SUST-BHND: A database of Bangla handwritten numerals

Shuvanon Razik
Dept. of Computer Science & Engineering
Shahjalal University of Science & Technology
Sylhet, Bangladesh
shuvanon.razik@student.sust.edu

Evan Hossain
Dept. of Computer Science & Engineering
Shahjalal University of Science & Technology
Sylhet, Bangladesh
cse.evan@gmail.com

Sabir Ismail
Dept. of Computer Science & Engineering
Shahjalal University of Science & Technology
Sylhet, Bangladesh
sabir-cse@sust.edu

Md Saiful Islam
Dept. of Computer Science & Engineering
Shahjalal University of Science & Technology
Sylhet, Bangladesh
saiful-cse@sust.ed

***Abstract*— This paper presents the development process of the SUST-Bangla Handwritten Numeral Database (SUST-BHND). We extracted handwritten Bengali digits from twenty-one hundred pre-designed form filled by different people. After data retrieval, cleaning, processing and error analysis we have created a database consisting of 101065 sample images. It provides a basic database for Bangla OCR and script identification research field. Finally, a deep convolutional neural network was trained by the database which led to an accuracy of around 99.4%.**

Keywords—SUST-BHND; Numeral image database; OCR; convolutional neural network; Handwritten digits;

I. Introduction

Bangla is used by more than 210 million people all over the world. That's why this language has much practical importance, especially in Bangladesh and maximum part of eastern India. Researchers are working on Bangla optical character recognition for more than 20 years in many ways[1-13]. Machine learning based OCR systems are attracting more and more attention of researchers in recent years, but the lack of large standard/benchmark public database for handwriting is the main obstacle in this field. There are many well known standard handwriting database for other major languages like nist, mnist[14], CENPAQM[15], CEDAR[16], PHOND[17], IRONOFF[18]. For Bangla only public database is ISI online and offline database[19-21]. But a small portion of this database is accessible online. This database has 23,392 isolated images of handwritten numerals. Offline data collected from postal code, job application and pre-designed form on the other hand online data collected using special software. Another handwritten numeral database is CMATERdb 3.1.1: Handwritten Bangla Numeral Database[22] which was used in many researches but now it is not accessible. The difference from other two databases is our database is collected from a pre-designed form and the main intention of developing this database is establish a standard database which can be used for any algorithm implementation.

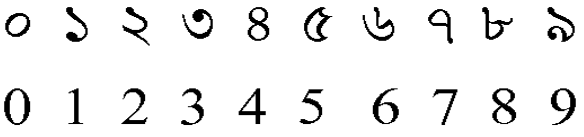


Figure 1. The ideal forms of Bangla numerals.

In this research phase, we are working on Bangla numerals because numerals are important in both research and application areas. A standard Bangla numerals database can open a vast window to design, comparison and evaluation of the performance of different algorithms for the Bangla OCR and script identification research fields. On the other hand, Bangla numerals are not only used in Bangla but also in Assamese and Manipuri language, so this database is going to be useful for three different languages and create new research opportunities. Figure 1 shows the ideal forms of Bangla numerals.

The SUST-Bangla Handwritten Numeral Database (SUST-BHND) provides researchers with a refined online Bangla database as training, testing and validating sample online.

. Rest of the paper is organized as follows: Section 2 describes the data collection process. Section 3 provides detailed information about data processing and finalization. In section 4 we have discussed the statistical analysis of the database. Section 5 and 6 explain classification using CNN and recognition analysis. Section 7 summarizes the present work. Finally, acknowledgments and references of literature are mentioned in the last two sections.

II. Data Collection

Data collection is the first step of our workflow. Before starting data collection we studied data collection process of the other database then we started our work.

A. Form design

Form design is one of the vital works for collecting good data. We have worked to create a suitable form which can help us to collect data in an optimal way from a writer and make

978-1-5090-6004-7/17/$31.00 ©2017 IEEE

workable data from scanning this without any problem. We have designed a one-page form where we have a table which has six rows and ten columns. The first row has example zero to nine and next 5 row is blank for writers to fill. The color of the forms is chosen to be white and the digits were written with black ink. Figure 2 shows a blank sample form.

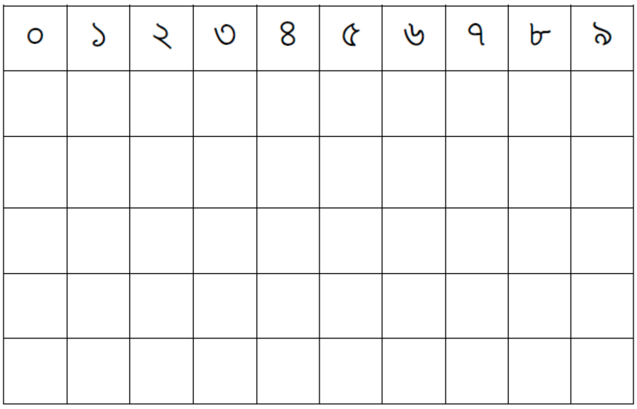
B. *Writer sampling*

We have collected data from three disconnected sample group. Every group has 700 writers. The first group consisted of 700 students from Shahjalal University of Science and Technology (SUST). The second group consisted of 700 students from Sylhet Polytechnic Institute and the third group consisted of 700 high school students.

Every writer was instructed to fill-up one form by writing 5 times zero to nine. In 5 to 10 minutes writing time writers were asked to write in their comfortable way but try not to write exactly same every time. There was no other restriction to the writers. Every writer uses their own writing instrument. Though we preferred pen but some writer user pencil and color pen for writing.

III. Data processing

After collecting data from writers we scanned every form and store in BMP format to avoid noise. Manually processing this amount of data is time-consuming so we developed some python tools for automating our data processing.



০	১	২	৩	৪	৫	৬	৭	৮	৯

Figure 2. A blank sample form.

Table 1
Statistics of writers and their contribution

Sample group	Number of writers	Error	Collected Images	Final Proportion
1	700	2.19%	34233	33.87%
2	700	3.22%	33873	33.52%
3	700	5.83%	32959	32.61%

A. *Preprocessing*

In preprocessing phase, first, we removed unused extra parts of the form using our automated tool which keeps only inside part of rectangular margin. After that, we used another small automated tool for slope correction based on parallel line and prepared for data retrieval. Figure 3 represents a form after preprocessing.

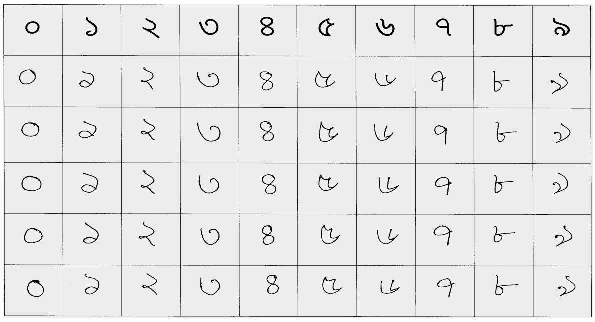
B. *Data retrieval*

In this phase first, we divided all form into parts based on row dividing line. Since the first row of our forms is printed example we removed this row. After that, every row is divided into 10 parts based on column dividing line and saved in corresponding labeled folder.

C. *Error correction*

After retrieving data from the forms, some images have shade or slit stain of borderline and some of the images have an unexpected mark. So we used another python tool for cropping the digits only. Figure 4 shows the states before and after correction of an image.

Finally, we roughly checked every image manually for bad or unclear handwriting and remove them. Figure 5 contains some sample images that we discarded manually due to noises or unreadability.



০	১	২	৩	৪	৫	৬	৭	৮	৯
০	১	২	৩	৪	৫	৬	৭	৮	৯
০	১	২	৩	৪	৫	৬	৭	৮	৯
০	১	২	৩	৪	৫	৬	৭	৮	৯
০	১	২	৩	৪	৫	৬	৭	৮	৯
০	১	২	৩	৪	৫	৬	৭	৮	৯

Figure 3. A preprocessed form.

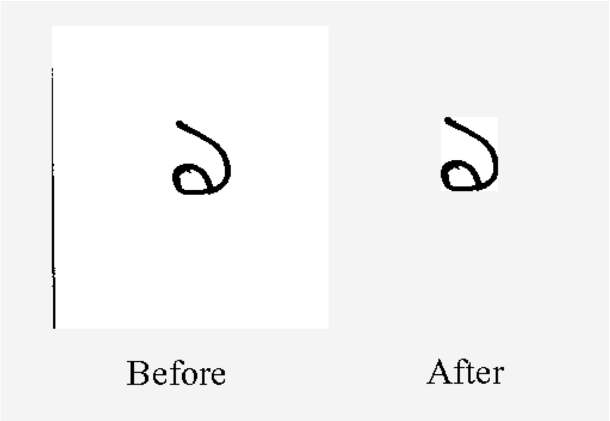


Figure 4. Before and after error correction of an image.

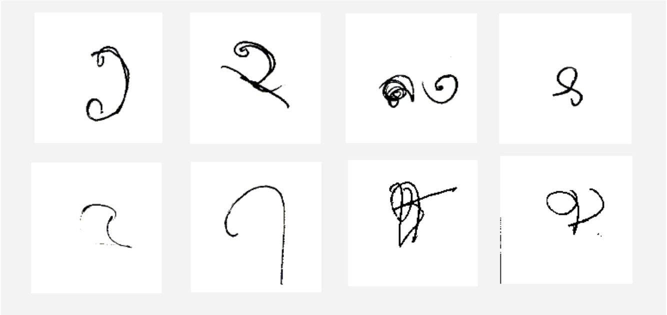


Figure 5. Some manually removed images.

D. *Finalized database*

In the final phase, we created three different set of the database for further use. The first database stores the raw images which we get after error correction. This database has 10 labeled folders that have BMP formatted images. The second database has normalized form of the data. For this case, we resized every image in 28*28 pixel and inverted color. The final case we convert all image in CSV format. The gray scale pixel values of 28*28 inverted images were flattened from 2 dimensional to 1-dimensional data, creating an array of 784 (28*28) values. Then these values were divided by 255.0 to normalize each pixel value in range 0 ~ 1. And lastly, we kept the label of the corresponding image as the 785th value of the array. We wrote a python tool that reads images from all the folders and randomly creates 3 different CSV files as stated above for training, validation and testing. Then these csv files can be fed into neural network models for benchmarking the database.

IV. Statistical analysis of database

Our database (SUST-BHND) contains 101065 sample images which were collected from 2100 different writers. For research purpose, we split this data into three set named training, test and validates. Sample images are distributed in this three-set 80:10:10 ratio. The train set contains 80851 sample images, test and validate contain 10107 sample images each. Distribution of images in files of Bangla digits is shown in table 2.

Table 2
Distribution of images in SUST_BHND

Digits	**Train**	**Test**	**Validate**	**Total**
0	8078	1010	1010	10098
1	8169	1021	1021	10211
2	8084	1011	1011	10106
3	8065	1008	1008	10081
4	8053	1007	1007	10067
5	8084	1010	1010	10104
6	8081	1010	1010	10101
7	8072	1009	1009	10090
8	8080	1010	1010	10100
9	8085	1011	1011	10107
Total	**80851**	**10107**	**10107**	**101065**

V. Classification using CNN

Convolutional neural network [23] has been used to benchmark the database. Most of the neural network algorithms transform images into a linear array of pixel values and use them as input nodes. Then connects each input node to all the hidden layer nodes. Thus missing an important feature-gaps among the input nodes. But Convolutional network takes that spatial structure of images into account and uses the original image matrix (28*28 in this case). And instead of connecting every pixel to every other node of hidden layers, this algorithm uses special filters called feature filter to extract a different kind of features. Local receptive fields, shared weights, and pooling are the three basic ideas of convolutional neural networks. Three basic ideas of convolutional neural networks are described in next sections.

A. *Local receptive fields*

Fully-connected networks depict input as a vertical line of neurons. But in a convolutional neural network, images are kept as original 28*28 square matrix and individual cells are treated as neurons.

Then these input pixels get connected to hidden layer neurons. But in this case, every input pixel is not connected to every hidden node. Other intermediate filters of small sizes are used instead. More precisely every 5*5 small region gets connected to one hidden layer node. This region in the input image is called the local receptive field for the hidden neuron.

Here in figure 6 a 2*2 small region is showed on the left side and corresponding hidden layer nodes are shown on the right side. Then if there are 28*28 input nodes and 5*5 local receptive fields, then there will be 24*24 neuron in the first hidden layer.

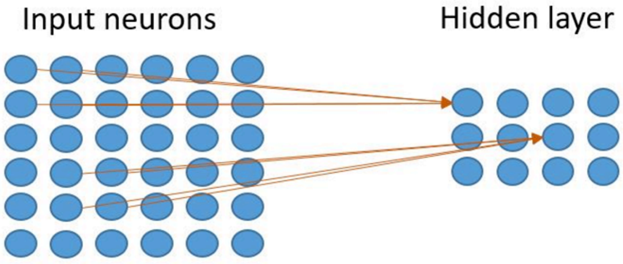


Figure 6. Local receptive field

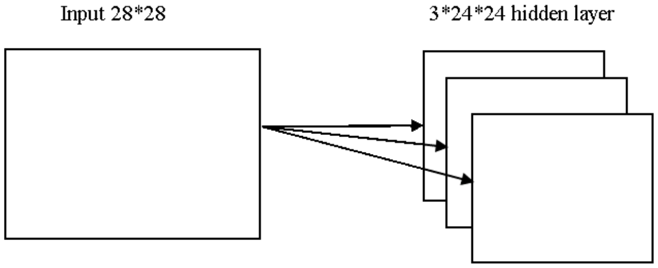


Figure 7. Hidden layer visualization

B. *Shared weights and biases*

Each of the 24*24 hidden layer nodes shares same weight and biases. That means every node in hidden layer detects the same feature from different locations of the input image. This map from input to hidden layer is also called feature map. As each hidden layer detects one kind of feature for all spots, we need to increase the features to ensure better recognition. So we use multiple feature maps for detecting different features over the input nodes.

Different filters of hidden layers have different random weights associated with them. But weights across the same layer is shared. So we have 5*5 shared weights and a single bias for each feature map. Then if we have 20 feature map, there are total 20*26 = 520 parameters in the convolutional layer. Previously we had 784 input nodes and 30 weights and also 30 biases totaling 23,550 parameters. So we have around 40 times fewer parameters in the convolutional network which make the system considerably faster.

C. *Pooling layer*

In addition to the layers described above, convolutional network [24] uses another layer called pooling layer. Usually pooling layer is smaller in size than local receptive fields. Pooling layer takes each feature map and makes a summary of that feature map. In our system, we have used 2*2 size pooling input region. These regions connect to pooling layer providing the maximum value in that area. This is call max-pooling. If we have 24*24 neurons in hidden layer, then we will have 12*12 nodes in pooling layer. This process can be visualized in the same way as in figure 6. Just the size will become 2*2 in this case.

D. *Training*

We have used two convolutional layers each with a pooling layer attached. The first convolutional layer contains 32 24*24 hidden layers with a feature map of size 5*5. Then there is the max pooling layer of size 2*2. Then these outputs of pooling layer go as input in the second convolutional layer containing 64 12*12 hidden layer. Then again there is another max pooling layer of size 2*2. This is the model summary.

Choosing proper validation data size is crucial we have kept validation size around 10,107 for around 80,851 training data. We will show the rate of change in accuracy over choosing proper training and validation data in recognition analysis section.

E. *Evaluation*

Training being done next step is the evaluation of test dataset. Deep neural networks are supposed to take significantly more time than simple networks. Yet the use of Tensorflow [25] improves the runtime significantly. For training around 80,851 images it takes around 16-18 minutes and running test dataset, it takes around 30 seconds. Using this method, we got less than 1% error for handwritten Bangla numerals.

VI. Recognition Analysis

We have applied CNN on the normalized image files without any feature extraction technique. All algorithm was implemented in python with Tensorflow. All programs are executed on a desktop machine (CPU: Intel Core i3 @ 3.2 GHz and RAM: 4 GB) in Window 10 (64bit) environment.

The accuracy of recognition depends on many variables. For this work, we worked on only a few variables which help us to define the dataset. Using CNN, we found 99.2% to 99.4% accuracy in best cases.

A. *Training data vs Recognition*

Training data is in our main focus in this experiment. We have tested our CNN system with various sizes of training data and the result is in figure 8.

This figure shows that increasing training dataset helps to improve accuracy but after an optimal point the improvement, the slope doesn't really increase. So from here, we can estimate how far we can improve solely depending on dataset size. Figure 9 shows some images from misclassified images with their level and CNN classification. In these images, left Bangla characters are the actual characters and right ones are predicted.

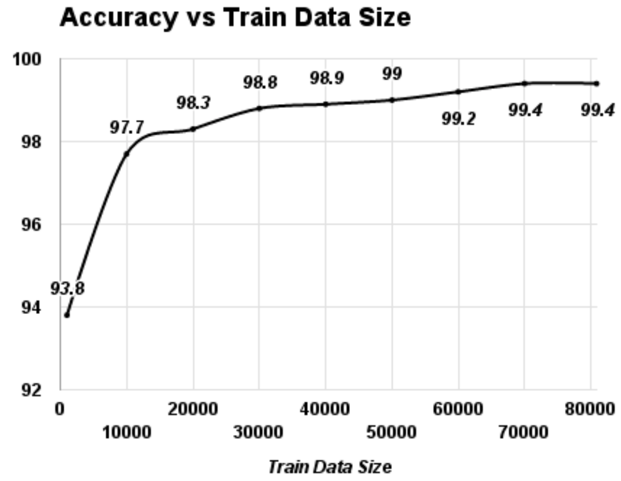


Figure 8. Accuracy vs data size graph

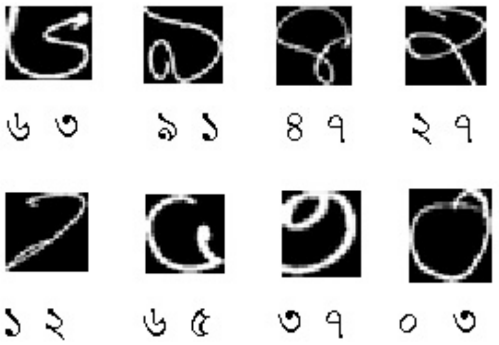


Figure 9. CNN error for handwritten data

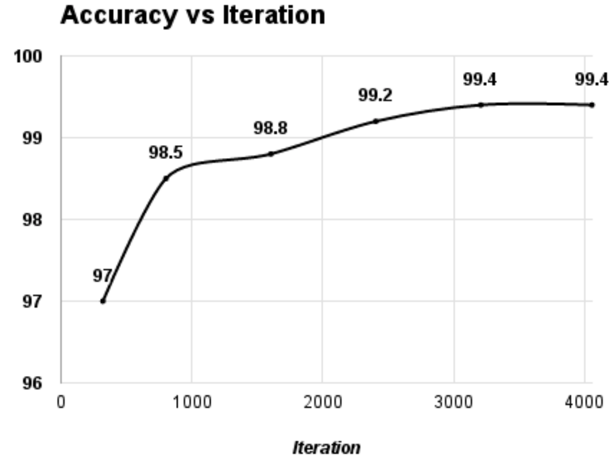


Figure 10. Accuracy vs iteration graph

B. Iteration vs Recognition

In convolutional neural networks, systems iteration is another important thing to impact on the result. We have increase iteration step by step and record the error rate. Figure 10 shows the relation between iteration and accuracy. As we saw in the case of training dataset size, iteration number also get fixed at some point. Increasing iteration after this step only requires more time, result change rate is negligible.

VII. Conclusion

In this paper, we have provided a detailed description of SUST-BHND development. An important infrastructure should be created for development and comparisons among various research on recognition of handwritten Bangla numerals by this database. The unique characteristics of this database are this is the largest handwritten Bangla numeral database and deep convolutional neural network works great on this database. Since this database is publicly available in three formats, researchers can use this however they want. Few researchers from Shahjalal University of Science and Technology have already started working on this database. If researchers use it properly it can make a great impact on Bangla optical character recognition.

Acknowledgements

We like to thankfully acknowledge Md. Ruhul Amin, assistant professor, Dept. of Computer Science and Engineering, Shahjalal University of Science and Technology for starting collecting data from school students and Md. Sazibur Rahman for coordinating data collection from Sylhet Polytechnic Institute. We are also thankful to all writers, volunteers and all other members of SUST NLP research group for their contribution and Department of Computer Science and Engineering, Shahjalal University of Science and Technology for providing infrastructural facilities during the progress of the work.

References

[1] Pal, U., and B. B. Chaudhuri. "OCR in Bangla: an Indo-Bangladeshi language." Pattern Recognition, 1994. Vol. 2-Conference B: Computer Vision &amp; Image Processing., Proceedings of the 12th IAPR International. Conference on. Vol. 2. IEEE, 1994.

[2] Chaudhuri, B.B., Pal, U.: A complete printed bangla ocr system. Pattern recognition. 31(5), 531–549 (1998)

[3] Bhowmik, T., Bhattacharya, U., Parui, S.: Recognition of bangla handwritten characters using an mlp classifier based on stroke features. In: Pal, N., Kasabov, N., Mudi, R., Pal, S., Parui, S. (eds.) Neural Inf. Process. Lecture notes in computer science, vol. 3316, pp. 814–819. Springer, Berlin (2004)

[4] Roy, K., Pal, U., Kimura, F.: Bangla handwritten character recognition. In: Prasad, B. (ed.) 2nd Indian international conference on artificial intelligence, pp. 431–443. Pune, India (2005)

[5] Bhattacharya, U., Parui, S.K., Shridhar, M., Kimura, F.: Two-stage recognition of handwritten Bangla alphanumeric characters using neural classifiers. In: Prasad, B. (ed.) 2nd Indian international conference on artificial intelligence, pp. 1357–1376. Pune, India (2005)

[6] Pal, U., Wakabayashi, T., Kimura, F.: Handwritten Bangla compound character recognition using gradient feature. In: 10th international conference on information technology-07, pp. 208–213 (2007).

[7] Basu, S., Das, N., Sarkar, R., Kundu, M., Nasipuri, M., Basu, D.K.: A hierarchical approach to recognition of handwritten bangla characters. Pattern Recognition. 42(7), 1467–1484 (2009).

[8] Bhowmik, T., Ghanty, P., Roy, A., Parui, S.: Svm-based hierarchical architectures for handwritten bangla character recognition. Int. J. Doc. Anal. Recognition International Journal on Document Analysis and Recognition (IJDAR). 12(2), 97–108 (2009)

[9] Das, N., Pramanik, S., Basu, S., Saha, P.K., Sarkar, R., Kundu, M., Nasipuri, M.: Recognition of handwritten Bangla basic characters and digits using convex hull based feature set. In: Dimitrios A. Karras, Z.M., Etienne E. Kerre, Chunping Li (eds.) International conference on artificial intelligence and pattern recognition, Orlando, Florida, USA, pp. 380–386. ISRST (2009)

[10] Das, N., Sarkar, R., Basu, S., Kundu, M., Nasipuri, M., Basu, D.K.: A genetic algorithm based region sampling for selection of local features in handwritten digit recognition application. Applied Soft Computing 12(5), 1592–1606 (2012)

[11] Garain, Utpal, David S. Doermann, and Douglas W. Oard. "Maryland at FIRE 2011: Retrieval of OCR'd Bengali." Multilingual Information Access in South Asian Languages. Springer Berlin Heidelberg, 2013. 205-213.

[12] Rahman, A. F. R., and M. Kaykobad. "A complete Bengali OCR: A novel hybrid approach to handwritten Bengali character recognition."

CIT. Journal of computing and information technology 6.4 (2015): 395-413.

[13] Chowdhury, Muhammed Tawfiq, et al. "Implementation of an Optical Character Reader (OCR) for Bengali language." 2015 International Conference on Data and Software Engineering (ICoDSE). IEEE, 2015.

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, 86(11):2278-2324, November 1998.

[15] C. Y. Suen, C. Nadal, R. Legault, T. A. Mai, L. Lam, "Computer recognition of unconstrained handwritten numerals", Proceedings of the IEEE, Vol. 80, No. 7, pp. 1162-1180, 1992.

[16] J. J. Hull, “A Database for Handwritten Text Recognition Research”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, 1994, pp. 550-554.

[17] Sajedi, Hedieh. "Handwriting recognition of digits, signs, and numerical strings in Persian." Computers & Electrical Engineering 49 (2016): 52-65.

[18] C. Viard-Gaudin, P.M. Lallican, S. Knerr, P. Binter, "The IRESTE On/Off (IRONOFF) Dual Handwriting Database", The 5th International Conference on Document Analysis and Recognition, 1999. pp. 455-458.

[19] Parui, S. K., et al. "A hidden Markov model for recognition of online handwritten Bangla numerals." Proc. of the 41st National Annual Convention of CSI. 2006.

[20] Bhattacharya, Ujjwal, and Bidyut Baran Chaudhuri. "Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals." IEEE transactions on pattern analysis and machine intelligence 31.3 (2009): 444-457.

[21] Chaudhuri, B. B. "A complete handwritten numeral database of Bangla– a major Indic script." Tenth International Workshop on Frontiers in Handwriting Recognition. Suvisoft, 2006.

[22] Das, N., Sarkar, R., Basu, S., Kundu, M., Nasipuri, M. and Basu, D.K., 2012. A genetic algorithm based region sampling for selection of local features in handwritten digit recognition application. Applied Soft Computing, 12(5), pp.1592-1606.

[23] LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. Nature, 521(7553), pp.436-444.

[24] Google, "Tensorflow," [Online]. Available: https://www.tensorflow.org/tutorials/mnist/pros/, January 18, 2017.

[25] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M. and Ghemawat, S., 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.