CNN Based Common Approach to Handwritten Character Recognition of Multiple Scripts

Durjoy Sen Maitra, Ujjwal Bhattacharya and Swapan K. Parui
Computer Vision & Pattern Recognition Unit, Indian Statistical Institute, 203 B. T. Road, Kolkata-108, India
durzoy@gmail.com, ujjwal@isical.ac.in, swapan@isical.ac.in

***Abstract*—There are many scripts in the world, several of which are used by hundreds of millions of people. Handwritten character recognition studies of several of these scripts are found in the literature. Different hand-crafted feature sets have been used in these recognition studies. However, convolutional neural network (CNN) has recently been used as an efficient unsupervised feature vector extractor. Although such a network can be used as a unified framework for both feature extraction and classification, it is more efficient as a feature extractor than as a classifier. In the present study, we performed certain amount of training of a 5-layer CNN for a moderately large class character recognition problem. We used this CNN trained for a larger class recognition problem towards feature extraction of samples of several smaller class recognition problems. In each case, a distinct Support Vector Machine (SVM) was used as the corresponding classifier. In particular, the CNN of the present study is trained using samples of a standard 50-class Bangla basic character database and features have been extracted for 5 different 10-class numeral recognition problems of English, Devanagari, Bangla, Telugu and Oriya each of which is an official Indian script. Recognition accuracies are comparable with the state-of-the-art.**

I. Introduction

Recognition of offline images of handwritten characters has been studied [1] by the document analysis research community during the last few decades. Several automatic systems of offline handwriting recognition are now available in the market. However, these systems provide solutions mainly for a few major scripts of the world such as English [2], Chinese [3], Arabic [4], Japanese [5] etc. Still the recognition problems of these scripts cannot be considered to be entirely solved. Due to the enormous variability in handwriting styles often the state-of-the-art handwriting recognition technologies fail to provide satisfactory performance on various types of handwriting samples. On the other hand, there are many scripts which are used by hundreds of millions of people in different parts of the world for which there does not exist any automatic system for recognizing their handwritten character images. Although some recognition studies have been made for offline handwritten characters of major Indian scripts such as Devanagari and Bangla [6], [7], [8], there is no system available yet in the market for their automatic recognition. Thus, recognition studies of handwritten character image samples still remain relevant because of their enormous application potentials.

Available approaches to handwriting recognition usually consist of various steps which often include preprocessing, feature extraction, classification, and postprocessing. However, feature extraction and classifier design are the two major steps of any recognition system. Related surveys can be found in [9]. Popularly used classifiers include multilayer perceptron (MLP) [10], radial basis function (RBF) classifier [11], modified quadratic discriminant function (MQDF) [12], support vector machine (SVM) classifier [13] etc. A comparative performance study of several classifiers on handwritten digit recognition can be found in Liu et al. [14]. Features providing the state-of-the-art accuracies in handwritten character recognition tasks are chaincode direction [15], gradient [16], and curvature [17] features. In a few character recognition studies, Gabor transform [18] and statistical/structural features [19] have also been successfully used. In a recent study [20] on handwritten Devanagari and Bangla character recognition, the wavelet transforms of input character image were subjected to three MLP classifiers corresponding to three coarse-to-fine resolution levels in a cascaded manner. An input character rejected at a lower resolution level, was passed to the MLP corresponding to the next higher resolution level and in case of its rejection even at the highest resolution level, a fourth MLP combined the outputs of all three previous MLP classifiers in the final attempt to recognize it.

Recent studies on convolutional neural networks (CNN) [21] have shown their power in document recognition tasks. The CNN based recognition approach has the effective advantage of not requiring a hand-crafted feature vector. This architecture is capable of learning the feature vector from the training character image samples in an unsupervised manner in the sense that no hand-crafting is employed to determine the feature vector. This fact prompted us to study whether one can skip the preprocessing and feature extraction steps for character recognition of a new script for which it is possible to collect at least some training samples with ground truth. The proposed approach assumes that there exists a CNN trained for a sufficiently large character class problem. We propose to feed the training samples of the new recognition problem (with a smaller number of classes) to this existing trained CNN. The values collected at the input layer of its MLP (placed at the top of the architecture) [22] are used to train an SVM for the recognition task. In our implementation, we trained a CNN (shown in Fig.1) for a 50 class character recognition problem and considered another five 10 class numeral recognition problems generating their feature vectors from the input layer of the MLP to train the respective SVM classifiers. The recognition accuracies are comparable to the state-of-the-art results.

II. Database details

Here we have used six standard databases available freely for research work. Among them MNIST

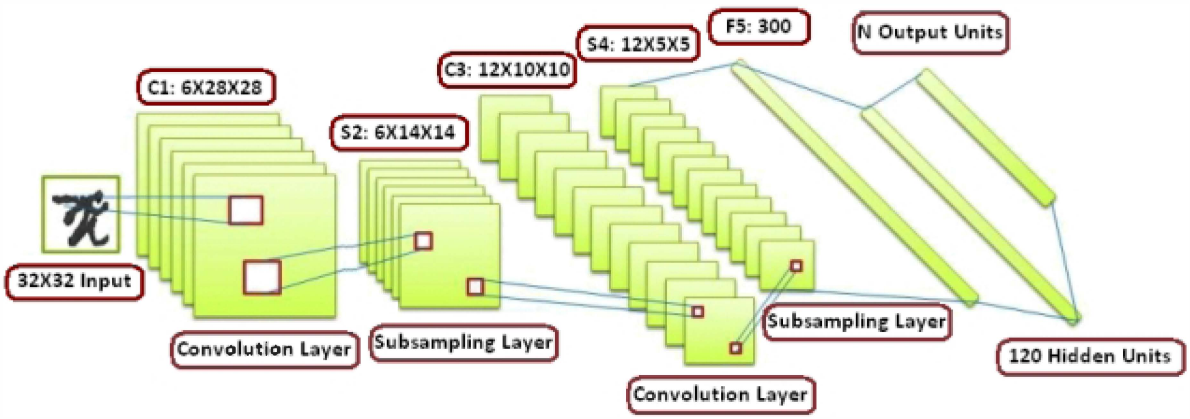

Fig. 1. 5-Layer Convolutional Neural Network (similar to LeNet-5) used in the present study.

numeral database is the most well-known and needs no description. The other 5 databases correspond to Indian script characters or numerals. Bangla numeral [23], Devanagari numeral [23], Oriya numeral [23] and Bangla basic character [8] databases are available from http://www.isical.ac.in/˜ujjwal/download/database.html where as the Telugu numeral database can be downloaded from [24]. The sample sizes of these databases are shown in Table I. Also, a few samples from them are shown in Fig. 2.

TABLE I. STATISTICS OF STANDARD DATABASES USED IN THE PRESENT STUDY

Database Name	Number of classes	Volume		
		Training	Test	Total
Bangla basic characters [8]	50	25000	12858	37858
Bangla numerals [20]	10	13392	4000	22533
Devanagari numerals [20]	10	18794	3762	22556
Oriya numerals [26]	10	4970	1000	5970
Telugu numerals [24]	10	2000	1000	3000
English (MNIST) numerals	10	60000	10000	70000

III. PROPOSED APPROACH

Usually, offline handwriting recognition approaches involve quite a few preprocessing steps to ensure robust recognition performance. On the other hand, for similar recognition problems based on convolutional neural networks, it is not necessary to design an efficient set of preprocessing steps to get rid of a portion of the variabilities present in the samples. The design of the architecture of CNN possesses an inherent mechanism to effectively take care of several sources of variation among the samples. In the proposed approach, the preprocessing step takes care of only normalization of the input image to the size of 32×32.

A major drawback in using CNN for character recognition purpose is that it needs considerable time and effort to fine tune its free parameters including the architecture. This has restricted the widespread use of CNN in character recognition problems. Hence, here we propose not to train a separate CNN for each individual problem. We trained only one CNN for a certain larger character class problem (in this case, 50 class Bangla basic characters). Here, we have used an architecture similar to the well-known LetNet-5 architecture and trained it with 10,000 iterations. We have restricted ourselves from fine tuning its parameters to achieve very good recognition accuracies by the same CNN on the test set of this 50-class Bangla basic character database. In an earlier study [8], we obtained 95.84% recognition accuracy on this database, and we terminated the training of the present CNN when its accuracy on the same test set reached the value 85% (approx.). The reason of this strategy is to avoid the overtraining of the CNN for this particular Bangla basic character recognition problem. For the above training we have used the deeplearning toolbox of Theano [25]. This is a Python library that has been primarily developed by academicians [27], [28]. It allows one to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently.

The architecture of this convolution network has the first Layer C1 (1st convolution layer) (shown in Fig. $3(a)$) consisting of 6 feature maps which are computed by using overlapping 5×5 kernel on the input 32×32 raw gray scale image. The input image is passed through two sets of convolution and subsampling layers and finally a multilayer perceptron with one hidden layer which is used for classification. Layer C1 provides 28×28 feature maps to prevent connections of inputs from falling off the boundary. At layer S2 (1st subsampling layer) (shown in Fig. $3(b)$), there are 6 feature maps each of size 14×14 obtained by subsampling based on maxpooling using non-overlapping 2×2 kernel on the output of C1 layer. At layer C3 (2nd convolution layer) (shown in Fig. $3(c)$), there are 12 feature maps which are computed by using overlapping 5×5 kernel on the output of Layer S2. It produces 10×10 feature maps. At layer S4 (2nd subsampling layer) (shown in Fig. $3(d)$), there are 12 feature maps each of size 5×5, computed by subsampling based on maxpooling using non-overlapping 2×2 kernel on the output of C3 Layer. Thus, we obtain 300 features at Layer F5 computed from these 12 kernels of size 5×5 as shown in Fig. $3(e)$.

In the next step, we used this CNN architecture trained as above to extract higher level of features from the input image at the Layer F5 and used them to train a support vector machine with RBF kernel. For implementation of the SVM, we used scikit-learn [29] toolbox under python environment. The values of the parameters C and γ were chosen by grid search strategy.

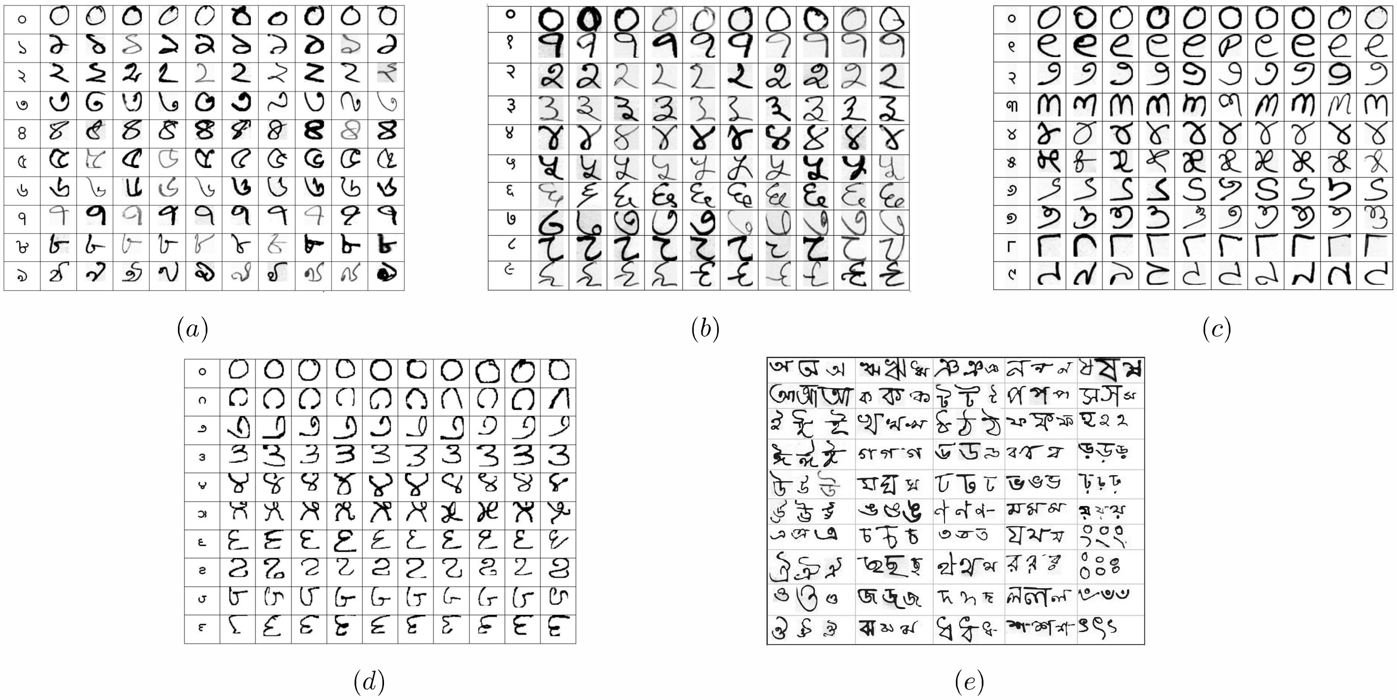

Fig. 2. Samples from 5 databases: (a) Bangla numerals, (b) Devanagari numerals, (c) Oriya numerals, (d) Telugu numerals and (e) Bangla basic characters.

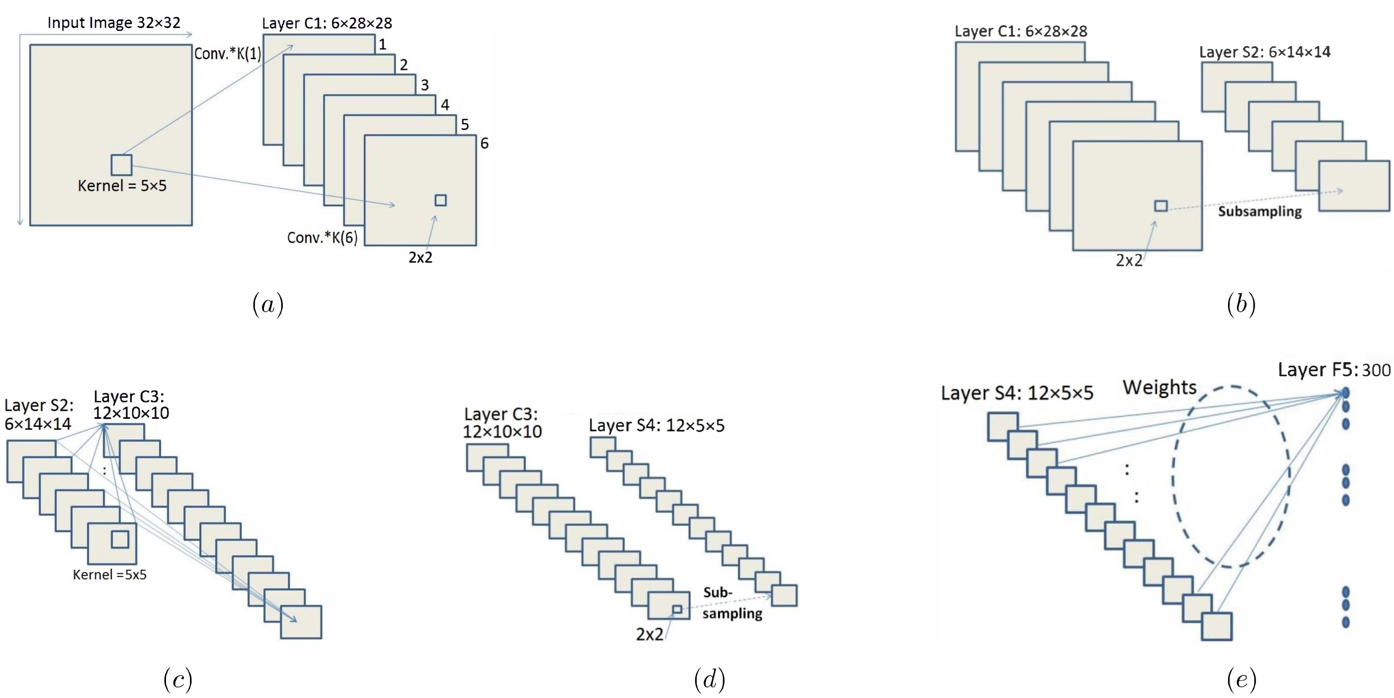

Fig. 3. Actions at different layers of the CNN: (a) Computation from input image to Layer C1, (b) Computation from Layer C1 to Layer S2, (c) Computation from Layer S2 to Layer C3, (d) Computation from Layer C3 to Layer S4 and (e) Computation from Layer S4 to Layer F5.

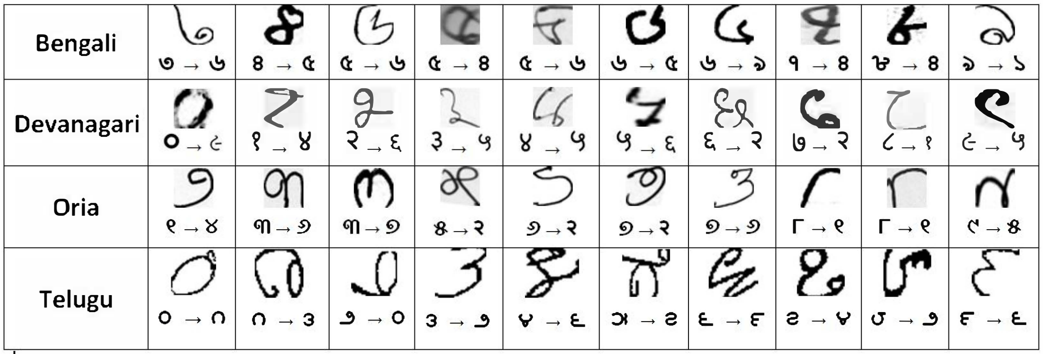

Fig. 4. Some misclassified samples of various numeral databases of four Indian scripts considered in the present study, $i \rightarrow j$ indicating that a sample belonging to Class-i is classified in Class-j.

TABLE II. ACCURACY FIGURES (%) OBTAINED BY THE PROPOSED STRATEGY

Bangla Basic Characters	Bangla Numeral	Devanagari Numeral	Oriya Numeral	Telugu Numeral	English Numeral
95.6	98.375	98.54	97.2	96.5	99.10

IV. Experimental results

The proposed approach to handwritten character recognition has been tested on 6 different character databases. These include English, Bangla, Devanagari, Oriya, Telugu numerals and Bangla basic characters. The first five databases consist of 10 classes and the sixth one has 50 classes. The recognition accuracies obtained on respective test sets are comparable with the state-of-the-art recognition accuracies on each of them. These have been summarized in Table II. Some samples misclassified by the four non-English numeral classifiers are shown in Fig. 4.

V. Conclusions and Future Scope

The goal of this study is to implement a feature extraction strategy which can be used in any character recognition problem. Here, we have shown that if a CNN is trained for a sufficiently large class problem, it can be used for extraction of features of any other character set and the resulting system is still capable of providing high recognition accuracies. In the present study, we used the popular SVM classifier. Other classifiers may be tested along with this feature vector in future studies. Also, in future, we plan to use a CNN trained with some other data different from character images and check whether the same remains capable of providing comparable accuracies on character recognition tasks. This strategy should help to train recognizers for a new script for which no such recognizer exists. Although it is now an established fact that CNN is sufficiently efficient in character recognition tasks, but it has not been widely used due to the difficulty faced in its training. However, in the present approach we did not allow the pain to train the CNN architecture through the fine-tuning of its parameter. On the other hand, we trained this architecture until we achieved only moderate recognition performance by its multilayer perceptron situated at the top of its Layer F5. In fact, whether fine-tuning of the network architecture on a particular training data set jeopardizes the entire training effort, will be explored in future studies.

The learning strategy implemented in the present work can be considered a type of “transfer learning” [30], in particular, a type of “inductive transfer learning”. In the traditional approaches of machine learning tasks, it is assumed that the training and test samples have the same feature distribution. On the other hand, there are many real-life tasks, for which not enough training data is available although it is not difficult to find some other domain with abundance of training data. In recent times, studies have been made on knowledge transfer to address the above situations improving the learning performance without concentrating on time and cost expensive data-labeling exercises. A survey of transfer learning approaches can be found in [30]. In particular, the proposed learning approach to handwritten character recognition of various scripts for which only limited volume of labelled samples are available, is called the “inductive transfer learning”, where the target task is different from the source task, irrespective of any relationship between the source and target domains.

Convolutional Neural Networks (CNN) based approaches to recognition tasks have recently been evaluated and comparisons of different such deep architectures have been made on a common ground [32]. Among various observations made in this recent survey work, it identified aspects of deep and shallow methods that can be successfully shared. A major bottleneck of learning CNNs for a given task is the requirement of a very large number of annotated image samples in order to estimate millions of its parameters. This has prevented application of CNNs to problems having only limited labelled training samples. However, in a recent work of object recognition [31], the authors have successfully shown that image representations learned with CNNs on large-scale annotated datasets could be transferred to other visual recognition tasks having only a limited amount of training data. Similar to our present study on document analysis tasks, the authors of [31] have shown that despite major differences in image characteristics and tasks in the two datasets, the transferred representation could lead to significantly improved results in a classification problem, outperforming the corresponding state of the art classifiers.

Acknowledgement

This research work has been partially supported by the Technology Development for Indian Languages (TDIL) Programme of the Dept. of Information Technology, Govt. of India.

REFERENCES

[1] N. Arica and F. Yarman-Vural, An Overview of Character Recognition Focused on Off-Line Handwriting, IEEE Trans. Systems, Man, and Cybernetics Part C: Applications and Rev., vol. 31, pp. 216-233, 2001.

[2] S. N. Srihari, E. Cohen, J. J. Hull, and L. Kuan, A System to Locate and Recognize ZIP Codes in Handwritten Addresses, Intl J. Research and Eng. - Postal Applications, vol. 1, pp. 37-45, 1989.

[3] J. Tsukumo and H. Tanaka, Classification of Handprinted Chinese Characters Using Nonlinear Normalization Methods, Proc. of Ninth Intl Conf. on Pattern Recognition, pp. 168-171, 1988.

[4] A. Amin and H. B. Al-Sadoun, Hand Printed Arabic Character Recognition System, Proc. of 12th Intl Conf. Pattern Recognition, pp.536-539, 1994.

[5] H. Yamada, K. Yamamoto, and T. Saito, A Non-Linear Normalization Method for Handprinted Kanji Character Recognition Line Density Equalization, Pattern Recognition, vol.23, pp.1023-1029, 1990.

[6] U. Bhattacharya, S. K. Parui, B. Shaw and K. Bhattacharya, Neural Combination of ANN and HMM for Handwritten Devanagari Numeral Recognition, Proc. 10th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR-2006), pp.613-618, 2006.

[7] T. K. Bhowmik, P. Ghanty, A. Roy and S. K. Parui, SVM-based Hierarchical Architectures for Handwritten Bangla Character Recognition, Int. Journal on Document Analysis and Recognition, vol.12(2), pp.97-108, 2009.

[8] U. Bhattacharya, M. Shridhar, S. K. Parui, P. K. Sen and B. B. Chaudhuri, Offline recognition of handwritten Bangla characters: an efficient two-stage approach, Pattern Analysis and Applications, vol.15(4), pp.445-458, 2012.

[9] R. Plamondon and S. N. Srihari, On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey, IEEE Trans. Pattern Analysis and Machine Intelligence, vol.22, pp.63-84, 2000.

[10] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Learning Internal Representations by Error Propagation, Inst. for Cognitive Science Report 8506, Univ. of California, 1985.

[11] C. M. Bishop, Neural Networks for Pattern Recognition, Clarendon, 1995.

[12] F. Kimura, K. Takashina, S. Tsuruoka, and Y. Miyake, Modified Quadratic Discriminant Functions and the Application to Chinese Character Recognition, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 9(1), pp.149-153, 1987.

[13] C. J. C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Knowledge Discovery and Data Mining, vol.2(2), pp.143, 1998.

[14] C. -L. Liu, H. Sako, and H. Fujisawa, Performance Evaluation of Pattern Classifiers for Handwritten Character Recognition, Intl J. Document Analysis and Recognition, vol. 4(3), pp. 191-204, 2002.

[15] F. Kimura, Y. Miyake, and M. Sridhar, Handwritten ZIP Code Recognition using Lexicon Free Word Recognition Algorithm, Proc. Third Intl Conf. on Document Analysis and Recognition, vol. 2, pp. 906-910, 1995.

[16] H. Liu and X. Ding, Handwritten Character Recognition using Gradient Feature and Quadratic Classifier with Multiple Discrimination Schemes, Proc. Eighth Intl Conf. on Document Analysis and Recognition, pp. 19-25, 2005.

[17] M. Shi, Y. Fujisawa, T. Wakabayashi, and F. Kimura, Handwritten Numeral Recognition Using Gradient and Curvature of Gray Scale Image, Pattern Recognition, vol. 35(10), pp. 2051-2059, 2002.

[18] A. Shustorovich, A Subspace Projection Approach to Feature Extraction: The Two-Dimensional Gabor Transform for Character Recognition, Neural Networks, vol. 7(8), pp. 1295-1301, 1994.

[19] L. Heutte, T. Paquet, J. V. Moreau, Y. Lecourtier, and C. Olivier, A Structural/Statistical Feature Based Vector for Handwritten Character Recognition, Pattern Recognition Letters, vol. 19(7), pp. 629-641, 1998.

[20] U. Bhattacharya and B. B. Chaudhuri, Handwritten Numeral Databases of Indian Scripts and Multistage Recognition of Mixed Numerals, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 31(3), pp. 444-457, 2009.

[21] P. Y. Simard, D. Steinkraus, and J. C. Platt, Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis, Proc. of Intl. Conf. on Document Analysis and Recognition, pp. 958-962, 2003.

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based Learning Applied to Document Recognition, Proc. of the IEEE, November 1998.

[23] U. Bhattacharya and B. B. Chaudhuri, Databases for Research on Recognition of Handwritten Characters of Indian Scripts, Proc. of the 8th Int. Conf. on Document Analysis and Recognition, vol. II, pp. 789-793, 2005.

[24] https://code.google.com/p/cmaterdb/downloads/detail?name=CMATERdb%203.4.1.rar&can=2&q= (as on 10 June, 2015)

[25] http://deeplearning.net/software/theano/ (as on 10 June, 2015)

[26] T. K. Bhowmick, S. K. Parui, U. Bhattacharya and B. Shaw, An HMM based Recognition Scheme for Handwritten Oriya Numerals, Proc. of the 9th International Conference on Information Technology (ICIT 2006), pp. 105-110, 2006.

[27] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley and Y. Bengio, Theano: A CPU and GPU Math Expression Compiler, Proc. of the Python for Scientific Computing Conference (SciPy). June 30 - July 3, Austin, TX, 2010.

[28] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley and Y. Bengio, Theano: New Features and Speed Improvements, Proc. NIPS 2012 Deep Learning Workshop, 2012.

[29] http://scikit-learn.org/

[30] S. J. Pan and Q. Yang, A Survey on Transfer Learning, IEEE Transactions on Knowledge and Data Engineering, vol. 22(10), pp.1345-1359, 2010.

[31] M. Oquab, L. Bottou, I. Laptev and J. Sivic, Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1717-1724, 2014.

[32] K. Chatfield, K. Simonyan, A. Vedaldi and A. Zisserman, Return of the Devil in the Details: Delving Deep into Convolutional Nets, arXiv preprint arXiv:1405.3531 (2014).