Image Augmentation by Blocky Artifact in Deep Convolutional Neural Network for Handwritten Digit Recognition

Md Shopon
Department of Computer Science and Engineering
University of Asia Pacific
Email: shopon.uap@gmail.com

Nabeel Mohammed
Department of Computer Science and Engineering
University of Liberal Arts Bangladesh
Email: nabeel.mohammed@ulab.edu.bd

Md Anowarul Abedin
Department of Computer Science and Engineering
University of Liberal Arts Bangladesh
Email: anowarul.abedin@ulab.edu.bd

***Abstract*—Deep Convolutional Neural Networks - also known as DCNN - are powerful models for different visual pattern classification problems. Many works in this field use image augmentation at the training phase to achieve better accuracy. This paper presents blocky artifact as an augmentation technique to increase the accuracy of DCNN for handwritten digit recognition, both English and Bangla digits, i.e., 0-9. This paper conducts a number of experiments on three different datasets: MNIST Dataset, CMATERDB 3.1.1 Dataset and Indian Statistical Institute (ISI) Dataset. For each dataset, DCNNs with the proposed augmentation technique give better results than those without such augmentation. Unsupervised pre-training with the blocky artifact achieves 99.56%, 99.83% and 99.35% accuracy respectively on MNIST, CMATERDDB and ISI datasets producing, in the process, so far the best accuracy rate for CMATERDB and ISI datasets.**

***Index Terms*—Augmentation, Image Augmentation, Artificial Neural Network, Convolutional Neural Network, Deep Learning, Handwritten Digit Recognition**

I. INTRODUCTION

This paper primarily aims at increasing the accuracy rate of a deep convolutional neural network (DCNN) for recognizing handwritten digits. In computer vision, handwritten digit recognition is a benchmark task. It is a task of optical character recognition; it has many significant usages in our regular life e.g., bank cheque processing, processing numeric fields of forms written by hand, identifying postal codes etc. These are sensitive tasks; tiny mistake can cause severe problem. For this reason, it is important to achieve a good accuracy in handwritten digit recognition.

As different persons write differently, the shapes and sizes of their digits vary. This is why handwritten digit recognition is a difficult task. Bangla digits are more complex than English digits for their shapes, sizes and rotations. Likewise, recognizing Bangla digits are harder than recognizing English digits. One of the two well-known Bangla datasets is prepared by Indian Statistical Institute; this dataset contains 23,299 images [1]. The other dataset is CMATERDB 3.1.1 consisting of 6000 images [1]. Compared to the dataset of English handwritten digits, the size of Bangla dataset is too small. MNIST is the largest dataset of handwritten English digits, which contains 60,000 images [2].

DCNNs are very popular and efficient for classifying handwritten digits [3]. The best accuracy rate on each of the three datasets - MNIST, ISI and CMATERDB - was achieved using DCNN [4], [1]. Accuracy of a DCNN model depends on the size of the training set - better accuracy can be achieved with more training images. However, it is difficult to get training data. As a result, data augmentation, i.e., modification or transformation of the features of the existing training images, is one of the best practices to increase the size of the training dataset. [5] demonstrated the effectiveness of augmentation to prevent overfitting and increase the accuracy of the model. [6], [7], [8] also used image augmentation in their works for increasing the accuracy.

Image augmentation can be done in many ways. A common process is to deform original data based on intra-class variation or prior knowledge. Color processing and geometrical variation or transformation such as rotation, resizing, shifting etc. are also used as augmentation techniques in visual recognition problems [9]. This paper focuses on a new augmentation method to improve the accuracy rate of DCNN. It proposes to create a blocky effect on the training images; such type of augmentation is unprecedented for handwritten digit recognition, and has not been found in other image classification problems as well. After augmentation of the training set by the blocky artifact, this work applies unsupervised pre-training using autoencoder and DCNN [1].

The rest of the paper is organized as follows: Section II contains the background of our current research work, Section III introduces our proposed method, Section IV describes the experiments and the datasets, Section V discusses the results and analysis and finally, conclusions are made in Section VI.

II. BACKGROUND

Different approaches have been taken so far for handwritten digit classification [4-8]. Among them Neural network, Statistical method, Fuzzy technique etc. are popular. The results that have been achieved using neural network have outperformed all the techniques available for classification methods [10]. [4] used "Dropconnect" method for classifying the MNIST dataset images and it is so far the best achieved accuracy in MNIST dataset which is 99.79%. [11] used multi column deep neural networks for MNIST which achieved 99.77%. A hierarchical bayesian network was used to recognize the images of CMATERDB with an accuracy of 87.5% [12]. Their work was based on [13]'s work. In [14], quad tree based

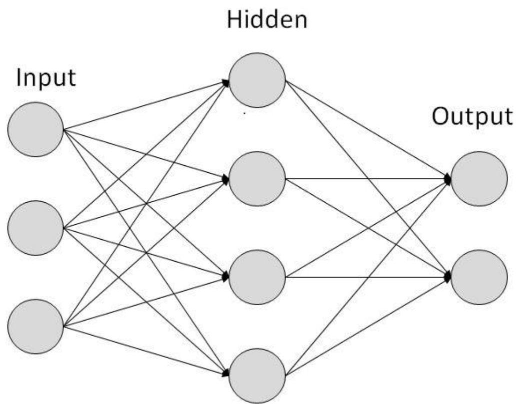


Fig. 1: Artificial Neural Network

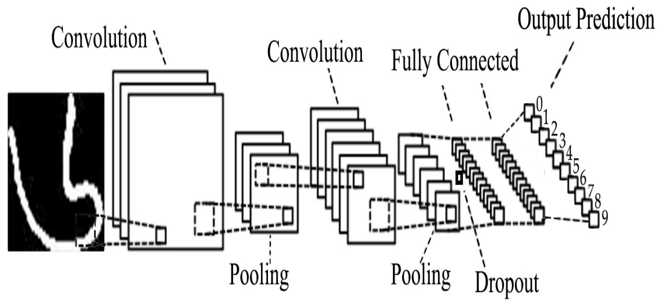


Fig. 2: Convolutional neural network

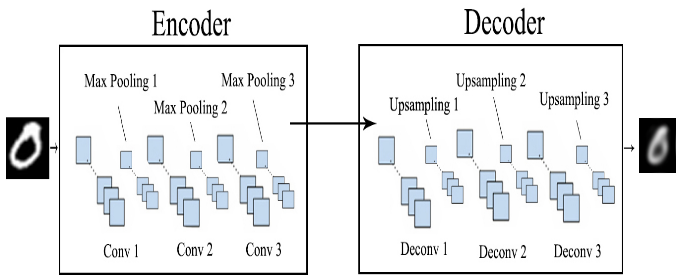


Fig. 3: Autoencoder

feature set was used in claffifying CMATERDB, they achieved 93.38% accuracy.

For ISI dataset ,[15] used a le-net like architecture to get an accuracy of 98.20%. Although [13] achieved better accuracy than [15], they used a fine tuned model. They used a similar network like [13] and augmented the image dataset in such roation range which would give better result only for ISI dataset. When the augmentation is for 10 ° their results reaches 98.98%. As it is similar to hand fitting, this paper claims the model proposed by [13] to be flawed.

In [16], denoising autoencoder was used for extracting features from images. [17] used deep autoencoder for content based image retrieval. In [18], unsupervised pre-training was used for classification of MNIST dataset and they achieved very good error rate. Inspired by their work, [1] also applied unsupervised pre-training using autoencoder and achieved 98.29% and 98.61% accuracy, which is so far the best result for ISI (apart from the work of [13]) and CMATERDB respectively. They also cross validated these two datasets and achieved an accuracy of 99.50% by training on ISI and testing on CMATERDB.

A. *Artificial Neural Network*

Artificial neural network(ANN) is a computational model that is inspired by the biological nervous system [19]. Fig 1 shows us the structure of artificial neural network. There are 3 layers in a artificial neural network: input layer, hidden layer and output layer. Artificial neural network works with one dimensional vectors. An image needs to be in one dimension before feeding it to an artificial neural network. An artificial neural network solves an optimization problem to choose some parameters which minimizes the error function in order to learn.

B. *Convolutional Neural Networks*

Convolutional neural networks(CNN) are special kind of neural networks that has learnable weights and biases. Its connectivity of neurons is inspired by the animal visual cortex. There are three basic ideas behind CNN [20].

- Local receptive fields
- Shared weights
- Pooling

Unlike ANN, CNN does not connect every input pixel to every hidden neurons. It connects localized regions of the input image. Each neuron of the first hidden layer will be connected to a small region of the input layer. The region that it gets connected is called local receptive field. CNN uses same weights and biases for all the other neurons. Pooling layer is another important part of CNN, which is used immediately after the convolutional layer. Pooling layer simplifies the output that it receives. Fig 2 shows a CNN.

C. *Autoencoder*

An autoencoder is a type of ANN which takes a raw image as input and then tries to reproduce the input image by encoding and decoding. Autoencoders are used for unsupervised pre-training which is very useful for deep learning. Autoencoders are surprisingly adept in feature extraction. They pre-train the dataset. [21] proves that unsupervised pre-training helps deep learning models. Figure 3 shows an image of autoencoder.

III. Proposed Method and Model

A. *Proposed Method*

In this paper, we are proposing a new augmentation technique to escalate the accuracy. We apply a blocky artifact on the images. We first reduce the dimension of the image to a certain value and then again increase the dimension to its

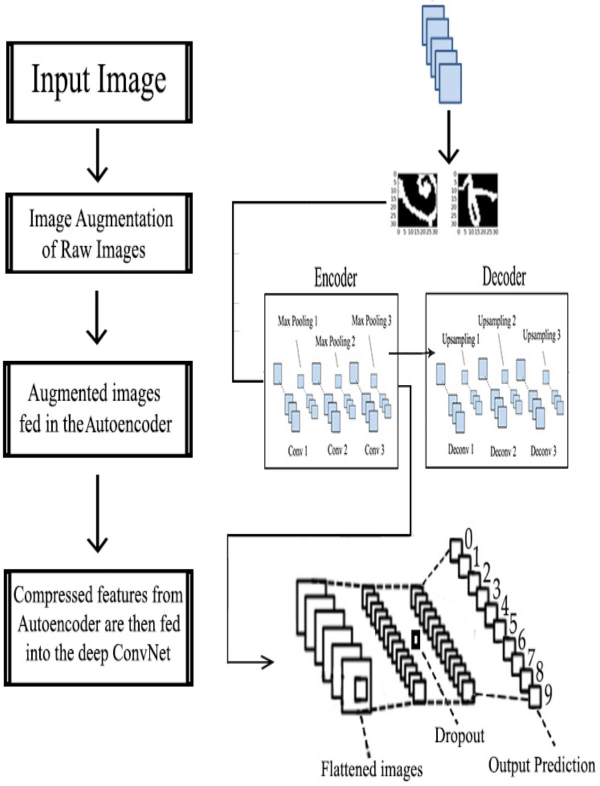


Fig. 4: Proposed Model

original size. We used python's sci-kit-image module for re-sizing the image [22]. It performs interpolation for up-size or down-size images. It down-samples a N-dimensional image by applying the arithmetic mean or sum. It uses linear interpolation for re-sizing the image. Linear interpolation produces blurred but jagged edges. It is mostly used for approximating the value of a function F by using two known values of that function F at other points. This approximation's error is defined as:

$$R_T = F(x) - G(x) \tag{1}$$

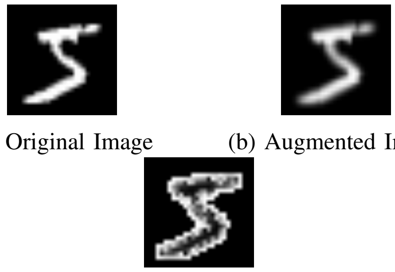
Here G is the linear interpolation polynomial which is defined in Equation 2.

$$G(x) = F(x_0) + \frac{F(x_1) - F(x_0)}{x_2 - x_1}(x - x_0) \tag{2}$$

B. Deep Convolutional Neural Network Model

We are using the model that was introduced in [1]. This model contains deep convolutional neural networks(DCNN) and artificial neural network(ANN). DCNN is such an ANN that contains more then one hidden layer. DCNN was used to implement autoencoder. We already discussed in section II that autoencoder is used for unsupervised pre-training. Autoencoder first encodes the image and then decodes the image. Their aim is to reproduce the input image.

This model first augments the images using rotation range of 10°. Encoder of the autoencoder contains three convolutional



(a) Original Image (b) Augmented Image

(c) Difference Between Image (a) & (b)

Fig. 5: An Image of CMATERDB

(a) 3 (b) 6

Fig. 6: MNIST Images

layers. Each convolutional layers has 32 filters and 3x3 kernel size. Each CNN is followed by a 2x2 max pooling layers.

Decoder also contains 3 CNN layers followed by 2x2 upsampling layers. Each CNN has 5 layers and 3x3 sized kernel. The encoder was later augmented again by rotation range of 10° and width range shift of 0.1%.

In our model decoder has no significant usage. Our target was to use the encoder. When autoencoder encodes the images, it extracts all the important features of that image. So after encoding the encoder consists of all the important features of the image. So we took the encoder and sent it to an fully connected layer which contains 128 layers. Next part contains the output layer with 10 layers, as we have 10 classes. The output layer classifies the image. Figure 4 shows the full architecture of the model.

IV. Datasets and Experiments

A. Datasets

1) MNIST: MNIST is database of English handwritten numerals. The dimension of the images are 28×28 pixels. This dataset consists total of 60,000 images. Figure 2(a),(b) shows us two image of MNIST dataset. This dataset is divided into two parts. 50,000 images were used for training and other 10,000 images were used for testing.

2) CMATERDB: CMATERDB is bangla numeral dataset which consists total of 6,000 images. The dataset was not split up in train and test set. We split the dataset into 7:3 ratio. 4,200 images were used for training and 1,800 images were used for testing purpose. The images were in 32×32 dimensions.

3) Indian Statstical Institute: Indian Statistical Institute is a Bangla numeral dataset just like CMATERDB. This dataset contains total 23,299 images. Training set contains 19,313 images and testing set contains 3,986. The images were in

(a) 3 (b) 8

Fig. 7: CMATERDB Images

(a) 3 (b) 8

Fig. 8: ISI Images

arbitrary dimensions. We reshaped the images into 32×32 pixels.

B. Experiments

We did a number of experiments with our proposed augmentation method. All the three datasets were augmented in different dimensions. But we are only mentioning the best achieved augmentations. In all the experiments the training set size was twice of its original size. We added the blocky artifact set with the original data set.

1) MNIST: Original size of the MNIST images are 28×28 pixels. Their dimension was first reduced to 25×25 and then again it was increased to 28×28.

2) CMATERDB: Images of CMATERDB are originally 32×32 pixels. The image was first reduced to 28×28 pixel and then again increased to 32×32.

3) ISI: Dimension of ISI images are same as CMATERDB. Although initially the images were in arbitrary dimension. We scaled all the images to 32×32 pixel. Then it was reduced to 28×28 pixel and again increased t o 32×32.

4) Cross Validation-1: In our first cross validation we trained the dataset with ISI images and then tested on CMATERDB test dataset.

5) Cross Validation-2: In our second cross validation we trained the CMATERDB train dataset and then tested on ISI test dataset.

There was two training phase in our used model. The first one was training the autoencoder. We trained autoencoder for 60 epochs with binary crossentropy as its cost function and adedelta as its optimizer. The convolutional layers of autoencoder has rectified linear unit (RELU) as their activation function. Next training phase was for fully connected layer. It was trained for 400 epochs with categorical crossentropy as its cost function and RMSprop as its optimizer. The first fully connected layer has RELU activation function. The last fully connected layer which is the output layer has softmax as its activation function.

V. Results and Analysis

Table I,II and III show us the previously achieved results on MNIST, CMATERDB and ISI dataset respectively. The results in the tables are sorted according to the accuracy rate. In 2013, [4] achieved best result so far on MNIST using neural network. [1] attained 99.50% on CMATERDB dataset which is the leading accuracy achieved. This accuracy was achieved by a cross validation approach. The model was trained by ISI images and then validated on CMATERDB. The best achieved accuracy on ISI dataset is 98.98% [23]. Although this result have some flaws in it which we already described in section II. Apart from this result [1] achieved 98.29% in 2016.

TABLE I: PAST RESULTS ON MNIST

Work	Accuracy
Liang et al. [24]	99.69%
Lee et al. [25]	99.71%
Chang et al. [26]	99.76%
Sato et al. [9]	99.77%
Ciregan et al. [27]	99.77%
Wan et al. [4]	99.79%

Table IV, V and VI discuss about the results we achieved using our proposed augmentation method respectively on MNIST, CMATERDB and ISI dataset. We achieved 99.56% on MNIST dataset. This accuracy was accomplished when we reduced the dimension of our image to 20×20 pixels. For CMATERDB we achieved 98.67% accuracy. We needed to reduce the image size to 28×28 pixels to achieve this result. On ISI dataset when we reduced the image dimension to 29×29 we achieved 99.35%. Our results surpasses the state-of-the-art of CMATERDB and ISI dataset.

Table VII,VIII discuss about cross validation-1 and cross validation-2. We achieved 99.83% accuracy in cross validation-1. This accuracy was achieved when the image was reduced to 28×28. In cross validation-2 we achieved 94.83% accuracy when we reduced the CMATERDB images to 25×25.

We used our proposed augmentation alongside the regular augmentation. These regular augmentations were also used in some past works. Such as, the state-of-the-art result of MNIST used rotation scaling augmentation.

In table IX we have discussed about another experiment we have conducted. We have done our experiments with only the regular augmentation. Our motive for this was to see how the accuracy rate changes if we don't use our proposed method. For every case blocky artifact augmentation gives better accuracy.

Figure 9 shows us the accuracy vs epoch graph of CMATERDB when it was reduced to 28×28 pixels. It can be observed from the image that the accuracy rate was not rugged; it was steady.

VI. Conclusions

This paper presents blocky artifact as an augmentation method for increasing the accuracy rate of handwritten English and Bangla digit recognition using a deep convolutional neural network model. To demonstrate the effectiveness and efficiency, we used three different datasets and conducted total 5 experiments. Our proposed augmentation method in

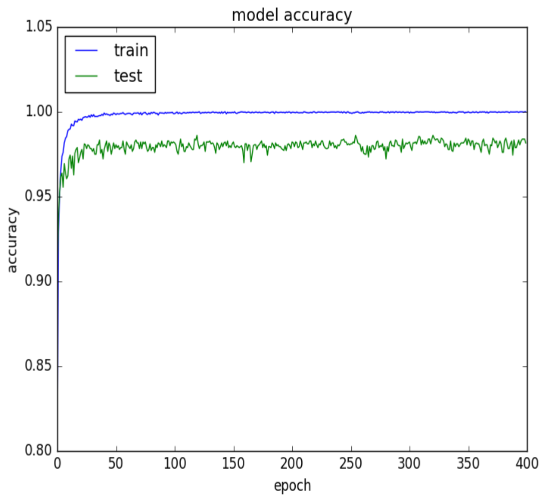


Fig. 9: Accuracy Vs Epoch Graph

TABLE II: PAST RESULTS ON CMATERDB

Work	Accuracy
Haider Adnan Khan et al. [28]	94%
Basu et al. [29]	95.1%
Hassan et al. [30]	96.7%
Basu et al. [31]	97.15%
Sarkhel el at. [32]	98.23%
Das et al. [33]	98.55%
Shopon et al. [1]	99.50%

TABLE III: PAST RESULTS ON ISI

Work	Accuracy
Nasir and Uddin [34]	96.80%
Wen and He [35]	96.91%
Das et al. [36]	97.70%
Akhnad et al. [37]	97.93%
Bhattacharya and Chaudhuri [38]	98.20%
Shopon et al. [1]	98.29%
CNNAP [23]	98.98%

TABLE IV: RESULTS ON MNIST FOR DIFFERENT REDUCTION

Reduction Size	Accuracy
25×25	99.51%
24×24	99.53%
23×23	99.52%
22×22	99.51%
21×21	99.48%
20×20	99.56%

conjunction with unsupervised pre-training outperforms the previous state-of-the-art of CMATERDB and ISI dataset. We achieved 98.67% and 99.35% accuracy respectively on these two datasets. Our accuracy rate for MNIST was 99.56%. Although it could not outperform the current state-of-the-art, still the accuracy rate is comparable with previous works and better than without using the blocky artifact. Future works include more detailed explanation about why and for what size the images give the best accuracy. Reproducing the current state-of-the-art of MNIST and applying the proposed augmentation in it can be another scope for future study.

TABLE V: RESULTS ON CMATERDB FOR DIFFERENT REDUCTION

Reduction Size	Accuracy
29×29	98.56%
28×28	98.67%
27×27	98.56%
26×26	98.61%
25×25	98.56%

TABLE VI: RESULTS ON ISI FOR DIFFERENT REDUCTION

Reduction Size	Accuracy
29×29	99.35%
28×28	99.05%
27×27	98.97%
26×26	99.10%
25×25	99.10%

TABLE VII: RESULTS ON CROSS VALIDATION-1 FOR DIFFERENT REDUCTION

Reduction Size	Accuracy
28×28	99.83%
27×27	99.78%
26×26	99.81%
25×25	99.76%
24×24	99.78%

TABLE VIII: RESULTS ON CROSS VALIDATION-2 FOR DIFFERENT REDUCTION

Reduction Size	Accuracy
28×28	94.27%
27×27	93.78%
26×26	93.92%
25×25	94.83%
24×24	94.44%

TABLE IX: RESULTS WITH AUGMENTATION AND WITHOUT AUGMENTATION

Dataset Name	With Augment	Without Augment
MNIST	99.56%	99.42%
CMATERDB	98.67%	98.07%
ISI	99.35%	98.76%

ACKNOWLEDGMENT

This project is supported by the ICT Division, Ministry of ICT, Government of the People's Republic of Bangladesh (Project ID: 56.00.0000.028.33.066.16-731)

REFERENCES

[1] M. Shopon, N. Mohammed, and A. Abedin, "Bangla handwritten digit recognition using autoencoder and deep convolutional neural network,"

in *International Workshop on Computational Intelligence*, 2016, accepted and Presented.

[2] Y. LeCun, C. Cortes, and C. J. Burges, "The mnist database of handwritten digits," 1998.

[3] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," *Neural computation*, vol. 22, no. 12, pp. 3207–3220, 2010.

[4] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 1058–1066.

[5] R. Wu, S. Yan, Y. Shan, Q. Dang, and G. Sun, "Deep image: Scaling up image recognition," *arXiv preprint arXiv:1501.02876*, vol. 7, no. 8, 2015.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[9] I. Sato, H. Nishimura, and K. Yokoi, "Apac: Augmented pattern classification with neural networks," *arXiv preprint arXiv:1505.03229*, 2015.

[10] L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, L. D. Jackel, Y. LeCun, U. A. Muller, E. Sackinger, P. Simard *et al.*, "Comparison of classifier methods: a case study in handwritten digit recognition," in *International conference on pattern recognition*. IEEE Computer Society Press, 1994, pp. 77–77.

[11] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3642–3649.

[12] J.-w. Xu, J. Xu, and Y. Lu, "Handwritten bangla digit recognition using hierarchical bayesian network," in *Intelligent System and Knowledge Engineering, 2008. ISKE 2008. 3rd International Conference on*, vol. 1. IEEE, 2008, pp. 1096–1099.

[13] D. George and J. Hawkins, "A hierarchical bayesian model of invariant pattern recognition in the visual cortex," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 3. IEEE, 2005, pp. 1812–1817.

[14] A. Roy, N. Mazumder, N. Das, R. Sarkar, S. Basu, and M. Nasipuri, "A new quad tree based feature set for recognition of handwritten bangla numerals," in *Engineering Education: Innovative Practices and Future Trends (AICERA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1–6.

[15] U. Bhattacharya and B. B. Chaudhuri, "Handwritten numeral databases of indian scripts and multistage recognition of mixed numerals," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 3, pp. 444–457, 2009.

[16] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[17] A. Krizhevsky and G. E. Hinton, "Using very deep autoencoders for content-based image retrieval." in *ESANN*, 2011.

[18] F. J. Huang, Y.-L. Boureau, Y. LeCun *et al.*, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *2007 IEEE conference on computer vision and pattern recognition*. IEEE, 2007, pp. 1–8.

[19] S.-C. Wang, "Artificial neural network," in *Interdisciplinary Computing in Java Programming*. Springer, 2003, pp. 81–100.

[20] M. A. Nielsen, "Neural networks and deep learning," *URL: http://neuralnetworksanddeeplearning. com/.(visited: 01.11. 2014)*, 2015.

[21] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *Journal of Machine Learning Research*, vol. 11, no. Feb, pp. 625–660, 2010.

[22] "Python scikit-image," https://goo.gl/gBXXDw, accessed: 2016-12-26.

[23] M. Akhand, M. Ahmed, and M. H. Rahman, "Convolutional neural network training with artificial pattern for bangla handwritten numeral recognition," in *Informatics, Electronics and Vision (ICIEV), 2016 5th International Conference on*. IEEE, 2016, pp. 625–630.

[24] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3367–3375.

[25] C.-Y. Lee, P. W. Gallagher, and Z. Tu, "Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree," in *International Conference on Artificial Intelligence and Statistics*, 2016.

[26] J.-R. Chang and Y.-S. Chen, "Batch-normalized maxout network in network," *arXiv preprint arXiv:1511.02583*, 2015.

[27] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3642–3649.

[28] H. A. Khan, A. Al Helal, and K. I. Ahmed, "Handwritten bangla digit recognition using sparse representation classifier," in *Informatics, Electronics & Vision (ICIEV), 2014 International Conference on*. IEEE, 2014, pp. 1–6.

[29] S. Basu, R. Sarkar, N. Das, M. Kundu, M. Nasipuri, and D. K. Basu, "Handwritten bangla digit recognition using classifier combination through ds technique," in *International Conference on Pattern Recognition and Machine Intelligence*. Springer, 2005, pp. 236–241.

[30] T. Hassan and H. A. Khan, "Handwritten bangla numeral recognition using local binary pattern," in *Electrical Engineering and Information Communication Technology (ICEEICT), 2015 International Conference on*. IEEE, 2015, pp. 1–4.

[31] S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri, and D. K. Basu, "A novel framework for automatic sorting of postal documents with multi-script address blocks," *Pattern Recognition*, vol. 43, no. 10, pp. 3507–3521, 2010.

[32] R. Sarkhel, N. Das, A. K. Saha, and M. Nasipuri, "A multi-objective approach towards cost effective isolated handwritten bangla character and digit recognition," *Pattern Recognition*, vol. 58, pp. 172–189, 2016.

[33] N. Das, J. M. Reddy, R. Sarkar, S. Basu, M. Kundu, M. Nasipuri, and D. K. Basu, "A statistical–topological feature combination for recognition of handwritten numerals," *Applied Soft Computing*, vol. 12, no. 8, pp. 2486–2495, 2012.

[34] M. K. Nasir and M. S. Uddin, "Hand written bangla numerals recognition for automated postal system," *IOSR Journal of Computer Engineering*, vol. 8, no. 6, pp. 43–48, 2013.

[35] Y. Wen and L. He, "A classifier for bangla handwritten numeral recognition," *Expert Systems with Applications*, vol. 39, no. 1, pp. 948–953, 2012.

[36] N. Das, R. Sarkar, S. Basu, M. Kundu, M. Nasipuri, and D. K. Basu, "A genetic algorithm based region sampling for selection of local features in handwritten digit recognition application," *Applied Soft Computing*, vol. 12, no. 5, pp. 1592–1606, 2012.

[37] M. M. Rahman, M. Akhand, S. Islam, P. C. Shill, and M. H. Rahman, "Bangla handwritten character recognition using convolutional neural network," *International Journal of Image, Graphics and Signal Processing (IJIGSP)*, vol. 7, no. 8, p. 42, 2015.

[38] U. Bhattacharya and B. B. Chaudhuri, "Handwritten numeral databases of indian scripts and multistage recognition of mixed numerals," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 3, pp. 444–457, 2009.