International Journal of Pure and Applied Mathematics
Volume 117 No. 21 2017, 663-678
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

A Novel Handwritten Gurmukhi Character Recognition System Based On Deep Neural Networks

[1]Neeraj Kumar and [2]Sheifali Gupta
[1]Dept of ECE, Chitkara University,
Himachal Pradesh, India.
Neeraj.kumar@chitkarauniversity.edu.in
[2]Dept of CURIN, Chitkara University,
Punjab, India.
Sheifali.gupta@chitkara.edu.in

Abstract

Deep learning (Deep Networks) is presently an exceptionally lively area of research in the field of machine learning and pattern recognition. It has gained massive success in anexpansivevicinity of applications like image processing and speech recognition. In this article a detailed methodology for recognition of handwritten Gurmukhi characters including broken characters using deep learning has been explained. Due to variations in handwriting styles and speed of writing it is very difficult to recognize the handwritten Gurmukhi characters. A majority of the work has already been reported on the online handwritten scripts like English, Bangla etc.Now research is being shifted towards the recognition of offline handwritten scripts. In the proposed work the feature extraction has been doneusing three types of features, namely Local binary pattern (LBP) featuresin addition to directional features andregionalfeatures. A total of 117 features have been extracted in order to correctly recognize the text. Furthermore, in order to map the Gurmukhi text with Devanagari text a suitable mapping technique has also been implemented.A total of 2700 samples have been taken for training and testing purpose. The proposed system is achieving an accuracy of 99.3%.

Key Words: Deep Learning, local binary patterns, directional features, regional features, softmax function, mapping.

1. Introduction

The handwritten characters written on an electronic surface or on plain paper are generally recognized by character recognition system (CRS). However, achievements acquired using printed CRS cannot be transmitted automatically to the handwritten CRS. Handwritten CRS (HCRS) can be categorized into two streams namely, online HCRSand offline HCRS. In online stream, user writes on an electronic surface by the aid of a unique pen and during the writing process, data is captured in terms of (x,y) coordinates.

A number of devices, including personal digital assistant and tablet PCs are available these days that can be used for data capturing. In these systems, characters are captured as a sequence of strokes. Features are then extracted from these strokes and strokes are recognized with the help of these features. Generally, a post-processing module helps in forming the characters from the stroke(s).OfflineHCRS converts offline handwritten text into such a form which can be easily understood by the machines. It involves the processing of documents containing scanned images of a text written by a user, usually on a plain paper. In this kind of systems, characters are digitized to obtain 2-dimensionalimages. Developing a realistic HCRS which is capable of retaining lofty recognition accuracy is still a very challenging task. Moreover, recognition accuracy depends upon the input text quality. Based on the literature studied so far, it is found that the majority of the work is being done on individual character recognition in different languages like English, Bangla, Devanagari, Gujrati etc.Now researchers are shifting towards the recognition of Gurmukhi script which is one of the most popularscripts in the whole world. Research in this script is currently limited to single character recognition only. Due to different writing styles and speed, discontinuities may arise in the characters which give wrong recognition results. So there is a need to solve this problem in an effective manner.

2. Gurmukhi Script

Generally, each Indian script is having some particular set of set of laws for the amalgamation of consonants, vowels and modifiers. In the Gurmukhi script there are a number of consonants, auxiliary signs, vowel bearers, vowel modifiers & half characters. There are a number of components which are used in Gurmukhi script .These components have been revealed in Fig 1.

ਚਛਜਝਞਟਠਡਸਹਕਖਗਘਙ
ਥਦਧਢਣਤਮਯਰਲਵੜ ਨਪਫਬਭ
Consonants
ਉਅਏ
Vowel bearers
ਸ਼ਜ਼ਖ਼ਫ਼ਗ਼
Additional consonants

Fig. 1: Components in Gurmukhi Script

3. How Recognition Takes Place

For doing the recognition, there are some steps which must be followed & these steps are explained as following:

i. **Image Acquisition:**Initially, some characters are written on the sheet of paper. Now the paper is scanned using a scanner. After scanning the image a bitmap image will be obtained. The process of converting the document typically in the handwritten form into an electronic format is termed as Digitization.

ii. **Pre-processing:**In this the input image is passed to the processing stage. Preprocessing involves a number of steps like binarization, detection of edges, dilation of images, and filling of holes present in the image. In binarization, with the help of thresholding the gray image is transformed into the binary image. The output of pre-processing stage will be a normalized bitmap image.

iii. **Segmentation:** An image may have sequence of words i.e. lines containing words. With the help of segmentation the image can be broken down into sub images containing individual characters.

iv. **Feature Extraction:** The efficiency of recognition relies a lot on the features extracted. There are various techniques through which features can be extracted such as LBP feature extraction, power & parabola arc fitting, diagonal feature extraction, transition feature extraction.

v. **Classification**: Classification is basically the phase where decision making is done. In this, we make use of the features that we extract in the feature extraction stage. Decision making is done in the classification using various classifiers like k-nearest neighbor (k-NN), Support Vector Machine (SVM), Artificial Neural Network (ANN) etc.k-NN classifier has been used by a number of researchers for the purpose of network ANN works in the similar fashion as our brain works.ANN is used in pattern recognition & data classification via some learning process. Information processing elements are neurons and ANN consists of neurons, which help in solving specific problems.

4. Literature Review

U. Garain et al [1] proposed a technique whichworks on fuzzy analysis and authors developed analgorithm to segment touching characters.M.Kumar et al [2]projected a scheme to recognize Gurmukhi characters which is based on transition & diagonal features extractionand the classifier used in the work is k-NN.To determine the k-nearest neighbors, the authors calculated the Euclidian distance b/w test point and reference point. Rajiv Kumar et al [3]showed how segmentation can be done in character recognition of Gurmukhi script. According to R.K Sharma et al [4] that with the help of various features like diagonal, directional & zoning & K-NN, Bayesian classifiers the handwriting of

writers can be compared. Narang et al [5] projected an approach based on parts or technique suitable for scene images. The author has taken the corner points which are served as parts and these parts are found to make a part based model. N. Kumar et al [6] has discussed a number of feature extraction techniques & classifiers like power arc fitting, parabola arc fitting, diagonal feature extraction, transition feature extraction, k-NN and SVM. M. Kumar et al[7]projected a techniqueto recognize isolated characters using horizontally & vertically peak extent features, centroid features &diagonal features. A. Dixit et al [8] proposed a feature extraction & classification approach named as the wavelet transform. The proposed technique achieves an accuracy of 70%. N.Arica et al [9] proposed a method for offline cursive handwriting recognition. Firstly, authors foundvarious parameters like baselines, slant angle, stroke width and height. Afterthat, character segmentation paths were found by merging both gray scale and binary information.R.Sharma et al [10] projected two stages for the recognition. In the first phase unidentified strokes are recognized and in the second stage the author has evaluated the characters with the help of strokes that are found in the first stage. Using the elastic matching a maximum recognition rate of 90.08% is achieved. M. Kumar et al [11] proposed two schemes for feature extraction namely,power and parabola curve fitting & the classifier used in the work is k-NN & SVM.

5. Proposed Technique &Methodology

In order to correctly recognize the characters a methodology has been developed and is divided into three main stages:

- Pre-Processing
- Processing
- Post-Processing

The flowchart for the proposed research methodology has been shown in fig.2 as following:

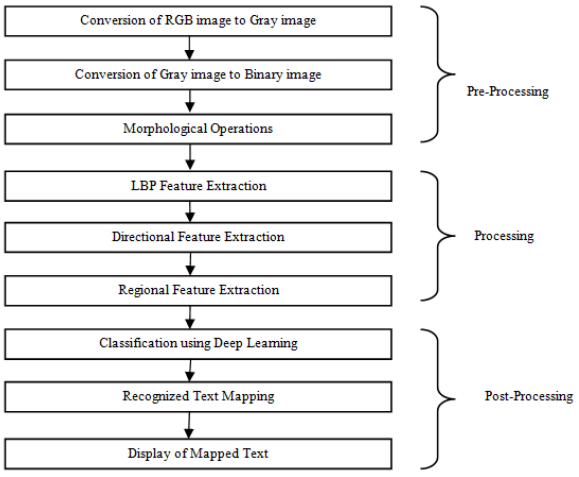

Fig. 2: Proposed Technique & Methodology

5.1. Pre-processing

Initially an image containing Gurmukhi text has been taken and this is passed to a preprocessing stage. Pre-processing involves a number of steps like conversion of images from RGB to gray, gray to binary, edge detection, area opening ,image dilation & holes filling etc. The flowchart for the proposed research methodology has been shown in fig.3.

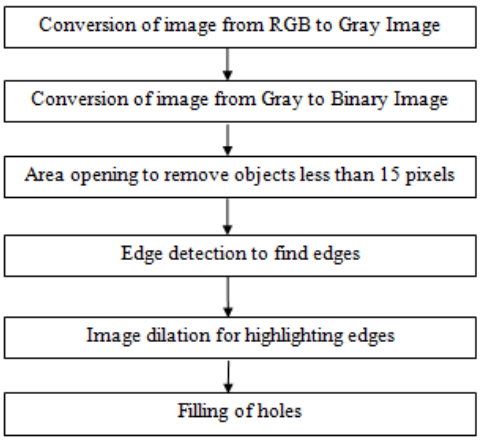

Fig. 3: Steps Involved in Preprocessing

The various operations involved in Pre-processing have been shown in fig.4 as following:

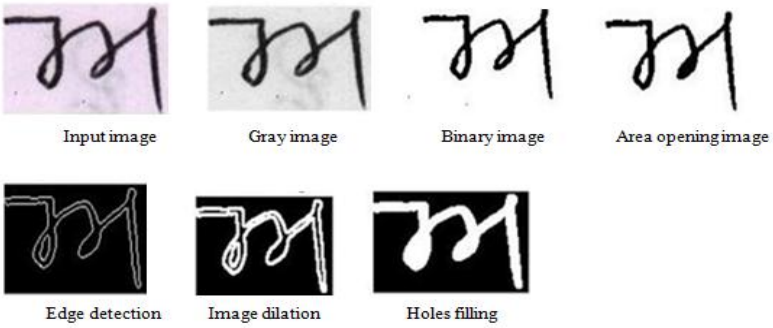

Fig.4: Preprocessing Operations on Input Image

5.2.Processing

After preprocessing stage, processing takes place in which feature extraction is performed. In the proposed work three types of features have been extracted namely LBP (Local binary pattern), directional features and regional features. A total of 117 features have been extracted in feature extraction stage. Out of 117 features 59 features are LBP features, 54 features are directional features and 4 features are regional features. So for each image 117 features have been extracted which will be utilized in the next stage.

5.2.1. LBP Features

For every pixel of an image, a 3x3 neighborhood is taken,&a binary code in terms of 0's & 1's is produced to make a new matrix with the new value (binary to decimal value). To determine the decimal value, the centre pixel is taken as reference.The following formula has been used to find the decimal values for LBP.

$$LBP_{p,r}(N_c) = \sum_{p=0}^{P-1} g(N_p - N_c)2^p \qquad (1)$$

Where

N_c---→is the center pixel value

N_P----→is the Neighbor pixels in each block thresholded by N_c

p---→ Sampling point (e.g. p=0, 1, 2------7 for 3x3 block, where P=8)

r----→ radius for 3x3 block, it is 1

Here g(x) is defined as

$$g(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 1 \end{cases} \qquad (2)$$

For 8 bit data, 2^8 i.e. 256 patterns can be found. These patterns can be categorized as: uniform patterns & non uniform patterns. A uniform pattern may have a maximum of two bitwise transitions i.e. 0 to 1 or 1 to 0.The patterns 11111111, 11110000 and 11001111 are uniform as these patterns have a maximum of 2 bitwise transitions. A non uniform pattern has more than 2 bitwise transitions. The uniform and uniform patterns can be understood by the fig.5 as following:

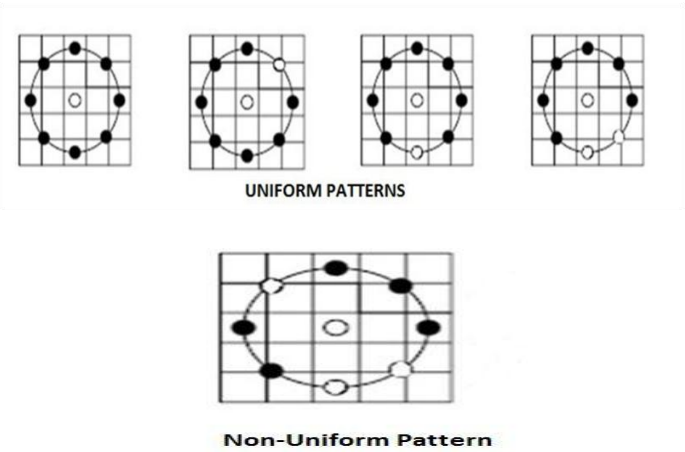

Fig.5: Uniform & Non uniform Patterns

Based on these patterns an LBP histogram is created. For every uniform pattern, a separate bin has been assigned and a single bin has been assigned to all non-

uniform patterns. By using these uniform & non uniform patterns, the length of the feature vector for a particular cell is reduced from 256 to 59.

The method for finding the local binary patterns for the centre pixel value '6' is shown in fig.6. Neighbors whose values are alike or bigger than '6' has been assigned a binary 1 & neighbor having a value less than the '6' have been assigned a binary zero. The LBP value for center pixel '6' is given by

LBP (6) = 1+4+16+32+64+128=245

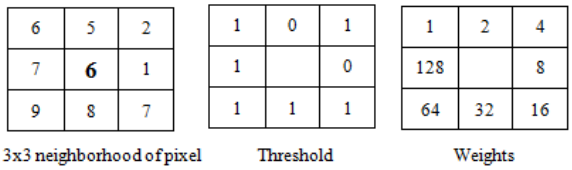

Fig.6: Method for Finding Local Binary Patterns

So for the centre pixel value '6' the binary pattern is given by 11110101, which is a non uniform pattern as it has more than two bitwise transitions. In the same way, the LBP values have been calculated for all the pixels of the image.

5.2.2. Directional Features

For finding the directional features the input image is partitioned into three equal row windows (R1, R2, and R3) and three equal column windows (C1, C2, C3).This is shownin fig.7

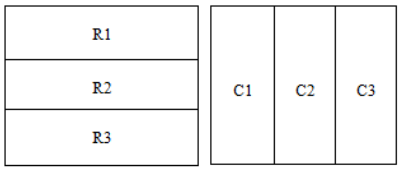

Fig. 7: (a) Row Image (b) ColumnImage

Each row and column windowhas further many rows &columns. It is well known that deep classifiers necessitate homogeneous vector size for training, so a new method has been created to develop suitable feature vectors. Initially, the character image is zoned into equal sized row and column windows. However, if the image size is not divisible by three, then additional background pixels are padded along the length of columns & rows of image [12]. From every row and column window, an vector is formed comprising of nine feature values as following: (a) Number of vertical lines (b) Number of horizontal lines (c) Number of right diagonal lines (d) Number of left diagonal lines (e) Total length of vertical lines (f) Total length of horizontal lines (g) Total length of right diagonal lines (h) Total length of left diagonal lines and (i) Number of intersection points. Here four features are corresponding to the number of lines and four features are for the length of lines and remaining one feature is for the number of intersection points. For the features corresponding to the number of lines, a value is calculated in the range of (-1, 1) using the following formula:

$$V = 1 - (\left(\frac{k}{10}\right) * 2) \quad (3)$$

WhereV=valueand k=number of lines.

And the length of each line segment is determined using the following formula:

$$L = {}^{n}/_{2w} (4)$$

Where
L=length of line segment
n= number of pixels in particular direction
w= length or width of window
So corresponding to six rows and columns windows, total of 54 features have been extracted.

5.2.3 Regional Features

In regional features, four features have been extracted namely Euler number, orientation, extent and eccentricity**.**All these values can be found by using a command in MATLAB named '**regionprops**'. The feature extraction for the Gurmukhi character **"ਅ" is** shown in table.1.

Table 1: FeaturesExtracted for Character "ਅ"

LBP Features			**Directional Features**			**Regional Features**
34	0	0	0.6	1	0.339286	26
11	0	0	0.8	0.05	0	0
0	0	0	0.6	0.475	0.064591	1
0	0	0	1	0.275	0.8	0
14	0	0	0.15	0	0.6	
0	0	0	0.425	0.046136	1	
0	0	2185	0.225	0.6	1	
0	0	35	0	0.8	0.296296	
0	0		0.046136	0.4	0.592593	
0	0		0	1	0	
0	0		1	0.192308	0	
11	0		0.4	0.307692	0.031142	
0	0		1	0.269231		
0	0		0.533333	0		
0	0		0	0.029988		
0	0		0.2	0.2		
0	0		0	0.6		
0	0		0.034602	0.2		
0	0		0.6	1		
0	0		0.2	0.25		
0	0		0.6	0.178571		

5.3. Post-Processing

In the post processing stage, classification & mapping operations take place. For the classification stage, deep neural networks have been used.

Deep Learningis basically a neural network with numerous layers of nodes between input and output .The function of these layers is to do feature identification and processing in a sequence of stages.Normal neural networks usually have one to two hidden layers whereas deep neural networks have several layers.

Here 117 hidden layers have been taken for training purpose in order to deep process the network. The hidden layers and the number of features extracted are equal in number.Training of classifier has been done using an auto encoder layer, followed by a Softmax layer. After that the character image to be recognized is passed through testing phase. The steps for training and testing phase are given below.

5.3.1 Training Phase

For training 1800 images of Gurmukhi text have been used. Inthis network training function has been taken namely, 'Trainscg' whose function is to update the weight and bias values. This function can train the network till its weight, net input, and transfer functions have derivative functions.

Steps for Training

Step 1: Input to neural network is Gurmukhi image with 117 features. In our work input is a matrix 'X' of size 117x 1800 showing that for training it is using 1800 images having 117 features for each image.

Step2: These features are passed to an auto-encoder. Theauto encoder is trained with a hidden layer of size 117. A stacked auto-encoder is trained to copy its input to its output. It contains a hidden layer 'h' that describes a code used to signify the input. Auto encoders are trained with the following options:

(i) Hidden size = 80 i.e. is the number of neurons in the hidden layer and is specified as positive number.

(ii) Sparsity Regularization: Coefficient that controls the impact of the sparsity regularize in the cost function, specified as the comma-separated pair consisting of 'Sparsity Regularization' and a positive scalar value. Its value has been set to 4.

(iii) Sparsity Proportion: Desired proportion of training examples which a neuron in the hidden layer of the auto-encoder should activate in response to. It must be in between 0 and 1.A low Sparsity proportion encourages higher degree of Sparsity. Its value has been set to 0.05.

(iv) Decoder Transfer Function: Three decoder transfer function can be used i.e. 'logsis', 'satlin', 'purelin'. In this algorithm, 'purelin' is used which is a linear transfer function and is defined by$f(z) = z$.

For training the autoencoder1, the MATLAB command used is

```
autoenc1 = trainAutoencoder(X, hiddenSize...
'SparsityRegularization', 4...
'SparsityProportion', 0.05...
'DecoderTransferFunction','purelin');
```

Step 3: Features are extracted in the hidden layer using the following MATLAB command:

```
features1 = encode (autoenc1, X);
```

Step 4: Now the second auto encoder is trained using the features extracted from the first auto encoder using the same command 'trainAutoencoder'.For this the parameters used are same as used in first auto encoder except the hidden size. The hidden size used in this case is 40.

Step5: Now the Features are extracted in the hidden layer of second auto encoder using the command 'encode'.

```
features2 = encode (autoenc2, features1);
```

Step6: Nowfor classification the Softmax layer is trained using the features extracted from the second auto encoder using the following MATLAB command:

```
softnet = trainSoftmaxLayer(features2,T,'LossFunction','crossentropy');
```

Where 'LossFunction' is an error between model output and measured response and 'crossentropy' is used to calculate neural network performance with given target and outputs

Step7: Now the encoders andSoftmax layer is stackedto make a deep network using the following command:

```
deepnet = stack(autoenc1,autoenc2,softnet);
```

Step8: Two auto-encoders and Softmax layer are stacked to form deep network. The first layer of a stacked auto encoder tends to learn first-order features in the raw input .The second layer of a stacked auto encoder tends to learn second-order features corresponding to patterns in the appearance of first-order features.

Step9: Deep network is trained on input and target data and accuracy is calculated using it.

The simulation snapshots involved in training are shown in fig.8

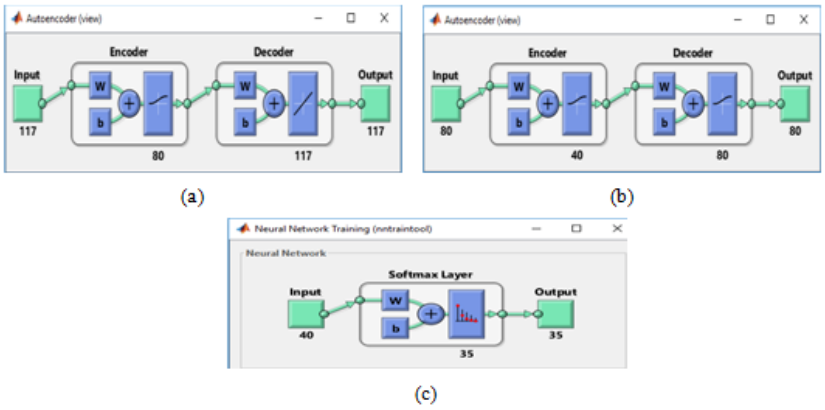

Fig. 8: Simulation Involved in Training (a) Training First Autoencoder (b) Training Second Autoencoder (c) Deep Network

5.3.2 Testing Phase

For testing, around 900 images have been taken which includes broken characters and characters written with different writing styles. For this, the image which is to be recognized is gone through various preprocessing steps as already discussed in section 5.1.After preprocessing, allthe 117 features i.e. directional features, LBP features and regional features are extracted from the character image as already discussed in section 5.2.Now the extracted features are passed to the trained network obtained in section 5.3.1 to classify the character image. The recognized character is mapped with the Devanagari text with the help of a look up table. The GUI for recognizing the character 'ग' is shown in fig.9 .Its mapping with Devanagari text is displayed in form of .txt file as shown in fig.10.In the same way, all the images have been tested and 99.3% accuracy has been achieved. The accuracy can be shown through confusion matrix .Confusion matrix is a table which is frequently used to depict the performance of a classification (or "classifier") on a set of test data. The confusion matrix for the proposed work is depicted in results and discussions part (section 6).

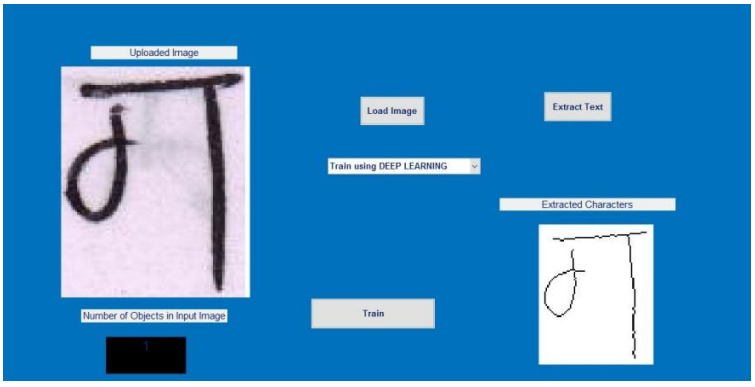

Fig.9: GUI for Recognizing Handwritten text "ग"

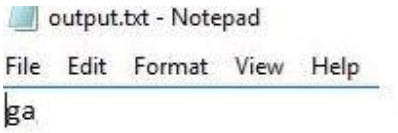

Fig.10: Mapping of Recognized Text with Devanagari Script

6. Results &Discussion

In order to correctly recognize the text, the data set used is of 2700 samples. Out of 2700 samples 1800 samples have been utilized in training & 900 samples have been used for testing purpose. The proposed system is achieving an accuracy of 99.3% and this accuracy has been depicted in confusion matrix as shown in fig.11.

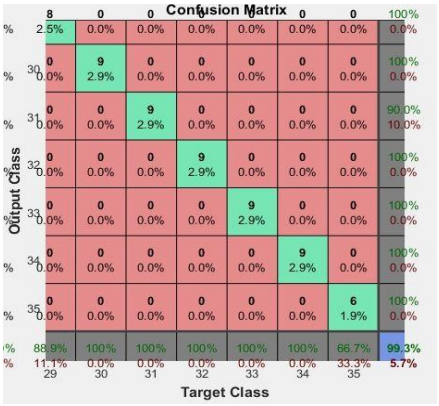

Fig. 11: Confusion Matrix

The proposedmethod is compared with the existing approaches and its comparison results are shown in table.2

Table2: Comparison of Proposed Method with Existing Approaches

Authors	Accuracy
Kumar M et.al [13]	94.12 % using K-NN
Sinha G et.al [14]	95.11% using SVM 90.64% using K-NN
Sahu N et.al [15]	75.6 % using ANN
Mahto MK et.al [16]	98.06% using joint horizontal and vertical projection feature extraction technique
Proposed Method With Deep Learning	99.3% using Deep neural networks

From the simulation results it can be seen that the proposed algorithm with the trained Softmax architecture of deep learning results in better accuracy in comparison with other existing techniques. The same can be inferred from the graph shown in fig12.

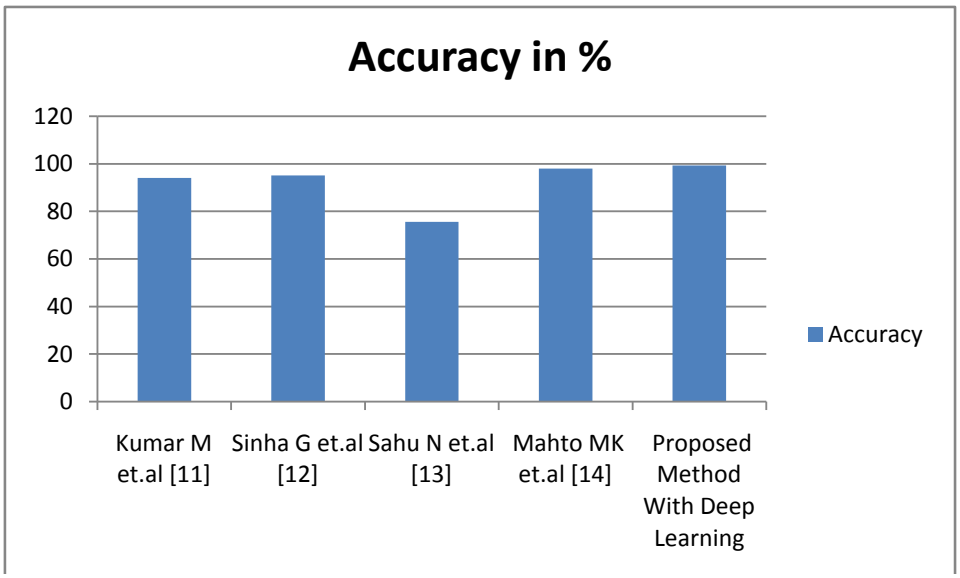

Fig.12: Comparison of Accuracy of Proposed Method with Previous Techniques

7. Conclusion &Future Scope

In this work, an efficient method for the recognition of offline handwritten Gurmukhi characters (including broken characters) has been proposed. The classifier used in the work is deep neural network that has been trained with 117 features. In the proposed work, three types of features, namely LBP features, Directional features and regional features have been extracted. Using these three types of features the recognition accuracy has been considerably increased. The proposed system is achieving an accuracy of 99.3%.Furthermore, the present work is limited to the recognition of single characters (including broken characters).This work can be extended to further word level recognition and in speech to text applications.

References

[1] Garain U., Chaudhuri B.B., Segmentation of touching characters in printed Devnagari and Bangla scripts using fuzzy multifactorial analysis, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 32(4) (2002), 449-459.

[2] Kumar M., Jindal M.K., Sharma R.K., k-nearest neighbor based offline handwritten Gurmukhi character recognition, International Conference on Image Information Processing, Himachal Pradesh (2011), 1-4.

[3] Kumar R., Singh A., Detection and segmentation of lines and words in Gurmukhi handwritten text, IEEE 2nd International Advance Computing Conference (2010), 353-356.

[4] Kumar M., Jindal M.K., Sharma R.K., Classification of characters and grading writers in offline handwritten Gurmukhi script, International Conference on Image Information Processing, Himachal Pradesh (2011), 1-4.

[5] Narang V., Roy S., Murthy O.V.R., Hanmandlu M., Devanagari Character Recognition in Scene Images, 12th International Conference on Document Analysis and Recognition, Washington (2013), 902-906.

[6] Kumar N., Gupta S., Offline Handwritten Gurmukhi Character Recognition: A Review, International Journal of Software Engineering and Its Applications 10(5) (2016), 77-86.

[7] Kumar M., Sharma R.K., Jindal M.K., A Novel Hierarchical Technique for Offline Handwritten Gurmukhi Character Recognition, National Academy Science Letters 37(6) (2014), 567-572.

[8] Surendar, A., Kavitha, M. “Secure patient data transmission in sensor networks”, (2017), Journal of Pharmaceutical Sciences and Research, 9 (2), pp. 230-232.

[9] Surendar, A.”FPGA based parallel computation techniques for bioinformatics applications”,(2017) International Journal of Research in Pharmaceutical Sciences, 8 (2), pp. 124-128. .

[10] Sharma A., Kumar R., Sharma R.K., Online Handwritten Gurmukhi Character Recognition Using Elastic Matching, Congress on Image and Signal Processing (2008), 391-396.

[11] Kumar M., Sharma R.K., Jindal M.K., Efficient Feature Extraction Techniques for Offline Handwritten Gurmukhi Character Recognition, National Academy Science Letters 37(4) (2014), 381-391.

[12] Blumenstein M., Verma B., Basli H., A novel feature extraction technique for the recognition of segmented handwritten characters, Seventh International Conference on Document Analysis and Recognition (2003), 137-141.

[13] Kumar M., Jindal M.K., Sharma R.K., k-nearest neighbor based offline handwritten Gurmukhi character recognition, International Conference on Image Information Processing, Himachal Pradesh (2011), 1-4.

[14] Sinha G.,Rani R., Dhir R., Handwritten Gurmukhi Character Recognition Using K-NN & SVM Classifier, International Journal of Advanced Research in Computer Science & Software Engineering 2(6) (2012), 288-293.

[15] Sahu N., Raman N.K., An efficient handwritten Devnagari character recognition system using neural network, International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (2013), 173-177.

[16] Mahto M.K., Bhatia K., Sharma R.K., Combined horizontal and vertical projection feature extraction technique for Gurmukhi handwritten character recognition, International Conference on Advances in Computer Engineering and Applications, Ghaziabad (2015), 59-65.