TECHNIUM

Technium Vol. 18, pp.65-74 (2023)
ISSN: 2668-778X
www.techniumscience.com

An Efficient Method for Bangla Handwritten Digit Recognition Using Convolutional Neural Network

Indronil Bhattacharjee

Department of Computer Science
New Mexico State University
indronil@nmsu.edu

Abstract. Handwritten digit recognition is a fundamental problem in the field of computer vision and pattern recognition. This paper presents a Convolutional Neural Network (CNN) approach for recognizing handwritten Bangla digits. The proposed method utilizes a dataset of handwritten Bangla digit images and trains a CNN model to classify these digits accurately. The dataset is preprocessed to enhance the quality of the images and make them suitable for training the CNN model. The trained model is then tested on a separate test dataset to evaluate its performance in terms of accuracy. With the Ekush: Bangla Handwritten Data - Numerals dataset, we tested our CNN implementation to determine the precision of handwritten characters. According to the test results, 25% of the images using a training set of more than 150,000 images from Ekush dataset had an accuracy of 98.3%.

Keywords. Handwritten character recognition, Bangla number, Computer vision, Deep learning, Convolutional neural networks, Classification, Image processing

1. Introduction

Handwritten digit recognition is a significant research area in the fields of computer vision and pattern recognition. It plays a crucial role in various applications including optical character recognition, document analysis, and postal automation systems. With the increasing availability of digital devices and the need for efficient digit recognition systems, the development of accurate and reliable methods for recognizing handwritten digits has gained immense importance.

In this paper, we focus on the specific task of recognizing handwritten Bangla digits, which presents unique challenges owing to the complexity and distinctiveness of the Bangla script. Bangla, also known as Bengali, is the official language of Bangladesh and is widely spoken in the Indian states of West Bengal, Tripura, and Assam. It has its own distinct script that consists of a set of complex characters and symbols.

Recognizing handwritten Bangla digits is particularly important for applications in various domains, including digit recognition systems in educational institutions, automated form processing, and digital-based document analysis. Despite the increasing demand for accurate Bangla digit recognition systems, existing methods often face difficulties in accurately recognizing handwritten Bangla digits because of their varying writing styles, distortions, and overlapping strokes.

In this study, we propose a Convolutional Neural Network (CNN) approach for recognizing handwritten Bangla digits. CNNs have proven highly effective in various computer vision tasks, including image classification, object detection, and digit recognition. By leveraging the power of deep learning and

CNNs, we aimed to develop a robust and accurate model capable of accurately recognizing handwritten Bangla digits.
Our methodology involves collecting a dataset of handwritten Bangla digit images, preprocessing them to enhance their quality, and preparing them for training. We utilized techniques such as thresholding, resizing, and color space conversion to transform the raw images into a suitable format for training the CNN model. The dataset was then split into training and testing sets to evaluate the performance of the trained model.
The CNN model architecture consists of multiple layers, including convolutional, max-pooling, and fully connected layers. These layers were designed to extract relevant features from the input images and perform classification based on these features. The model was trained using the training dataset, optimized using stochastic gradient descent, and evaluated using the testing dataset to measure its accuracy and performance.
Through extensive experiments and evaluations, we demonstrated the effectiveness of our proposed CNN approach in accurately recognizing handwritten Bangla digits. We analyzed the classification results using a confusion matrix to gain insight into the performance of the model. The promising results obtained from our methodology highlight the potential of CNN-based approaches for handwritten Bangla digit recognition and their applications in various real-world scenarios.

2. Background Study

Due to its applicability in a variety of fields, handwritten character recognition has attracted substantial attention in the fields of pattern recognition and machine learning. Several methodologies for character recognition have been proposed, with a substantial focus on converting text content from physical documents into digital formats for enhanced accessibility and storage efficiency [1].
The architecture of Convolutional Neural Networks (CNNs) emulates the communication patterns observed in the human brain's neurons, drawing inspiration from the arrangement found in the visual cortex. This design enables CNNs to capture spatial dependencies within an image, facilitating recognition tasks. The architecture systematically aligns with the source image, identifying distinct features critical for classification. During the training phase, weights, parameters, and biases are refined to transform the original image into a feature vector, enriching the model's understanding of the image's nature [2].
Figure 1 [1] illustrates the CNN process, consisting of key components such as convolution, pooling, and fully connected neural networks, which are elaborated as follows:

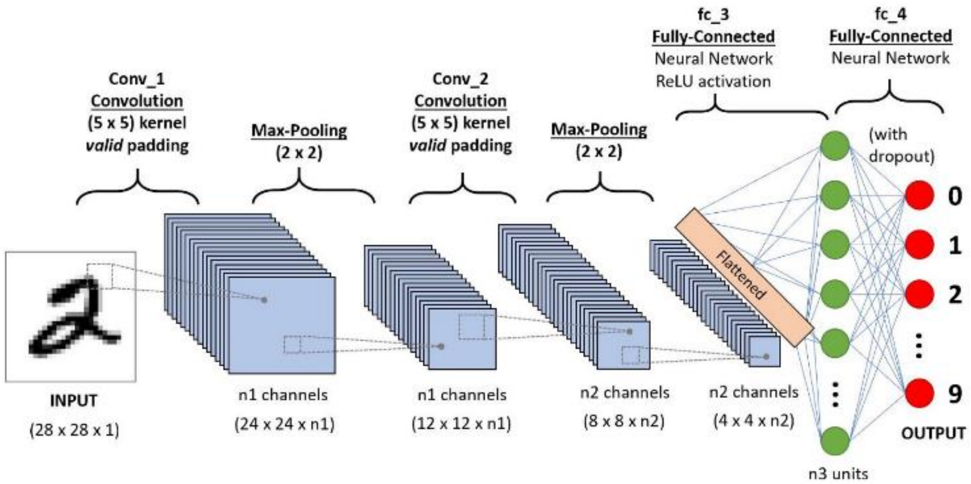

Fig. 1. Convolutional Neural Network

2.1. Convulation

Convolution, a process widely employed for image manipulation tasks such as image smoothing and sharpening, forms a fundamental part of CNNs. This operation involves applying a matrix (kernel or filter) to an image, modifying the image's structure. Each pixel in the image submatrix is multiplied by its corresponding value in the kernel, and the results are summed to generate a single pixel value for the filtered image. This operation is repeated across the entire image. Convolution captures low-level features like edges, gradients, and color, and can be further adapted to higher-level feature extraction through additional layers [3].

2.2. Pooling

Another critical component of CNNs is pooling, which aids in increasing the spatial scale of the image representation, reducing computational complexity. Different types of pooling, such as max-pooling, average pooling, and adaptive pooling, can be applied based on the application's requirements. Max-pooling, a common technique, retains dominant features while discarding noise, thereby contributing to efficient model training. It preserves rotational and translational variations in the features, enhancing the model's robustness. On the other hand, average pooling is beneficial for noise suppression and dimensionality reduction [3].

2.3. Fully connected neural network

The final phase of CNN involves a fully connected neural network. This network, composed of interconnected artificial neurons, operates on the results obtained from the previous convolution and pooling stages. During training, the network's weights are adjusted. This network takes the outcome of the convolution and pooling operations and predicts the most fitting label that characterizes the image. The process involves multiplying the outputs from the preceding stages by weights and passing them through an activation function [4].

In summary, CNNs harness convolution, pooling, and fully connected neural networks to progressively capture image features, enabling accurate and effective recognition tasks. These networks mimic the human brain's visual processing patterns and have proven to be powerful tools in various image-based applications.

3. Related Works

Despite being a challenging research problem, several studies on Bangla Handwritten Digit and Character Recognition have been conducted. DenseNet [5] was used by Md. Zahangir Alom et al. [6] used 12000 photos for training and 3000 images for testing. They were able to identify 95.13% of Handwritten Bangla digits. Alif et al. [7] employed modified ResNet-18 and achieved 95.10% accuracy. They employed CMATERdb and BanglaLekha-Isolated as datasets. Rumman et al. [8] also used CNN to the BanglaLekha dataset and obtained 91.81% accuracy. Khan, et al. [9] employed SE-ResNeXt and achieved an average accuracy of 96.82 percent.

Using Bangla scripts, Hazra et al. [10] constructed a new dataset where they obtained 94.12% on CMATERdb. By Saha, Faisal, and Rahman [11], a DCNN model known as BBCNET-15 attained 96.40% on CMATERdb. Numerous studies have also been conducted using solely numbers. On NumtaDB, Shawon et al. [12] scored 92.72%. Another paper by Saha et al. [13] used CMATERdb and yielded 93.24% accuracy for Bangla digits. Noor et al. [14] received a NumtaDB score of 96.98%. Using autoencoder and deep ConvNet on CMATERdb, Shopon et al. [15] were able to achieve 99.50% accuracy. Using the KNN classifier, on the CMATERdb database, Hassan et al. [16] got 96.7% on Bangla digits.

4. The Proposed System

The proposed method utilizes a CNN architecture for handwritten Bangla digit recognition. The dataset used for training and testing the model consists of images of handwritten Bangla digits. The images are

Technium Vol. 18, pp.65-74 (2023)
ISSN: 2668-778X
www.techniumscience.com

preprocessed to improve their quality and remove any noise or artifacts. The preprocessing steps include thresholding, resizing, and color space conversion.

The CNN model consists of several layers, including convolutional layers, max-pooling layers, and fully connected layers. The convolutional layers extract relevant features from the input images, while the max-pooling layers down sample the feature maps to reduce the computational complexity. The fully connected layers perform classification based on the extracted features.

The model is trained using a training dataset and optimized using the stochastic gradient descent (SGD) optimizer. The loss function used is the sparse categorical cross-entropy, which is suitable for multiclass classification problems. The model is trained for multiple epochs to improve its accuracy.

4.1. Dataset

The first step is to collect a dataset of handwritten Bangla digit images. In this study, the dataset is obtained from Kaggle datasets, where each subdirectory corresponds to a specific digit label. To address the issue, we worked with a personalized convolutional neural network using the Ekush: Bangla Handwritten Data - Numerals dataset [17].

There are ten files overall in the Ekush dataset, each having more than 3,000 each of the handwritten Bangla numbers. In this study, we trained our CNN model on classes 0 through 9. There are around 3000 png-format pictures in each folder. We have 31,130 example photos in 10 classifications. We use 75% of the sample data during training and the remaining 25% during model testing. Additionally, we put our model to the test using information gathered from 300 randomly selected individuals.

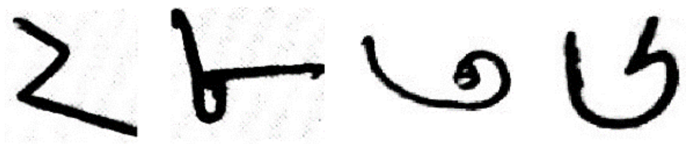

Fig. 2. Unprocessed Images

4.2. Data Preprocessing

Each image in the dataset undergoes a series of preprocessing steps to enhance the quality and prepare them for training the CNN model.

First, the image is converted to a binary image using a threshold value of 55. This step helps to separate the foreground (digits) from the background. The image is then resized to a fixed size of 56x56 pixels for developing the computational time.

Additionally, the color channels of the image are converted from BGR to RGB format. The preprocessed images are stored in the image list.

Fig. 3. Preprocessed Data

4.3. Data Augmentation

Data augmentation is used to enlarge datasets without actually adding new data. Several methods, including resizing, rotation, zooming in or out, etc. are used to produce new data from the old dataset for picture data [18]. In our research, we expanded the amount of our dataset using a method known as zooming and rotation. Every image was taken and given a 30-degree right and left rotation, along with zoom in and out. Thus, it basically turned one example image into five new ones. Ekush, our main dataset, initially had 31,130 images as samples. Our dataset had 155,650 sample pictures after data augmentation.

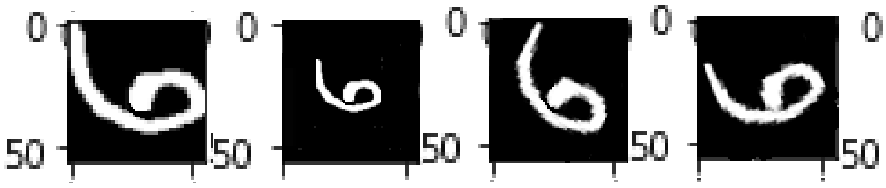

Fig. 4. Augmented Data

4.4. Model Description

4.4.1. Input Layer

a) Conv2D Layer (16 filters, tanh activation): This layer performs a 2D convolution operation on the input image. It uses 16 filters to extract features from the image. The activation function used here is the hyperbolic tangent (tanh).

b) MaxPooling2D Layer (2x2 pool size, stride 2x2): This layer performs max pooling, which downsamples the feature maps obtained from the previous Conv2D layer. It reduces the dimensions of the feature maps by selecting the maximum value in each 2x2 window.

4.4.2. Convolution Layer

a) Conv2D Layer (32 filters, ReLU activation): This is another convolutional layer that uses 32 filters to further extract more complex features from the downsampled feature maps. The activation function used here is the Rectified Linear Unit (ReLU).

4.4.3. Max Pooling Layer

a) MaxPooling2D Layer (2x2 pool size): This is another max pooling layer that further downsamples the feature maps obtained from the previous Conv2D layer. It again reduces the dimensions of the feature maps by selecting the maximum value in each 2x2 window. Pooling helps to make the model more robust to variations in position and scale of features in the input images [19].

4.4.4. Flatten Layer:

a) Flatten Layer: This layer converts the 2D feature maps from the previous layer into a 1D vector, which can be used as input for fully connected layers.

4.4.5. Fully Connected Layers:

a) Dense Layer (1000 neurons, ReLU activation): This fully connected layer has 1000 neurons. It performs a linear transformation on the input data and then applies the Rectified Linear Unit (ReLU) activation function.

b) Dense Layer (10 neurons, softmax activation): This is the final fully connected layer with 10 neurons, which corresponds to the number of classes in the classification task. The softmax activation function converts the output into class probabilities.

In summary, the Conv2D layers extract features from the input image, the MaxPooling2D layers downsample the feature maps, the Flatten layer prepares the data for fully connected layers, and the Dense layers perform classification based on the extracted features. The architecture is designed for image classification using Convolutional Neural Networks (CNNs).

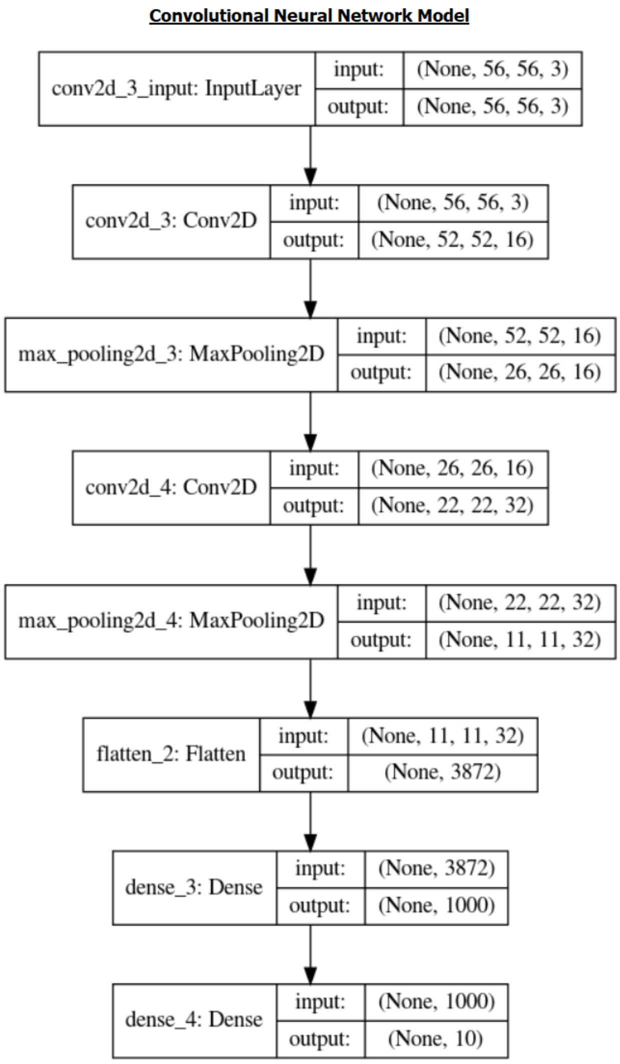

Fig. 5. Proposed Model of the Study

5. Results

We utilized Kaggle services with GPU features to process our sample images and train the CNN model we developed for this study. Our analyses were carried out on GPU runtime. In the backend, we utilized tensorflow and keras too.

5.1. Experimental Results

We carried out our experiment on the Ekush dataset with augmented data. For training and testing, 40 epochs were performed. We trained the model with 75% of our data and validated it with the rest 25%. Moreover, randomly collected handwritings were also tested with the model where 300 sample pictures were used. As seen in Figure 6 and 7, our training accuracy exceeds the predicted validation accuracy since our prediction model was not overfitted.

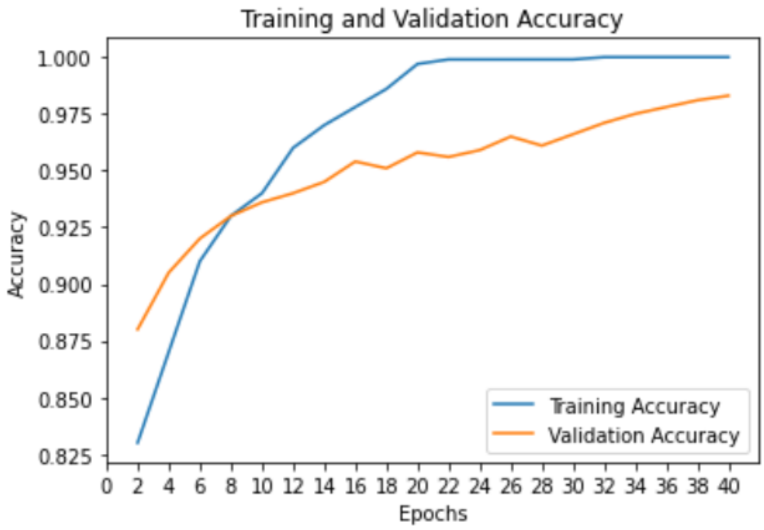

Fig. 6. Training vs Validation Accuracy of the model

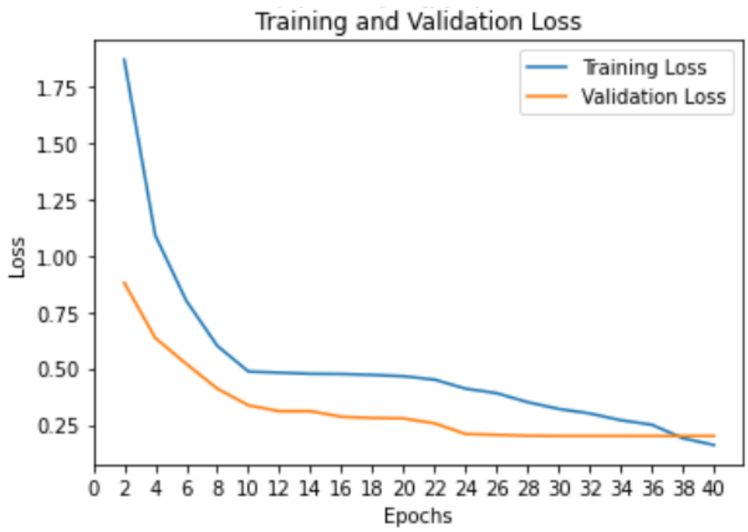

Fig. 7. Training vs Validation Loss of the model

For the Ekush dataset, the model's attained accuracy is 98.3%. This remarkable accuracy highlights the efficiency and dependability of the suggested method, highlighting its capacity to carry out high-precision character recognition jobs. Given that effective character recognition is essential for a variety of applications, the model's ability to achieve such a high accuracy rate highlights its significance and usefulness in real-world situations.

Moreover, a random handwriting dataset of 300 images was used to validate the model too. In this case, the model correctly detected 296 digits and failed to get 4. Accuracy for this dataset was 98.7%. The confusion matrix for the random dataset was like the following:

Model Recognized \ Actual	0,	1,	2,	3,	4,	5,	6,	7,	8,	9
0	[30,	0,	0,	0,	0,	0,	0,	0,	0,	1],
1	[0,	28,	0,	0,	0,	0,	0,	0,	0,	0],
2	[0,	0,	30,	0,	0,	0,	0,	0,	0,	0],
3	[0,	0,	0,	30,	0,	0,	0,	0,	0,	0],
4	[0,	0,	0,	0,	30,	1,	0,	0,	0,	0],
5	[0,	0,	0,	0,	0,	29,	0,	0,	0,	0],
6	[0,	0,	0,	0,	0,	0,	30,	0,	0,	0],
7	[0,	0,	0,	0,	0,	0,	0,	30,	0,	0],
8	[0,	0,	0,	0,	0,	0,	0,	0,	30,	0],
9	[0,	2,	0,	0,	0,	0,	0,	0,	0,	29]

Fig. 8. Confusion Matrix for Random Dataset

5.2. Result Comparison

The result of the experiment is presented by comparing with related works. In Table I, the comparison among this work and other relevant works is shown.

Table 1. Comparison Of Accuracy With The Related Works

Related Works	Method Used	Dataset	Data Size	Accuracy
This work	CNN	Ekush	155,650	98.3%
Hassan et al. [16]	KNN	CMATERdb	6000	96.7%
S. Alam et al. [12]	CNN	NumbtaDB	85000	92.7%
Huda et al. [18]	CNN	NumbtaDB	72000	97.9%
Khan et al. [9]	Sparse Representation	CMATERdb	6000	94.2%

6. Future Work

We utilized Kaggle services with GPU features to process our sample images and train the CNN model we developed for this study. Our analyses were carried out on GPU runtime. In the backend, we utilized tensorflow and keras too. In the pursuit of advancing the scope and capabilities of our research, our future endeavors will be focused on extending our methodology to encompass the complexities presented by the Bangla alphabet and its intricate combination of characters. This expansion holds immense promise as it introduces a new dimension of challenges and opportunities in the field of character recognition.

Working with Bangla alphabets and complex letter structures presents unique challenges due to the intricate nature of their composition. The inherent intricacies of Bangla script, including the presence of compound characters and variations in letter forms, necessitate a refined and adaptive approach to character recognition. Our future research will involve developing specialized models and techniques tailored to handle these complexities effectively.

One of the primary goals of this future work is to enhance the versatility of our recognition system to accurately decipher a wide range of Bangla characters, including standalone letters, compound characters, and ligatures. This involves exploring strategies for handling variable character shapes, sizes, and orientations, which are characteristic of Bangla script.

References

[1] Khandokar, I., Hasan, M., Md, Ernawan, F., Islam, S., Md, & Kabir, M. N. (2021). Handwritten character recognition using convolutional neural network. In Journal of Physics: Conference Series (Vol. 1918, Issue 4, p. 042152). IOP Publishing. https://doi.org/10.1088/1742-6596/1918/4/042152.

[2] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, L., Wang, G., Cai, J., & Chen, T. (2015). Recent Advances in Convolutional Neural Networks (Version 6). arXiv. https://doi.org/10.48550/ARXIV.1512.07108.

[3] Ptucha, R., Petroski Such, F., Pillai, S., Brockler, F., Singh, V., & Hutkowski, P. (2019). Intelligent character recognition using fully convolutional neural networks. In Pattern Recognition (Vol. 88, pp. 604–613). Elsevier BV. https://doi.org/10.1016/j.patcog.2018.12.017.

[4] Farabet, C., Couprie, C., Najman, L., & LeCun, Y. (2013). Learning hierarchical features for scene labeling. IEEE transactions on pattern analysis and machine intelligence, 35(8), 1915-1929.

[5] Rashid, M. M., Rokon, I., Ahmed, T., & Islam, M. R. (2020). Performance Evaluation of Various Feature Extraction Techniques for Handwritten Bangla Digit Recognition. In 2020 12th International Conference on Software, Knowledge, Information Management and Applications (SKIMA) (pp. 1-6). IEEE.

[6] Md Zahangir Alom, Paheding Sidike, Mahmudul Hasan, Tarek M. Taha, Vijayan K. Asari, "Handwritten Bangla Character Recognition Using the State-of-the-Art Deep Convolutional Neural Networks", Computational Intelligence and Neuroscience, vol. 2018, Article ID 6747098, 13 pages, 2018. https://doi.org/10.1155/2018/6747098.

[7] M. Al Rabbani Alif, S. Ahmed and M. A. Hasan, "Isolated Bangla handwritten character recognition with convolutional neural network," 2017 20th International Conference of Computer and Information Technology (ICCIT), 2017, pp. 1-6, doi: 10.1109/ICCITECHN.2017.8281823..

[8] R. R. Chowdhury, M . S. Hossain, R. ul Islam, K. Andersson and S. Hossain, "Bangla Handwritten Character Recognition using Convolutional Neural Network with Data Augmentation," 2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR), 2019, pp. 318-323, doi: 10.1109/ICIEV.2019.8858545.

[9] Mohammad Meraj Khan, Mohammad Shorif Uddin, Mohammad Zavid Parvez, Lutfur Nahar, A squeeze and excitation ResNeXt-based deep learning model for Bangla handwritten compound character recognition, Journal of King Saud University Computer and Information Sciences, 2021, ISSN 1319 -1578, https://doi.org/10.1016/j.jksuci.2021.01.021..

[10] Hazra, A., Choudhary, P., Inunganbi, S. et al. Bangla-Meitei Mayek scripts handwritten character recognition using Convolutional Neural Network. Appl Intell 51, 2291–2311 (2021). https://doi.org/10.1007/s10489-020-01901-2.

[11] C. Saha, R. H. Faisal and M. M. Rahman, "Bangla Handwritten Basic Character Recognition Using Deep Convolutional Neural Network," 2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd

International Conference on Imaging, Vision & Pattern Recognition (icIVPR), 2019, pp. 190-195, doi: 10.1109/ICIEV.2019.8858575.

[12] S. Alam, T. Reasat, R.M. Doha, and A.I Humayun, “NumbtaDB Assembled Bengali Handwritten Digits”, arXiv:1806.02452[cs.CV], 6 June, 2018.

[13] C. Saha, R. H. Faisal and M. Mostafijur Rahman, "Bangla Handwritten Character Recognition Using Local Binary Pattern and Its Variants," 2018 International Conference on Innovations in Science, Engineering and Technology (ICISET), 2018, pp. 236-241, doi: 10.1109/ICISET.2018.8745645.

[14] R. Noor, K. Mejbaul Islam and M. J. Rahimi, "Handwritten Bangla Numeral Recognition Using Ensembling of Convolutional Neural Network," 2018 21st International Conference of Computer and Information Technology (ICCIT), 2018, pp. 1 -6, doi: 10.1109/ICCITECHN.2018.8631944.

[15] M. Shopon, N. Mohammed and M. A. Abedin, "Bangla handwritten digit recognition using autoencoder and deep convolutional neural network," 2016 International Workshop on Computational Intelligence (IWCI), 2016, pp. 64-68, doi: 10.1109/IWCI.2016.7860340.

[16] T. Hassan and H. A. Khan, "Handwritten Bangla numeral recognition using Local Binary Pattern," 2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), 2015, pp. 1-4, doi: 10.1109/ICEEICT.2015.7307371.

[17] Rabby, A. S. A., Haque, S., Islam, Md. S., Abujar, S., & Hossain, S. A. (2019). Ekush: A Multipurpose and Multitype Comprehensive Database for Online Off-Line Bangla Handwritten Characters. In Communications in Computer and Information Science (pp. 149–158). Springer Singapore. https://doi.org/10.1007/978-981-13-9187-3_14.

[18] Huda, Hasibul & Fahad, Md & Islam, Moonmoon & Das, Amit. (2022). Bangla Handwritten Character and Digit Recognition Using Deep Convolutional Neural Network on Augmented Dataset and Its Applications. 1-7. 10.1109/IMCOM53663.2022.9721634.

[19] R. A. Tuhin, B. K. Paul, F. Nawrine, M. Akter and A. K. Das, "An Automated System of Sentiment Analysis from Bangla Text using Supervised Learning Techniques," 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), Singapore, 2019, pp. 360-364.