted in figure 7. The detail of each block is as follows:

**Data set:** It is the data set of off line optical characters [50]. The dataset consists of 884 grey scale images of small printed characters from 'a' to 'z' with 34 samples of each character shown in figure 1. The dimension of each image is 24 x 42 pixels.
Preprocessing: This process converts the grey scale images of the dataset into binary images.

Features extraction: CSM and GLCM features are extracted for each binarized image. The dimension of CSM features is $1 \times 884$ and the dimension of GLCM features is $1 \times 4$ for each binarized image. The dimension of combined CSM and GLCM features is $1 \times 888$ for each binarized image. The dimension of combined CSM and GLCM features for the dataset (features matrix of the dataset) is $884 \times 888$.

Division: The ratio ($r$) is used to divide the features matrix of the dataset into two parts: features sub-matrix of the training dataset and features sub-matrix of the testing dataset. The size of a training dataset is $[884 \times r]$ and the size of a testing dataset is $884 - [884 \times r]$. The dimension of the features sub-matrix of the training dataset is $[884 \times r] \times 884$ and the dimension of features sub-matrix of the testing dataset is $(884 - [884 \times r]) \times 884$. In this work, the used ratio of size of training dataset to testing dataset is approximately $10:90$ ($r = 0.1$), $20:80$ ($r = 0.2$), $30:70$ ($r = 0.3$), $40:60$ ($r = 0.4$), $50:50$ ($r = 0.5$), $60:40$ ($r = 0.6$), $70:30$ ($r = 0.7$), $80:20$ ($r = 0.8$) and $90:10$ ($r = 0.9$).

Trained model: Appropriate classification algorithm (as discussed in section 3) is implemented on the features sub-matrix of the training dataset and corresponding labels to obtain the trained model.

Classification: Trained model predicts the labels of the images of training/test data-set by using the features sub-matrix of the training/testing dataset.

5.2 Accuracy

The general formula for calculating the accuracy of classifier is as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100$$  \hspace{1cm} (34)

Figure 7. Block diagram of the procedure used in experiments

Figure 8. Block diagram of the accuracy of classifier

Figure 9. Impact of regularization parameter $\gamma$ in SVM based classification

Where TP = True positive, TN = True negative, FP = false positive, FN = false negative. The block diagram to compute the accuracy is shown in figure 8.

5.3 Experiment 1: Analysis of results of support vector machine classifier

We studied the impact of $(C, \gamma)$ on the recognition accuracy of the testing dataset. $C$ is fixed as 1, the ratio $r$ is fixed as 0.9 and $\gamma$ is varied in the range $(0, 1)$. We observed that change in the testing accuracy with respect to $\gamma$ is very less for low range values of the $\gamma$. However, the best testing accuracy of 78.9% is achieved at $\gamma = 0.032$. The testing accuracy significantly falls down for high range values of the $\gamma$. Graph of accuracy versus $\gamma$ is shown in figure 9. We recorded the least testing accuracy of $\approx 35\%$ at $\gamma = 0.998$.

The effect of ratio ($r$) on the testing accuracy is provided in the table 3. The accuracy values are corresponding to $(C, \gamma) = (1, 0.019)$. Accuracy value of each character for different $r$ values is provided in the table. The maximum average accuracy is 78.2% for $r = 0.9$ $(90:10)$ and the least average accuracy is 38.4% for $r = 0.1$ $(10:90)$. The best accuracy of thirteen characters is 100% for $r = 0.9$ and the least accuracy of two characters is 0% for $r = 0.1.$
The effect of ratio \((r)\) on the training and testing accuracies for \((C, \gamma) = (1, 0.019)\) is provided in the figure 10. Training accuracy is almost constant with respect to \(r\). Testing accuracy improves with respect to \(r\). Testing accuracy is less than training accuracy. Testing accuracy converges to training accuracy with \(r\).

5.4 Experiment 2: Analysis of results of artificial neural network classifier

The effect of ratio \((r)\) on the training and testing accuracies is provided in figure 11. The maximum average accuracy is 78.57% for \(r = 0.8\) (80:10) and the least average accuracy is 37.6% for \(r = 0.2\) (20:80). The best accuracy of individual character is 100% for \(r = 0.9\), 6% for \(r = 0.3\). The least accuracy of individual character is 0% for \(r = 0.1\).

The effect of ratio \((r)\) on the training and testing accuracies is provided in figure 12. The figure is corresponding to one hidden layer and \(N_2 = 31\). Accuracy value of each character for different \(r\) values is provided in the table. The maximum average accuracy is 57% for \(r = 0.7\) and the least average accuracy is 5.4% for \(r = 0.9\). The best accuracy of seven characters is 100% for \(r = 0.9\) and the least accuracy of individual character is 0% for \(r = 0.1\).

The effect of ratio \((r)\) on the training and testing accuracies is provided in figure 13. Training accuracy decreases with \(r\), while testing accuracy improves with respect to \(r\). Testing accuracy is less than training accuracy. Testing accuracy converges to training accuracy with \(r\). The best training accuracy of \(\approx 100\%\) is achieved at \(r = 0.9\), and the least training accuracy of 83.3% is achieved at \(r = 0.1\).

5.5 Experiment 3: Analysis of results of Naive Baye’s classifier

The effect of ratio \((r)\) on the testing accuracy is provided in the table 5. The maximum average accuracy is 65.36% for \(r = 0.7\) and the least average accuracy is 26.4% for \(r = 0.1\). The best accuracy of seven characters is 100% for \(r = 0.9\) and the least accuracy of individual character is 0% for \(r = 0.1, 0.8, 0.9\).

The effect of ratio \((r)\) on the testing accuracy is provided in figure 13. Training accuracy decreases with \(r\), while testing accuracy improves with respect to \(r\). Training accuracy is less than training accuracy. Testing accuracy converges to training accuracy with \(r\). The best training accuracy of \(\approx 100\%\) is achieved at \(r = 0.1\) and the least training accuracy of \(\approx 74\%\) is obtained at \(r = 0.9\).

5.6 Experiment 4: Analysis of results of random forest classifier

The effect of ratio \((r)\) on the testing accuracy is provided in the table 6. The maximum average accuracy is 82.05% at \(r = 0.9\) and the least average accuracy is 52.1% at \(r =
Recognition of Degraded and Non Degraded Roman Characters Using Different Classifiers

The best accuracy of individual character is 100% at \( r = 0.9, 0.8, 0.7, 0.6, 0.5 \). The least accuracy of individual character is 9.7% at \( r = 0.1 \).

The effect of ratio \( r \) on the training and testing accuracies is provided in the figure 14. Training accuracy is almost constant with respect to \( r \). Testing accuracy improves with respect to \( r \). Testing accuracy is less than training accuracy. Testing accuracy converges to training accuracy with \( r = 0.15 \).

The ANN is studied at \( N_2 = 31 \) and \( r = 0.8 \). The naive Bayes is studied at \( r = 0.7 \) and random forest is studied at \( r = .9 \).

The observed accuracies with noise level are presented in the figure 15 and table 2. The accuracy of each classifier falls down with noise level as expected. Further, few important observations are as follows:

- The random forest and the artificial neural network have almost same accuracy. In the low range levels of impulsive noise, random forest and the artificial neural network outperforms.
- The SVM outperforms in the high range levels of impulsive noise.
- The accuracy of the Naive Bayes is the least except in the mid range levels of the impulsive noise.

### Table 1. Comparison of accuracy of four classifiers on a clean and noisy dataset. Clean dataset has 0% noise

| Classifier    | Noise 0% | Noise 15% | Noise 25% | Noise 50% | Noise 75% |
|---------------|----------|------------|-----------|-----------|-----------|
| SVM           | 78.20    | 70.51      | 69.23     | 57.69     | 48.71     |
| Naive Bayes   | 65.38    | 65.1       | 65.0      | 62.38     | 40        |
| RF            | 82.05    | 75.64      | 74.35     | 65.38     | 41.03     |
| ANN           | 78.57    | 75.82      | 74.72     | 64.83     | 41.2      |

### Table 2. Comparison of accuracy of four classifiers on a clean and noisy dataset. Clean dataset has 0% noise

| Classifier    | Noise 0% | Noise 15% | Noise 25% | Noise 50% | Noise 75% |
|---------------|----------|-----------|-----------|-----------|-----------|
| SVM           | 78.20    | 70.51      | 69.23     | 57.69     | 48.71     |
| Naive Bayes   | 65.38    | 65.1       | 65.0      | 62.38     | 40        |
| RF            | 82.05    | 75.64      | 74.35     | 65.38     | 41.03     |
| ANN           | 78.57    | 75.82      | 74.72     | 64.83     | 41.2      |

### 5.7 Experiment 5: Analysis of results of noisy dataset

In this experiment, we have studied the effect of impulsive noise on the testing accuracy. We have used the parameters corresponding to the best observed testing accuracy. The used impulsive noise level for testing dataset is 15%, 25%, 50% and 75%. The SVM is studied at \( (C, \gamma) = (1, 0.019) \), and \( r = .9 \).
Table 3. Recognition accuracy (%) of SVM using different parameters, $c = 1, \gamma = 0.019$, av:average accuracy

| Ratio (r) | 90:10 | 80:20 | 70:30 | 60:40 | 50:50 | 40:60 | 30:70 | 20:80 | 10:90 |
|-----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Characters | 100   | 71.4  | 80    | 76.9  | 70.5  | 57.1  | 54.1  | 33.3  | 12.9  |
| a         | 33.3  | 42.8  | 60    | 46.1  | 52.9  | 61.9  | 58.3  | 55.5  | 25.8  |
| b         | 66.6  | 85.7  | 40    | 61.5  | 41.1  | 42.8  | 54.1  | 29.6  | 22.5  |
| c         | 66.6  | 71.4  | 90    | 92.3  | 94.1  | 95.2  | 66.6  | 29.6  | 45.1  |
| d         | 100   | 42.85 | 60    | 46.1  | 58.8  | 57.1  | 62.5  | 74.0  | 0     |
| e         | 33.3  | 85.7  | 70    | 76.9  | 70.5  | 57.1  | 37.5  | 29.6  | 51.6  |
| f         | 33.3  | 71.4  | 60    | 76.9  | 41.1  | 33.3  | 25    | 18.5  | 25.8  |
| g         | 66.6  | 71.4  | 80    | 53.8  | 47.05 | 57.1  | 37.5  | 37.03 | 38.7  |
| h         | 100   | 85.7  | 80    | 84.6  | 64.70 | 71.4  | 54.16 | 44.4  | 48.3  |
| i         | 100   | 85.7  | 80    | 69.23 | 70.5  | 76.19 | 79.1  | 77.7  | 80.6  |
| j         | 66.6  | 85.7  | 70    | 76.9  | 82.3  | 85.7  | 87.5  | 88.8  | 29.0  |
| k         | 66.6  | 85.7  | 90    | 69.2  | 70.5  | 61.9  | 66.6  | 70.3  | 64.5  |
| l         | 100   | 100   | 100   | 100   | 94.1  | 90.4  | 87.5  | 77.7  | 54.8  |
| m         | 100   | 85.7  | 70    | 76.9  | 58.8  | 28.57 | 66.6  | 29.6  | 0     |
| n         | 66.6  | 57.1  | 70    | 76.9  | 88.2  | 76.19 | 37.5  | 18.5  | 9.6   |
| o         | 100   | 71.4  | 60    | 76.9  | 76.4  | 76.1  | 75    | 74.0  | 3.2   |
| p         | 100   | 71.4  | 80    | 84.6  | 82.3  | 80.9  | 83.33 | 74.07 | 32.25 |
| q         | 100   | 71.4  | 80    | 69.2  | 52.9  | 47.6  | 50    | 59.2  | 12.9  |
| r         | 100   | 85.7  | 90    | 92.3  | 88.2  | 76.19 | 75    | 44.4  | 22.58 |
| s         | 100   | 85.7  | 40    | 46.1  | 23.5  | 42.8  | 41.6  | 44.4  | 35.4  |
| t         | 33.3  | 57.14 | 70    | 76.9  | 52.9  | 66.7  | 66.67 | 55.55 | 41.93 |
| u         | 66.7  | 71.3  | 80    | 84.2  | 82.8  | 76.9  | 75    | 55.6  | 54.8  |
| v         | 66.7  | 85.71 | 90    | 92.31 | 94.6  | 90.6  | 75    | 77.8  | 70.7  |
| w         | 100   | 85.71 | 90    | 69.3  | 70.9  | 76.6  | 79.7  | 81.1  | 58.4  |
| x         | 100   | 100   | 90    | 84.62 | 70.9  | 47.9  | 33.33 | 37.7  | 38.9  |
| y         | 100   | 100   | 80    | 69.3  | 76.4  | 80.95 | 83.3  | 74.7  | 41.93 |
| z         | 87.2  | 76.9  | 74.2  | 74.4  | 68.8  | 65.9  | 62.78 | 53.56 | 38.45 |
| av        | 90:10 | 80:20 | 70:30 | 60:40 | 50:50 | 40:60 | 30:70 | 20:80 | 10:90 |
Table 4. Recognition accuracy (%) of ANN at 31 number of hidden nodes using different parameters, av: average accuracy

| Ratio (r) | 90:10 | 80:20 | 70:30 | 60:40 | 50:50 | 40:60 | 30:70 | 20:80 | 10:90 |
|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Characters |       |       |       |       |       |       |       |       |       |
| a        | 100   | 71.4  | 80    | 76.9  | 64.70 | 71.4  | 59.2  | 33.3  | 3.2   |
| b        | 33.3  | 57.1  | 90    | 38.4  | 58.8  | 52.38 | 74.07 | 51.8  | 38.70 |
| c        | 66.6  | 71.4  | 50    | 61.5  | 70.5  | 52.3  | 88.8  | 74.0  | 9.6   |
| d        | 66.7  | 57.7  | 90    | 84.6  | 52.9  | 66.7  | 51.8  | 40.7  | 38.7  |
| e        | 66.7  | 85.7  | 80    | 36.7  | 76.4  | 66.7  | 88.9  | 77.8  | 19.71 |
| f        | 33.3  | 100   | 80    | 76.92 | 100   | 52.8  | 37.4  | 29.3  | 45.32 |
| g        | 100   | 71.3  | 70    | 76.92 | 47.3  | 66.7  | 55.6  | 7.7   | 25.61 |
| h        | 100   | 85.71 | 100   | 100   | 58.81 | 85.7  | 37.03 | 40.7  | 22.5  |
| i        | 100   | 71.48 | 90    | 84.6  | 64.7  | 85.7  | 40.7  | 48.14 | 58.06 |
| j        | 100   | 85.7  | 60    | 76.92 | 64.70 | 66.7  | 77.8  | 74.7  | 77.4  |
| k        | 66.7  | 71.3  | 60    | 61.5  | 64.5  | 76.9  | 81.8  | 88.9  | 93.71 |
| l        | 66.7  | 85.7  | 80    | 84.6  | 76.4  | 61.9  | 70.37 | 81.46 | 67.7  |
| m        | 100   | 100   | 100   | 100   | 85.7  | 85.18 | 85.18 | 25.80 |       |
| n        | 66.7  | 85.7  | 70    | 92.3  | 76.4  | 85.7  | 48.5  | 55.6  | 41.9  |
| o        | 66.7  | 85.71 | 60    | 53.5  | 64.5  | 66.7  | 25.9  | 3.7   | 45.2  |
| p        | 100   | 57.14 | 60    | 92.3  | 76.4  | 71.4  | 66.7  | 55.6  | 64.3  |
| q        | 66.7  | 71.3  | 70    | 69.23 | 88.12 | 76.19 | 85.19 | 74.07 | 74.9  |
| r        | 66.7  | 85.7  | 90    | 76.9  | 58.8  | 71.4  | 66.7  | 70.7  | 70.9  |
| s        | 100   | 85.7  | 90    | 76.9  | 76.47 | 95.2  | 88.9  | 85.9  | 64.3  |
| t        | 66.7  | 71.4  | 60    | 61.4  | 52.9  | 61.9  | 51.8  | 48.14 | 3.2   |
| u        | 33.3  | 71.4  | 80    | 76.9  | 58.8  | 42.8  | 59.25 | 48.14 | 0     |
| v        | 66.7  | 85.7  | 70    | 69.3  | 64.7  | 66.7  | 77.8  | 59.6  | 54.8  |
| w        | 66.7  | 71.43 | 60    | 61.5  | 60.79 | 76.9  | 70.7  | 70.7  | 64.5  |
| x        | 66.7  | 85.7  | 80    | 92.31 | 64.7  | 76.9  | 100   | 88.9  | 0     |
| y        | 100   | 100   | 100   | 92.3  | 76.4  | 33.3  | 33.3  | 37.4  | 61.8  |
| z        | 66.7  | 85.7  | 80    | 76.9  | 64.7  | 66.7  | 51.8  | 48.5  | 48.7  |
| av       | 73.07 | 78.57 | 76.92 | 76.92 | 69.05 | 68.49 | 64.38 | 37.60 | 43.05 |
Table 5. Recognition accuracy (%) of Naïve Bayes using different parameters

| Ratio (r) | 90:10 | 80:20 | 70:30 | 60:40 | 50:50 | 40:60 | 30:70 | 20:80 | 10:90 |
|-----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Characters | av    |       |       |       |       |       |       |       |       |
| a         | 100   | 57.14 | 70    | 76.9  | 70.5  | 57.4  | 41.7  | 33.3  | 3.2   |
| b         | 0     | 0     | 30    | 53.8  | 35.5  | 57.4  | 58.3  | 33.3  | 12.91 |
| c         | 33.3  | 28.7  | 10    | 15.8  | 11.8  | 19.5  | 41.7  | 33.3  | 16.6  |
| d         | 66.7  | 71.3  | 70    | 76.9  | 70.9  | 61.9  | 37.5  | 18.5  | 19.1  |
| e         | 66.7  | 42.6  | 70    | 61.5  | 64.7  | 61.9  | 62.5  | 70.7  | 12.9  |
| f         | 33.3  | 71.4  | 70    | 76.9  | 64.7  | 42.8  | 37.5  | 25.9  | 25.8  |
| g         | 0     | 14.9  | 10    | 15.8  | 17.82 | 28.7  | 20.8  | 25.9  | 12.9  |
| h         | 66.7  | 71.4  | 80    | 61.5  | 64.7  | 57.4  | 37.5  | 22.2  | 38.7  |
| i         | 100   | 85.7  | 80    | 76.9  | 70.5  | 66.7  | 37.5  | 37.4  | 25.8  |
| j         | 33.3  | 57.4  | 60    | 53.8  | 47.3  | 57.4  | 62.5  | 40.7  | 48.7  |
| k         | 66.7  | 85.7  | 60    | 76.9  | 82.3  | 80.9  | 79.7  | 85.9  | 6.4   |
| l         | 66.7  | 71.3  | 80    | 61.5  | 64.7  | 61.9  | 70.8  | 81.4  | 35.4  |
| m         | 100   | 100   | 90    | 84.6  | 82.3  | 76.19 | 75    | 70.8  | 32.25 |
| n         | 33.3  | 57.14 | 70    | 61.5  | 29.4  | 28.5  | 41.7  | 33.3  | 90.5  |
| o         | 33.3  | 57.4  | 70    | 69.2  | 70.9  | 61.9  | 16.7  | 18.2  | 6.3   |
| p         | 100   | 57.14 | 70    | 76.9  | 76.4  | 47.6  | 41.7  | 29.6  | 41.9  |
| q         | 66.7  | 57.4  | 70    | 76.9  | 82.3  | 71.4  | 75    | 70.3  | 22.5  |
| r         | 66.7  | 85.7  | 60    | 53.8  | 35.5  | 33.3  | 41.7  | 48.5  | 80.9  |
| s         | 100   | 85.7  | 80    | 76.92 | 82.8  | 71.4  | 75    | 51.8  | 12.91 |
| t         | 66.7  | 42.8  | 30    | 30.77 | 47.05 | 42.8  | 54.7  | 48.5  | 38.70 |
| u         | 33.3  | 57.14 | 70    | 69.2  | 52.9  | 57.4  | 66.7  | 77.7  | 9.5   |
| v         | 66.7  | 57.4  | 70    | 69.2  | 64.7  | 71.3  | 66.7  | 55.6  | 41.9  |
| w         | 66.7  | 85.71 | 90    | 76.9  | 70.9  | 66.7  | 54.7  | 66.67 | 0     |
| x         | 100   | 85.7  | 80    | 69.2  | 58.8  | 66.7  | 70.8  | 77.8  | 0     |
| y         | 100   | 100   | 80    | 84.6  | 70.9  | 57.14 | 33.3  | 33.3  | 22.16 |
| z         | 66.7  | 71.4  | 80    | 69.3  | 58.8  | 80.9  | 87.5  | 66.7  | 29.6  |
| av        | 62.42 | 63.73 | 65.38 | 64.492| 59.502| 57.14 | 53.36 | 48.29 | 26.42 |
| Ratio (r) | 90:10 | 80:20 | 70:30 | 60:40 | 50:50 | 40:60 | 30:70 | 20:80 | 10:90 |
|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| **Characters** |       |       |       |       |       |       |       |       |       |
| a        | 100   | 71.43 | 80    | 76.9  | 58.8  | 66.7  | 62.5  | 48.5  | 9.64  |
| b        | 66.67 | 57.4  | 60    | 46.5  | 41.9  | 66.7  | 66.7  | 62.96 | 48.87 |
| c        | 66.7  | 85.71 | 80    | 84.6  | 47.53 | 57.14 | 70.8  | 48.5  | 32.2  |
| d        | 100   | 100   | 90    | 100   | 88.2  | 80.9  | 70.8  | 48.1  | 32.2  |
| e        | 100   | 85.71 | 70    | 61.54 | 64.75 | 57.4  | 62.5  | 74.07 | 80.6  |
| f        | 100   | 85.71 | 80    | 76.9  | 64.75 | 47.2  | 45.8  | 33.3  | 41.9  |
| g        | 33.3  | 57.1  | 40    | 38.46 | 17.6  | 33.3  | 44.4  | 29.0  |       |
| h        | 100   | 57.1  | 80    | 69.2  | 41.17 | 61.9  | 29.16 | 29.6  | 38.9  |
| i        | 100   | 85.7  | 90    | 84.62 | 70.9  | 66.7  | 54.7  | 51.8  | 58.0  |
| j        | 100   | 85.7  | 80    | 69.3  | 64.7  | 66.7  | 79.7  | 77.8  | 74.193|
| k        | 66.6  | 85.7  | 70    | 76.9  | 76.4  | 85.7  | 87.5  | 88.9  | 87.6  |
| l        | 66.7  | 71.4  | 80    | 61.5  | 64.7  | 66.7  | 62.5  | 77.8  | 64.513|
| m        | 100   | 100   | 100   | 100   | 88.2  | 85.71 | 91.7  | 89.9  | 61.290|
| n        | 100   | 100   | 70    | 69.23 | 17.62 | 42.86 | 62.5  | 48.15 | 16.129|
| o        | 33.3  | 71.4  | 40    | 61.5  | 94.11 | 66.6  | 20.83 | 22.2  | 19.3  |
| p        | 100   | 71.4  | 70    | 76.9  | 100   | 66.7  | 62.5  | 70.37 | 45.1612|
| q        | 100   | 71.42 | 80    | 69.23 | 94.06 | 80.9  | 87.5  | 77.78 | 74.19 |
| r        | 66.7  | 85.7  | 70    | 69.23 | 94.706| 47.6  | 50    | 51.8  | 77.4  |
| s        | 100   | 85.7  | 90    | 92.3  | 100   | 90.48 | 83.3  | 85.19 | 87.096|
| t        | 66.67 | 71.3  | 70    | 46.7  | 94.6  | 42.8  | 45.8  | 55.6  | 61.290|
| u        | 66.6  | 57.4  | 90    | 69.3  | 94.06 | 42.86 | 50    | 51.85 | 25.806|
| v        | 66.67 | 85.7  | 90    | 84.6  | 94.6  | 66.67 | 79.7  | 70.37 | 77.419|
| w        | 66.6  | 85.7  | 90    | 92.3  | 100   | 85.7  | 83.3  | 81.8  | 80.6  |
| x        | 66.67 | 85.7  | 90    | 84.6  | 94.6  | 71.4  | 79.167| 81.8  | 48.7  |
| y        | 100   | 100   | 90    | 100   | 100   | 57.14 | 33.3  | 25.9  | 38.7  |
| z        | 100   | 100   | 100   | 69.23 | 94.6  | 80.9  | 83.3  | 77.8  | 45.161|
| av       | 82.051| 80.769| 78.46 | 74.26 | 75.339| 64.835| 62.480| 60.54 | 52.109|
6 Conclusions

In this paper, four different classifiers viz SVM, Naive Bayes, random forest and neural network are studied for classification of offline alphabetic characters. Experiments have been done on clean dataset and impulsive noise dataset. The CSM and GLCM features have been extracted from the dataset. The experimental results show that, on clean dataset, random forest classifier provides the highest accuracy of ≈ 84% and Naive Bayes classifier provides the least accuracy of ≈ 65.38%. SVM and artificial neural network classifiers have comparable accuracies of 78% on the clean dataset. Random forest outperforms on the clean dataset. For noisy dataset, the random forest and the artificial neural network have almost same accuracy. In the low range levels of impulsive noise, random forest and the artificial neural network outperforms. However, the SVM outperforms in the high range levels of impulsive noise.

REFERENCES

[1] Michiel Hagedoorn. Pattern matching using similarity measures. PhD thesis, PHDThesis, 2000.

[2] Soumen Bag, Gaurav Harit, and Partha Bhowmick. Recognition of bangla compound characters using structural decomposition. Pattern Recognition, 47(3):1187–1201, 2014.

[3] Wangmeng Zuo, Kuanquan Wang, and David Zhang. Bi-directional pca with assembled matrix distance metric. In Image Processing, 2005. ICIP 2005. IEEE International Conference on, volume 2, pages II–958. IEEE, 2005.

[4] Vaishali Khurana, Pradeep Kumar, Rajkumar Saini, and Partha Pratim Roy. Eeg based word familiarity using features and frequency bands combination. Cognitive Systems Research, 49:33–48, 2018.

[5] Subrahmanyam Murala, RP Maheshwari, and R Balasubramanian. Local tetra patterns: a new feature descriptor for content-based image retrieval. IEEE transactions on image processing, 21(5):2874–2886, 2012.

[6] William R Uttal. Masking of alphabetic character recognition by dynamic visual noise (dvn). Perception & Psychophysics, 6(2):121–128, 1969.

[7] Simon Kahan, Theo Pavlidis, and Henry S Baird. On the recognition of printed characters of any font and size. IEEE Transactions on pattern analysis and machine intelligence, 2(2):274–288, 1987.

[8] Paolo Comelli, Paolo Ferragina, Mario Notturno Granieri, and Flavio Stabile. Optical recognition of motor vehicle license plates. IEEE transactions on Vehicular Technology, 44(4):790–799, 1995.

[9] Kunihiko Fukushima and Nobuaki Wake. Handwritten alphanumeric character recognition by the neocognitron. IEEE transactions on Neural Networks, 2(3):355–365, 1991.

[10] Amrita Hirwani, Neelmani Verma, and Sandeep Gonnade. Efficient handwritten alphabet recognition using lbp based feature extraction and nearest neighbor classifier. International Journal of Advanced Research in Computer Science and Software Engineering, 4(11), 2014.

[11] Laurent Heutte, Thierry Paquet, Jean-Vincent Moreau, Yves Lecourtier, and Christian Olivier. A structural/statistical feature based vector for handwritten character recognition. Pattern recognition letters, 19(7):629–641, 1998.

[12] Z-C Li and Ching Y Suen. The partition–combination method for recognition of handwritten characters. Pattern Recognition Letters, 21(8):701–720, 2000.

[13] Scott D Connell, RMK Sinha, and Anil K Jain. Recognition of unconstrained online devanagari characters. In Pattern Recognition, 2000. Proceedings. 15th International Conference on, volume 2, pages 368–371. IEEE, 2000.

[14] Laurence Likforman-Sulem, Jérôme Darbon, and Elisa H Barney Smith. Enhancement of historical printed document images by combining total variation regularization and non-local means filtering. Image and vision computing, 29(5):351–363, 2011.

[15] Priyanka Bavane, Snehal Gadariye, S chatturvedi, and AA Khurshid. Object and character recognition using spiking neural network. Materials Today: Proceedings, 5(1):360–366, 2018.

[16] Nisha Vasudeva, Hem Jyotsana Parashar, and Singh Vijendra. Offline character recognition system using artificial neural network. International Journal of Machine Learning and Computing, 2(4):449, 2012.

[17] Guoqiang Peter Zhang. Neural networks for classification: a survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 30(4):451–462, 2000.

[18] Fritz Albregtsen et al. Statistical texture measures computed from gray level coocurrence matrices. Image processing laboratory, department of informatics, university of oslo, 5, 2008.

[19] P Mohanaiah, P Sathyanarayana, and L GuruKumar. Image texture feature extraction using glcm approach. International Journal of Scientific and Research Publications, 3(5):1, 2013.

[20] Djeddi Chawki and Souici-Meslati Labiba. A texture based approach for arabic writer identification and verification. In Machine and Web Intelligence (ICMWI),
2010 International Conference on, pages 115–120. IEEE, 2010.

[21] Chirag I Patel, Ripal Patel, and Palak Patel. Handwritten character recognition using neural network. International Journal of Scientific & Engineering Research, 2(5):1–6, 2011.

[22] Nikola Dojčinović, Igor Mihajlović, Jugoslav Joković, Vera Marković, and Bratislav Milovanović. Neural network based optical character recognition system. In Neural Network Applications in Electrical Engineering (NEUREL), 2012 11th Symposium on, pages 111–114. IEEE, 2012.

[23] Youssof Chherawala, Partha Pratim Roy, and Mohamed Cheriet. Combination of context-dependent bidirectional long short-term memory classifiers for robust offline handwriting recognition. Pattern Recognition Letters, 90:58–64, 2017.

[24] Minako Sawaki and Norihiro Hagita. Text-line extraction and character recognition of document headlines with graphical designs using complementary similarity measure. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(10):1103–1109, 1998.

[25] John D Hobby and Tin Kam Ho. Enhancing degraded document images via bitmap clustering and averaging. In Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference on, volume 1, pages 394–400. IEEE, 1997.

[26] Anna Tonazzini, Stefano Vezzosi, and Luigi Bedini. Analysis and recognition of highly degraded printed characters. Document Analysis and Recognition, 6(4):236–247, 2003.

[27] Laurence Likforman-Sulem and Marc Sigelle. Recognition of degraded characters using dynamic bayesian networks. Pattern Recognition, 41(10):3092–3103, 2008.

[28] Abderrahmane Namane, El Houssine Soubari, and Patrick Meyrueis. Degraded dot matrix character recognition using csm-based feature extraction. In Proceedings of the 10th ACM symposium on Document engineering, pages 207–210. ACM, 2010.

[29] DR Ramesh Babu, M Ravishankar, Manish Kumar, Kevin Wadera, and Aakash Raj. Degraded character recognition based on gradient pattern. In Second International Conference on Digital Image Processing, volume 7546, page 754605. International Society for Optics and Photonics, 2010.

[30] Abderrahmane Namane, Abderrazak Guessoum, El Houssine Soubari, and Patrick Meyrueis. Csm neural network for degraded printed character optical recognition. Journal of visual communication and image representation, 25(5):1171–1186, 2014.

[31] Jitendra Virmani, Vinod Kumar, Naveen Kalra, and Niranjana Khandelwal. Svm-based characterisation of liver cirrhosis by singular value decomposition of glcm matrix. International Journal of Artificial Intelligence and Soft Computing, 3(3):276–296, 2013.

[32] Kl Kim Kim, KI Kim, JB Kim, and Hang Joon Kim. Learning-based approach for license plate recognition. In Neural Networks for Signal Processing X, 2000. Proceedings of the 2000 IEEE Signal Processing Society Workshop, volume 2, pages 614–623. IEEE, 2000.

[33] Shen-Zheng Wang and Hsi-Jian Lee. Detection and recognition of license plate characters with different appearances. In Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE, volume 2, pages 979–984. IEEE, 2003.

[34] Xiang Pan, Xiuzi Ye, and Sanyuan Zhang. A hybrid method for robust car plate character recognition. Engineering Applications of Artificial Intelligence, 18(8):963–972, 2005.

[35] Basilios Gatos, Ioannis Pratikakis, and Stavros J Perantonis. Adaptive degraded document image binarization. Pattern recognition, 39(3):317–327, 2006.

[36] Partha Pratim Roy, Eduard Vazquez, Josep Lladós, Ramon Baldrich, and Umapada Pal. A system to segment text and symbols from color maps. In International Workshop on Graphics Recognition, pages 245–256. Springer, 2007.

[37] Partha Pratim Roy, Umapada Pal, and Josep Lladós. Recognition of multi-oriented touching characters in graphical documents. In Computer Vision, Graphics & Image Processing, 2008. ICVGIP’08. Sixth Indian Conference on, pages 297–304. IEEE, 2008.

[38] Binu P Chacko, VR Vimal Krishnan, G Raju, and P Babu Anto. Handwritten character recognition using wavelet energy and extreme learning machine. International Journal of Machine Learning and Cybernetics, 3(2):149–161, 2012.

[39] Narasimha Reddy Soora and Parag S Deshpande. Robust feature extraction technique for license plate characters recognition. IETE Journal of Research, 61(1):72–79, 2015.

[40] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In Proceedings of the fifth annual workshop on Computational learning theory, pages 144–152. ACM, 1992.

[41] Corinna Cortes and Vladimir Vapnik. Support-vector networks. Machine learning, 20(3):273–297, 1995.

[42] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. ACM transactions on intelligent systems and technology (TIST), 2(3):27, 2011.
[43] https://www.csie.ntu.edu.tw/ cjlin/libsvm/ (last accessed on 5 June 2019).

[44] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. Cognitive modeling, 5(3):1, 1988.

[45] Christopher M Bishop. Pattern recognition and machine learning. springer, 2006.

[46] Duda RO Hart PE Pattern. Classification and Scene Analysis. John Wiley and Sons, New York, 1973.

[47] Pat Langley, Wayne Iba, Kevin Thompson, et al. An analysis of bayesian classifiers. In Aaai, volume 90, pages 223–228, 1992.

[48] Harry Zhang. The optimality of naive bayes. AA, 1(2):3, 2004.

[49] Leo Breiman. Random forests. Machine learning, 45(1):5–32, 2001.

[50] Chetan Arora, Nikhil Arora, Hriday Goyal, and Verma Gaurav. Sixth sense dictionary. Proceedings of the IEEE International Conference on Computing for Sustainable Global Development”, pages 5860–5864, 2017.