BanglaHandwritten: A Comparative Study among Single, Numeral, Vowel Modifier, And Compound Characters Recognition Using CNN

Conference Paper · October 2022

DOI: 10.1109/ICCCNT54827.2022.9984562

7 authors, including:

Nusrat Nabi
Daffodil International University

Md. Sazzadur Ahamed
Daffodil International University

Gazi Hadiuzzaman
Bangladesh University of Textiles

BanglaHandwritten: A Comparative Study among Single, Numeral, Vowel Modifier, And Compound Characters Recognition Using CNN

Sadia Jaman
Dept. name of CSE
Daffodil International University
Dhaka,Bangladesh
sadia15-10710@diu.edu.bd

Mehadi Hassan Sovon
Dept. name of CSE
Daffodil International University
Dhaka,Bangladesh
mehadi15-10712@diu.edu.bd

Syed Raihanuzzaman
Dept. name of CSE
Daffodil International University
Dhaka,Bangladesh
syed15-10568@diu.edu.bd

Md. Mehadi Hasan
Dept. name of CSE
Daffodil International University
Dhaka,Bangladesh
mehedi.shesher.1380@gmail.com

Nusrat Nabi
Dept. name of CSE
Daffodil International University
Dhaka,Bangladesh
nusrat15-10524@diu.edu.bd

Md. Sazzadur Ahamed
Dept. name of CSE
Daffodil International University
Dhaka,Bangladesh
sazzad.cse@diu.edu.bd

Gazi Hadiuzzaman
Dept. name of Apparel Engineering
Bangladesh University of Textiles
Dhaka,Bangladesh
gazihadiuzzaman@gmail.com

***Abstract*— The difficulty of handwritten character identification varies by language, owing to differences in shapes, lines, numbers, and size of characters. There are several studies for the identification of handwritten characters accessible for English in comparison with other significant languages like Bangla. In their recognition procedures, existing technologies use multiple techniques such as classification tools and feature extraction. CNN has recently been shown to be proficient in handwritten character recognition in English. A Handwritten Bangla character identification system based on CNN has been examined in this research. Using CNN, the suggested approach for feature, labelling and normalizing the handwritten character of images, as well as categorizing different characters. It doesn't use a feature extraction approach like previous research in the field. This research used almost 4,50,000 unique handwritten characters in a variety of styles. The recommended model has been proved to have a high recognition accuracy level and outperforms some of the most widely used methods already in use. In this research, identify the Bangla handwritten character and digits with the use of 189 classes consisting of 50 fundamental characters, 119 compound characters, 10 numerals, and 10 modifiers. The accuracy rate of basic characters is 84.62%, numerals 94%, modifiers 96.46%, compound characters 77.60% using created new model.**

Keywords—Bangla Handwritten, Convolutional Neural Network, Recognition, Vast Dataset, and Classes.

I. INTRODUCTION

The world's population has extended to 7 billion across 195 countries. Almost every country has its own language and alphabet. Recognition of characters and alphabets is a broad area of research. Presently, recognition of characters has been accorded massive attention in interim printed forms and handwritten. Printed forms are easier than handwritten text recognition because different types of people have different types of handwriting. Bangla language holds the 5th rank in the world's spoken languages and is the mother tongue of Bangladesh. More than 200M people exert Bangla for speaking along with writing purposes. UNESCO declared February 21st as Mother Language Day Internationally to honour the martyrs who fought for their mother tongue in 1952 Bangladesh. Therefore, now is the right time to computerize the Bengali language. The Bangla language has 50 characters, including digits, vowel modifiers, and many compound letters. There are some similar shapes of characters and compound letters in the Bangla alphabet. That's why achieving a better result is so difficult. The handwriting recognition system consists of two main steps: extracting features from a dataset and classifying individual characters using learning tools. Convolutional Neural Networks (CNN) have grown popular in recent years for complicated visual identification because of their architecture and several research studies on utilizing deep CNN to detect handwritten numbers [1], characters [2] [3], and other complex optical recognition. Because of its unique characteristics, CNN has been deemed effective in recognizing handwritten characters.

II. RELATED WORK

Many previous experiments for handwritten character identification in a different language, such as Hindi and English, have had significant success. A few studies on the handwritten fundamental digit, character, and compound character recognition in Bangla are available. Some literature on the identification of Bangla characters has been published in recent years. Addressing various kinds of research papers in the section of the literature review. Some of them use machine learning approaches like SVM classifier, and gradient feature and those use deep learning approaches like CNN with less amount dataset. Also, it is observed that most of the work is done separately and used a pre-trained model. Hence, it motivates us to work on a CNN-based handwritten recognition system that uses basic characters, compounds, modifiers, and digits combined in a paper and focused on creating a new model, using a vast amount of datasets by combining two separate datasets.

In recent years, some literature has documented the recognition of Bangla characters as " Using SVM classifier and MLP Basic Bangla and Compound handwritten character recognition [4]." The recommended models worked with their own datasets, which consisted of 250 participants, and achieved the recognition rates of MLP and SVM of around 79.25% and 80.51%, respectively. The study of Md. Mahbubar Rahman et al. [5], worked on Bangla's 50 isolated characters, normalizing the dataset and using the CNN approach for classification. The dataset consisted of 20,000 samples only, resized samples of each character in the 28X28 dimension. The founding rate of recognition is 85.96%. In [6], the authors mainly discussed the BanglaLekha-Isolated dataset consisting of over 2,00,000 data which was expanded by data augmentation. After preprocessing, the selected data was 1,66,105, consisting of 50 characters, ten digits, and 24 compound letters. After implementing the CNN, the model achieved an accuracy of 95.25%. But the paper focused on only 50 characters. However, using digits and compound letters would have been more appreciable. In another study [7], the researchers proposed and evaluated a CNN-based handwritten character recognition technique on the BanglaLekha-Isolated database. The accuracy rate of numerals is 98.66%, vowels 94.99%, compound letters (20 classes) 91.60%, alphabets 91.23%, and the accuracy rate of Bengali characters is around 80 classes, 89.93%. However, adding more classes would have given more accuracy. In [8], the authors focused on BBCNet-15 for recognizing handwritten characters, and it's the pre-trained model of DCNN (Deep Convolutional Neural Network). Using the dataset of this scheme CMATERdb 3.1.2. Rate of the accuracy based on Bangla basic 50 characters is 96.40%. However, the numerals and compound letters will be attached to it.

In [9], the researchers used a 9-layer sequential CNN model which is used to detect 60 Bangla handwritten characters, 17,645 Numerals data of 10 digits, 83,458 data of 50 basic characters are collected from the BanglaLekha-Isolated dataset for the training validation, and used as their own dataset for a test set where 25 students participate. The result of the existing dataset is 99.44%, and the test set is 95.16%, which is prepared. In another study [2], the authors introduced the CNN model for identifying Handwriting Letters, including 50 fundamental Bangla characters. Experiments were conducted using three datasets: BanglaLekha-Isolated, ISI dataset, and CMATERdb. Using the BornoNet model, validation accuracy of CMATERdb is 98%, ISI dataset 96.81%, BanglaLekha-Isolated dataset 95.71%, and combined dataset 96.40%. In [3], in this research, the researcher described the EkushNet diagram to identify Bangla handwritten fifty primary characters, ten modifiers, ten digits, and fifty-two compound characters used mostly. For validation and cross-validate Ekush dataset and CMATERdb dataset were used, respectively. The accuracy for the Ekush dataset is 97.73% and 95.01% for the CMATERdb dataset. However, the result would have been better if more prominent architecture could have been used. The study of Tandra Rani Das et al. [10], developed an extended CNN model to identify Bangla handwritten characters. The model has experimented with the "Bangla Lekha-Isolated" dataset consisting of 10 numerals, 39 consonants, and 11 vowels. The accuracy of Bangla numerals is 99.50%, vowels 93.18%, consonants 90.00%, and combined classes 92.25%. However, using more classes and compound letters would have been great. In another study [11], authors came up with a Handwritten system that used both CNN and BiLSTM (Bidirectional long short-term memory). The feature was tested on the CMATERdb 3.1.3.3 dataset consisting of 171 classes. The accuracy of the model is 98.50%. The model was only tested with compound letters. However, adding digits, simple characters, and modifiers would have been more promising. In a study [18], using CNN and VGG-16 models the authors got 93.07%. In [19], the authors used the CNN approach in three different datasets and achieved 87%, 89.6%, and 83.1% from CMATERdb, BanglaLekha-Isolated, and Ekush respectively. In another study [20], authors used machine learning techniques and the method SVM as well as got 91% accuracy on average.

Table 1. Some exclusive reviews of research work

Author's Name	**Used Algorithm**	**Accuracy**	**Year**
Md. Mahbubar Rahman et al.[5]	CNN (50 classes)	85.96%	2015
Rumman Rashid Chowdhury et al.[6]	CNN (84 classes)	95.25%	2019
Bishwajit Purkayastha et al.[7]	CNN (80 classes)	89.93%	2017
Chandrika Saha et al.[8]	CNN (50 classes)	96.40%	2019
Tandra Rani Das et al.[10]	CNN (60 classes)	92.25%	2021

When going through these exclusive reviews, it is observed that most of the work is done by basic characters, compounds, modifiers, and digits separately, using a pre-trained model, and a low quantity of data and classes. Hence, it motivates,

- To work on a CNN-based handwritten recognition system using basic characters, compounds, modifiers, and digits combined in a paper.
- To use customized models.
- To work with huge data samples of around 450000, under 189 classes.

III. METHODOLOGY

To develop the study firstly collect huge data from two different datasets. After that, pre-process the dataset to reduce noise and invert all the images. Then resize all the images in 60X60 pixels and complete this section applied it to feature extraction. Lastly, to do the labelling of the classes according to features. Here applied the CNN approach with created model.

A. Proposed Methodology

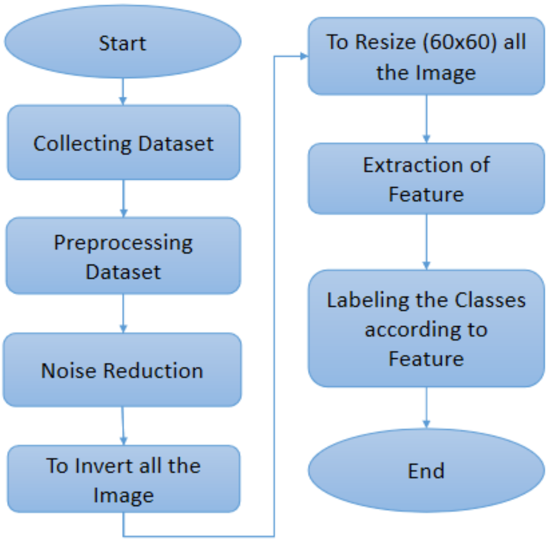


Fig.1. Proposed methodology flow diagram.

B. Collection of Data

For the purpose of this study, there were two types of datasets utilized in this study, called the "EkushNet"[12] and "BanglaLekha Isolated Dataset"[13]. After combining datasets, 50 characters consisting of 11 vowels and 39 consonants shown in (eg, Fig.2) and (eg, Fig.3), ten numerals shown in (eg, Fig.4), ten modifiers shown in (eg, Fig.6), and 119 compound letters shown in (eg, Fig.5) were collected, modified, and preprocessed. Hence, the total number of data samples is around 450000, down from 500000, which is under 189 classes. The Ekush dataset's image resolutions are different because of character size and diverse people's writing styles, the dataset comprises a wide range of unique characters. Several of these character representations are shaped in a highly complicated way and are tightly connected. Bangla's writing starts from left to right. The majority of Bangla characters feature an aclinic line in the top section, as can be seen. This line is known as Matra/Shirorekh[2] When a consonant or else vowel after a consonant has a compound ortho-graphic form, it is called a compound character[2]. Compound characters may be made up of two consonants or a consonant and a vowel. Bangla allows for the compounding of multi-characters. In Bangla, the number of compound characters is 200. This study looks at recognizing such complex-shaped compound characters, using 119 prominent Bangla compound characters as examples.

1	2	3	4	5	6	7	8	9	10
অ	আ	ই	ঈ	উ	ঊ	এ	ঐ	ও	ঔ

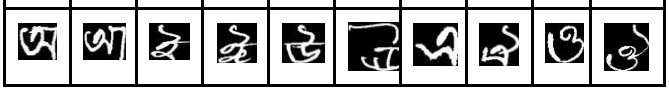


Fig.2. Bangla basic vowel characters

1	2	3	4	5	6	7	8	9	10
ক	খ	গ	ঘ	ঙ	চ	ছ	জ	ঝ	ঞ

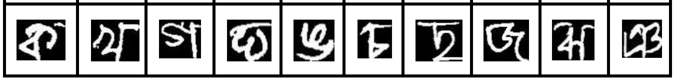
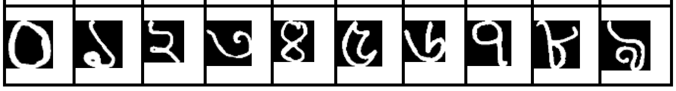


Fig.3. Bangla basic consonant characters

0	1	2	3	4	5	6	7	8	9



s

Fig.4. Bangla numerals

1	2	3	4	5	6	7	8	9	10
ন্ড	ড্ড	ষ্ঠ	ল্প	ষ্প	ন্দ	ন্ধ	ম্ম	ণ্ঠ	প্ত

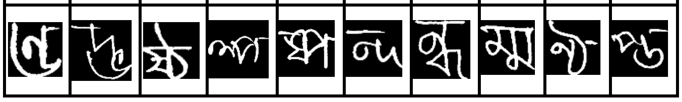


Fig.5. Bangla compound characters

1	2	3	4	5	6	7	8	9	10
া	ি	ী	ু	ূ	ে	ো	ৈ	ৃ	ৌ

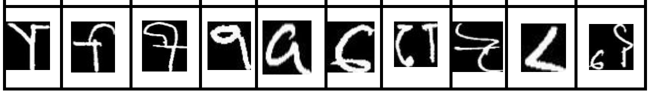


Fig.6. Bangla vowel modifiers

C. Pre-processing of Dataset

Preprocessing transforms random pictures into a consistent shape or form that may be fed into classifiers. In deep learning, data preparation plays a vital role. Data is ubiquitous, but the issue is that it isn't processed. The proposed model uses "BanglaLekha Isolated Dataset" and "EkushNet". The character image of the EkushNet dataset is black, and the backdrop is white. First of all, the character of all images is switched to white, and the backdrop is switched to black (Fig.7), representing that the black pixels are 0, and the white pixels are 255. Both dataset images' size, height, and widthwise are not the same. As a result, the datasets were resized to 60 x 60 pixels (Fig.8).

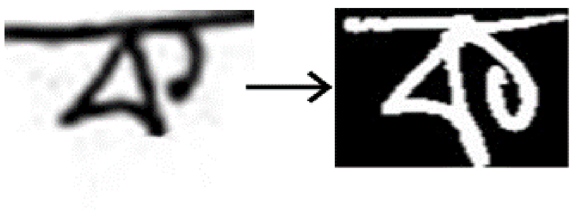


Fig.7. Inverted image

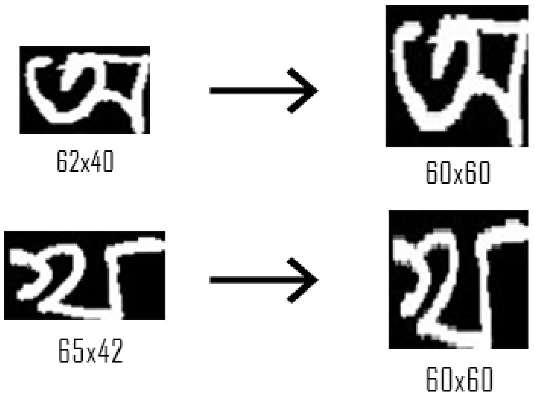


Fig.8. Resize Image

D. Feature & Label Extraction

The entire data set has been feature extracted after resizing. Here no specific algorithm is used for feature extraction. Then, according to feature extraction, find the image’s label. After that, all features and labels are stored in .npy file.

E. CNN Architecture

To classify Bangla handwritten character identification were used the CNN approach with multilayer in this proposed method. Convolution, Max pooling layer, fully connected dense layer, and regularization techniques such as batch normalization [14] and dropout [15] were employed in this model.

A softmax layer is the ultimate output layer.

$$softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}} \qquad (1)$$

The output layer provides a distribution of probabilities across a predetermined set of categories. the network recommends the category which has the most significant probability. The first convolutional layer collects information from the picture directly, allowing them to be utilized in a subsequent discrimination task. this layer searches the picture for receptive fields. The characteristics would be included in these fields. The characteristics are then transmitted to the relu activation mechanism after scanning for receptive fields.

$$Relu(x) = \max(0, x) \qquad (2)$$

a) Architecture for Character

This proposed method firstly used the Convolution 2D layer using a 32 filter size where kernel volume is 3x3 and the layer used the 'relu'() activation with the same padding. Then those layers are connected with a max-pooling layer. After that, for regularization, 20% dropout layers were used. Then the output of the previous layer goes through another Convolution 2D layer using a 64 filter size with a 3x3 kernel size, where the activation is 'relu,' linked to a maximum-pooling layer with a 20% dropout rate. Layer 7, the final convolution 2D layer, takes input from the last layer.

It has 128 filters using 3x3 kernel volume, where activation is 'relu,' linked to a maximum-pooling layer with a 20% dropout rate. Then, after these nine procedures, the result is flattened within an array and routed through a fully connected dense layer of 128 filters, and for regularization, 20% dropout was used. Using the Softmax() activation, the preceding layer's output will be connected in a fully-connected dense layer with 50 nodes, which will be the model's final layer.

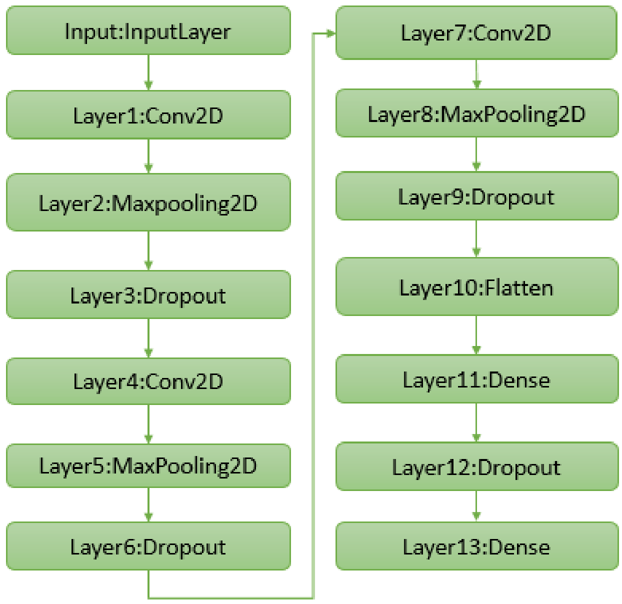


Fig. 9. Architecture of character

b) Architecture for Digit

Initially, this method used the Convolution 2D layer using 16 filter size where kernel volume is 3x3 and the layer used the 'relu'() activation with the same padding. Then those layers are connected with a max-pooling layer. After that, for regularization 20% dropout layers were used. Then the output of the previous layer goes through another Convolution 2D layer using 32 filter size with a 3x3 kernel volume, where the activation is 'relu,' linked to a maximum-pooling layer with a 20% dropout rate. Layer 7, the final convolution 2D layer, takes input from the last layer.

It has 64 filters using 3x3 kernel volume, where activation is 'relu,' linked to a maximum-pooling layer with a 20% dropout rate. Then, after these nine procedures, the result is flattened within an array and routed through a fully connected dense layer of 16 filters, and for regularization, 20% dropout was used. Then the output of the previous layer went in a second fully connected dense layer of 32 filters. Using the Softmax() activation, the preceding layer's output will be connected in a fully-connected dense layer with 10 nodes, which will be the model's final layer.

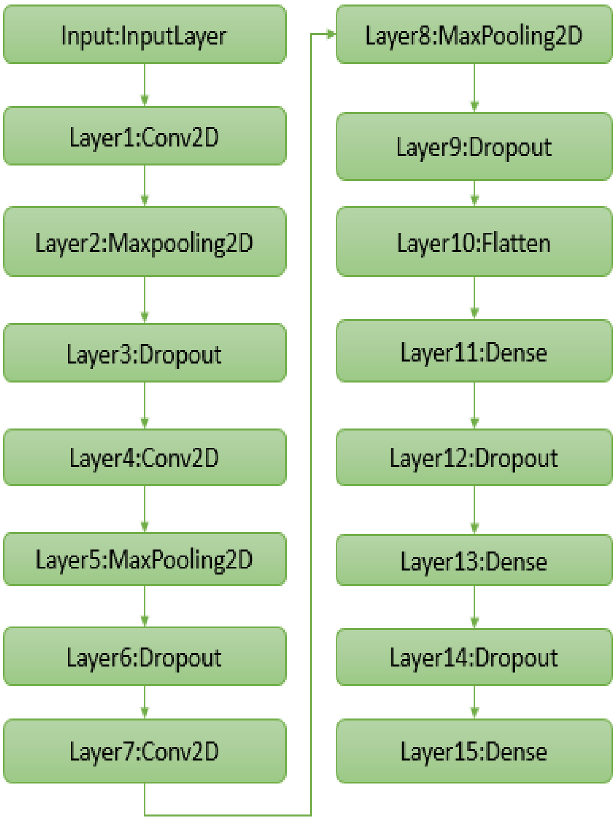


Fig.10. Architecture of digit

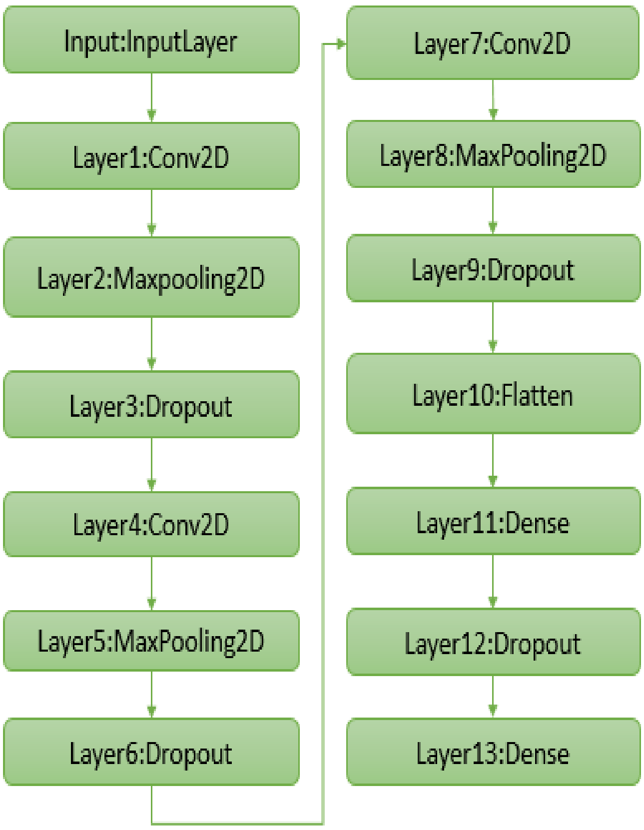


Fig.11. Architecture of modifier

c) Architecture for Modifier

Our proposed method firstly used the Convolution 2D layer using 16 filter size where kernel volume is 3x3 and the layer used the 'relu'() activation with the same padding. Then those layers are connected with a max-pooling layer. After that, for regularization, 20% dropout layers were used. Then the output of the previous layer goes through another Convolution 2D layer using 32 filter size with a 3x3 kernel volume, where the activation is 'relu,' linked to a maximum-pooling layer with a 20% dropout rate. Layer 7, the final convolution 2D layer, takes input from the last layer.

It has filters of 64 using 3x3 kernel volume, where the activation is 'relu,' linked to a maximum-pooling layer with a 20% dropout rate. Then, after these nine procedures, the result is flattened within an array and routed through a fully connected dense layer of 16 filters, and for regularization, 20% dropout was used. Using the Softmax() activation, the preceding layer's output will be connected in a fully-connected dense layer with 10 nodes, which will be the model's final layer.

d) Architecture for Compound Character

This proposed method firstly used the Convolution 2D layer using 16 filter size where kernel volume is 3x3 and the layer used the 'relu'() activation with the same padding. Then those layers are connected with a max-pooling layer. After that, for regularization, 20% dropout layers were used then the output of the previous layer goes through another Convolution 2D layer using 32 filter size with a 3x3 kernel volume, where the activation is 'relu,' linked to a maximum-pooling layer with a 20% dropout rate. Layer 7, the final convolution 2D layer, takes input from the last layer.

It has filters of 64 using 3x3 kernel volume, where the activation is 'relu,' linked to a maximum-pooling layer with a 20% dropout rate. Then, after these nine procedures, the result is flattened within an array and routed through a fully connected dense layer of 64 filters, and for regularization, 20% dropout was used. Using the Softmax() activation, the preceding layer's output will be connected in a fully-connected dense layer with 119 nodes, which will be the model's final layer.

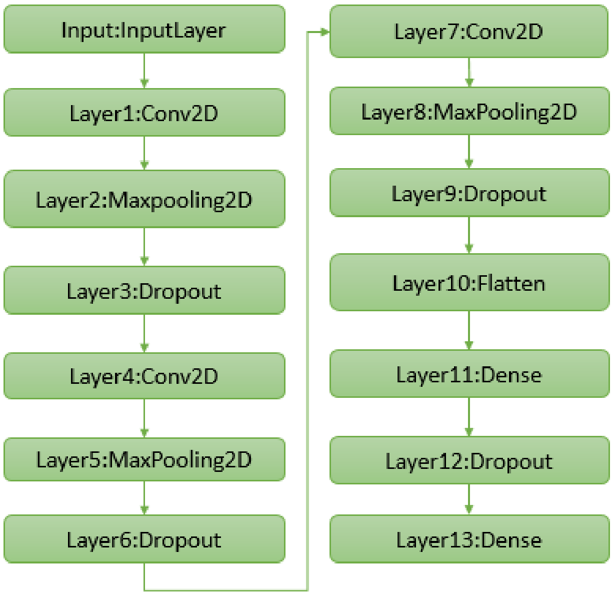


Fig.12. Architecture of compound character

E. *Optimizer*

CNN techniques benefit from optimization algorithms since they help to reduce error. The Adam optimizer [16] was applied in the proposed model. In the training phase, the Adam optimization strategy may be employed to repeatedly adjust network weights. Adam is a more advanced variant of the stochastic gradient descent algorithm. It is commonly utilized in deep learning-based research because of its superior performance.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}}+\varepsilon}\widehat{mt} \tag{3}$$

For implementing the categorical cross-entropy function to determine the error for the optimization procedure. According to the latest studies, cross-entropy outperforms other functions such as mean squared error [17] and classification error.

$$L_i = -\sum_j t_{i,\ j} \log(p_{i,\ j}) \tag{4}$$

IV. RESULTS & DISCUSSION

On the dataset, the proposed model was trained and verified. And provide positive outcomes on a train, test, and validation set.

A. *Test,Train and Validation Split*

Test, train, and validation splits were developed to measure the model's achievement. All features and labels have been linked together. The character dataset has 101368 data images and this dataset is split into four parts each part has contained 25342 characters, this proposed method applied 75% of data for training purposes and 25% of data for test purposes. Same as the character dataset, the digit dataset has 19697 data images. The dataset is split into two parts and each part contains 9848 digits; which means 50% of image data. And applied 50% of the data for the train and 50% for the test. The modifier dataset has 30550 data images. Like other datasets, the dataset is split into two parts and each part contains 15275 modifier images which mean 50% of the data. And here the model applied 50% data for the train and 50% for the test. The compound character dataset has 303888 data images and this dataset is split into eight parts each part contains 37986 characters, in this proposed method applied 265902 images, which means 87.5% for training purposes and 12.5% of data for testing purposes. A summary of these train, test and validation splits is shown in (eg, Table 2).

Table 2. Split the dataset

Type	**Data**	**Train%**	**Test%**
Single characters	101368	75%	25%
Numerals	19697	50%	50%
Modifiers	30550	50%	50%
Compound letters	303888	87.5%	12.5%

B. *Model Performance*

In the single-character model was used 40 epochs. After that, got a training result of 95.71% and the test result of 84.62%. Then for numerals, 50 epochs were used. The training result of 92.77% and the test result of 94.00%. After 40 epochs modifiers got the training result of 95.46% and the test result of 96.46%. Finally, for the compound letters, epoch 30 was used and also got the training result of 74.98% and the test result of 77.60%.

Table 3. Train accuracy of models

TYPE	**CLASS**	**TRAIN ACCURACY**	**TRAIN LOSS**
Single characters	50	95.71%	0.1251
Numerals	10	92.77%	0.2207
Modifiers	10	95.46%	0.1299
Compound letters	119	74.98%	0.8208

In above Table 3 shows the training loss, train accuracy, and how many class are used for single characters, modifiers, numerals, and compound letters.

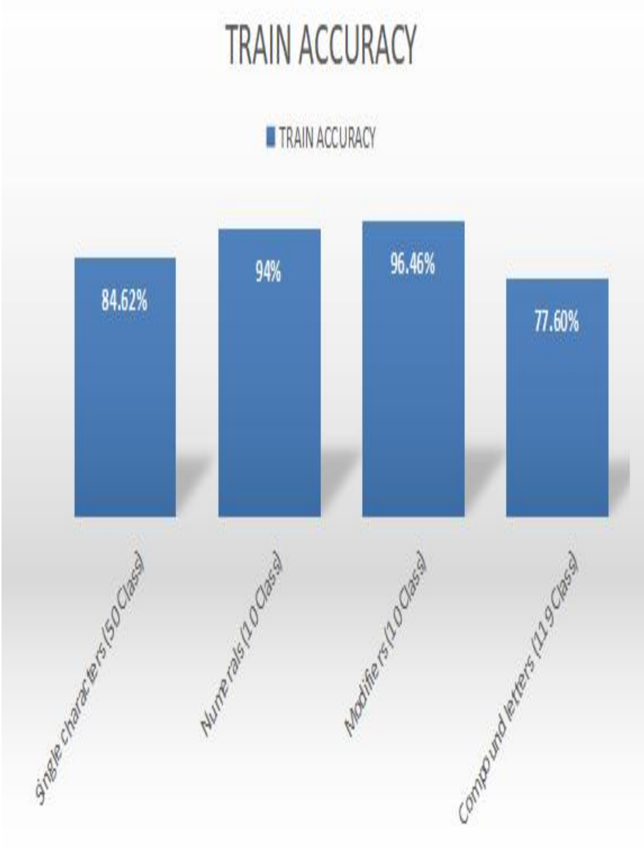


Fig.13. Performance analysis on Train Accuracy

Above Fig.13 represent the training accuracy and how many classes are used for single characters, modifiers, numerals, and compound letters.

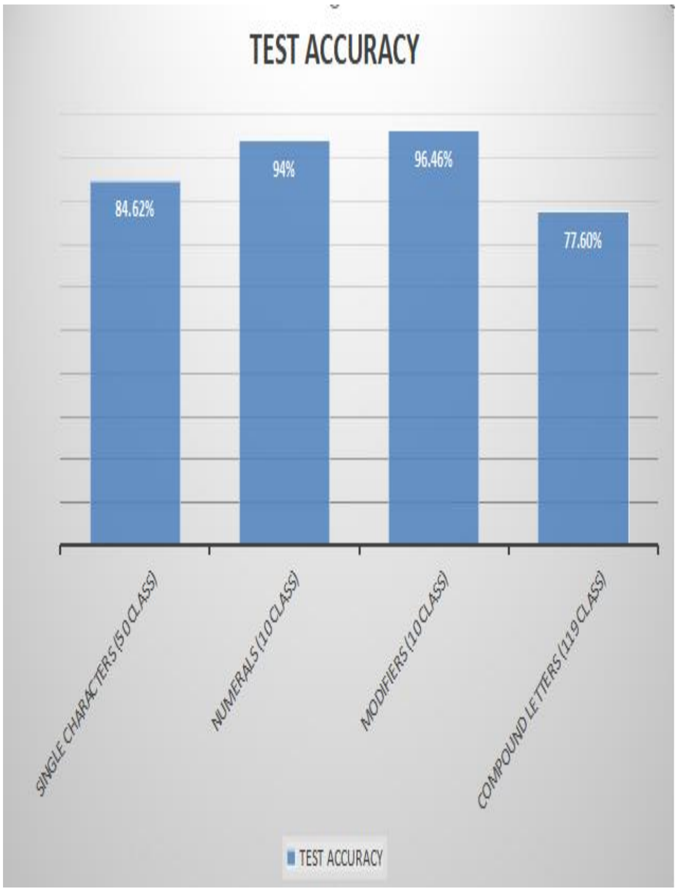


Fig.14. Performance analysis on Test Accuracy

Fig.14 represent the test accuracy and how many classes are used for single characters, modifiers, numerals, and compound letters.

Table 4. Test accuracy of models

TYPE	CLASS	TEST ACCURACY	TEST LOSS
Single characters	50	84.62%	0.7554
Numerals	10	94%	0.2094
Modifiers	10	96.46%	0.1482
Compound letters	119	77.60%	0.7967

Table 4 shows the test loss, test accuracy, and how many class are used for single characters, modifiers, numerals, and compound letters.

Table 5. Validation accuracy of models

TYPE	CLASS	VALIDATION ACCURACY	VALIDATION LOSS
Single characters	50	85.34%	0.7248
Numerals	10	96.24%	0.1355
Modifiers	10	96.27%	0.1406
Compound letters	119	77.55%	0.7825

Table 5 shows the validation loss, validation accuracy, and how many class are used for single characters, modifiers, numerals, and compound letters.

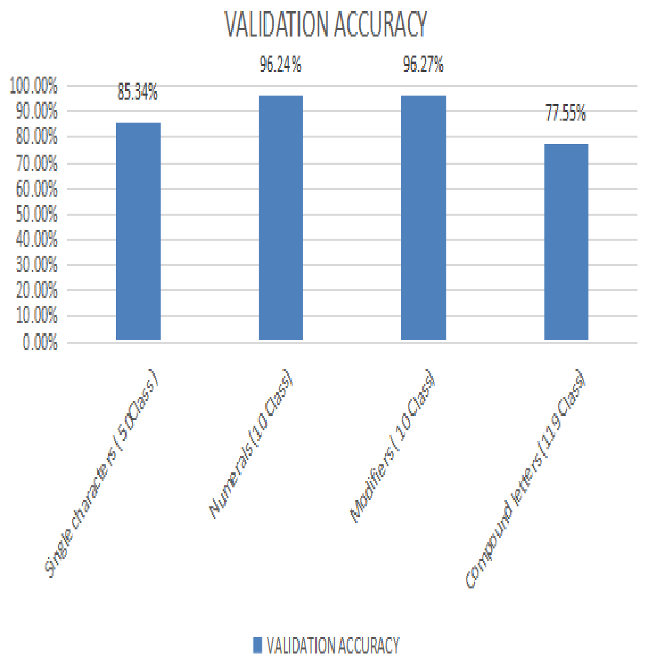


Fig.15. Performance analysis Validation accuracy

Fig.15 represent the validation accuracy and how many classes are used for single characters, modifiers, numerals, and compound letters.

V. Conclusion & Future Work

This research aims to see how well CNN can classify Bangla handwritten characters, compound letters, digits, and up/lower signs. A convolutional neural network (CNN) can identify optical patterns directly away from pixel images with minimum preparation. As a result, the paper examines a CNN structure for handwritten pattern classification with only the raw feature. It also showed that utilizing a larger quantity of data with variance might aid the model's learning of the classes' attributes or qualities. The accuracy of the proposed work is excellent because it is done with the created model. In the future, utilize more deep layers or pre-trained models.

VI. Acknowledgements

Would like to express gratitude to Daffodil International University's Department of Computer Science and Engineering. Also want to express gratitude to all of the supervisors for their assistance.

VII. Reference

[1] A. K. M. Rabby, S. Abujar, S. Haque, and S. A. Hossain, "Bangla handwritten digit recognition using convolutional neural network," In *Emerging Technologies in Data Mining and Information Security*, pp. 111-122, Springer, Singapore, 2019.

[2] A. K. M. Rabby, S. Abujar, S. Haque, and S. A. Hossain, "Bornonet: Bangla handwritten characters recognition using convolutional neural network," *Procedia computer science*, *143*, pp.528-535,2018.

[3] A. K. M. Rabby, S. Abujar, S. Haque, and S. A. Hossain, "Ekushnet: using convolutional neural network for bangla handwritten recognition." *Procedia computer science, 143*, pp. 603-610, 2018.

[4] N. Das, B. Das, R. Sarkar, S. Basu, M. Kundu and M. Nasipuri, "Handwritten Bangla basic and compound character recognition using MLP and SVM classifier," *arXiv preprint arXiv:1002.4040*, 2010.

[5] M. M. Rahman, M. A. H. Akhand, S. Islam, P. C. Shill and M. M. Hafizur Rahman, "Bangla handwritten character recognition using convolutional neural network," *International Journal of Image, Graphics and Signal Processing*, *7*(8), 42, 2015, doi: 10.5815/ijigsp.2015.08.05.

[6] R. R. Chowdhury, M. S. Hossain, R. ul Islam, K. Andersson, & S. Hossain, "Bangla handwritten character recognition using convolutional neural network with data augmentation," In *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, pp. 318-323, IEEE, May 2019.

[7] B. Purkaystha, T. Datta & M. S. Islam, "Bengali handwritten character recognition using deep convolutional neural network," In *2017 20th International conference of computer and information technology (ICCIT)*, pp. 1-5, IEEE, December 2017.

[8] C. Saha, R. H. Faisal & M. M. Rahman, "Bangla handwritten basic character recognition using deep convolutional neural network," In *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, pp. 190-195, IEEE, May 2019.

[9] S. A. Hakim, "Handwritten bangla numeral and basic character recognition using deep convolutional neural network," In *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pp. 1-6, IEEE, February 2019.

[10] T. R. Das, S. Hasan, M. R. Jani, F. Tabassum & M. I. Islam, "Bangla Handwritten Character Recognition Using Extended Convolutional Neural Network," *Journal of Computer and Communications*, *9*(3), 158-171, 2021.

[11] M. J. Hasan, M. F. Wahid & M. S. Alom, "Bangla compound character recognition by combining deep convolutional neural network with bidirectional long short-term memory," In *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*, pp. 1-4, IEEE, December 2019.

[12] A. S. A. Rabby, S. Haque, M. S. Islam, S. Abujar & S. A. Hossain, "Ekush: A multipurpose and multitype comprehensive database for online off-line bangla handwritten characters," In *International Conference on Recent Trends in Image Processing and Pattern Recognition*, pp. 149-158, Springer, Singapore, December 2018.

[13] M. Biswas, R. Islam, G. K. Shom, M. Shopon, N. Mohammed, S. Momen & A. Abedin, "Banglalekha-isolated: A multi-purpose comprehensive dataset of handwritten bangla isolated characters," Data in brief, 12, 103-107, 2017.

[14] S. Ioffe & C. S. B. Normalization,"Accelerating Deep Network Training by Reducing Internal Covariate Shift," *arXiv preprint arXiv:1502.03167*.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever & R. Salakhutdinov,"Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, *15*(1), 1929-1958, 2014.

[16] D. P. Kingma & J. Ba," Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[17] K. Janocha & W. M. Czarnecki," On loss functions for deep neural networks in classification," *arXiv preprint arXiv:1702.05659*, 2017.

[18] P. Chakraborty, A. Islam, M. A. Yousuf, R. Agarwal & T. Choudhury, " Bangla Handwritten Character Recognition Using Convolutional Neural Network," In *Machine Intelligence and Data Science Applications*, pp. 721-731. Springer, Singapore, 2022.

[19] R. Jadhav, S. Gadge, K. Kharde, S. Bhere & I. Dokare, " Recognition of Handwritten Bengali Characters using Low Cost Convolutional Neural Network," In *2022 Interdisciplinary Research in Technology and Management (IRTM)*, pp. 1-6. IEEE, 2022.

[20] S. Ahsan, S. T. Nawaz, T. B. Sarwar, M. S. U. Miah, A. Bhowmik, "A machine learning approach for Bengali handwritten vowel character recognition," *Int J Artif Intell* 11, pp. 1143-1152, 2022.