Performance Analysis of State of the Art Convolutional Neural Network Architectures in Bangla Handwritten Character Recognition

Article *in* Pattern Recognition and Image Analysis · January 2021

DOI: 10.1134/S1054661821010089

CITATIONS
16

READS
313

5 authors, including:

Tapotosh Ghosh
United International University
19 PUBLICATIONS **310** CITATIONS
SEE PROFILE

Md. Min-ha-zul Abedin
Bangladesh University of Professionals
5 PUBLICATIONS **52** CITATIONS
SEE PROFILE

Md Hasan Al Banna
Bangladesh University of Professionals
15 PUBLICATIONS **289** CITATIONS
SEE PROFILE

Nasirul Mumenin
Bangladesh University of Professionals
4 PUBLICATIONS **16** CITATIONS
SEE PROFILE

Performance Analysis of State of the Art Convolutional Neural Network Architectures in Bangla Handwritten Character Recognition

Tapotosh Ghosh[a],*, Min-Ha-Zul Abedin[b],, Hasan Al Banna[a],***, Nasirul Mumenin[a],****, and Mohammad Abu Yousuf[c],*****

[a] *Department of Information and Communication Technology, Bangladesh University of Professionals, Dhaka, 1216 Bangladesh*

[b] *Department of Information and Communication Engineering, Bangladesh Army University of Engineering and Technology, Qadirabad Cantonment, Natore, 6431 Bangladesh*

[c] *Institute of Information Technology, Jahangirnagar University, Savar, Dhaka, 1342 Bangladesh*

** e-mail: 16511038@student.bup.edu.bd*

*** e-mail: 16511024@student.bup.edu.bd*

**** e-mail: alifhasan39@gmail.com*

***** e-mail: nmmouno@gmail.com*

****** e-mail: yousuf@juniv.edu*

Abstract—Bangla handwritten character recognition is a popular research topic as its difficulty is higher than the recognition of other languages because of multiple formats of compound characters. State of the art Convolutional neural network (CNN) architectures are very much useful in computer vision applications. Some works have been carried out in Bangla handwritten character recognition but most of them either not very efficient or they can not classify a lot of characters. In this work, state of art pre-trained CNN architectures is used to classify 231 different Bangla handwritten characters using CMATERdb dataset. The images were first converted to B&W form with white as the foreground color. The size of the images is reduced to 28 × 28 form. These images are used as input to the CNN architectures. The weights of the state-of-the-art CNN models are kept as it was. The training learning rate was set to 0.001 and categorical cross-entropy as the error function. After 50 epochs, InceptionResNetV2 achieved the best accuracy (96.99%). DenseNet121 and InceptionNetV3 also provided remarkable recognition accuracy (96.55 and 96.20%, respectively). We also considered combination of trained InceptionResNetV2, InceptionNetV3 and DenseNet121 architectures which provided better recognition accuracy (97.69%) than other single CNN architectures but it is not feasible for using as it requires a lot of computation power and memory. The models were tested in the cases where characters look confusing to humans, but all the architectures showed equal capability in recognizing these images. Considering computational complexity, memory and capability of recognizing confused characters, InceptionResNetV2 can be said as the best performing model.

Keywords: convolutional neural network, Bangla character, classification

DOI: 10.1134/S1054661821010089

1. INTRODUCTION

Bangla is used as a native language by 228 million people which is the seventh most spoken language but lacks a well-performing optical character recognition (OCR) system because of underperforming compound character recognition methods. Bangla character recognition is a difficult job as it has a wide variety of compound characters which are written in different forms. A complete handwritten character recognition system should recognize the basic form of characters, their compound structures, and numeric characters written in any form. This kind of recognition system can be used in handwritten manuscript conversion to digital format, postal automation, shopping list generation from the handwritten list, and many other sectors.

Recognition of handwritten characters is not a very young research field since much successful research has been done on different other languages like English, French, Japanese, and Arabic. These well-established models could easily be employed in Bangla if the variety of compound characters were not present. There are 50 simple characters, 10 numeric characters, and over 400 composite characters in Bangla

Received June 18, 2020; revised October 16, 2020; accepted October 20, 2020

Language. If this character recognition problem is considered as a classification problem, then there will be more than 460 output classes. For dealing with this kind of problem artificial neural network (ANN) is a common choice. In general, ANN systems are made up of an input layer, several hidden layers, and an output layer. The layers have connected neurons with weights attached to them. An error function computes the residue between the output value and the actual value. Based on that the weights are updated till an optimized position is reached. This search of optimized weights can stick to local minima. Normally, for handwritten character recognition, the image data is considered as input to a recognition system which is a combination of pixels. From these pixels, it is quite difficult to calculate handcrafted features. Convolutional neural networks (CNN) can compute thousands of features from images without human intervention. CNN's have many hidden layers that use methods like depth wise convolution to calculate these complex features. Since CNN performs a convolution of many layers the training procedure is very long.

To overcome the problem of long training times pre-trained models can be very helpful. There are some state of art pre-trained CNN structures like-InceptionV3 [1], ResNet50 [2], ResNet50V2 [3], DenseNet121 [4], VGG16 [5], MobileNetV2 [6], IncetionResnetV2 [7], EfficientNetB3 [8], NasNet [9], etc. These models are trained with millions of images of different objects which are used for object recognition systems. These pre-trained models can also be used in handwritten character recognition problems. Since these models are built for object recognition systems the weights of the networks should be kept unchanged but while training the whole model should be trained with the Bangla character images. This procedure can reduce the time of training as the model is in near optimized form. Moreover, these models can produce high output accuracy. So, this method should be explored.

In this research, we tried to implement the state of the art CNN architecture in Bangla handwritten character recognition and evaluate their performance in this field. We have selected 231 different Bangla characters (10 numerals, 50 basic, 171 compound) classes from CMATERdb as these characters are mostly used in Bangla words and rest of them are either obsolete or not that much used in Bangla language [10]. We have also tested combination of the trained CNN models in this research. The rest of the paper is organized in the following sections: the problem statement and contribution section will illustrate the research question and our contributions. In the related works section, the works in this field are discussed. A brief discussion on the state of art CNN architectures is described in the following section. Methodology and performance analysis sections illustrate the proposed methodologies and performance analysis respectively. The conclusion section concludes the paper with some discussion of future scopes in this field.

2. RELATED WORKS

Bangla Handwritten Character Recognition (BHCR) has gained quite a lot of movement in recent times. OCR is a very necessary tool in different applications, such as NLP, automatic character interpretation, automated text entry, etc. BHCR research was confined only to the numeral and basic characters before the year 2010. The interest of research with compound characters came after 2015. It needs more attention to establish a reliable recognition system.

Das et al. [11] suggested a process of classifying 93 characters using tree feature set, MLP and SVM classifier. Sarkhel et al. [12] classified 384 characters with 72.87% accuracy by using region sampling and SVM. Pramanik et al. [13] classified 171 compound characters using shape decomposition where basic characters were found out from the compound characters and classification task was carried out using MLP. This method achieved 88.74% precision in compound character classification. A convex hull-based method was adopted by Das et al. [14] to distinguish between 10 numeric and 50 basic characters. An updated feature set containing 132 different features was suggested by Das et al. [15] to enhance recognition efficiency in recognizing basic characters. Basu et al. [16] took word segmentation approach where they suggested a different feature descriptor. Bhowmik et al. [17] used a combination of three different methods. SVM, RBF, and MLP based method was used for classifying 45 basic character classes that performed better than SVM. In this method, they at first recognized a character group and then classified the actual class label. Parui et al. [18] suggested a hidden Markov model (HMM) and implemented a stroke-based approach where they identified 54 stroke groups, generated 6 stroke groups and each stroke was assigned with a different HMM. Roy [19] suggested a handwritten character-based stroke database and proposed an architecture to build Bangla characters from strokes. Sazal et al. [20] developed a deep belief approach to recognize 10 numerals and 50 basic Bangla characters where supervised methods were implemented for fine tuning of images and classification was done by unsu-

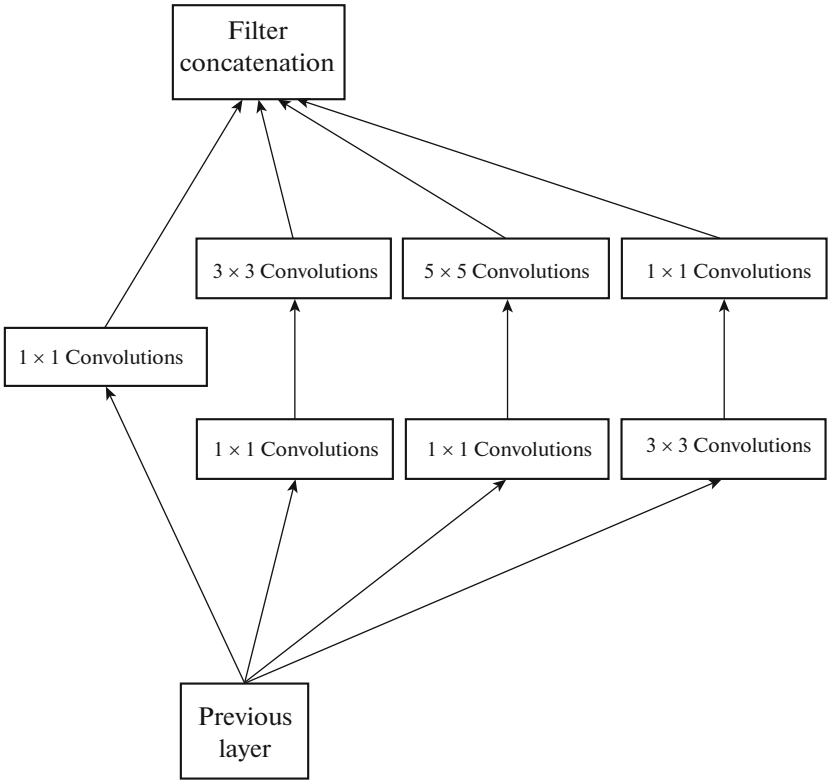

Fig. 1. Inception block.

pervised learning approach. RMSprop optimizer and layer-wise training was adopted in a neural network to classify 173 Bangla characters by Roy et al. [21] where faster convergence was obtained by RMSprop.

Ashiquzzaman et al. [22] developed a CNN model where dropout and ELU filter was merged to get rid of overfitting and gradient vanishing problem to classify 171 classes. Fardous et al. [23] developed a CNN model that had 4 pooling layers, 8 convolutional layers, and 2 dense layers to classify 171-character classes. ReLU was used as the activation function as it can introduce non-linearity. In this case, dropout was used to reduce overfitting. Saha et al. [24] classified 84 character classes by following GOOGLENET where they added various number of filters in the layers of the proposed architecture. Rabby et al. [25] classified 122 classes by introducing a 22 layered CNN architecture where they got a significant recognition accuracy. Alif et al. [26] utilized ResNet in 173-character classification where dropouts were added and optimizer was Ghosh et al. [27] implemented MobileNetV1 architecture to classify a large number of character classes.

3. STATE OF THE ART CNN ARCHITECTURES

3.1. InceptionV3

InceptionV3 came as an extension and upgraded version of InceptionV2 and InceptionV1. The inception model is a micro-architecture module that works as a multi-level feature extractor. Within the same module of the network, it computes convolutions 1×1, 3×3, 5×5. Then before being passed to the next layer, the results found from the filters are piled along the dimension of the channel. InceptionV3 is lighter than ResNet and VGG. It upgrades the recognition efficiency of ImageNet by adopting some updates in inception block. The default input size for this model is 299×299. Figure 1 illustrates the inception block.

3.2. ResNet50 and ResNet50V2

ResNet stands for Deep Residual Network. ResNet provides the solution to find the right number of hidden layers that have to be used for a deep neural network. It is based on skipping connections between layers or shortcuts that jumps over several layers. The goal of these architectures to reduce vanishing or exploding

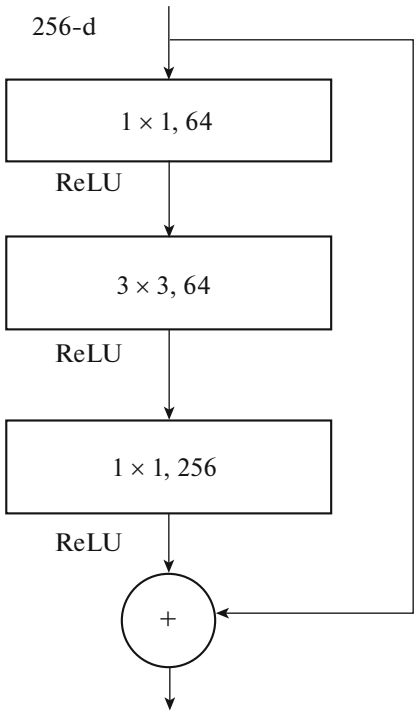

Fig. 2. Residual block.

gradient problem. It solves the problem by reusing the activations from the layer before it. It continues till the weight is learned by an adjacent layer. The default input size for this model is 224 × 224. It was the winner in ILSVRC 2015. Figure 2 illustrates the residual block.

3.3. DenseNet121

DenseNet stands for Dense Convolutional Network which is a logical extension of ResNet. ResNet is the fundamental building block in which additive layers are merged with a future layer. DenseNet proposes concatenating outputs from the previous layers instead of using the summation. The default input size for this model is 224 × 224. DenseNet possesses some pros such as no vanishing gradient problem, encourage feature reuse, strengthened feature propagation, reduced number of parameters. Basic architecture of DenseNet is illustrated in Fig. 3.

3.4. VGG16 and VGG19

VGGNet was the runner up of ILSVRC 2014. It was build using only 3 × 3 convolution and these layers were stacked. It is very much recommended for feature extraction from images. VGG19 has 3 more convolutional layers than VGG16. The input tensor of VGGNet is 224 × 224 × 3.

3.5. MobileNetV2

MobileNetV2 is an extension and improvement over MobileNetV1 which uses depthwise convolution. The default input size for this model is 224 × 224. Linear bottleneck was introduced between the layers. The bottlenecks contained shortcut connection among them as they introduced two features for encoding intermediate inputs and outputs. Higher level descriptors were achieved from the lower level concepts through the model. It utilizes basic residual connection blocks to achieve better accuracy and faster training capability. Figure 4 depicts the basic architecture of MobileNetV2.

3.6. IncetionResNetV2

InceptionResNetV2 is an updated version of the InceptionV3 which took some idea from ResNet. It consists of 164 layers and trained on the ImageNet dataset. It is more capable of acquiring better result than other CNN architectures. Residual connections allow shortcuts in the model which has led this architecture to gain even better performance. Significant simplification of the Inception blocks has also been enabled by it. This architecture is a combination of inception and residual block which enhances the performance. The required input tensor of this architecture is 299 × 299 × 3.

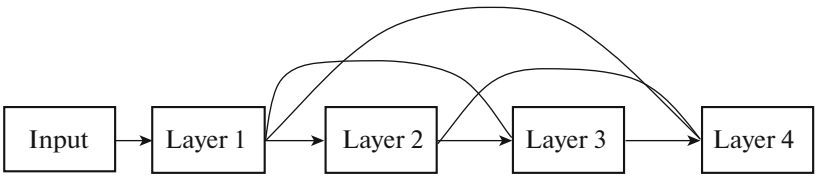

Fig. 3. DenseNet architecture.

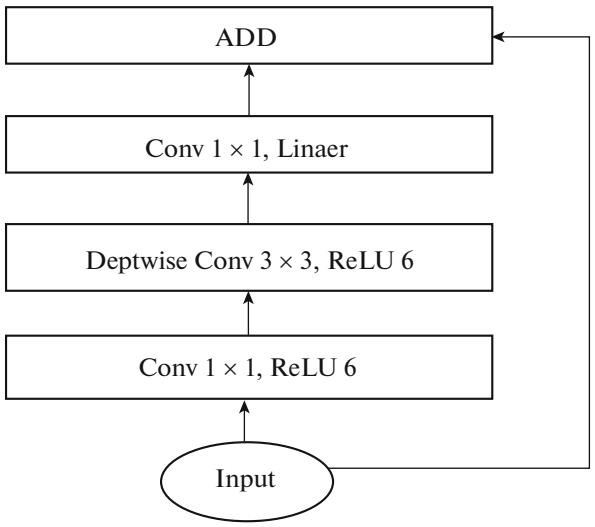

Fig. 4. MobileNetV2 basic building block.

3.7. EfficientNetB3

EfficientNet is one of the newest CNN architecture which was released by Google. It uses a simple and highly efficient procedure where a fixed set of coefficients is implemented to scale the dimensions in a structured manner. This model performed better than others CNN models and provided a very good accuracy in ImageNet.

3.8. NasNet

NAS is mainly an architecture for making automated ANN architecture and is related to automated machine learning. NAS designs ANN networks that perform similar or better than hand-design models. It is related to hyperparameter optimization. It is capable of achieving good accuracy with smaller model size and lower complexity (FLOPs). The default input size for the NASNetMobile model is 224 × 224. Figure 5 illustrates the NASNet architecture.

Table 1 provides a list of parameters, input tensor, and size of the model of the above-described models.

4. METHODOLOGY

The whole research work followed the flowchart illustrated in Fig. 6. At first, a dataset containing 231 Bangla handwritten character classes was collected and images were preprocessed. 20% of the total images were isolated and used for testing purposes. Then the state of art CNN architecture models was trained using the training portion of the dataset and tested using the separated testing portion.

4.1. Dataset Preparation

CMATERdb is one of the largest and most used datasets of Bangla handwritten character image [28]. We merged the CMATERdb compound character, CMATERdb basic character, and CMATERdb numeric character databases and made a final database consisting of 231 classes. The images were converted into B&W where white was used as a foreground color and resized to 28 × 28 pixels. Figure 7 shows the result of the preprocessing task which was carried out in this research. Table 2 shows the number of images

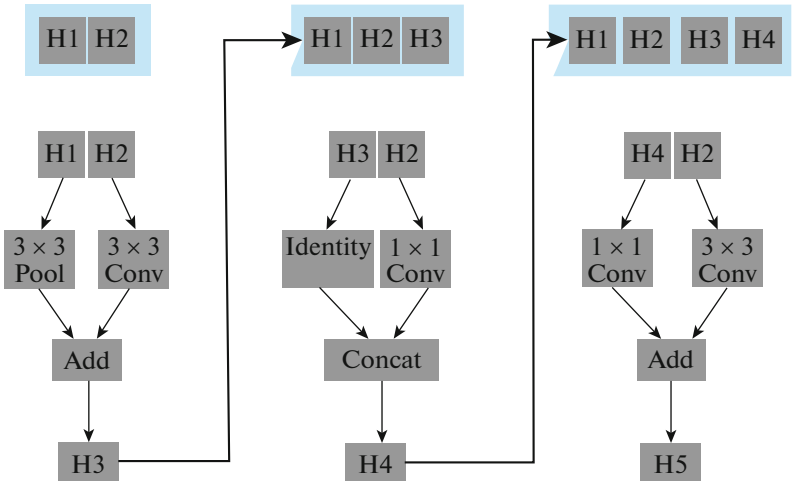

Fig. 5. NasNet search space.

Table 1. Comparison of different state of art CNN architecture

Model	Parameters	Size, Mb	Input tensor
InceptionResNetV2	55873736	215	224 × 224 × 3
DenseNet121	8062504	33	224 × 224 × 3
InceptionV3	23851784	92	224 × 224 × 3
NASNet	5326716	23	224 × 224 × 3
MobileNetV2	3538984	14	224 × 224 × 3
ResNet50	25636712	98	224 × 224 × 3
ResNet50V2	25613800	98	224 × 224 × 3
EfficientNetB3	11138575	131	224 × 224 × 3
VGG19	143667240	549	224 × 224 × 3
VGG16	138357544	528	224 × 224 × 3

Table 2. Dataset splitting

Dataset	No. of classes	Training images	Validate images	Test images
Compound	171	27486	6793	8123
Basic	50	9411	2345	2896
Numeral	10	1910	474	484
Total	**231**	**38807**	**9612**	**11503**

Table 3. Time required to train (s/epoch)

Model	Time required to train (s/epoch)
EfficientNetB3	284
VGG16	329
MobileNetV2	363
VGG19	365
ResNet50V2	372
ResNet50	469
InceptionV3	592
DenseNet121	620
NASNet	674
InceptionResNetV2	1188

Table 4. Performance obtained by diferent CNN models

Model	Accuracy, %	Avg. recall	Avg. precision
InceptionResNetV2	96.99	0.97	0.97
DenseNet121	96.55	0.96	0.97
InceptionV3	96.20	0.96	0.96
NASNet	95.85	0.96	0.96
MobileNetV2	95.56	0.96	0.96
ResNet50	94.91	0.95	0.95
ResNet50V2	93.80	0.94	0.94
EfficientNetB3	92.80	0.93	0.93
VGG19	92.07	0.92	0.92
VGG16	90.70	0.90	0.91

and splitting of the dataset into training, testing, and validation. The training, testing, and validation set was fixed in this research.

4.2. Training

InceptionV3, ResNet50, ResNet50V2, DenseNet121, VGG16, VGG19, MobileNetV2, IncetionResnetV2, EfficientNetB3, NasNet models were trained with the training portion of the dataset and no of neurons of the final dense layers were set to 231. We set the following parameters for all models: loss = categorical crossentropy, epochs = 50, and learning rate = 0.001.

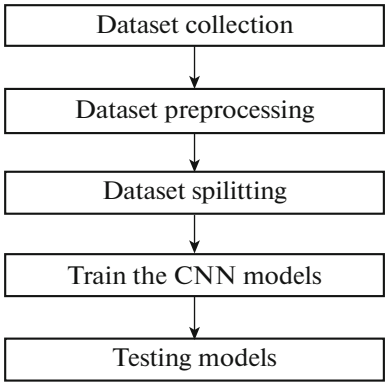

Fig. 6. Workflow followed in this research.

InceptionV3, ResNet50, ResNet50V2, DenseNet121, MobileNetV2, IncetionResnetV2, EfficientNetB3, NasNet models were trained using Adam optimizer and VGG16, VGG19 were trained using SGD optimizer. The whole work was conducted using Kaggle kernel. EffecientNetB3 took the lowest amount of time to complete an epoch where InceptionResnetV2 took the highest amount of time. Table 3 shows the required time to complete an epoch of the above-mentioned models.

4.3. Testing

All the models were further tested with a separated testing set of the dataset. It was found that InceptionResNetV2 obtained the highest accuracy (96.99%) and VGG16 model obtained the lowest accuracy (90.70%). Accuracy, avg. precision, and avg. recall obtained by the state of art architectures are stated in Table 4.

(a) (b)

Fig. 7. (a) Before and (b) after preprocessing.

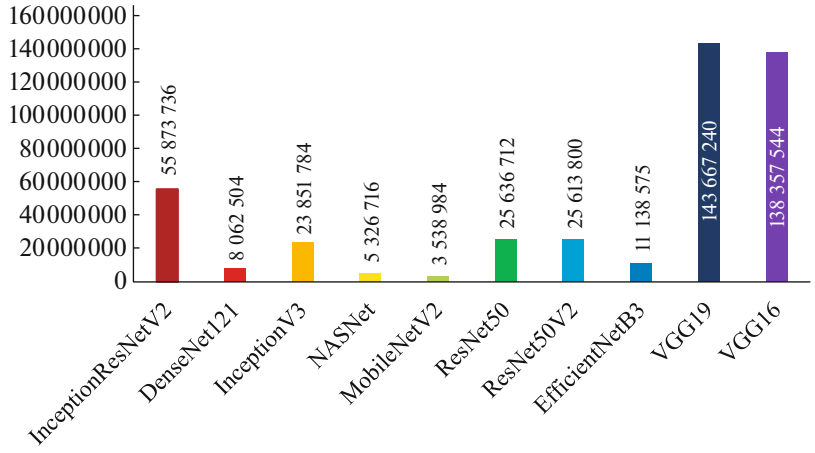

Fig. 8. Comparison of state of art CNN architectures in terms of number of parameters.

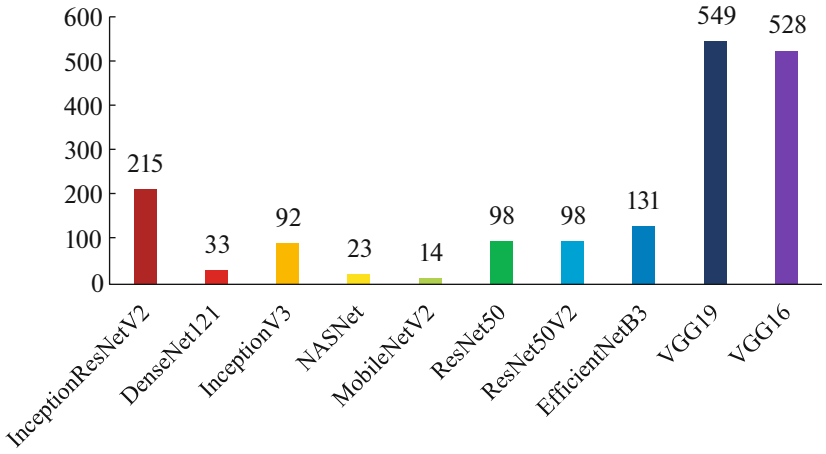

Fig. 9. Comparison of state of art CNN architectures in terms of model size (Mb).

5. PERFORMANCE ANALYSIS

InceptionV3, ResNet50, ResNet50V2, DenseNet121, VGG16, VGG19, MobileNetV2, IncetionResNetV2, EfficientNetB3, NasNet models were trained and tested for Bangla Handwritten Character Recognition. Among these models, VGG19 and VGG16 have the largest number of parameters and sizes where NASNet, MobileNetV2, and DenseNetV2 contain a low number of parameters and size. Figures 8 and 9 show the comparison of the state of art CNN architectures in terms of the number of parameters and size of the model. Among these models, InceptionResNetV2, InceptionV3, and DenseNet121 took the highest amount of time to test where EfficientNetB3 and MobileNetV2 took less time. Though InceptionV3, DenseNet121, and InceptionResNetV2 contain a low number of parameters than the VGG models, they took more time to test. A comparison between these models is shown in Fig. 10 which is based on the time required to test the testing set.

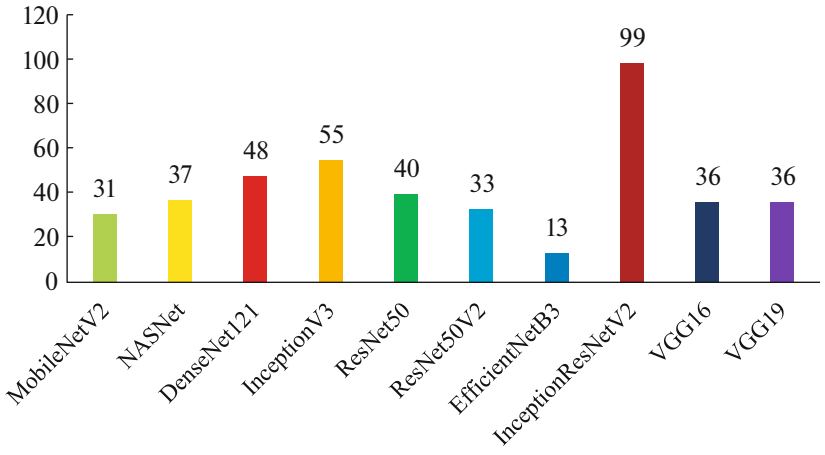

Fig. 10. Comparison of state of art CNN architectures in terms of required testing time (s).

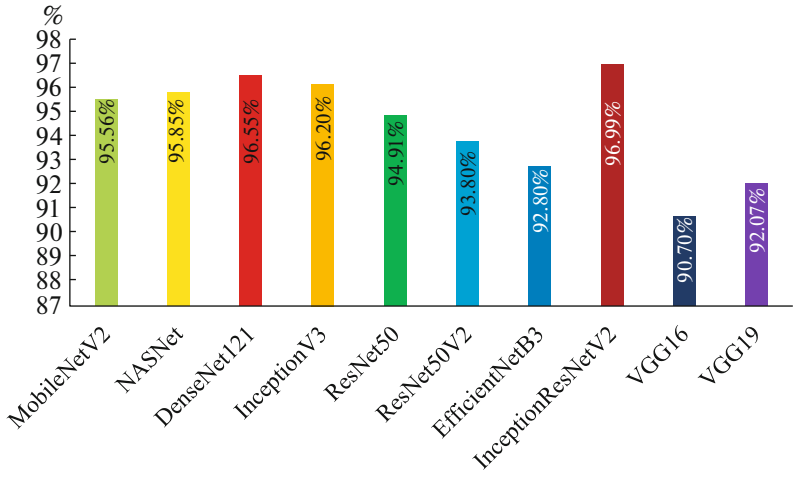

Fig. 11. Comparison of state of art CNN architectures in terms of acquired accuracy.

Among these models, InceptionResNetV2 outperformed all the other models by acquiring 96.99% accuracy in the 231 classes classification. InceptionResNetV2 is a very bulky model and requires a larger time to train and test. However, DenseNet121 which is comparatively small-sized and required less time to train than most of the other models achieved 96.55% accuracy which is the second highest.

InceptionResNetV2 required the largest time to test. DenseNet 121 needed more time to test than most other models. ResNets and VGGs did not perform well in this classification task. Other low sized models such as NASNet and MobileNet which are suitable for mobile devices could not achieve remarkable accuracy.

EffecientNetB2 which is one of the latest state of the art CNN architecture could not classify Bangla handwritten characters well compared to other architectures, if we consider model size, performance and required time to test and train, DenseNet121 performed the best but just in case of performance InceptionResNetV2 proved its efficiency in 231 Bangla handwritten character classification. The performance comparison of these models according to accuracy is illustrated in Fig. 11.

We have conducted another experiment where we used the trained InceptionResNetV2, DenseNet121, and InceptionNetV3 models. Figure 12 shows the pipeline of the experiment. We at first preprocessed the testing set according to input shape requirements of the models. Then this samples were tested using these models and these models predict the samples individually. If any of the 2 model predict same class, the sample is labelled as the predicted class by the 2 models. In case of predicting 3 different class by 3 classifiers, the sample was labelled as per the predicted class of InceptionResNetV2 as it provided the

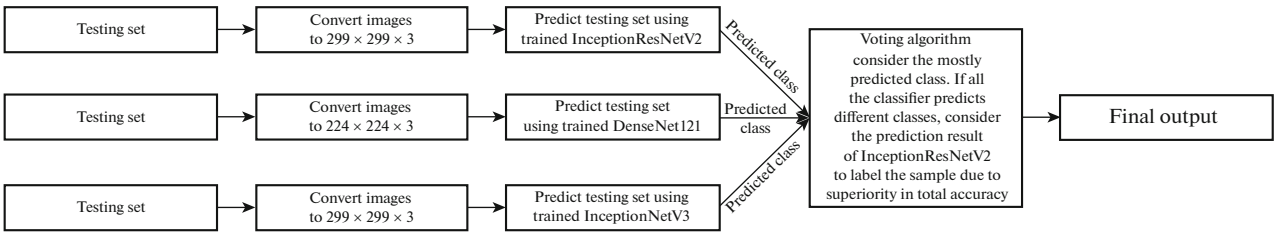

Fig. 12. Experimental process of combined state of the art CNN architecture.

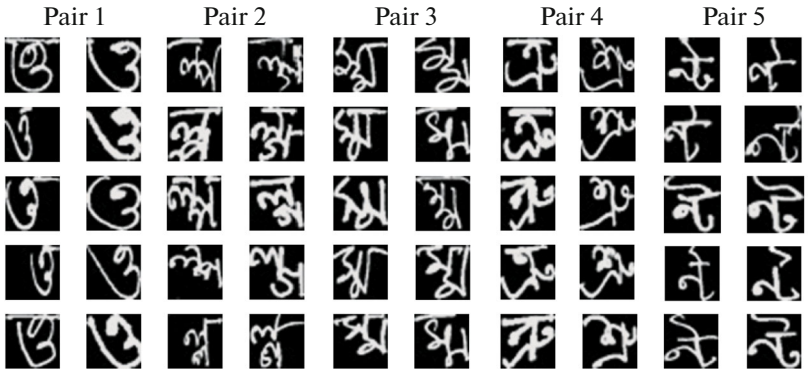

Fig. 13. Selected samples of experiment similar character pairs.

best performing accuracy. This combination of architecture method correctly classified 11238 out of 11503 character images and achieved 97.69% accuracy. But the size of the whole architecture is around 340 Mb and required computational cost is higher than single CNN classifiers. So, it will not be a feasible solution for such devices that does not have GPU and requires lighter models to run. However, if precision is considered above all the other parameters, this combined architecture can be used though it requires 2 times higher time and memory than single CNN classifiers.

Handwritten characters are very hard to understand sometimes even by the human. There are some pairs of character classes that very much alike in shape with each other. We have picked 5 of this kind of pairs. We have taken 5 character images of each classes that were predicted by the InceptionResNetV2, DenseNet121, and the combined architecture. Figure 13 shows the selected samples that were used in this experiment. From Fig. 13, it can be seen that images are very confusing. However, InceptionResNetV2, DenseNet121 and the combined architecture correctly classified all the character samples of pairs 1–4. However, they all misclassified 3 out of 10 samples of pair 5 classes. So, they correctly classified 47 out of 50 samples and achieved 94% accuracy in the confusing cases.

Some of the works have been already carried out for BHCR using the CMATERdb database. InceptionResNetv2, InceptionNetV3, DenseNet121, and combination of trained CNN architectures performed better than all the existing models. Figure 14 provides a performance comparison of the works that were carried out using the CMATERdb dataset which prove that state of art architectures outperformed all the existing research works that used CMATERdb in terms of accuracy and number of classes classified.

We have also compared the performance of the state of the art CNN architecture with the research works that takes different approaches for detecting handwritten Bangla characters. In these researches, researchers used several methods such as: shape decomposition, different feature extractor, hidden Markov model, convex hull, region sampling etc. but could not achieve accuracy greater than 91%. Number of classified character classes is also not also very high. From Table 5, it can be clearly seen that proposed CNN classifiers performed better than the other proposed architectures.

CONCLUSIONS

Because of the vast number of compound characters and their complicated structure, Bangla handwritten

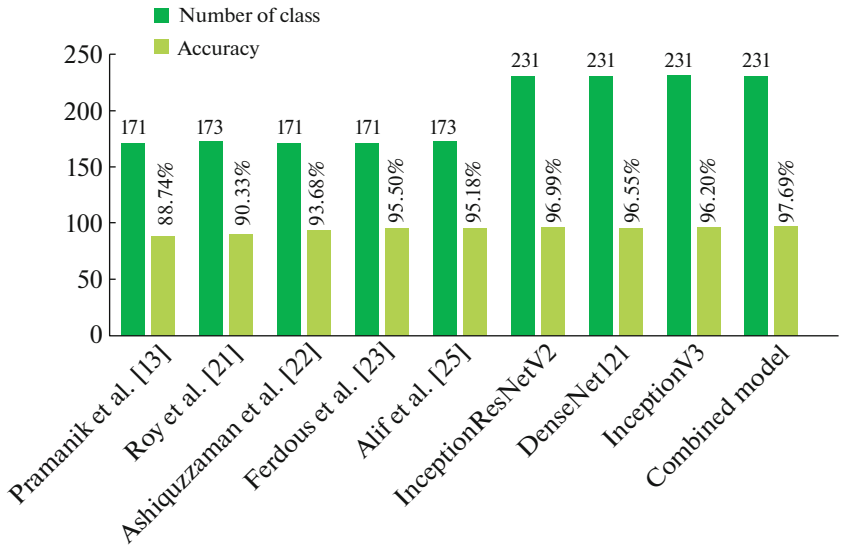

Fig. 14. Performance comparison of state of art CNN architecture and existing architecture of Bangla handwritten character recognition using CMATERdb dataset.

Table 5. Performance comparison of state of the art CNN architectures with the research work that did not use neural network

Research work	Method	Number of classes	Accuracy, %
Das et al. [11]	Shadow, longest, quad tree feature set, MLP, SVM	93	80.86
Sarkhel et al. [12]	Region sampling, SVM	221	72.87
Das et al. [14]	Convex Hull feature extractor	50	76.86
Das et al. [15]	132 features, MLP	50	85.40
Bhowmik et al. [17]	SVM based hierarchical classification	45	88.02
Parui et al. [18]	Hidden Markov model and stroke based	50	87.7
Sazal et al. [20]	Deep belief network	60	90.27
InceptionResNetV2	**InceptionResNetV2**	**231**	**96.99**
DenseNet121	**DenseNet121**	**231**	**96.55**
InceptionV3	**InceptionV3**	**231**	**96.20**
Combination Model	**InceptionResNetV2, DenseNet121, InceptionV3**	**231**	**97.69**

character recognition is a very difficult research question. This research measured the efficiency of the state-of-the-art CNN architectures in identifying 231 handwritten Bangla characters. Although the combination of InceptionResNetV2, DenseNet121, and InceptionV3 offers the highest accuracy (97.69%), but the computational cost of this voting process is really high. InceptionResNetV2 achieved significant testing accuracy with a much lower expense therefore, being a better choice for this classification problem. On testing confusing characters, this model performed equally as the combined voting algorithm. Since the recognition of a single compound character reached a very good accuracy, the future researches should concentrate in processing whole texts and reducing the computational costs. The proposed CNN architecture can be a great starting point for processing larger texts in the future and can significantly improve this field of research.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

1. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015).

2. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).
3. K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," *in European Conference on Computer Vision(ECCV)* (2016).
4. G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).
5. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition" (2014). arXiv:1409.1556 [cs.CV]
6. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018).
7. C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "InceptionV4, InceptionResNet, and the impact of residual connections on learning," in *31st AAAI Conf. Artif. Intell. (AAAI 2017)* (2017), pp. 4278–4284.
8. M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *36th Int. Conf. Mach. Learn. (ICML 2019)* (2019), pp. 10691–10700.
9. B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* (2018), pp. 8697–8710.
10. N. Das, K. Acharya, R. Sarkar, S. Basu, M. Kundu, and M. Nasipuri, "A benchmark data base of isolated Bangla handwritten compound characters," IJDAR **17**, 413–431 (2014).
11. 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," J. Comput. **2** (2), 109–115 (2010).
12. R. Sarkhel, A. K. Saha, and N. Das, "An enhanced harmony search method for Bangla handwritten character recognition using region sampling," in *Proc. 2015 IEEE 2nd Int. Conf. Recent Trends Inf. Syst. (ReTIS 2015)* (2015), pp. 325–330.
13. R. Pramanik and S. Bag, "Shape decomposition-based handwritten compound character recognition for Bangla OCR," J. Vis. Commun. Image Represent. **50**, 123–134 (2018).
14. N. Das et al., "Recognition of handwritten Bangla basic characters and digits using convex hull-based feature set," in *Int. Conf. Artif. Intell. Pattern Recognit. 2009 (AIPR 2009)* (2009), pp. 380–386.
15. N. Das, S. Basu, R. Sarkar, M. Kundu, M. Nasipuri, and D. Kumar Basu, "An improved feature descriptor for recognition of handwritten Bangla alphabet" (2015). arXiv:1501.05497 [cs.CV]
16. S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri, and D. K. Basu, "A hierarchical approach to recognition of handwritten Bangla characters," Pattern Recognit. **42** (7), 1467–1484 (2009).
17. T. Bhowmik, P. Ghanty, A. Roy, and S. Parui, "SVM-based hierarchical architectures for handwritten Bangla character recognition," Doc. Anal. Recognit. **12**, 97–108 (2009).
18. S. K. Parui, K. Guin, U. Bhattacharya, and B. B. Chaudhuri, "Online handwritten Bangla character recognition using HMM," in *2008 19th International Conference on Pattern Recognition* (2008), pp. 1–4.
19. K. Roy, "Stroke-database design for online handwriting recognition in Bangla," Int. J. Mod. Eng. Res. **2** (4), 2534–2540 (2012).
20. M. M. R. Sazal, S. K. Biswas, M. F. Amin, and K. Murase, "Bangla handwritten character recognition using deep belief network," in *2013 Int. Conf. Electr. Inf. Commun. Technol. (EICT 2013)* (2013), pp. 1–5.
21. 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 Recognit. Lett. **90**, 15–21 (2017).
22. Ashiquzzaman, A. K. Tushar, S. Dutta, and F. Mohsin, "An efficient method for improving classification accuracy of handwritten Bangla compound characters using DCNN with dropout and ELU," in *Proc. 2017 3rd IEEE Int. Conf. Res. Comput. Intell. Commun. Networks (ICRCICN 2017)* (2017), pp. 147–152.
23. A. Fardous and S. Afroge, "Handwritten isolated Bangla compound character recognition," in *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)* (Cox's Bazar, Bangladesh, 2019), pp. 1–5.
24. S. Saha and N. Saha, "A lightning fast approach to classify Bangla handwritten characters and numerals using newly structured deep neural network," Procedia Comput. Sci. **132**, 1760–1770 (2018).
25. A. K. M. S. Azad Rabby, S. Haque, S. Abujar, and S. A. Hossain, "Ekushnet: Using convolutional neural network for Bangla handwritten recognition," Procedia Comput. Sci. **143**, 603–610 (2018).
26. M. A. R. Alif, S. Ahmed, and M. A. Hasan, "Isolated Bangla handwritten character recognition with convolutional neural network," in *20th Int. Conf. Comput. Inf. Technol. (ICCIT 2017)* (2018), pp. 1–6.
27. T. Ghosh et al., "Bangla handwritten character recognition using MobileNet V1 architecture," Bull. Electr. Eng. Inf. **9** (6), 2547–2554 (2020).
28. T. Ghosh, S. M. Chowdhury, M. A. Yousuf, et al., "A comprehensive review on recognition techniques for Bangla handwritten characters," in *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)* (2019), pp. 1–6.

Tapotosh Ghosh was born in Dhaka, Bangladesh on November 3, 1998. He received his B.Sc. degree in Information and Communication Technology from Bangladesh University of Professionals, Dhaka, in 2020. Currently, he is persuading a master's degree from Bangladesh University of Professionals. He has published a conference papers and a journal paper on the field of handwritten character recognition and currently working in several computer vision-related research works. His research interest includes the application of deep learning, machine learning, and natural language processing.

Md. Min-Ha-Zul Abedin was born in Kushtia, Bangladesh on August 13, 1997. He received B.Sc. in Information and Communication Technology from Bangladesh University of Professionals in 2020. He is now working as Lecturer at the Department of Information and Communication Engineering, Bangladesh Army University of Engineering and Technology, Bangladesh. He has published a conference paper and a journal paper on the field of handwritten character recognition and currently working in several computer vision-related research works. His research interests include computer vision, pattern recognition, and artificial intelligence.

Md. Hasan Al Banna was born in Dhaka, Bangladesh in 1997. He received his B.Sc. degree in Information and Communication Technology from Bangladesh University of Professionals, Dhaka, in 2019. Currently, he is persuading a master's degree from Bangladesh University of Professionals and working as a teaching assistant in the same university. He has published a conference paper on camera model identification and currently working on earthquake prediction. His research interest includes the application of artificial intelligence and machine learning. He was awarded a fellowship from Bangladesh ICT division for his master's thesis.

Nasirul Mumenin was born in Dhaka, Bangladesh on April 12, 1997. Currently, he is persuading his B.Sc. degree in Information and Communication Technology from Bangladesh University of Professionals, Dhaka, in 2020. His research interest includes artificial intelligence, machine learning, NLP, and IOT. He is a member of academies, scientific societies, and editorial boards and journals of BUP IEEE student branch.

Dr. Mohammad Abu Yousuf received the B.Sc. (Engineering) degree in Computer Science and Engineering from Shahjalal University of Science and Technology, Sylhet, Bangladesh in 1999, the Master of Engineering degree in Biomedical Engineering from Kyung Hee University, South Korea in 2009, and the PhD degree in Science and Engineering from Saitama University, Japan in 2013. In 2003, he joined as a Lecturer in the Department of Computer Science and Engineering, Mawlana Bhashani Science and Technology University, Tangail, Bangladesh. In 2014, he moved to the Institute of Information Technology, Jahangirnagar University. He is now working as Professor at the Institute of Information Technology, Jahangirnagar University, Savar, Dhaka, Bangladesh. His research interests include medical image processing, human-robot interaction, and computer vision.