Handwritten Bangla Character Recognition using Convolutional Neural Networks: A Comparative Study and New Lightweight Model

Md. Nahidul Islam Opu[1], Md Ekramul Hossain[2,3] and Muhammad Ashad Kabir[2]

[1]Department of Computer Science and Engineering, Chittagong University of Engineering and Technology (CUET), Chattogram, 4349, Bangladesh.
[2]School of Computing, Mathematics and Engineering, Charles Sturt University, Bathurst, NSW 2795, Australia.
[3]Complex Systems Research Group, Faculty of Engineering, The University of Sydney, Darlington, NSW 2008, Australia.

Contributing authors: u1604073@student.cuet.ac.bd; mdhossain@csu.edu.au; akabir@csu.edu.au;

Abstract

Handwriting is a crucial way to enhance character recognition and learn new words. However, the Bangla characters consist of very complex shapes and similar patterns. Deep learning (DL) techniques have become a prominent solution for handwritten Bangla character recognition (HBCR) due to their ability to extract high-level features from complex data. Several DL techniques have been proposed for HBCR, but they are computationally expensive and large in model size and thus not suitable for use in resource-constrained devices such as smartphones. In this study, we have evaluated the state-of-the-art DL models for HBCR. For this, we have used four existing datasets and created a merged dataset (by combining the four) for cross-dataset evaluation. We have provided a comparative performance analysis of the state-of-the-art DL models for HBCR. Additionally, we have proposed a new lightweight DL model for HBCR and evaluated its performance. The proposed DL model consists of 74 layers, including sub-layers, and its architecture is divided into five similar blocks. It includes the convolutional layers of (3,3) and (5,5) kernels, (1,1) stride, and the max-pooling layer of (2,2) pool size. The proposed model achieved accuracy, model size, loading, and testing times of 96.87%, 13 MB, 9.11 sec, and 7.95 sec, respectively. The experimental results show that our model outperformed state-of-the-art models in terms of efficiency (loading and testing time) and model size with competitive accuracy.

Keywords: Handwritten character recognition, deep learning, convolutional neural networks, Bangla character, Bangla alphabet, Bangla digit

1 Introduction

Handwritten character recognition involves identifying handwritten symbols of a specific language and representing them in digital form using Unicode or ASCII code. It has many applications, including the input method of different portable devices, alphabet and digit learning, national identity (NID) number recognition, licence plate recognition, and many more [1–3]. The demand for a personalised handwritten character recognition tool is growing rapidly. However, handwritten character recognition is a difficult task due to the variability of handwriting in terms of person, place, time, and situation. These challenges need to be identified and solved for accurate handwritten character recognition systems.

Bangla is one of the most widely spoken languages in the world with about 300 million native speakers [4]. It is the national and mother language of Bangladesh as well as the West Bengal, Tripura, and Assam states in India. Bangla characters comprise 10 digits and an alphabet of 50 letters (11 vowels and 39 consonants) [5] (Figure 1). The shapes of Bangla characters are complicated and similar in appearance (Figure 2). Handwritten Bangla characters are also influenced by the writer's mood, writing style, and writing components' quality (e.g., pen and paper). For this reason, automatic HBCR is considered a significant challenge for researchers. In addition, the recognition of these complex characters with high accuracy is in demand for many applications related to the Bangla language, e.g., a mobile app to learn Bangla characters [6].

In the literature, a significant number of handwritten Bangla character recognition systems have been developed to classify the Bangla letters and digits [7–9]. Traditional machine learning (ML) models were extensively used to recognise handwritten Bangla characters [9–11]. In ML-based handwritten character recognition models, a feature extractor is used to extract the features from handwritten character images. Then, the features are fed into an ML model to train it. Finally, the trained model is tested to evaluate the recognition performance. However, traditional ML-based models cannot handle a large amount of data, and do not work on unstructured data. In recent years, DL techniques have gained attention from researchers for HBCR [5, 12, 13].

Vowel	অ	আ	ই	ঈ	উ	ঊ	ঋ	এ	ঐ	ও	ঔ
	ô	a	i	ee	u	oo	ri	e	oi	o	ou
Consonant	ক	খ	গ	ঘ	ঙ	চ	ছ	জ	ঝ	ঞ	ট
	kô	khô	gô	ghô	ngô	chô	chhô	ǰô	ǰhô	ñô	ṭô
	ঠ	ড	ঢ	ণ	ত	থ	দ	ধ	ন	প	ফ
	ṭhô	ḍô	ḍhô	ṇô	tô	thô	dô	dhô	nô	pô	phô
	ব	ভ	ম	য	র	ল	শ	ষ	স	হ	ড়
	bô	bhô	mô	jô	rô	lô	shô	ṣô	shô	hô	ṛô
	ঢ়	য়	ৎ	◌ং	◌ঃ	◌ঁ					
	ṛhô	yô	ṭ	ng	h	ñ					
Digit	০	১	২	৩	৪	৫	৬	৭	৮	৯	
	shunnô	ēk	dui	tin	chā	pāṅch	chhôy	shat	āṭ	nôy	
	0	1	2	3	4	5	6	7	8	9	

Fig. 1: Bangla letters (vowel and consonants) and digits. Pronunciations are provided below each character.

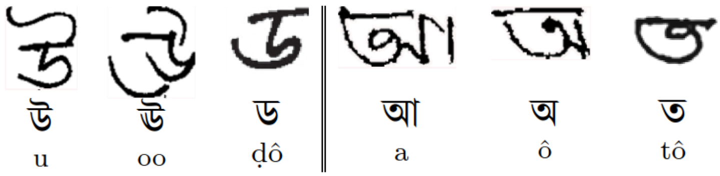


Fig. 2: Examples of similar appearance in Bangla characters.

In DL models, a separate feature extraction method is not needed as DL algorithms can automatically extract high-level features from the dataset and learn from those features. DL models, such as VGGNet [14], ResNet [15], and AlexNet [16] have performed well with higher accuracy. However, few studies have been completed on Bangla handwritten character recognition using DL techniques [5, 12, 13]. Although most models have good recognition accuracy, they are not lightweight in terms of size and efficiency.

In this study, the widely used state-of-the-art DL techniques are explored for HBCR. For this, we have used four existing datasets for the performance evaluation of the models. We also created a new merged dataset by combining all four datasets. We have provided a comparative analysis of the performance of the existing DL models with cross-dataset evaluation. In addition, we have proposed a new lightweight handwritten Bangla character recognition model using CNN. The proposed CNN model consists of 74 layers, including sub-layers. The architecture is divided into five similar blocks. It includes the convolutional layers of (3,3) and (5,5) kernals, (1,1) stride, and max-pooling layer of (2,2) pool size. The combined dataset was used to evaluate the performance of the proposed model. Our model was compared to the existing DL models in terms of accuracy, efficiency (model loading and execution time), and model size. The results presented in this study can be used as a benchmark for comparison in the future. The contributions of this paper are summarised as follows:

1. We have compared five state-of-the-art DL models by evaluating their performance for HBCR.
2. We have created a merged dataset by combining the existing four Bangla character datasets and used those five datasets for cross-dataset evaluation (train in one dataset and test the model in other datasets).
3. We have proposed a new lightweight deep learning model for HBCR to use in resource-constrained devices such as smartphones.
4. We have evaluated our proposed model in terms of accuracy, efficiency, and model size, and compared it with state-of-the-art models.

The paper is organised as follows. First, we describe the related works in Section 2. Next, Section 3 introduces the methodology for this study, including, dataset preparation, a description of the state-of-the-art DL architecture, and procedures for the proposed model. Then, we present the experimental results and discussions in Section 4. Finally, the conclusion with future research directions are drawn in Section 5.

2 Related Works

In the literature, numerous research studies have been conducted on Bangla handwritten character recognition. This section describes the existing studies based on Bangla handwritten character recognition using different approaches, including conventional, machine learning (ML), and deep learning (DL).

Hasnat et al. [17] proposed an optical character recognition (OCR) method based on machine-printed and handwritten characters. In 1994, Pal and Chaudhuri [18] presented an OCR method to identify the character shapes using a feature-based tree classifier. Chowdhury et al. [19] developed an Android application to recognise the Bangla characters from images. They used the Tesseract engine to recognise the characters by using the Leptonica image processing library. Pal and Chaudhuri [20] developed automatic handwritten Bangla digits based on features obtained from topological and statistical feature extraction methods. Rakshit et al. [7] developed a multiple-user recognition system for handwritten Bangla characters and digits. They created a dataset including isolated Bangla characters and numbers gathered from various people. They used the Tesseract OCR engine to train the data samples and develop the recognition model.

There are several remarkable studies about Bangla handwritten digit recognition. Liu and Suen [9] proposed a new benchmark to recognise handwritten Bangla numeral characters using different ML algorithms. They used the ISI digit dataset [8] to evaluate the performance of several ML techniques, including support vector machine (SVM) and multilayer perceptron (MLP). Bhattacharya and Chaudhuri [21] proposed a Bangla handwritten digit recognition system using MLP neural networks based on wavelet features. Surinta et al. [10] developed a handwritten character recognition system for handwritten Bangla numeral characters. They used a contour angular technique to extract features from handwritten Bangla digit images and then fed the features into an SVM classifier to recognise the numerical characters. The model obtained an accuracy of 96.8%. A similar MLP-based handwritten Bangla digit recognition system was proposed by Basu et al. [22]. Xu et al. [23] presented a model for handwritten Bangla digit recognition using a Bayesian hierarchical network. They used a dataset with 2000 handwritten sample images. Instead of extracted features, they directly used the images as inputs to the model. The model achieved an average recognition accuracy of 87.5%. Another study used SVM to classify simple Bangla characters [11]. The model obtained an accuracy of 88.13%.

Recently, DL techniques have gained popularity for use in handwritten Bangla character recognition [24–27]. Rabby et al. [12] developed and implemented a lightweight convolutional neural network (CNN) to classify handwritten Bangla digits. The CNN model was trained and tested with three publicly available handwritten Bangla character datasets (i.e., CMATERdb, BanglaLekha-Isolated, and ISI). The model obtained the best performance on various datasets and was found to be very lightweight. Another CNN-based model was proposed for handwritten Bangla and Odia digit recognition [28]. Alom et al [29] used several latest Deep CNN models for recognising Bangla characters and digits separately. Shawon et al. [30] proposed a handwritten Bangla digit classification system using an eighteen-layer CNN. They used the BanglaLekha-isolated and EMNIST datasets to evaluate their model performance. Saha et al. [31] proposed a DCNN-based model, namely, BBCNet-15 for recognising handwritten Bangla characters. The performance of the study was evaluated on the CMATERdb basic alphabet and digit dataset of 50 classes. Hakim et al. [13] proposed a CNN model inspired by VGG-16 for classifying handwritten Bangla characters and digits. They trained the model with the BanglaLekha-isolated dataset and tested it with their own dataset. Rabby et al. [5] proposed a Bangla handwritten character recognition system named BornoNet using a simple lightweight CNN model. Their proposed model was trained and tested on three different datasets. The same research group used a mixed dataset of alphabets, digits, modifiers, and compound characters to train a deep neural network (DNN) named EkushNet [32].

In recent years, there has been a growing interest in developing lightweight CNNs to address the computational and memory requirements associated with DL models [33–35]. These lightweight CNN architectures aim to reduce the model size, loading, and testing time as well as increase model performance and efficiency. Wang et al. [34] proposed a CNN pruning approach that can identify the structural redundancy of a network and prune the unnecessary filters in the selected layers. Wang and Li [36] developed a lightweight model to optimise the filter redundancy in channel pruning with lookahead search guided reinforcement learning. Knowledge distillation is an efficient approach for deep neural network compression that is used in machine learning to transfer knowledge from a large, complex model (the teacher) to a smaller, more compact model (the student) [37]. Recently, several studies used the Knowledge distillation technique for their model compression in terms of deep neural networks [38–41].

HBCR has been an area of active research, and deep learning (DL) techniques have emerged as a prominent solution due to their ability to extract high-level features from complex data. In the literature, many DL models have been proposed for HBCR, showing their effectiveness in recognising the intricate shapes and similar patterns present in Bangla characters. These models have demonstrated competitive accuracy, which is crucial for character recognition tasks. However, a common limitation among existing models is their computational expense and large model sizes, rendering them unsuitable for deployment on resource-constrained devices, such as smartphones. In this study, we aim to address these gaps in the literature by proposing a new lightweight DL model for HBCR. Our model consists of 74 layers, organised into five similar blocks, and incorporates a novel combination of convolutional layers with (3,3) and (5,5) kernels, (1,1) stride, and max-pooling layers with a (2,2) pool size. The architecture is specifically designed to achieve a balance between efficiency and accuracy, providing an optimal solution for HBCR that outperforms existing models in terms of loading and testing times, model size, and overall efficiency while maintaining a competitive accuracy. Through this work, we contribute a valuable approach to HBCR, offering an efficient solution that is well-suited for real-world applications, particularly on resource-constrained devices.

3 Materials and Methods

3.1 Datasets and Preprocessing

In this study, we used four existing datasets: CMATERdb [42], BanglaLekha-Isolated [43], ISI [44] and Ekush [45]. We also combined all four datasets to create a merged dataset for an extensive evaluation. Table 1 presents the different attributes of the datasets. Preprocessing has a significant impact on recognition accuracy. The goal is to make the datasets consistent and enhance the quality of the samples using noise reduction, inversion, grayscaling, and thresholding. Therefore, when necessary, the images were inverted to white writing on a black background. Grayscaling, thresholding, and noise reduction have been done on all the images.

3.2 Comparative analysis

In this study, we use and explore five high-performance state-of-the-art CNN architectures, BornoNet [5], EkushNet [32], DenseNet, Xception, and MobileNetV2, for a comparative analysis. We extensively evaluated the performance of these five CNN architectures on all five datasets. BornoNet and

Table 1: Bangla handwritten character datasets used in this study.

Dataset	Type	Number of classes	Samples	Total samples
CMATERdb	Digit	10	6000	21000
	Alphabet	50	15000	
BanglaLekha – Isolated	Digit	10	19748	118698
	Alphabet	50	98950	
ISI	Digit	10	23368	61250
	Alphabet	50	37858	
Ekush	Digit	10	30785	186400
	Alphabet	50	155570	
Merged dataset	Digit	10	79901	387279
	Alphabet	50	307378	

EkushNet are custom CNN architectures proposed for Bangla handwritten character recognition. A brief description of these two architectures is given below.

BornoNet [5] consists of 13 layers with 2 sub-layers (Fig. 3). This network is made with three layers: convolutional, pooling, and fully connected with dropout as a regularisation method. They used three dropout layers to avoid overfitting, and ReLU as an activation function in the network with a softmax function in the final output layer.

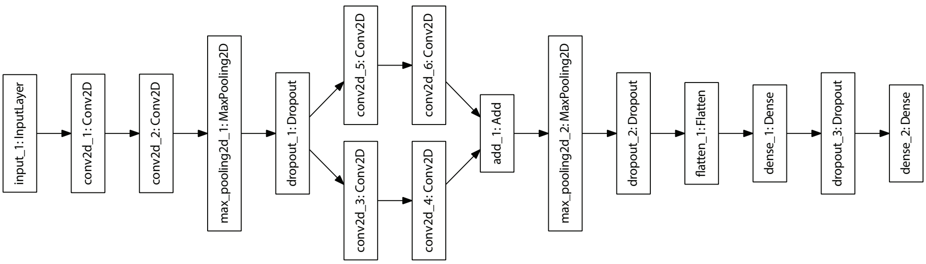


Fig. 3: BornoNet architecture.

EkushNet [32] consists of 23 layers containing ten sub-layers (Fig. 4). Convolutional, max-pooling, and fully connected layers were used in this network. Various regularisation techniques are used, including batch normalisation and dropout. The network is divided into two parts after the fifth layer. The first has four layers, while the second has six. These two sections are merged on the 16th layer. The final layer required the number of output nodes with Softmax activation.

3.3 Proposed New Architecture

A model needs to be lightweight (while providing high accuracy) for use in real-world applications, particularly, resource-constrained scenarios, such as

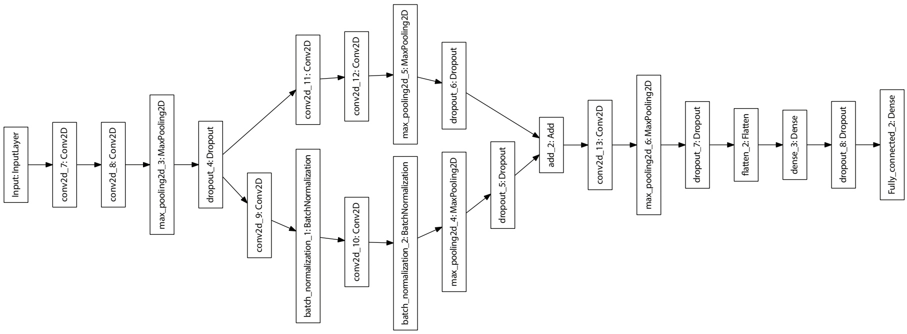


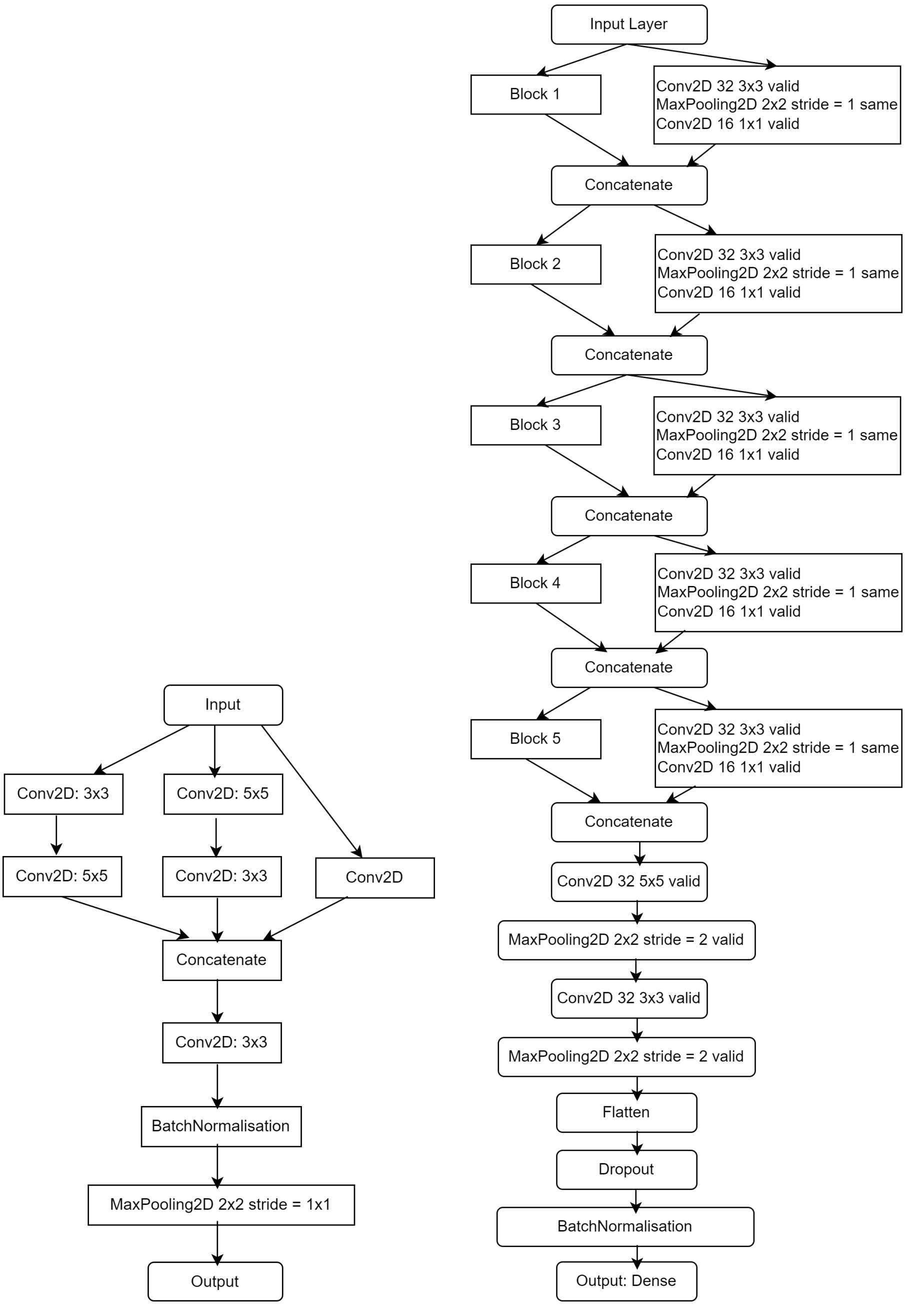
Fig. 4: EkushNet architecture.

smartphones. Considering this, we develop a new CNN architecture for Bangla handwritten character recognition (shown in Fig. 5). Our proposed architecture consists of 74 layers, including sub-layers. The architecture is divided into five similar blocks, where the input of each block is divided into three sub-layers. First, the sub-layer has two consecutive convolutional layers of (3,3) and (5,5) kernels. The second sub-layer has two consecutive convolutional layers of (5,5) and (3,3). And the third sub-layer has a convolutional layer. All these sub-layers use (1,1) stride, 'same' padding, and 'relu' as an activation function. These three sub-layers are then concatenated into a layer followed by a convolutional layer of (3,3) kernel with 'valid' padding and 'relu' activation. The last two block layers are batch normalisation and a max-pooling layer of (2,2) pool size and 'same' padding with stride 1. The number of filters in all layers and the kernel size of the last layer are fixed in each block and different for different blocks.

From the second block, each block has two inputs concatenated. These two inputs are the output of its previous block and the input of its previous block. The kernel size and number of kernels for each block are [3×3,64], [5×5, 64], [3×3, 32], [5×5, 32], and [3×3, 32]. After the fifth block, there is a pair of convolutional layers with 32 filters of (5×5) size and a max pool layer of 2×2 pool size, followed by another pair of convolutional layers with 32 filters of (3×3) size and a max pool layer of 2×2 pool size. The flatten layer is followed by a 25% dropout layer and a batch normalisation. The final layer is a fully connected layer with softmax activation.

4 Experimental Evaluations and Discussions

In this study, we used five state-of-the-art DL models using different datasets and compared their performances with each other. In addition, we evaluated the proposed model and compared its performance with other existing DL models for all datasets used in this study.



(a) Building lock of proposed model (b) Proposed model architecture

Fig. 5: Building block and architecture of the proposed model.

4.1 Experimental Settings

In this study, each dataset is divided into three subsets (i.e., training, validation, and testing) with a split ratio of 80:10:10. Based on the initial testing performance, the size of the input images to the DL models was selected as 28×28. This study employed ADAM as an optimisation algorithm for updating the parameters in the network, where the algorithm is popular in stochastic optimisation due to its faster computational time compared to other optimisation algorithms [46]. The initial learning rate for DL models was set to 0.001. However, the learning rate needs to be updated for fast convergence and better results. A lower learning rate will take more time to converge; on the other hand, a higher learning rate can even diverge instead of connecting. This study used an adaptive learning rate reduction method [47] to reduce the learning rate by monitoring validation accuracy. We applied data augmentation in our experiment as it reduces loss and increases accuracy, and thus prevents overfitting [48]. Three augmentation techniques were used to augment the datasets, including rotation of 90°, 10% zoom, and shifting height and width. The total epoch for each model was set to 40 with a batch size of 100.

4.2 Evaluation of Five Existing DL Models for Bangla Character Recognition

For comparative analysis, we used five existing DL models selected for this study. The selected DL models were trained using one of the datasets separately and tested against the others. Thus, we can verify whether the model trained with one dataset works well, when tested with other datasets. This study separated digits and letters from the datasets. Thus, we evaluated the performance of DL models using digits and letters separately.

4.2.1 Performance of DL Models Trained on CMATERdb Dataset

Table 2 shows the performance results of five DL models (i.e., BornoNet, EkushNet, DenseNet, Xception, and MobileNetV2) using alphabet and digit datasets. The models were trained based on the CMATERdb dataset and tested with the BanglaLekha, ISI, Ekush, and merged datasets. In the training phase, the models were evaluated using several metrics, including train loss, validation loss, train accuracy, and validation accuracy. In the testing phase, the accuracy metric was calculated to evaluate the performance of the DL models. In terms of evaluation metrics, the best models are shown in bold. The accuracy for the EkushNet model was evaluated to be the highest for all test datasets in terms of the alphabet, at 88.67% for BanglaLekha, 93.84% for ISI, 89.50% for Ekush, and 90.08% for merged. For the CMATERdb digit training dataset, the DenseNet model obtained the highest accuracy of 93.04%, 90.27%, and 92.85% for the BanglaLekha, Ekush, and merged test datasets, respectively. For the ISI test dataset, the BornoNet model achieved the highest accuracy of 93.05%. Overall, this study found the best accuracy for the digit datasets compared to alphabet datasets.

Table 2: Performance of models trained on CMATERdb dataset and tested on other datasets.

Dataset	Model	Train loss	Val loss	Train acc. (%)	Val acc. (%)	Test accuracy (%)			
						BanglaLekha	ISI	Ekush	Merged
CMATERdb alphabet	BornoNet	0.0389	**0.0581**	98.74	**98.50**	87.40	93.24	86.82	88.73
	EkushNet	0.0598	0.0631	97.74	98.00	**88.67**	**93.84**	**89.50**	**90.08**
	DenseNet	**0.0211**	0.0639	99.42	97.58	88.07	92.48	88.89	89.42
	Xception	0.0361	0.0663	**98.81**	97.75	87.60	93.01	87.22	88.50
	MobileNetV2	0.1428	0.2587	95.20	92.25	74.99	84.05	73.06	76.76
CMATERdb digit	BornoNet	0.0230	**0.0107**	99.30	**99.83**	92.38	**93.05**	88.59	92.33
	EkushNet	0.0250	0.0155	99.11	99.50	92.57	91.68	89.66	92.42
	DenseNet	**0.0150**	0.0262	**99.54**	99.33	**93.04**	92.55	**90.27**	**92.85**
	Xception	0.0209	0.0173	99.31	99.33	92.70	90.36	90.15	92.43
	MobileNetV2	0.0506	0.0766	98.17	97.34	90.34	84.54	87.76	89.21

4.2.2 Performance of DL Models Trained on BanglaLekha Dataset

The performance results of DL models trained on the BanglaLekha dataset and tested with the CMATERdb, ISI, Ekush and merged datasets, are presented in Table 3. The accuracy of the DenseNet model was evaluated to be the highest for the ISI, Ekush, and merged datasets for the alphabet, at 95.49%, 95.77%, and 96.83%, respectively. For CMATERdb, the EkushNet model achieved the highest accuracy of 97.13%. For digits, the EkushNet model obtained the highest accuracy of 98.11% and 98.64% for the CMATERdb and ISI, respectively. The DenseNet model obtained the highest accuracy of 96.14% for Ekush. The Xception model performed best for the Merged dataset.

Table 3: Performance of models trained on BanglaLekha dataset and tested on other datasets.

Dataset	Model	Train loss	Val loss	Train acc. (%)	Val acc. (%)	Test accuracy (%)			
						CMATERdb	ISI	Ekush	Merged
BanglaLekha alphabet	BornoNet	0.1499	0.2085	95.45	94.98	96.83	95.10	93.73	95.26
	EkushNet	0.1256	**0.1902**	96.13	95.50	**97.13**	94.85	95.38	96.20
	DenseNet	**0.0391**	0.2293	**98.71**	**95.58**	96.80	**95.49**	**95.77**	**96.83**
	Xception	0.0833	0.2054	97.26	95.06	96.87	94.82	95.50	96.28
	MobileNetV2	0.1421	0.2184	95.54	94.07	95.80	93.60	93.96	94.85
BanglaLekha digit	BornoNet	0.0265	0.0449	99.18	98.68	98.10	98.53	95.85	97.96
	EkushNet	0.0234	**0.0349**	99.24	**98.94**	**98.11**	**98.64**	95.83	97.98
	DenseNet	**0.0133**	0.0436	**99.63**	98.84	97.91	98.23	**96.14**	97.86
	Xception	0.0204	0.0402	99.38	98.84	97.86	98.51	96.01	**98.01**
	MobileNetV2	0.0316	0.0608	98.92	98.18	96.83	97.06	93.84	96.84

4.2.3 Performance of DL Models Trained on ISI Dataset

Table 4 shows the performance results of DL models trained on the ISI dataset and tested on the CMATERdb, BanglaLekha, Ekush, and merged datasets. The different evaluation metrics were calculated during the training and testing phases. We performed the comparison of different DL models based on test accuracy, although all evaluation metrics used in this study are shown in Table 4. The accuracy for DenseNet model was evaluated to be the highest for all test datasets for the alphabet, at 96.34% for the CMATERdb, 90.45% for the BanglaLekha, 71.74% for the Ekush, and 92.14% for the merged dataset. For the ISI digit training dataset, the DenseNet model obtained the highest accuracy of 97.85%, 96%, and 98.02% for the BanglaLekha, Ekush, and merged test datasets, respectively. The EkushNet model obtained the highest accuracy of 99.11% for the CMATERdb test dataset.

Table 4: Performance of models trained on ISI dataset and tested on other datasets.

Dataset	Model	Train loss	Val loss	Train acc. (%)	Val acc. (%)	Test accuracy (%)			
						CMATERdb	BanglaLekha	Ekush	Merged
ISI alphabet	BornoNet	0.0463	0.1112	98.40	96.97	96.10	89.59	90.22	91.25
	EkushNet	0.0674	**0.1040**	97.79	**96.86**	96.31	90.09	90.99	91.84
	DenseNet	**0.0133**	0.1316	**99.61**	96.69	**96.34**	**90.45**	**91.74**	**92.14**
	Xception	0.0342	0.1239	98.89	96.73	96.02	89.67	91.57	91.64
	MobileNetV2	0.0823	0.2020	97.26	93.60	92.26	84.80	85.87	87.19
ISI digit	BornoNet	0.0075	0.0097	99.71	99.64	99.06	97.59	94.16	97.26
	EkushNet	0.0116	**0.0069**	99.60	**99.74**	**99.11**	97.67	95.28	97.57
	DenseNet	**0.0061**	0.0147	**99.76**	99.48	98.96	**97.85**	**96.00**	**98.02**
	Xception	0.0100	0.0098	99.64	99.69	98.85	97.76	95.61	97.71
	MobileNetV2	0.0155	0.0189	99.47	99.38	98.90	96.61	93.67	96.62

4.2.4 Performance of DL Models Trained on Ekush Dataset

The performance results of DL models trained on the Ekush dataset and tested on the CMATERdb, BanglaLekha, ISI, and merged datasets are shown in Table 5. The performance evaluation metrics were calculated for DL models to identify the best model in terms of test accuracy. According to the performance, the EkushNet model obtained the highest accuracy for all test datasets for the alphabet, and the accuracy was 96.36% for the CMATERdb, 93.49% for the BanglaLekha, 93.80% for the ISI, and 95.49% for the merged test datasets. For the Ekush digit training dataset, the EkushNet model obtained the highest accuracy of 98.90% and 98.94% for the CMATERdb and merged test datasets, respectively. For the BanglaLekha test dataset, the DenseNet model obtained the highest accuracy of 97.99%. The Xception model performed best for the ISI test dataset with an accuracy of 98.16%.

4.2.5 Performance of DL Models Trained on Merged Dataset

To evaluate the performance of DL models for Bangla character recognition, we trained the models using the merged dataset and tested them using the

Table 5: Performance of models trained on Ekush dataset and tested on other datasets.

Dataset	Model	Train loss	Val loss	Train acc. (%)	Val acc. (%)	Test accuracy (%)			
						CMATERdb	BanglaLekha	ISI	Merged
Ekush alphabet	BornoNet	0.0968	0.1173	97.28	97.46	95.51	92.25	92.70	94.53
	EkushNet	0.0674	**0.0929**	98.04	**97.99**	**96.36**	**93.49**	**93.80**	**95.49**
	DenseNet	**0.0201**	0.1096	**99.40**	97.89	95.17	92.73	92.99	95.27
	Xception	0.0403	0.1054	98.79	97.70	92.55	90.34	90.56	93.42
	MobileNetV2	0.0570	0.1142	98.29	97.82	93.99	91.37	92.00	94.11
Ekush digit	BornoNet	0.0104	0.0264	99.60	99.58	98.63	97.81	97.41	98.82
	EkushNet	0.0119	0.0188	99.63	**99.74**	**98.90**	97.80	97.93	**98.94**
	DenseNet	**0.0045**	0.0279	**99.82**	99.45	98.18	**97.99**	97.82	98.81
	Xception	0.0140	**0.0155**	99.52	99.51	98.83	97.94	**98.16**	98.86
	MobileNetV2	0.0176	0.0222	99.41	99.22	98.28	97.47	96.76	98.44

CMATERdb, BanglaLekha, ISI, and Ekush datasets. In the training and testing phases, we calculated different evaluation metrics which are presented in Table 6. The results show that the DenseNet model achieved the highest accuracy of 98.4%, 99.32%, and 98.66% for the BanglaLekha, ISI, and Ekush test datasets, respectively, when it was trained using the merged alphabet dataset. However, Xception obtained the highest accuracy of 98.9% for the CMATERdb test dataset. For the merged digit dataset as a training dataset, the best-performing model is Xception for the CMATERdb, ISI, and Ekush test datasets as the model achieved the highest accuracy. For the Ekush test dataset, EkushNet and DenseNet performed equally well, with an accuracy of 99.91% when the models were trained using the merged digit dataset.

Table 6: Performance of models trained on Merged dataset and tested on other datasets.

Dataset	Model	Train loss	Val loss	Train acc. (%)	Val acc. (%)	Test accuracy (%)			
						CMATERdb	BanglaLekha	ISI	Ekush
Merged alphabet	BornoNet	0.1594	0.1591	95.43	96.12	98.8	96.43	98.29	97.76
	EkushNet	0.1198	**0.1370**	96.61	**96.67**	99.17	97.66	98.79	98.32
	DenseNet	**0.0399**	.01622	**98.69**	96.62	99.40	**98.40**	**99.32**	**98.66**
	Xception	0.0814	0.1460	97.36	96.58	**99.42**	98.02	99.17	98.45
	MobileNetV2	0.1098	0.1556	96.62	95.81	98.75	96.74	98.30	97.76
Merged digit	BornoNet	0.0178	0.0216	99.40	99.49	99.86	99.66	99.79	99.90
	EkushNet	0.0169	**0.0190**	99.47	99.42	99.83	99.65	99.79	**99.91**
	DenseNet	**0.0076**	0.0253	**99.75**	**99.55**	99.85	99.67	99.80	**99.91**
	Xception	0.0125	0.0216	99.54	99.52	**99.91**	**99.72**	**99.84**	99.90
	MobileNetV2	0.0153	0.0255	99.47	99.25	99.90	99.49	99.73	99.80

4.3 Performance of the Proposed Model

In this study, the performance of the proposed model was compared to the performance of five existing DL models (i.e., BornoNet, EkushNet, DenseNet, Xception, and MobileNetV2). We selected the merged dataset for training and testing the models. To evaluate the performance, we calculated several evaluation metrics during the training and testing phases, including training accuracy, validation accuracy, test accuracy, loading time, testing time, and model size as shown in Table 7. Loading time is calculated by loading each model ten times and measuring the mean and standard deviation. Testing time is the time to test mixed datasets on each model. The model size is the amount of memory consumed when a model is saved on a memory device. According to the results, DenseNet showed the highest accuracy for Bangla character recognition, although the proposed model outperformed the other DL models. In terms of model size, loading, and testing time, the proposed model outperformed all the existing DL models considered in this study. Thus, the proposed model can process the data faster and consumes less memory.

Table 7: Comparison between several existing DL models and the proposed model

Model	Train acc. (%)	Val acc. (%)	Test acc. (%)	Loading time (sec)	Testing time (sec)	Model size (MB)
BornoNet	94.70	96.14	96.28	1.974 ± 0.034	8.813 ± 0.179	51.6
EkushNet	96.40	96.70	96.71	2.801 ± 0.0307	9.235 ± 0.084	18.6
DenseNet	**98.51**	**96.88**	**96.90**	36.35 ± 0.75	98.879 ± 0.829	144.3
Xception	97.46	96.58	96.63	48.42 ± 1.10	58.956 ± 1.898	403.2
MobileNetV2	96.80	96.14	96.22	20.956 ± 0.588	26.05 ± 1.68	28.4
Proposed model	97.84	96.79	96.87	$\mathbf{1.911 \pm 0.112}$	$\mathbf{7.956 \pm 1.918}$	**13.2**

4.4 Limitation of the Study and Future Research Directions

One limitation of this study is the generalizability of the findings to a new real-world dataset for handwritten Bangla characters. While we evaluated the proposed lightweight DL model on four existing datasets and their merged dataset, the performance may vary on unseen data from different sources. To address this limitation, future research should include evaluations on diverse and larger datasets, encompassing a wide range of writing styles and variations in handwriting. Another limitation is that the current study did not include a comparison of FLOPs (Floating Point Operations) in the evaluation of the proposed approach against existing works. FLOPs provide valuable insights into the computational efficiency and resource requirements of deep learning models. Incorporating FLOPs comparison would give a more comprehensive understanding of the trade-offs between model accuracy and computational cost.

In the future, we plan to conduct an ablation study to further validate the effectiveness of the proposed approach. This study will involve systematically removing blocks from the proposed model and evaluating its performance under different configurations. By analyzing how the reduction in the number of blocks affects the model size and recognition accuracy, we can gain insights into the trade-offs between model complexity and performance. This will help optimize the proposed approach for specific deployment scenarios and resource constraints. Additionally, we aim to extend the performance comparison of the proposed model with state-of-the-art models using FLOPs.

Furthermore, we will explore the robustness of the proposed model by conducting evaluations on larger and more diverse datasets. This will allow us to assess its ability to handle variations in handwriting styles, noise, and different writing conditions commonly encountered in real-world scenarios. Lastly, to ensure the proposed approach's usability in practical applications, we will consider real-time implementation and deployment on resource-limited devices such as smartphones, for instance, by extending the Lear2Write app [6].

5 Conclusions

This study presented a comparative analysis of the existing DL models and proposed a new lightweight CNN model for HBCR. For this, we used four public datasets for Bangla letters and digits. In addition, we created a combined dataset by merging all four datasets. For the comparative performance analysis, we trained each existing DL model on one dataset and tested it on the rest of the datasets. Thus, we examined the consistency of the models' performance for all datasets. The results showed that DenseNet outperformed the other existing models used in this study for handwritten Bangla alphabets and digits recognition. It is also observed that the models were trained well with the ISI alphabet dataset as the models obtained the highest test accuracy for the rest of the alphabet datasets. To evaluate the performance of the proposed model, this study calculated model size, loading, and testing time in addition to the accuracy. We compared our proposed model to the existing DL models in terms of these evaluation parameters. The results showed that the proposed model outperformed all existing models in terms of processing time and model size. In terms of accuracy, our model achieved the highest accuracy compared to most of the existing models used in this study for handwritten Bangla character recognition. Thus, the proposed lightweight Bangla handwritten recognition model could be useful for real-world applications related to the Bangla language.

Declarations

Conflicts of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability Statements

The datasets generated during and/or analysed during the current study are available from the corresponding author upon reasonable request.

References

[1] Du, S., Ibrahim, M., Shehata, M., Badawy, W.: Automatic license

plate recognition (alpr): A state-of-the-art review. IEEE Transactions on circuits and systems for video technology **23**(2), 311–325 (2012)

[2] Srikantan, J.F.G., Srihari, S.: Handprinted character/digit recognition using a multiple feature/resolution philosophy. In: Proc. Fourth Int'l Workshop Frontiers in Handwriting Recognition, p. 57 (1994)

[3] Weng, Y., Xia, C.: A new deep learning-based handwritten character recognition system on mobile computing devices. Mobile Networks and Applications **25**(2), 402–411 (2020)

[4] Simons, G.F., Fennig, C.D.: Summary by language size. Ethnologue: Languages of the world **12** (2017)

[5] Rabby, A.S.A., Haque, S., Islam, S., Abujar, S., Hossain, S.A.: Bornonet: Bangla handwritten characters recognition using convolutional neural network. Procedia computer science **143**, 528–535 (2018)

[6] Opu, M.N.I., Islam, M.R., Kabir, M.A., Hossain, M.S., Islam, M.M.: Learn2write: Augmented reality and machine learning-based mobile app to learn writing. Computers **11**(1) (2022). https://doi.org/10.3390/computers11010004

[7] Rakshit, S., Ghosal, D., Das, T., Dutta, S., Basu, S.: Development of a multi-user recognition engine for handwritten bangla basic characters and digits. arXiv preprint arXiv:1003.5897 (2010)

[8] Chaudhuri, B.: A complete handwritten numeral database of bangla–a major indic script. (2006)

[9] Liu, C.-L., Suen, C.Y.: A new benchmark on the recognition of handwritten bangla and farsi numeral characters. Pattern Recognition **42**(12), 3287–3295 (2009)

[10] Surinta, O., Schomaker, L., Wiering, M.: A comparison of feature and pixel-based methods for recognizing handwritten bangla digits. In: 2013 12th International Conference on Document Analysis and Recognition, pp. 165–169 (2013). IEEE

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

[12] Rabby, A.S.A., Abujar, S., Haque, S., Hossain, S.A.: Bangla handwritten digit recognition using convolutional neural network. In: Emerging Technologies in Data Mining and Information Security, pp. 111–122. Springer,

??? (2019)

[13] Hakim, S.A., *et al.*: 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 (2019). IEEE

[14] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

[15] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

[16] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)

[17] Hasnat, M.A., Habib, S.M., Khan, M.: A high performance domain specific ocr for bangla script. In: Novel Algorithms and Techniques In Telecommunications, Automation and Industrial Electronics, pp. 174–178. Springer, ??? (2008)

[18] Pal, U., Chaudhuri, B.: Ocr in bangla: an indo-bangladeshi language. In: Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3-Conference C: Signal Processing (Cat. No. 94CH3440-5), vol. 2, pp. 269–273 (1994). IEEE

[19] Chowdhury, A., Foysal, A., Islam, S.: Bangla character recognition for android devices. International Journal of Computer Applications **136**(11) (2016)

[20] Pal, U., Chaudhuri, B.: Automatic recognition of unconstrained off-line bangla handwritten numerals. In: International Conference on Multimodal Interfaces, pp. 371–378 (2000). Springer

[21] Bhattacharya, U., Chaudhuri, B.: A majority voting scheme for multiresolution recognition of handprinted numerals. In: Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings., pp. 16–20 (2003). IEEE

[22] Basu, S., Das, N., Sarkar, R., Kundu, M., Nasipuri, M., Basu, D.K.: An mlp based approach for recognition of handwrittenbangla'numerals. arXiv preprint arXiv:1203.0876 (2012)

[23] Xu, J.-w., Xu, J., Lu, Y.: Handwritten bangla digit recognition using hierarchical bayesian network. In: 2008 3rd International Conference on

Intelligent System and Knowledge Engineering, vol. 1, pp. 1096–1099 (2008). IEEE

[24] Cireşan, D., Meier, U.: Multi-column deep neural networks for offline handwritten chinese character classification. In: 2015 International Joint Conference on Neural Networks (IJCNN), pp. 1–6 (2015). IEEE

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

[26] Maitra, D.S., Bhattacharya, U., Parui, S.K.: Cnn based common approach to handwritten character recognition of multiple scripts. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 1021–1025 (2015). IEEE

[27] Chowdhury, R.R., Hossain, M.S., Ul Islam, R., Andersson, K., Hossain, S.: 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 (2019). IEEE

[28] Sethy, A., Patra, P.K., Nayak, S.R.: Offline Handwritten Numeral Recognition Using Convolution Neural Network. Machine Vision Inspection Systems **1**, 197–212 (2020). https://doi.org/10.1002/9781119682042.ch9

[29] Alom, M.Z., Sidike, P., Hasan, M., Taha, T.M., Asari, V.K.: Handwritten bangla character recognition using the state-of-the-art deep convolutional neural networks. Computational intelligence and neuroscience **2018** (2018)

[30] Shawon, A., Rahman, M.J.-U., Mahmud, F., Zaman, M.A.: Bangla handwritten digit recognition using deep cnn for large and unbiased dataset. In: 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), pp. 1–6 (2018). IEEE

[31] Saha, C., Faisal, R.H., Rahman, M.M.: 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 (2019). IEEE

[32] Rabby, A.S.A., Haque, S., Abujar, S., Hossain, S.A.: Ekushnet: Using convolutional neural network for bangla handwritten recognition. Procedia computer science **143**, 603–610 (2018)

[33] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018)

[34] Wang, Z., Li, C., Wang, X.: Convolutional neural network pruning with structural redundancy reduction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14913–14922 (2021)

[35] Wang, Z., Li, C., Wang, X., Wang, D.: Towards efficient convolutional neural networks through low-error filter saliency estimation. In: PRICAI 2019: Trends in Artificial Intelligence: 16th Pacific Rim International Conference on Artificial Intelligence, Cuvu, Yanuca Island, Fiji, August 26–30, 2019, Proceedings, Part II 16, pp. 255–267 (2019). Springer

[36] Wang, Z., Li, C.: Channel pruning via lookahead search guided reinforcement learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 2029–2040 (2022)

[37] Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)

[38] Wang, Z.: Zero-shot knowledge distillation from a decision-based black-box model. In: International Conference on Machine Learning, pp. 10675–10685 (2021). PMLR

[39] Li, C., Wang, Z., Qi, H.: Online knowledge distillation by temporal-spatial boosting. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 197–206 (2022)

[40] Wu, G., Gong, S.: Peer collaborative learning for online knowledge distillation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 10302–10310 (2021)

[41] Wang, Z.: Data-free knowledge distillation with soft targeted transfer set synthesis. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 10245–10253 (2021)

[42] Sarkar, R., Das, N., Basu, S., Kundu, M., Nasipuri, M., Basu, D.K.: Cmaterdb1: a database of unconstrained handwritten bangla and bangla–english mixed script document image. International Journal on Document Analysis and Recognition (IJDAR) **15**(1), 71–83 (2012)

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

103–107 (2017)

[44] Bhattacharya, U., Chaudhuri, B.B.: Handwritten numeral databases of indian scripts and multistage recognition of mixed numerals. IEEE transactions on pattern analysis and machine intelligence **31**(3), 444–457 (2008)

[45] Rabby, A.S.A., Haque, S., Islam, M.S., Abujar, S., Hossain, S.A.: 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 (2018). Springer

[46] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

[47] Schaul, T., Zhang, S., LeCun, Y.: No more pesky learning rates. In: International Conference on Machine Learning, pp. 343–351 (2013)

[48] Perez, L., Wang, J.: The effectiveness of data augmentation in image classification using deep learning. arXiv preprint arXiv:1712.04621 (2017)