Bangla Handwritten Character Recognition using Convolutional Neural Network with Data Augmentation

Rumman Rashid Chowdhury
Department of Computer Science
and Engineering
University of Chittagong
Chittagong, Bangladesh
rumman179@gmail.com

Mohammad Shahadat Hossain
Department of Computer Science
and Engineering
University of Chittagong
Chittagong, Bangladesh
hossain_ms@cu.ac.bd

Raihan Ul Islam
Department of Computer Science, Electrical
and Space Engineering
Luleå University of Technology
Skellefteå, Sweden
raihan.ul.islam@ltu.se

Karl Andersson
Department of Computer Science, Electrical
and Space Engineering
Luleå University of Technology
Skellefteå, Sweden
karl.andersson@ltu.se

Sazzad Hossain
Department of Computer Science and Engineering
University of Liberal Arts Bangladesh
Dhaka, Bangladesh
sazzad.hossain@ulab.edu.bd

Abstract—**This paper proposes a process of Handwritten Character Recognition to recognize and convert images of individual Bangla handwritten characters into electronically editable format, which will create opportunities for further research and can also have various practical applications. The dataset used in this experiment is the BanglaLekha-Isolated dataset [1]. Using Convolutional Neural Network, this model achieves 91.81% accuracy on the alphabets (50 character classes) on the base dataset, and after expanding the number of images to 200,000 using data augmentation, the accuracy achieved on the test set is 95.25%. The model was hosted on a web server for the ease of testing and interaction with the model. Furthermore, a comparison with other machine learning approaches is presented.**

Keywords — Convolutional Neural Network, handwritten character recognition, Bangla handwritten characters, Data augmentation.

I. INTRODUCTION

Bangla is the second-most spoken language in this Indian subcontinent. As it is the language of above 250 million people of this subcontinent and over 300 million from all around the world, Bangla holds the sixth position among the most spoken languages of the world. Furthermore, it is the official and most widely spoken language of Bangladesh and second most widely spoken among the languages of India. Bangla is a part of the Indo-European Language class, its primary roots originate from Sanskrit and this language has evolved over a thousand years since its existence. The current version of the language consists of 50 basic alphabets among which there are 11 vowels and 39 consonants. In spite of being such a widely popular language, there has not been much research conducted on the handwriting recognition of this language, compared to English. This research focuses on one of the fundamental tasks of handwriting recognition, which is, classifying individual characters of a language.

In the last few decades, the primary method of storing information has switched from handwritten copies of documents to digital formats. The digital format of the documents are more reliable and easy to store. Despite the switchover to the new form of document storage, a large segment of the older documents are stored in handwritten form. The problem lies in the process of converting these documents, where traditional techniques rely on manually typing the whole literature. This process is tedious and requires a huge amount of time to successfully convert the documents, and also requires substantial amount of manpower to create accurate copies of the documents. Furthermore, Bangla characters have a complex arrangement of curvatures which are more complicated than that of other languages such as English or German. One of the most challenging tasks in handwritten character classification is, dealing with the huge variety of handwritings by different people.

The focal point of this research lies on a fundamental problem of handwriting recognition, which is, recognition of individual characters/alphabets of the Bangla language. By providing a method for individual character recognition, this research opens up the path for further development in the Bangla handwriting recognition sector. When combined with appropriate image processing techniques, Bangla handwriting recognition systems might have some practical use cases such as: sorting addresses written on letters at postal offices, handwritten cheque recognition in the banking sector etc.

The primary goal of this research is to build a model that can train on a large amount of data and classify new image samples in real time. The objectives of this research are, first of all, developing a handwritten character recognition model using Convolutional Neural Network (CNN). Then, observing the effect of using Data Augmentation on the dataset. After that, performing a comparison with other Machine Learning methods (such as Linear Support Vector Machine, Long Short Term Memory [2] etc) on the same dataset. There arose some research questions, which were answered in this research. The experiments were performed to see whether CNN is a suitable approach to effectively solve the handwritten character recognition problem. There was also an observation of how the amount of data affects a Deep Learning model's performance.

The remainder of this article is structured as follows: Section II covers related work on Bangla handwritten character recognition, while Section III provides an overview of the methodology and system architecture. Section IV briefly discusses about the dataset used in this experiment. Section V describes the experiment process and Section VI presents an analysis of the results. Finally, Section VII concludes the paper and discusses some future scopes of improvement.

II. Related Work

Although a lot of research has been done on English handwriting recognition there has not been much research performed on Bangla handwriting recognition. But there has been some notable researchwork conducted by computer scientists in the Indian subcontinent.

A Bangla handwritten character recognition system using CNN was developed by Rahman et al. [3] in 2015. Here, the CNN method achieved 85.96% accuracy on 50 classes. A custom dataset was prepared with total samples: 20,000 and 400 sample images for each class. Image resolution was 28x28 in this experiment.

A deep learning approach was applied on Bangla numeric digits by Zahangir et al. [4]. This approach managed a 98.8% accuracy on the 10 classes. The dataset used in this experiment was CMATERdb3.1.1 with 6000 images for Bangla Numerals [10 numeric digit classes]. Image resolution was 32x32 in this experiment.

A research on offline handwritten numeral recognition was conducted by Bhattacharya et al. [5] and the approached used in this case was, an ensemble of MLPs (Multi Layer Perceptrons) [6] [7] which were combined by Adaboost. This paper provides a detailed comparison between handwritten Bangla, Oria and Devnagari numeric digit recognition using the Multi-Layer Perceptron.

Chowdhury et al. [8] presented a research work of Bangla handwritten word recognition using fuzzy logic. There was no specific dataset used as the data was collected as time ordered sequence of coordinates. The mouse-downs and mouse-ups at various points were collected sequentially. This method only works for mouse input, and it's not compatible with image data.

Another online Bangla handwriting recognition system was developed by K. Roy et al. [9]. This approach uses mouse input and touch screen input to collect online handwriting data, and the main focus of this paper is to apply sequential information obtained from the movements of the pen on the touch screen surfaces to the quadratic classifier for recognition. The system was tested on 2500 numeric Bangla digits and 12500 Bangla characters. This method obtained 98.42% accuracy on 10 numeric data classes and 91.13% accuracy on 50 classes of character data.

An offline or image based Bengali handwritten character recognition system was implemented using Deep Convolutional Neural Network by Bishwajit et al. [10] in 2017. This experiment achieved 91.23% classification accuracy on 50 alphabets. Dataset used in this experiment was the Banglalekha-isolated dataset, which contains binary images of isolated Bangla alphabets. Image resolution used in this case was 28x28 and only 5% of the dataset was used for validation.

III. Methodology and System Architecture

Among different types of neural networks, CNN [11] is one of the best methods to perform classification on image data [12]. CNN was inspired by the visual cortex of the human brain [13] [14]. The layers of the human brain can detect complex features in order to recognize what is seen. In case of computers, the CNN image classifier takes an input image, processes it and classifies it under certain categories. The model that has been used to classify the Bangla handwritten characters was implemented using Keras [15] with Tensorflow [16] as backend. It contains a convolutional layer with 32 filters and the kernel size is 5x5. The activation function used in this layer is the Relu [17] activation function which introduces non-linearity. For the first layer, the input dimensions had to be specified, which, in this case, is 50x50x1, which means each image will be a grayscale 50x50 image. This layer is followed by a Max Pooling layer with pool size 2x2.

Then there is another convolutional layer with 32 filters and kernel size 5x5. The activation function used in this layer is the Relu activation function. This layer is also followed by a Max Pooling layer with pool size 2x2. The values obtained from the previous layers are then flattened. This one dimensional array is then provided to the fully connected layers.

Next, there are three fully connected layers with 1024 nodes in the first two layers and 512 nodes in the 3rd layer. The activation function for all three fully connected layers is Relu. The first two layers have a Dropout [18] value of 0.5 which means each node will have a dropout probability of 50%. This is helpful for preventing the overfitting issue. And the last fully connected layer is followed by the output layer. The output layer has 50 nodes, each corresponding to the 50 classes of Bangla alphabets. The activation function for the output layer is Softmax [19], which provides probabilistic values for each class.

This classifier model is then compiled, and the optimization function used in this model is the Stochastic Gradient Descent (SGD) [20] optimizer, with a learning rate of 0.001, momentum of 0.9 and nesterov set to True. The loss function used in this case is the Categorical Crossentropy [21], as there are more than two classes. The metrics of accuracy are also specified to be shown for the model, for evaluation purposes.

This architecture was chosen empirically by trying various combinations and number of nodes. Among all other models, this architecture was able to achieve the best accuracy in classifying the Bangla handwritten digits.

Summary of the architecture is displayed on Table I.

TABLE I
CNN ARCHITECTURE

Model Content	Details
First Convolution Layer	32 filters of size 5x5, ReLU
First Max Pooling Layer	Pooling Size 2x2
Second Convolution Layer	32 filters of size 5x5, ReLU
Second Max Pooling layer	Pooling size 2x2
First Fully connected Layer	1024 nodes, ReLU, Dropout = 0.5
Second Fully Connected Layer	1024 nodes, ReLU, Dropout = 0.5
Third Fully Connected Layer	512 nodes, ReLU
Output Layer	50 nodes for 50 classes, SoftMax
Optimization Function	Stochastic Gradient Descent (SGD)
Learning Rate	0.001
Metrics	Loss, Accuracy

The system was implemented on Google Colab, which is a cloud based web interface to run machine learning experiments, and is available for machine learning researchers for free. The system has 12 Gigabytes of memory, Intel Xeon CPU running at 2.20GHz clock speed and a robust GPU (Nvidia Tesla K80). It provides access to an online python notebook which acts as the user interface. To provide a graphical interface for the trained model, the Flask library of Python was utilized, which provides the backend of a web server where the model will be hosted. For the frontend, HTML, CSS and Javascript was used, where the user can interact with a canvas and write Bangla alphabets. The server, hosted at localhost:5000 by Flask, receives an image input and resizes it to 50x50 which the model expects, then it goes through the trained model and predicts the probability of each class. The one with the maximum probability is selected to be the letter which was drawn. The result is printed out to the webpage.

IV. DATASET

The dataset used in this experiment is named BanglaLekha-Isolated [1], which is a collection of Bangla handwritten isolated character samples. It contains sample images of 10 Bangla numeric digits, 50 Bangla basic characters and 24 compound characters. Approximately 2000 samples of handwriting for each characters were collected, pre-processed and digitized. After rejecting the mistakes and uninterpretable data samples, 1,66,105 handwritten character images were selected for the final dataset. The collection of this dataset was funded by the ICT Division of the Ministry of ICT, Bangladesh and it is publicly available for all sorts of research [1]. The dataset is expected to be used for pattern recognition and machine learning related tasks. From this dataset 75000 images of the 50 character samples have been used in this experiment, and excluded some noisy data which might hamper the learning process.

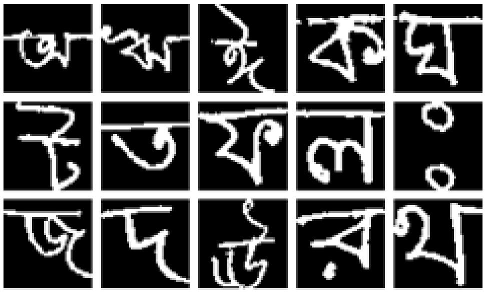

Fig. 1. Sample images from the dataset

V. EXPERIMENT

A. Training Phase with the base dataset

The dataset was converted to a csv file. There are 2501 columns and 75000 rows in the csv file as there are 75000 grayscale images of 50x50 resolution. Total number of classes is 50, so there are 1500 images for each class. The last column indicates which class the image belongs to. As this is a convolutional neural network model, this model expects image data as input, which are represented as 2D arrays. So, the data received from the csv file is reshaped as 50x50 arrays. Also the pixel values are normalized by dividing each pixel value by 255. Two separate arrays were created from the csv file, one containing the image data and the other containing the labels of the data. Then a train_test_split is performed, to create a validation set. which will be used during the training phase for validating the accuracy of the model on unknown data. 10% of the total dataset was used as validation set in this experiment, so 67500 samples were used for training and 7500 samples were used for validation. The training process took about 35 minutes, approximately 23 seconds per epoch. The process was stopped at 92 epochs by the EarlyStopping callback as the accuracy was decreasing due to overfitting.

B. Training Phase with expanded dataset

Augmented data means applying some changes (rotation, position shifting, zoom, shear etc.) to the existing data to generate some new data. This can be implemented using the ImageDataGenerator function from the Keras [15] library.

By using Data Augmentation the number of samples provided for training can be increased, which is essential for the model to generalize well on new data. Moreover, data augmentation introduces variation in the training images, which also helps the model to learn the characteristics of the alphabets well in the training phase. Some samples of Augmented Data are shown in figure 2.

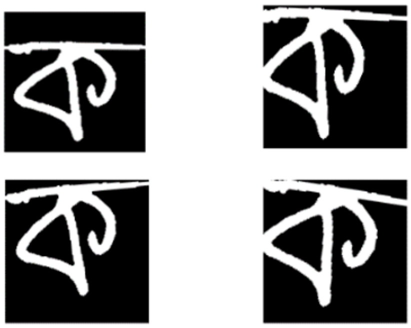

Fig. 2. Three augmented image samples created from one original image

The training process was performed once again, this time with 200,000 images using the augmentation process. For validation set 20,000 samples were used. Therefore, each class had 4000 images for training. The image resolution and the number of color channels were kept the same as the previous training for consistency. The model architecture was also the same. The training process was executed for 70 epochs. Each epoch required approximately 100 seconds, therefore taking almost 2 hours in total. Each epoch consisted of 780 steps with each step taking 125ms to execute. Although it took longer to train, some improvement over the previous training was observed.

To verify the robustness of the architecture that was used, the experiment was run again with a larger portion of the dataset used as test data. 65% of the dataset was used for training and 35% was used for validation.

This architecture was also applied to other datasets to validate the overall performance. Furthermore, LSTM and SVM based systems were implemented on the Bangla handwritten character datasets and a comparative study is presented in the Result Analysis section.

VI. Result Analysis

A. Validation Accuracy

Compared to the first training phase with the base dataset, the validation accuracy achieved with augmented data, was much better even though the number of epochs were less in the latter stage. From the graph in figure 3, It can be observed that the curve increases higher than the previous phase in the early iterations and reaches a steady state around 95% which is an improved result than the earlier phase.

B. Validation Loss

The validation loss with augmented data, is lower than the base dataset. The augmented data experiment curve goes downwards rapidly in the earlier epochs. The curve becomes almost stable at around 0.2 which is better than the experiment without augmented data, which achieved a loss value of slightly more than 0.3

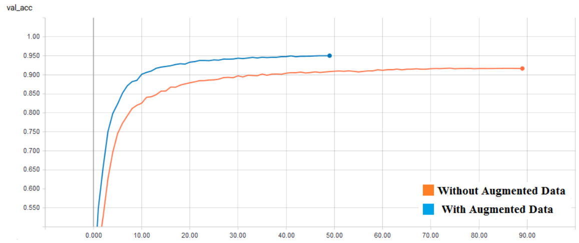

Fig. 3. Validation accuracy graph comparison

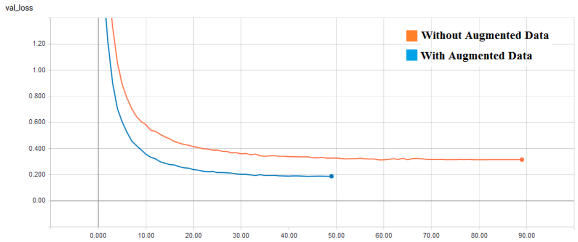

Fig. 4. Validation loss graph comparison

Some other factors were also considered for evaluating the performance of the model, they are: Recall, Precision and F1 score. The weighted average of the recall value of each class was 0.95. In case of precision the weighted average was also 0.95. For F1-score the weighted average was 0.95. There was very little fluctuation among the scores of each of the 50 classes.

Performing the same experiment with 65% of the dataset as training data and the remaining 35% as test data, this model achieves an accuracy metric of 94.576% and 0.204 loss value. The result is not much different than the previous experiment, which indicates that the model performs similarly well on a larger test set.

To check the versatility of the proposed architecture, the system was applied to some similar datasets, such as MNIST English handwritten digit dataset [22], CMATERdb 3.1.2 [23] and Ekush Bangla dataset [24]. The results of implementing the system on the new datasets are portrayed in a tabular form below:

TABLE II
Comparison between the performance of the system on different datasets

Dataset	Class	Accuracy	Loss
Banglalekha-isolated	50	91.81%	0.20
MNIST	10	99.25%	0.03
Ekush	59	95.07%	0.19
CMATERdb 3.1.2	50	93.37%	0.32

Using the same dataset on the Support Vector Machine [7] algorithm with the Linear kernel, the system achieved only a 51% accuracy. From this result, it can be interpreted that almost half of each of the classes were misclassified, although some classes had good recognition rates, this sort of performance is not reliable at all. The possible reason for such results from the SVM algorithm might be because of the nature of the data. The same dataset that was used for the CNN model was also applied here. But SVM with Linear kernel could not adapt to the non-linear nature of the human handwriting data. The result of SVM might improve if some sort of preprocessing was applied. For example: centering the alphabets in the image so that the pixels follow a similar spatial pattern in all the samples. But for raw data, SVM might not be suitable.

An LSTM based image classification system was built using Keras [15] and was applied to two Bangla handwritten character datasets for comparison. This architecture consists of one LSTM layer with 128 units, activation function was specified as 'ReLU', 'hard_sigmoid' was used as recurrent activation function and a dropout value of 0.25 was applied to this layer. The performance of the system is presented on the following table :

TABLE III
PERFORMANCE OF THE LSTM BASED SYSTEM

Dataset	Class	Accuracy	Loss
Banglalekha-isolated	50	87.41%	0.43
Ekush	59	93.06%	0.25

From the above experiment results, the different approaches that were used can be compared. Convolutional Neural Networks perform significantly better than Linear approaches like Linear Support Vector Machine (LSVM) when there is a huge amount of non-preprocessed data and a high number of classes. Data augmentation further improves the CNN classifier by supplying more data samples. Although LSTM provides respectable performance, the results of the experiments indicate that CNN is still the more preferable approach for handwritten character recognition.

TABLE IV
COMPARISON BETWEEN THE APPROACHES USED

Classifier	Accuracy
CNN	91.81%
SVM (Linear Kernel)	51%
Long Short Term Memory	87.41%
CNN with Augmented Data	95.25%

VII. Conclusion and Future Work

This research explores the opportunities and methods that are required to classify Bangla handwritten characters. The research proves that the CNN method is more efficient compared to other machine learning approaches. The proposed model also performs well on other datasets, which indicates the versatility of the system. It was also observed that using a larger amount of data with variation can help the model to learn the features or characteristics of the classes more effectively. The web interface also provides an easy way to interact with the model and perform real time validation.

Although the system is providing admirable results in classifying individual letters of the Bangla alphabet, there is no scope to detect a sequence of characters. Recognizing a sequence of alphabets or words as a whole, can be considered as future scope, as it requires more complex methods to implement, namely Recurrent Neural Networks (RNN [25]), Long Short Term Memory (LSTM) [2] architectures etc. The dataset regarding the computation can also be further increased, by adding more classes for numeric digits and compound letters. It will be interesting to see how the model performs while classifying an extensive number of Bangla characters (including all compound letters [26], numeric [27] digits). Some other extensions are also possible. For example, by applying knowledge driven methodology such as Belief Rule Base (BRB), which is widely used where uncertainty becomes an issue [28] [29] [30] [31] [32].

Acknowledgment

This study was funded by the Swedish Research Council under grant 2014-4251.

References

[1] Mar 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340917301117

[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[3] M. M. Rahman, M. Akhand, S. Islam, P. C. Shill, M. Rahman *et al.*, "Bangla handwritten character recognition using convolutional neural network," *International Journal of Image, Graphics and Signal Processing (IJIGSP)*, vol. 7, no. 8, pp. 42–49, 2015.

[4] M. Z. Alom, P. Sidike, T. M. Taha, and V. K. Asari, "Handwritten bangla digit recognition using deep learning," *arXiv preprint arXiv:1705.02680*, 2017.

[5] T. Jindal and U. Bhattacharya, "Recognition of offline handwritten numerals using an ensemble of mlps combined by adaboost," in *Proceedings of the 4th International Workshop on Multilingual OCR*. ACM, 2013, p. 18.

[6] U. Bhattacharya, M. Shridhar, S. K. Parui, P. Sen, and B. Chaudhuri, "Offline recognition of handwritten bangla characters: an efficient two-stage approach," *Pattern Analysis and Applications*, vol. 15, no. 4, pp. 445–458, 2012.

[7] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[8] K. Chowdhury, L. Alam, S. Sarmin, S. Arefin, and M. M. Hoque, "A fuzzy features based online handwritten bangla word recognition framework," in *2015 18th International Conference on Computer and Information Technology (ICCIT)*. IEEE, 2015, pp. 484–489.

[9] K. Roy, N. Sharma, T. Pal, and U. Pal, "Online bangla handwriting recognition system," in *Advances In Pattern Recognition*. World Scientific, 2007, pp. 117–122.

[10] B. Purkaystha, T. Datta, and M. S. Islam, "Bengali handwritten character recognition using deep convolutional neural network," in *2017 20th International Conference of Computer and Information Technology (ICCIT)*. IEEE, 2017, pp. 1–5.

[11] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *null*. IEEE, 2003, p. 958.

[12] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1717–1724.

[13] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, no. 1, pp. 106–154, 1962.

[14] R. M. Cichy, A. Khosla, D. Pantazis, A. Torralba, and A. Oliva, "Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence," *Scientific reports*, vol. 6, p. 27755, 2016.

[15] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[16] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

[17] W. Shang, K. Sohn, D. Almeida, and H. Lee, "Understanding and improving convolutional neural networks via concatenated rectified linear units," in *international conference on machine learning*, 2016, pp. 2217–2225.

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[19] R. A. Dunne and N. A. Campbell, "On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function," in *Proc. 8th Aust. Conf. on the Neural Networks, Melbourne*, vol. 181. Citeseer, 1997, p. 185.

[20] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[21] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of operations research*, vol. 134, no. 1, pp. 19–67, 2005.

[22] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[23] [Online]. Available: https://code.google.com/archive/p/cmaterdb/

[24] A. S. A. Rabby, S. Haque, S. Islam, S. Abujar, and S. Hossain, "Ekush: A multipurpose and multitype comprehensive database for online off-line bangla handwritten characters," 01 2019.

[25] S. Banerjee and S. Banerjee, "An introduction to recurrent neural networks – explore artificial intelligence – medium," May 2018. [Online]. Available: https://medium.com/explore-artificial-intelligence/an-introduction-to-recurrent-neural-networks-72c97bf0912

[26] S. Roy, N. Das, M. Kundu, and M. Nasipuri, "Handwritten isolated bangla compound character recognition: A new benchmark using a novel deep learning approach," *Pattern Recognition Letters*, vol. 90, pp. 15–21, 2017.

[27] 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.

[28] M. S. Hossain, S. Rahaman, A.-L. Kor, K. Andersson, and C. Pattinson, "A belief rule based expert system for datacenter pue prediction under uncertainty," *IEEE Transactions on Sustainable Computing*, vol. 2, no. 2, pp. 140–153, 2017.

[29] R. Karim, K. Andersson, M. S. Hossain, M. J. Uddin, and M. P. Meah, "A belief rule based expert system to assess clinical bronchopneumonia suspicion," in *2016 Future Technologies Conference (FTC)*. IEEE, 2016, pp. 655–660.

[30] M. S. Hossain, M. S. Khalid, S. Akter, and S. Dey, "A belief rule-based expert system to diagnose influenza," in *2014 9Th international forum on strategic technology (IFOST)*. IEEE, 2014, pp. 113–116.

[31] R. Ul Islam, K. Andersson, and M. S. Hossain, "A web based belief rule based expert system to predict flood," in *Proceedings of the 17th International conference on information integration and web-based applications & services*. ACM, 2015, p. 3.

[32] M. S. Hossain, S. Rahaman, R. Mustafa, and K. Andersson, "A belief rule-based expert system to assess suspicion of acute coronary syndrome (acs) under uncertainty," *Soft Computing*, vol. 22, no. 22, pp. 7571–7586, 2018.