

# Recognition of online Handwritten Bangla Characters using Hierarchical System with Denoising Autoencoders

Arghya Pal, Dr. J. D. Pawar

Department of C. S. T

Goa University

Goa 403 206, India

E-mail: arghyapal5@gmail.com

**Abstract**—This work describes the recognition of online handwritten Bengali characters using Deep Denoising Autoencoder with Multilayer Perceptron (MLP) trained through backpropagation algorithm [1]. Initial pre-training has been done to the Denoising Autoencoder with MLP trained through backpropagation algorithm, to bring the weights of the Deep network to some good solution and then pre-trained Denoising Autoencoders are stacked to form a Deep Denoising Autoencoder (DDA). A final classification layer makes DDA to a Deep Classifier (DC) followed by a final fine-tune that gives the best classifier for the job of classification of Bengali characters. The overall system is hierarchical in nature and the system has been trained in two phase where the first phase has trained a broad classifier and in the second phase class specific recognizer has been trained. At the testing phase in this hierarchical approach, first a broad classifier has been used to recognize broad classes like Vowel, Consonant, Special Symbol and Numeral for a novel test sample. Once the broad class gets recognized then a class specific recognizer has been used to recognize the exact character the test sample belongs. Recognition performance of the hierarchical system is 93.12%.

**Keywords**-MLP; Denoising Autoencoder; Deep Network; Bengali Character; Classification.

## I. INTRODUCTION

Autoencoders are machine learning tools trained to capture meaningful information from the input data distribution. A traditional Autoencoder has three basic parts: one input layer, one hidden layer and one output layer. If the number of hidden layer in an Autoencoder is more than one the Autoencoder is considered to be Deep. Recognition of Online Handwritten Bengali Numeric character using Deep Denoising Autoencoder using Multilayer Perceptron (MLP) trained through back propagation algorithm has been proposed in [1]. In that experiment Denoising Autoencoder has been developed by first partially corrupting a fixed number of input vector elements using a stochastic mapping and then this input representation has been passed to a traditional Autoencoder. Traditional Autoencoder with MLP trained through backpropagation algorithm proposed in [2] uses a nonlinear mapping function that makes a novel representation at hidden layer from which the reconstruction has been made. In [3] it is stated that

Denoising Autoencoders “are still minimizing the same reconstruction loss between a clean  $X$  and its reconstruction from  $Y$ ”, but forces the learning to capture more robust features from input distribution. Following Deep representation in mind that has been proposed in [4] pre-trained Denoising Autoencoders has been stacked to form a Deep Denoising Autoencoder. After addition of a classifier Deep Denoising Autoencoder has been converted as a Deep Classifier followed by a final fine-tuning. In [2] Deep Autoencoder network with MLP trained through backpropagation algorithm is different from greedy layer wise pre-training proposed in [5] that have used Deep Autoencoder with Restricted Boltzmann Machine (RBM). This present study has used Deep Denoising Autoencoder (DDA) with MLP trained through backpropagation algorithm has been proposed in [1] to recognize online handwritten Bengali characters such as vowel, consonants, special symbols and numerals. A hierarchical approach has been developed in this experiment, like a broad classifier has been developed to first recognize broad categories like vowel, consonants, special symbols and numerals. So, for a given test sample if the test sample has been predicted by the broad classifier as one of the broad class then a class specific recognizer has been used to recognize which Bengali character the test pattern belongs.

ANN called multilayer perceptrons has been applied in the field of handwritten character recognition to recognize benchmark dataset MNIST [6]. After that using novel elastic training image deformation [7] Convolution Neural Network (CNN) has achieved a record breaking performance. In the Indian context some benchmark research like; [8] have employed a Multi Layer Perceptron (MLP) based classifier to recognize Bangla and Arabic numerals, [9] have proposed a hierarchical approach for handwritten Bangla characters recognition just to mention a few. Though there is a small amount of work in the new paradigm of ANN that is Deep Neural Network to classify on-line handwritten character recognition systems. In [10] it is stated that to learn complex functions which represents high-level abstractions like Natural language processing, Machine Vision etc. deep architectures are needed.

Several research have been done to recognize online handwritten characters from different perspectives include HMM based handwriting recognition systems for Bangla

[11], Malayalam [12], Telugu [13], Tamil [14], and Assamese [15], using support vector machines (SVMs) for Telegu and Devnagari [16], Principal Component Analysis (PCA) for Tamil script [16]. In [17] HMM and SVM based system for Telegu script has been depicted. In [18] has used HMMs as feature extractors and then the HMM output features together with global features were combined and passed to SVM for classification. Hybrid SVM-HMM systems have also been developed in [19] and [20]. There are some ground breaking researches on handwritten Bengali characters using MLP [8] [22] but those mainly focus on offline recognition,

In this present study, classification of online handwritten Bengali numerals is performed by using a hierarchical system constituted with the DDA based on MLP as a classifier. The input features consist of pre-processed (x, y) co-ordinates along with the first and second derivatives of each pre-processed point proposed in [23]. Initially layer-wise pre-training of the Denoising Autoencoder is employed to bring the weights of the layers to some good solution. Then the pre-trained Denoising Autoencoders are unrolled to build a deep autoencoder following a backpropagation to fine tune the weights of the network. The addition of the final output layer converts the DDA to a DC.

The paper is organized as follows: Section II describes the data collection and feature extraction method. Development of DDA classifier is described in section III. Section IV explores the experiment with DDA. The summary and conclusion is given in section V.

## II. DATA COLLECTION AND FEATURE EXTRACTION

Data has been collected for 11 vowels, 35 consonants, 10 numeric characters and 10 special symbols depicted in Fig. 1. Temporally ordered sequence of pen coordinates (x, y) for an isolated character has been taken using a digitizer such as electronic tablet-stylus without imposing any constraint to the writer. Isolated characters can be written in one or more stroke (stroke is the locus from one PEN-DOWN to PEN-UP). But, this present study has considered the full character and not recognition by analyzing strokes for a given sample. To perform this experiment 100 writers data has been considered for training the system, 50 writers data for validation and 30 writers data has been considered to test the system and no mixture of training, test and validation data has been done. A well mixture of frequent writer (i.e. regular writer), non-frequent writer and collection from different age group, profession been collected aiming to make the system more robust. The data has been collected using two sessions and each session each writer has given one single character from the total characters in Fig. 1. So, this implies that for each character there are 100 training data, 50 validation data and 30 testing data. Each class consists of same number of training sample to mitigate biasness among classes. For choosing parameter values and number of epochs in the fine-tuning phase validation set is used.

After collection of raw data next crucial step is preprocessing. In order to find a fixed length feature vector for the purpose of discriminating patterns, preprocessing

|     |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |    |    |
|-----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|----|----|
| 1   | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11  | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| ০   | ১  | ২  | ৩  | ৪  | ৫  | ৬  | ৭  | ৮  | ৯  | অ   | আ  | ই  | ঈ  | উ  | ঊ  | ঋ  | এ  | ঐ  | ঔ  |
| 21  | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31  | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| ক   | খ  | গ  | ঘ  | ঙ  | চ  | ছ  | জ  | ঝ  | ট  | ঠ   | ড  | ঢ  | ণ  | ত  | থ  | দ  | ধ  | ন  | প  |
| 41  | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51  | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| ব   | ভ  | ম  | য  | র  | ল  | শ  | ষ  | স  | হ  | ক্ষ | য় | ড় | ঢ় | !  | '' | -  |    |    |    |
| 61  | 62 | 63 | 64 | 65 | 66 | 67 |    |    |    |     |    |    |    |    |    |    |    |    |    |
| ( ) | @  | □  | ?  | €  | ·  |    |    |    |    |     |    |    |    |    |    |    |    |    |    |

Figure 1. Bengali Numerals (1-10), Vowels (11-21), Consonants (22-57), Special Characters (58-67).

steps like: normalization, smoothing, removal of duplicate points and re-sampling has been done to the collected raw data, as described in [23]. To make the description complete, all the processes mentioned above is discussed below.

1) Size normalization of raw data: Online handwritten characters can have large variations of shape, size and position due to great degrees of freedom in writing styles. So, an initial normalization has been done to perform further operations.

2) Smoothing: Following the literature value described in [23] an average filter of size three has been moved all over the image, in x and y direction separately for the purpose of smoothing.

3) Elimination of unwanted points: Unwanted points like overwritten points, repetition points, redundant point, unwanted noise from the part of user (nail, bangle etc.) has been removed in that process.

4) Re-sampling: In this step first cumulative distance has been calculated along the trajectory followed by interpolation of missing points by the calculated cumulative distance until the distance between any two points are less than one. But starting and end point has been unaltered as they contain vital information.

After preprocessing of raw data a fixed 60 preprocessed  $(x_p, y_p)$  coordinates along with first derivative  $(x'_p, y'_p)$  and second derivative of each preprocessed point  $(x''_p, y''_p)$  makes feature set following the literature [?]. So, input dimension is 360 that is  $(x_p, y_p, x'_p, y'_p, x''_p, y''_p)$  and each of them has 60 points, so  $60 \times 6 = 360$  both for modeling and testing. This set of parameters has already been studied in [23] and have reported a major overall information gain to discriminate numeric characters. In the next section using these extracted features experiment with DDA is explained.

## III. TRAINING OF HIERARCHICAL RECOGNIZER USING DDA WITH MLP TRAINED THROUGH BACKPROPAGATION

DDA have been built using the open source platform SNNS [24] that interact with users through a graphical interface and **batchman** (A command interpreter for batch jobs, included in SNNS) to deal minibatch creation and

noise input during pre-training process. The hierarchical recognizer has been built by following steps:

Step 1: First creating a broad classifier that enables the system to recognize broad class such as: Vowel, Consonants, Numerals and Special Symbol using a DDA with MLP trained through backpropagation algorithm. So input for this Broad recognizer is the sample and output class consists of 5 classes. Broad classifier has been trained using training samples, in this case vowel class has 1100 (i. e. 11 X 100), numeric characters has 1000 (i. e. 10 X 100), special characters has 1000 (i. e. 10 X 100) and consonants has 3500 (i. e. 35 X 100) samples. That means in this Broad classifier all Vowels in Fig. 1 has been considered in one class, all Numerals has been considered in one class, all Consonants has been classified in one Consonant class and so on.

Step 2: Once a sample has been recognized by the Step 1 broad classifier, the sample has been passed by the hierarchical system to a class specific recognizer. Below from Step 2.1 to Step 2.4 each time the system should take one of those steps to recognize the sample.

Step 2.1: A Vowel recognizer has been developed with DDA with 11 classes to discriminate. So, if the sample gets recognized by the broad classifier as Vowel then it should pass to the Vowel character recognizer. Vowel recognizer has been trained with 100 samples each and validated by using 50 samples each class of the Vowel classifier. Confusion Matrix is depicted in Fig. 8.

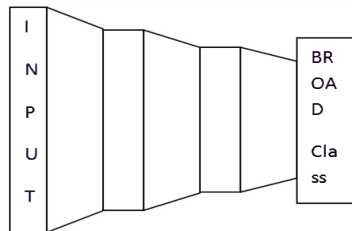


Figure 2. Broad class recognizer to recognize Vowel, Consonants, Numerals and Special Symbols.

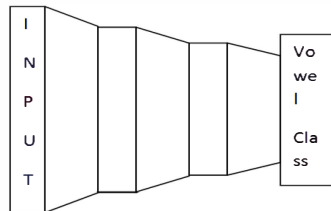


Figure 3. Vowel class recognizer to recognize Vowel character from the input pattern.

Step 2.2: If the sample gets recognized by the broad classifier as Numeral then it should pass to the Numeric character recognizer and Numeral has been developed with DDA with 10 classes to discriminate, see Fig. 3. Numeral classifier has been trained using 100 samples each numeric character and validated by using 50 samples each numeric character. Confusion matrix is depicted in Fig. 9.

Step 2.3: Similarly for Special symbol recognizer has been developed with DDA with 10 classes to discriminate see Fig. 4. Special Symbol classifier has been trained using 100 samples each Special Symbol character and validated

by using 50 samples each Special Symbol character. Confusion matrix is depicted in Fig. 10.

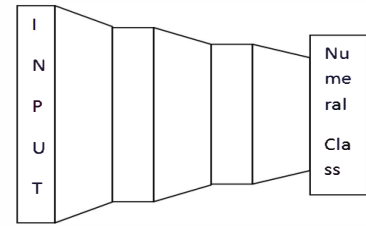


Figure 4. Numeral class recognizer to recognize Numeric characters from the input pattern.

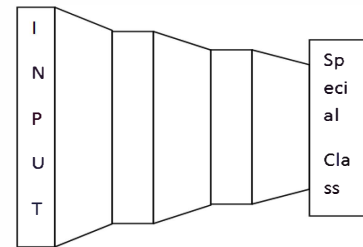


Figure 5. Special class recognizer to recognize Special characters from the input pattern.

Step 2.4: If the sample is recognized as consonants then Step 2.4 has been called and this recognizer has 35 classes to discriminate. Consonant classifier has been trained using 100 samples each consonant character and validated by using 50 samples each consonant character.

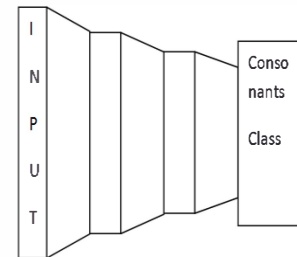


Figure 6. Special class recognizer to recognize Special characters from the input pattern.

Backpropagation algorithm free parameters are: Learning Rate (LR), Momentum (M), Flat Spot Elimination (FSE) and Tolerance (T). Graphics of SSE, validation set performance, initial typical values were analyzed in order to get smoother graphics and deeper local minimum with some good LR, M, FSE, T parameter values. Validation set data has been used to fix those parameter values.

#### IV. TESTING WITH DDA WITH MLP TRAINED THROUGH BACKPROPAGATION AND RESULTS

An experiment has been carried out with the test data of the hierarchical recognizer. DDA has been built using the line of [1] that is a fully connected network having one input layer, two hidden layers and one output layer and this structure has been maintained for all recognizers described in Section III. Testing of the Broad classifier with 1050 (30 X 35 = 1050, i. e. 30 users data and 35 total

Bengali characters) data that has been collected during data collection process and is depicted in Table I and confusion matrix is shown in Fig. 7. Testing of the Vowel classifier with 330 (30 X 11 = 330, i. e. 30 users data and 11 vowel characters) data that

TABLE I. DIFFERENT CONFIGURATIONS STUDIED USING DDA TO RECOGNIZE BROAD CLASSIFIER

| Hidden Layers                    | Configurations (360:input dimension, 5:output class) | Performance  |
|----------------------------------|--|--------------|
| Two hidden layer $h_1$ and $h_2$ | 360+150+30+5   | 99.01%       |
|                                  | 360+100+80+5   | 99.04%       |
|                                  | <b>360+90+40+5</b>                                   | <b>99.9%</b> |
|                                  | 360+200+100+5  | 99.27%       |

TABLE II. DIFFERENT CONFIGURATIONS STUDIED USING DDA TO RECOGNIZE VOWEL CLASSIFIER

| Hidden Layers                    | Configurations (360:input dimension, 11:output class) | Performance  |
|----------------------------------|---|--------------|
| Two hidden layer $h_1$ and $h_2$ | 360+150+30+11   | 89.01%       |
|                                  | 360+100+80+11   | 90.42%       |
|                                  | <b>360+100+40+11</b>                                  | <b>93.9%</b> |
|                                  | 360+200+100+11  | 91.27%       |

TABLE III. DIFFERENT CONFIGURATIONS STUDIED USING DDA TO RECOGNIZE NUMERIC CLASSIFIER

| Hidden Layers                    | Configurations (360:input dimension, 10:output class) | Performance  |
|----------------------------------|---|--------------|
| Two hidden layer $h_1$ and $h_2$ | 360+150+30+10   | 92.68%       |
|                                  | 360+100+80+10   | 93.22%       |
|                                  | <b>360+100+40+10</b>                                  | <b>94.9%</b> |
|                                  | 360+200+100+10  | 92.97%       |

TABLE IV. DIFFERENT CONFIGURATIONS STUDIED USING DDA TO RECOGNIZE SPECIAL CHARACTERS CLASSIFIER

| Hidden Layers                    | Configurations (360:input dimension, 10:output class) | Performance  |
|----------------------------------|---|--------------|
| Two hidden layer $h_1$ and $h_2$ | 360+150+30+10   | 94.81%       |
|                                  | 360+100+80+10   | 94.94%       |
|                                  | <b>360+80+40+10</b>                                   | <b>96.3%</b> |
|                                  | 360+200+100+10  | 95.07%       |

TABLE V. DIFFERENT CONFIGURATIONS STUDIED USING DDA TO RECOGNIZE CONSONANT CHARACTERS CLASSIFIER

| Hidden Layers                    | Configurations (360:input dimension, 35:output class) | Performance  |
|----------------------------------|---|--------------|
| Two hidden layer $h_1$ and $h_2$ | 360+150+30+35   | 88%          |
|                                  | 360+100+80+35   | 87.03%       |
|                                  | <b>360+80+40+35</b>                                   | <b>89.7%</b> |
|                                  | 360+200+100+35  | 86.7%        |

|     | (0) | (1) | (2)  | (3) | (4) |
|-----|-----|-----|------|-----|-----|
| (0) | 100 | -   | -    | -   | -   |
| (1) | -   | 100 | -    | -   | -   |
| (2) | -   | 0.5 | 99.5 | -   | -   |
| (3) | -   | -   | -    | 100 | -   |
| (4) | -   | -   | -    | -   | 100 |

Figure 7. Confusion Matrix with Broad classifier using 360+90+40+5

|      | (0) | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) | (10) |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|
| (0)  | 92  | -   | -   | -   | -   | -   | -   | -   | -   | -   | -    |
| (1)  | -   | 96  | -   | -   | -   | -   | -   | -   | -   | -   | -    |
| (2)  | -   | -   | 90  | -   | -   | -   | -   | -   | -   | -   | -    |
| (3)  | -   | -   | -   | 99  | -   | -   | -   | -   | -   | -   | -    |
| (4)  | -   | -   | -   | -   | 92  | -   | -   | -   | -   | -   | -    |
| (5)  | -   | -   | -   | -   | -   | 96  | -   | -   | -   | -   | -    |
| (6)  | -   | -   | -   | -   | -   | -   | 99  | -   | -   | -   | -    |
| (7)  | -   | -   | -   | -   | -   | -   | -   | 90  | -   | -   | -    |
| (8)  | -   | -   | -   | -   | -   | -   | -   | -   | 95  | -   | -    |
| (9)  | -   | -   | -   | -   | -   | -   | -   | -   | -   | 90  | -    |
| (10) | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | 94   |

Figure 8. Confusion Matrix with Vowel classifier using 360+100+40+11

|     | (0) | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| (0) | 96  | -   | -   | -   | -   | -   | -   | -   | -   | -   |
| (1) | -   | 96  | -   | -   | -   | -   | -   | -   | -   | -   |
| (2) | -   | -   | 93  | -   | -   | -   | -   | -   | -   | -   |
| (3) | -   | -   | -   | 99  | -   | -   | -   | -   | -   | -   |
| (4) | -   | -   | -   | -   | 94  | -   | -   | -   | -   | -   |
| (5) | -   | -   | -   | -   | -   | 94  | -   | -   | -   | -   |
| (6) | -   | -   | -   | -   | -   | -   | 92  | -   | -   | -   |
| (7) | -   | -   | -   | -   | -   | -   | -   | 90  | -   | -   |
| (8) | -   | -   | -   | -   | -   | -   | -   | -   | 93  | -   |
| (9) | -   | -   | -   | -   | -   | -   | -   | -   | -   | 93  |

Figure 9. Confusion Matrix with Numeric classifier using 360+100+40+10

|     | (0) | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| (0) | 96  | -   | -   | -   | -   | -   | -   | -   | -   | -   |
| (1) | -   | 96  | -   | -   | -   | -   | -   | -   | -   | -   |
| (2) | -   | -   | 95  | -   | -   | -   | -   | -   | -   | -   |
| (3) | -   | -   | -   | 97  | -   | -   | -   | -   | -   | -   |
| (4) | -   | -   | -   | -   | 97  | -   | -   | -   | -   | -   |
| (5) | -   | -   | -   | -   | -   | 95  | -   | -   | -   | -   |
| (6) | -   | -   | -   | -   | -   | -   | 96  | -   | -   | -   |
| (7) | -   | -   | -   | -   | -   | -   | -   | 99  | -   | -   |
| (8) | -   | -   | -   | -   | -   | -   | -   | -   | 97  | -   |
| (9) | -   | -   | -   | -   | -   | -   | -   | -   | -   | 95  |

Figure 10. Confusion Matrix with Special Characters classifier using 360+80+40+10

has been collected during data collection process and is depicted in Table II. The overall recognition performance of the system is found to be 93.12%. It has got a higher performance than a system without hierarchy using DDA with MLP trained through backpropagation algorithm. A system without hierarchy has got only 85% performance and both the training time and testing time for this system is comparably higher than the system with hierarchy.

## V. CONCLUSION AND DISCUSSION

Recognition of Bengali characters is a more towards Man-Machine-Interaction and is a part-and-parcel of Machine Vision, Artificial Intelligence and Machine Learning. Bengali character recognition finds its place in the field of census data collection, several form filling applications, question answering systems and many more. A writer can enter complex conjuncts or consonants etc from a terminal rather than typing characters from keyboard as for Indian characters entry through keyboard is a very cumbersome job. On the other hand it saves pen-and-paper and terminals like key board, as typos could be found in entry through keyboard [25].

Using this present method a hierarchical system has been built that first recognize a broad class and after recognition of the broad class a class specific recognizer has been used to recognize the exact class of the test sample.

For a future enhancement this process can be further modified to make a more robust system. Another dimension could be to analyze the present system using Dropout Autoencoder that stochastically eliminates hidden nodes to capture more robust feature from the input data distribution.

## REFERENCES

- [1] Pal, Bengali Handwritten Numeric Character Recognition Using Denoising Autoencoders, accepted in 2015 IEEE International Conference on Engineering and Technology (ICETECH'15), 20th March 2015, Coimbatore, TN, India, (accepted and in process).
- [2] G.E. Hinton, R.R. Salakhutdinov, Reducing the Dimensionality of Data with Neural Networks, 28 JULY 2006, VOL 313, SCIENCE, www.sciencemag.org.
- [3] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion, Journal of Machine Learning Research 11 (2010) 3371-3408.
- [4] G.E. Hinton, S. Osindero, Y. Teh, A fast learning algorithm for deep belief nets, Neural Computation, VOL. 18, NO. 7, PP. 1527-1554, 2006.
- [5] N.E. Cibau, E.M. Albornoz, H.L. Rufiner, Speech emotion recognition using a deep autoencoder, XV Reunión de Trabajo en Procesamiento de la Información y Control, 2013
- [6] Werbos, P. J, Beyond regression: New tools for prediction and analysis in the behavioral sciences, 1974, Unpublished doctoral dissertation, Harvard University
- [7] Simard, P. Y., Steinkraus, D., Platt, J.C., Best practices for convolutional neural networks applied to visual document analysis. In Intl. Conf. Document Analysis and Recognition (pp. 958-962), 2003, San Mateo, CA: IEEE Computer Society Press
- [8] T.K. Bhowmik, U. Bhattacharya, S.K. Parui, Recognition of Bangla Handwritten Characters Using an MLP Classifier Based on Stroke Features, ICONIP 2004, LNCS 3316, pp. 8148-19, 2004
- [9] A.F.R. Rahman, R. Rahman, M.C. Fairhurst, "Recognition of Handwritten Bengali Characters: a Novel Multistage Approach," Pattern Recognition, vol. 35, p.p. 997-1006, 2002
- [10] Y. Bengio, Learning Deep Architectures for AI, Foundations and Trends in Machine Learning, vol. 2, no. 1, pp. 1-127, 2009
- [11] S.K. Parui, K. Guin, U. Bhattacharya, and B.B. Chaudhuri, Bangla Handwritten Character Recognition using HMM, in Proc. 19th Int. Conf. on Pattern Recognition (ICIP), pp. 1-4, Tampa FL, 2008.
- [12] A. Arora and A.M. Namboodiri, Hybrid Model for Recognition of Online Handwriting in Indian Scripts, Proc. of Int. Conf. on Frontiers in handwriting Recognition, pp. 433-438, Kolkata, 2010.
- [13] V.J. Babu, L. Prasanth, R.R. Prasanth, R.R. Sharma, G.V.P. Rao and A. Bharath, HMM-based online handwriting recognition system for telugu symbols, Proc. 9th Int. Conf. on Document Analysis and Recognition (ICDAR), Curitiba, Brazil, 2007, pp. 63-67
- [14] K. Shashikiran, K.S. Prasad, R. Kunwar, A.G. Ramakrishnan, Comparison of HMM and SDTW for Tamil handwritten character recognition, Proc. Int. Conf. on Signal Processing and Communications, pp. 1-4, IISc Bangalore, India, 2010
- [15] G.S. Reddy, B. Sarma, R.K. Naik, S.R.M. Prasanna and C. Mahanta, Assamese Online Handwritten Digit Recognition System using Hidden Markov Models, Workshop on Document Analysis and Recognition, 2012
- [16] H. Swethalakshmi, A. Jayaraman, V.S. Chakravarthy and C. Chandra Sekhar, Online Handwritten Character Recognition of Devnagiri and Telegu Characters using Support Vector Machines, International Workshop on Frontiers in Handwriting Recognition, 2006.
- [17] Deepu V., S. Madhvanath, A.G. Ramakrishnan Principal Component Analysis for Online Handwritten Character Recognition, Proceedings of the 17th International Conference on Pattern Recognition, 2004.
- [18] A. Jayaraman, C. Chandra Sekhar and V.S. Chakravarthy, Modular Approach to Recognition of Strokes in Telegu Script, in Proc. of 9th International Conference on Document Analysis and Recognition, pp. 501-505, 2007
- [19] B.Q. Huang, C.J. Du, Y.B. Zhang and M.T. Kechadi, A Hybrid HMM, SVM Method for Online Handwriting Symbol Recognition, in 6th International Conference on Intelligent Systems Design and Applications (ISDA'06), vol. 1, pp. 887-891, 2006
- [20] M.F. Valstar and M. Pantic, Combined Support Vector Machines and Hidden Markov Models for Modeling Facial Action Temporal Dynamics, Human-Computer Interaction, Lecture Notes in Computer Science, Volume 4796, pp 118-127, 2007
- [21] P. Hu, W. Liu and W. Jiang, Combining frame and segment based models for environmental sound classification, in 13th Annual Conference of the International Speech Communication Association, 2012
- [22] U. Bhattacharya, M. Shridhar, and S.K. Parui, On Recognition of Handwritten Bangla Characters, ICVGIP 2006, LNCS 4338, pp. 817-828, 2006.
- [23] G.S. Reddy, B. Sarma, R.K. Naik, S.R.M. Prasanna and C. Mahanta, Assamese Online Handwritten Digit Recognition System using Hidden Markov Models, Workshop on Document Analysis and Recognition, 2012
- [24] <http://www.ra.cs.uni-tuebingen.de/SNNS/>
- [25] A. Pal, Handle Bengali Numeral(s) in Rule based Bengali Spell Checker, International Journal of Advanced Research in Computer Science and Software Engineering, pp. 989-991, 2014.