Multimedia Tools and Applications manuscript No.
(will be inserted by the editor)

RATNet: A Deep Learning Model for Bengali Handwritten Characters Recognition

Md Shafiqul Islam[1] · **Md Moklesur Rahman**[1] · **Md Hafizur Rahman**[2] · **Massimo Walter Rivolta**[3] · **Md Aktaruzzaman**[4]

Received: date / Accepted: date

Abstract The Bengali language is based on a set of symbols for basic characters, modifiers, compound characters, and numerals. The recognition rates of handwritten basic characters and numerals are very high. However, the recognition rates of compound characters and modifiers are still poor. This might be due to their large class size with huge writing styles, much similarity, and unavailability of sufficient data for deep learning. In fact, there are some compound characters which appear very rare in practice. A proper selection of frequently used characters may reduce class size, and hence improving the accuracy. In this study, we performed a statistics on the frequency of compound characters, we developed two datasets for modifiers and compound characters, and finally we proposed a heterogeneous deep learning model (RATNet) for characters recognition. A statistics was performed on two daily Bengali newspapers, and characters with frequency $\geq 5\%$ were selected. The handwriting of selected characters was collected from 130 writers of different ages and professions. The performance of RATNet model was evaluated on the proposed datasets and also three other existing datasets (*i.e.*, ISI, CMATERdb, BanglaLekha-Isolated). In addition, the performance of RATNet was also compared with LeNet-5, VGG-16, ResNet-50, and DenseNet-121 models. We selected 87 out of 107 compound characters. The proposed RATNet model outperforms other models providing 99.66%, 99.27%, 98.78%, and 97.70% accuracy, respectively for the recognition of numerals, basic characters, modifiers, and compound characters on the CMATERdb dataset while keeping the number of parameters relatively low likely due to layer heterogeneity.

Keywords Bengali · Handwritten characters recognition · Dataset · Convolutional neural network · Residual attention model · Deep learning

1 Introduction

The development of optical character recognition (OCR) systems for handwritten characters (HCR) of some languages recently achieved human-level error rates, especially after the introduction of deep learning (DL) techniques [3, 6, 13]. In particular, the performance for recognizing handwritten characters in European scripts (based on the Roman alphabet) [39] and several Asian languages like Chinese [48], Japanese [44] and Korean [32], achieved accuracies $> 95\%$. However, HCR systems for languages of the Indian subcontinent presents additional challenges, mainly due to the large amount of characters and variability in the writing styles among individuals [18].

Bengali is the second mostly spoken language in the Indian subcontinent and the official language of Bangladesh [14]. It is also ranked fifth based on the number of people who speak Bengali worldwide [46]. In Bengali, there are 10 basic symbols in its decimal number system and 50 basic characters (11 vowels, 39 consonants). Also, 12 different modifiers may change the shape of a vowel or some consonants, and might appear on the top, bottom, left, or right of each single character. In addition to this already large set of different characters, there are also between 160 and 334 compound characters currently in use [17], which are formed by combining two or more consonants together. Another complication is that the shape of such combined character is different from the shape of its members simply

✉ Md Aktaruzzaman (Corresponding Author)
aktaruzzaman@iu.ac.bd
[1] Department of Computer Science and Engineering, The People's University of Bangladesh, Dhaka-1207, Bangladesh.
[2] Department of Electrical and Electronics Engineering, Islamic University, Kushtia-7003, Bangladesh.
[3] Dipartimento di Informatica, Università degli Studi di Milano, Italy.
[4] Department of Computer Science and Engineering, Islamic University, Kushtia-7003, Bangladesh.

aligned. Therefore, in a survey [17], considering all possible characters, the development of a complete Bengali HCR system led to a classification problem of 232 classes.

The research in Bengali HCR has been improved recently for recognition of basic characters and numerals using DL-based methods [2, 4, 35, 36]. A few works reported promising results (although with comparatively low accuracy) for the recognition of compound characters [5, 38]. This limited amount of research might be due to the unavailability of sufficient data of compound characters. The major reasons for the unavailability of datasets of compound characters are the large variety of classes and very rare occurrences of some compound characters. Furthermore, these studies suffer from one major limitation, *i.e.*, the use of a subset of Bengali alphabet. To the best of our knowledge, no research is found in the literature that has considered the complete set of Bengali alphabet. As a consequence, there are many works, but no work to adopt in practice. Moreover, the free datasets of compound characters and modifiers are very rare, thus limiting the use of DL techniques.

The objectives of this study are: i) to develop a dataset of handwritten Bengali compound characters and modifiers based on an extensive survey of the comparative rate of occurrence of the compound characters; ii) to develop a DL architecture with the aim of improving the recognition accuracy of existing recognition methods; iii) to develop a complete recognition system of handwritten Bengali basic characters, compound characters, modifiers, numerals; and finally iv) to compare the performance of the proposed method with well-established DL architectures adapted for Bengali HCR.

2 Related Works

Convolutional neural networks (CNNs) have been found outperforming feature-based classifiers for Bengali HCR of both numeral and basic characters (*e.g.*, [33, 35] vs [11, 18, 30]). By this time, a few researchers [1, 2, 4, 41] have explored the power of CNNs for Bengali HCR. In 2015, Rahman *et al.* [35] proposed the use of CNN for handwritten Bengali characters recognition. They proposed a CNN model that normalizes the handwritten character images which are fed to a CNN to classify individual characters of their own generated database. They reported about 85.35% recognition accuracy on a dataset of 20,000 samples. Sazal *et al.* [40] proposed a DL approach for Bengali handwritten characters recognition. Their proposed deep belief network (DBN) takes the input images, and the network is trained in two steps. Differently from other models, DBN can generate samples which can be used for data augmentation for other classification problems. They used Bengali numerals [9] and other basic characters [10] datasets collected from Indian Statistical Institute. The average classification accuracy for DBN was around 90%. Shopon *et al.* [41] used autoencoders and deep CNNs. The deep network was trained with 19,313 samples of the ISI training dataset [9] and tested with images of the CMATERdb3.1.1 dataset [8], and which led to an accuracy of 99.50%. Akhand *et al.* [1] employed CNNs for handwritten Bengali isolated numerals recognition on a self-generated dataset of 17,000 numeral images. They used 13,000 samples for training the CNNs and the remaining 4,000 to assess the model performance. They reported 99.40% and 97.93% recognition accuracies on the train and test sets, respectively. Later, Akhand *et al.* [2] has proposed a different CNNs based method for handwritten Bengali numerals recognition. In their study, three different CNNs with the same architecture were used for the experiments. Each CNN was trained with a different training set prepared from the samples of handwritten images, and the final decision was made by combining the decisions of the CNNs. One CNN was trained with the ordinary images of the handwritten numerals. A simple rotation based approach was used to generate training sets for the other two CNNs. A set of 18,000 images (out of the 19,392) of the ISI numeral dataset was considered for training one CNN. The performance of the developed system was tested on the separated test set, of 4,000 samples, of the ISI database. Their method with multiple CNNs reported accuracy of 98.80% and 99.51% on the test and training sets, respectively, and currently, it is the best performance available in the literature on Bengali HCR. It is a great achievement that seems closer to human perception. However, the test set comprised of only 4,000 samples. Alom *et al.* [5] published a work where they have implemented a set of CNNs for Bengali handwritten characters (including digits, alphabets, and special characters) recognition. Very recently, Modhej *et al.* [29] proposed an intelligent model for handwritten character recognition of five languages based on the computational function of the dentate gyrus of the brain's hippocampus. The proposed model uses two excitation and inhibition steps which play vital roles in increasing character recognition accuracy. They evaluated the model with six datasets from five languages, and reported an accuracy of 99.65% and 99.60%, which is the highest accuracy till found for handwritten Telugu and Devanagari numerals recognition on CMATERdb datasets.

3 Properties of Bengali Scripts

The Bengali alphabet was originated from an ancient Brahmi alphabet and it got its present form through various transformations. It is enriched with 11 vowels and 39 consonants. Unlike in the Roman alphabet, there is no case style (uppercase and lowercase) in Bengali, but it is read and written from left to right likewise. The set of vowels and

স্বর বর্ণ (Basic Vowel)										
অ	আ	ই	ঈ	উ	ঊ	ঋ	এ	ঐ	ও	ঔ
A	A,	I	I,	U	U,	R.	E	(AI)	O	(AU)
ব্যঞ্জন বর্ণ (Consonants)										
ক	খ	গ	ঘ	ঙ	চ	ছ	জ	ঝ	ঞ	ট
K	KH	G	GH	N,	C	CH	J	JH	N+	T,
ঠ	ড	ঢ	ণ	ত	থ	দ	ধ	ন	প	ফ
T,H	D,	D,H	N*	T	TH	D	DH	N	P	PH
ব	ভ	ম	য	র	ল	শ	ষ	স	হ	ড়
B	BH	M	J,	R	L	S.	S,	S	H.	R,
ঢ়	য়	ৎ	ং	ঃ	ঁ					
R*	Y	T.	M,	H,	N.					

(a)

(b)

(c)

Fig. 1: Example of printed basic characters including all Bengali vowels and consonants (a). Some samples of compound characters and modifiers of IUBCCdb and IUBMCdb dataset. The first seven rows represent compound characters and the last row indicates modifiers (b). Example of the different parts of a Bengali sentence (c).

consonants is called together the basic characters. The set of basic characters is shown in Fig. 1a.

All vowels (except the letter A is shown in Fig. 1a) following a consonant and several consonants may take modifiers (Fig. 1b). Some vowels, *e.g.*, "U," and "R.", may also take different shapes when used with consonants. In addition, two or more basic characters can be compounded together to construct a compound character (Fig. 1b).

Figure 1c showed that the shape of Bengali alphabet could be divided into three parts: i) upper part: the part above an upper horizontal line called "matra" (although matra does not exist for all characters); ii) lower part: the bottom-most part where modifiers are attached to the vowels and consonants; and iii) middle part: the part between upper and lower parts. Interestingly, modifiers do not deform the shapes of the middle part of a consonant while the shape of the middle part is deformed in the compound characters.

The matra is an important characteristic that is used to segment a character from the script and classify it. Some characters have an extended part above the matra, which makes it different from other similar characters. Even though matra has been reported as one of the most important features for character segmentation and recognition from printed texts [37], it results less reliable for HCR due to its inappropriate/partial presence in the scripts.

Another important property of some Bengali alphabet characters is that there are vertical lines at left or right side of the character. Some characters have double vertical lines as well. Similar to the matra, vertical lines were identified as important features for printed character recognition [15] but reduced their efficacy for HCR due to the variability of people's handwriting styles.

4 Data Description

In this study, we proposed two datasets, *e.g.*, i) set of modifiers and ii) set of compound characters, respectively. In addition, for the training of the classifiers, we considered three other datasets. The description of all datasets are given in the sections below.

4.1 Proposed Datasets

As we stated above, the frequency of usage of compound characters differs between each other. In this study, we intended to quantify the frequency of usage, as well as creating a dataset suitable for training an automatic classifier. We collected a total of $9,140$ samples of compound characters and $7,909$ samples of the modifiers from the handwriting of a convenience sample comprising of 130 individuals. The subjects involved in the handwriting were chosen from different professions of both males and females aged between 13 to 55 years.

Data were collected as follows. First, we considered the text scripts of two popular Bengali newspapers, *i.e.*, The Daily Prothom-Alo and The Daily Ittefaq in Bangladesh, for 15 days. Second, the news were segmented into different sections based on the news headings. The paper cuttings were distributed among 30 volunteers selected from both undergraduate and graduate programs of the Department of Computer Science and Engineering, Islamic University, Kushtia, Bangladesh. Third, the volunteers prepared a list of compound characters and modifiers found in the news cutting, and asked the writers to write the selected characters on a plain A4 white paper. The writers were selected from students of schools, colleges and university, shopkeepers, and officials of private and public offices. Fourth, these handwritten characters were scanned by a flatbed scanner at resolution 300×300 dpi at the Department of Computer Science and Engineering, Islamic University, Bangladesh. Finally, the individual characters were automatically segmented from the scanned images by a program written in Matlab 2015b, and saved as 256 color bmp file. We called these set of modifiers and compound characters as IUBMCdb (Islamic University Bengali Modifier Characters) and IUBCCdb (Islamic University Bengali Compound Characters), respectively. Writers wrote an average of 130 characters each. The datasets are available at: `https://github.com/Aktaruzzaman78/Handwritten-Bengali-Compound-Characters-and-Modifiers-Dataset`

4.2 Other Datasets

In addition to the proposed IUBMCdb and IUBCCdb dataset, we considered three other datasets of Bengali handwritings: i) the ISI Bengali handwritten dataset [9, 10]; ii) the CMATERdb dataset (we used the CMATERdb 3.1.1, CMATERdb 3.1.2, CMATERdb 3.1.3.3.7z and CMATERdb 3.1.4) [8, 16, 19] ; and iii) the BanglaLekha-Isolated dataset [12]. The ISI dataset consisted of about 23,392 samples of Bengali handwritten numerals and 37,858 basic characters collected from the handwritings of 1,106 subjects. The CMATERdb dataset consisted of approximately 67,063 samples among numerals, basic characters, modifiers, and compound characters. In addition to the basic characters and numerals, the CMATERdb included compound characters and modifiers as well. The BanglaLekha-Isolated dataset consisted of about 166,105 samples of handwritten isolated numerals, basic and compound characters. The ISI and CMATERdb datasets have been already splitted into training, validation and test sets by their curators. A brief description of the datasets has been summarized in Table 1.

5 Recognition Method

The methodological steps involved in the proposed classification (or recognition) method are described by the block diagram in Fig. 2. The input to the classifier was a symbol of handwritten Bengali character. The input could be a numeral, a basic character, a modifier or a compound character belong to the Bengali language. The input was an 8 bit 256

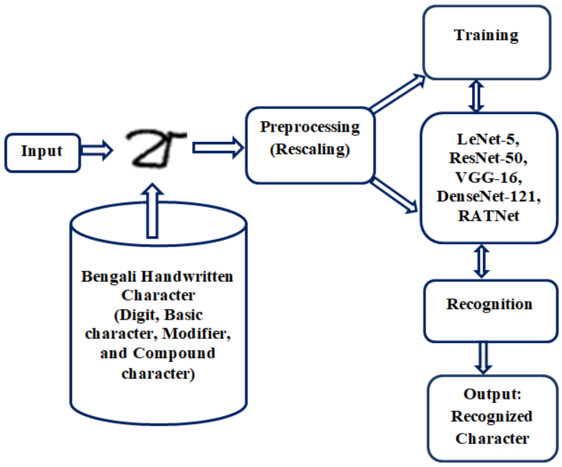


Fig. 2: Block diagram of the Bengali HCR system.

gray level image of any resolution. In this study, we have considered two resolutions of images (32×32 and 64×64). The rescaled image has been fed into a classification model based on the DL architecture described in the following sections.

5.1 Model Architecture

DL architecture has become an important part of computer vision research since the performance in an application depends on how well a network is designed. A wide range of network architectures has been proposed since the successful implementation of a CNN. For example, a Residual Network (ResNet) [23] makes possible to train up to hundreds or even more layers and provides compelling performance in many computer vision applications, object detection, and face recognition [21]. Increasing the depth of a network does not always work by simply stacking the layers together, and are hard to train due to vanishing gradient problem. The main idea behind ResNet is introducing identity shortcut connection between layers and skip one or more layers without sacrificing the network performance.

Most of the DL architectures target on three factors: depth, width, and cardinality. However, some researchers [31, 45, 47] have also recently investigated on another important concept called "attention mechanism" to further enhance the performance of CNNs, while reducing the complexity of the model. The importance of attention mechanism has been already described in several previous studies [7, 28] for object recognition. It does not only tell where is the focusing points of an image but also improves the region of interests. Inspired by this attention mechanism and recent advances in DL, we have proposed an architecture of CNN that employs both residual unit and attention mechanisms in this study. Here, we have termed the proposed model as a

Table 1: Statistics of the different datasets used in our study. The symbol – means information was not available.

Dataset	Category	Training samples	Validation samples	Testing samples	Total samples	Number of classes
ISI	Numeral	19,392	—	4,000	23,392	10
	Basic character	18,000	7,000	12,858	37,858	50
CMATERdb	Numeral	6,000	—	—	6,000	10
	Basic character	12,000	—	3,000	15,000	50
	Modifier	2,927	—	729	3,656	13
	Compound	34,439	—	8,520	42,959	171
BanglaLekha-Isolated	Numeral	19,748	—	—	19,748	10
	Basic character	98,950	—	—	98,950	50
	Compound	47,407	—	—	47,407	24
IUBMCdb	Modifier	7,909	—	—	7,909	13
IUBCCdb	Compound	9,140	—	—	9,140	87

residual attention network (RATNet). It is a Residual CNN with an attention mechanism that adopts a spatial attention mechanism in a very deep structure. The architecture of the model is shown in Fig. 3. A brief description of each component of the proposed RATNet model is given here:

5.1.1 *Convolutional Block and Dense Block*

We designed two blocks, *i.e.*, the convolutional and dense blocks, both of them are comprised of a set of layers. Each convolutional block comprised single convolutional layers with a kernel size of 3×3 and a different number of filters, batch normalization layer and ReLU activation functions. The dense block is defined as a trio of operations: fully connected layer (FC), batch normalization, and ReLU activation. Figure 3 shows the number of filters used in the convolutional layers and the dense blocks along with a 50% dropout layer.

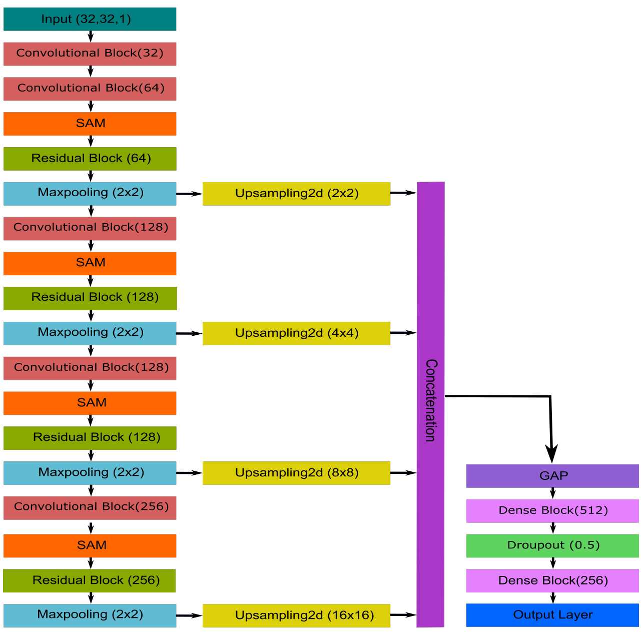


Fig. 3: The architecture of our proposed RATNet model.

5.1.2 *Spatial Attention Module*

The spatial attention module (SAM) mainly focuses on the dependence between spatial neighboring pixels and the noteworthy features on the input data. In other words, the SAM focuses on "where" is an informative part and "which" information is most relevant to an information part. In our network, we implemented a modified version of a specific spatial attention module called Convolutional Block Attention Module (CBAM) [47]. Briefly, CBAM is composed by two submodules, *i.e.*, the channel attention module and spatial attention module. In this study, we utilized only the spatial module to further reduce the complexity of the proposed model. The spatial attention module of CBAM worked as follows. First, the input tensor $X \in \mathbb{R}^{H\times W\times C}$ was manipulated along the channel dimension by computing the average and max-pooling operations, thus obtaining the two tensors X_{avg} and $X_{\text{max}} \in \mathbb{R}^{H\times W\times 1}$. Second, these two tensors have been concatenated to obtain the tensor $X_{\text{conc}} \in \mathbb{R}^{H\times W\times 2}$. Third, a standard convolutional layer with a mask 7×7 was applied to X_{conc} and a sigmoid function σ applied element-wise. Finally, the output tensor $\text{SAM}(X)$ was $\in \mathbb{R}^{H\times W\times 1}$. The equation was as follows:

$$\text{SAM}(X) = \sigma(f^{7\times7}([\text{AvgPool}(X);\text{MaxPool}(X)])) \tag{1}$$

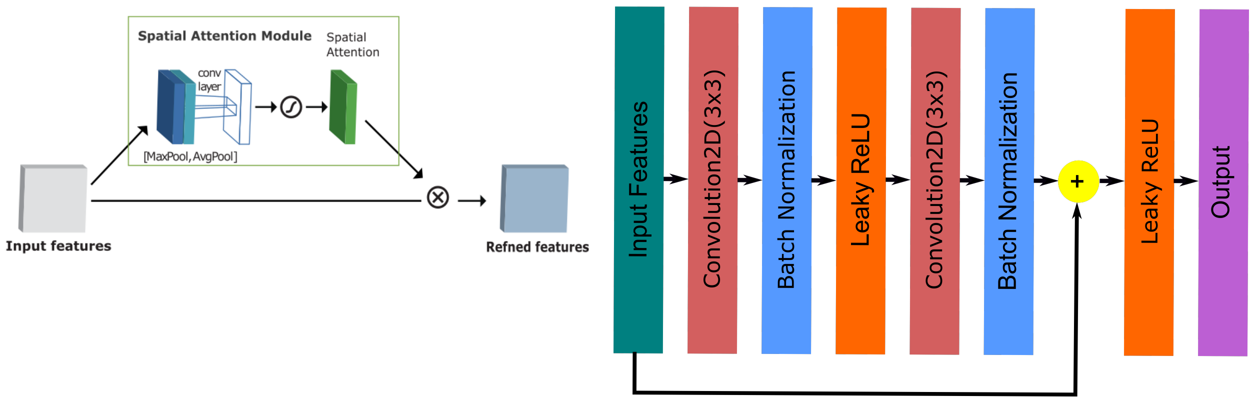
A schematic diagram of the SAM is shown in Fig. 4a.

5.1.3 *Residual Block*

In our work, we implemented residual blocks and skip connections as follows:

$$R(x) = \text{LeakyReLU}(x + f(x)) \tag{2}$$

where x is the input tensor, $f(x)$ is the residual component, and $R(x)$ is the output of the residual block. In this block,



(a) (b)

Fig. 4: Schematic diagrams of both spatial attention module (a) and residual block (b), respectively.

$f(x)$ was defined as two consecutive copies of a trio of operations: convolution with a filter of size 3×3, batch normalization and a LeakyReLU activation function (LeakyReLU activation was not used in second copy). Then, the feature map from $f(x)$ was added to the input x. Finally, the LeakyReLU activation function is performed on concatenated features. The whole block of $R(x)$ is referred to as a residual block. The schematic diagram of the residual block is shown in Fig. 4b.

5.1.4 Hypercolumn Technique and Global Average Pooling

The typical network architecture using CNN involves the use of a FC layer dedicated to the classification task. However, due to sequences of max-pooling, the information in the last FC layer may be too coarse spatially to allow precise localization. Besides, the first layers may be spatially precise, but these include a lack of semantic information. To trade-off between spatial precision and amount of semantic information, the Hypercolumn Technique (HT) has been implemented [22]. HT comprises of upsampling with bilinear interpolation and concatenation layers. The bilinear upsampling determines the values of new neighboring pixels by using all nearby pixel values. In order to select important features, hypercolumn features feed to a global average pooling (GAP) layer. The usage of GAP gives better robustness to the network while maintaining high sensitivity to texture information in images [34]. Besides, the GAP summing out the activations of every feature make the network learn to be invariant to those spatial transformations. We feed the spatial pooling of features maps to the dense block.

5.2 Training, evaluation and experiments

The training and evaluation of the model performance were performed by means of a hold-out strategy. Depending on the dataset, we employed the training and test sets already in place (see Table 1), or in case on no split, we partitioned the entire dataset into train and test sets by the random selection of samples at the ratio of 70/30.

After defining the model hyperparameters (see sec. 5.4), the model was trained using the training data of a dataset. Then, its performance was evaluated on the test data of that dataset. The performance of the model was evaluated for each type of sample of each dataset, *i.e.*, digit, basic characters, modifiers and compound characters, respectively, hereafter called individual performance.

The performance was also evaluated for the case when all types of data of a specific dataset were combined together, hereafter defined as combined performance. In determining combined performance, the training samples of all categories were combined to form the train set, and the set samples were combined to form the test set.

We performed several experiments to i) assess the performance of the model; ii) to quantify the effect of the image resolution on the performance; iii) compare performance with those of other state-of-the-art architectures; and iv) assess whether the spatial attention mechanism was relevant for the task.

For assessing the individual performance, the output layer of our model was adjusted based on the classification task to perform (*e.g.*, 10 neurons for digit recognition). The model was trained with input images whose resolution was 32×32 and 64×64 pixels to assess the effect of a higher resolution on the performance. In addition, we quantified the difference in performance with and without the attention mechanism in our model, hereafter called RATNet-No-SAM-RB. The performance was quantified on each dataset separately. In a similar manner, the combined performance was quantified by adjusting the number output neurons. The other network architectures we considered were: LeNet-5 [26], VGG-16 [43], ResNet-50 [23], and DenseNet-121 [24]. Even in this case, the input and output layers of all models were adjusted depending on the classification task. The training for all models was performed from scratch using a random initialization of the weights.

The experiments were carried out on a desktop computer with Intel Core i7 3.90 GHz CPU and NVIDIA Titan XP Pro GTX1080Ti 12 GB GPU, 1TB HDD, and 8 GB RAM.

5.3 Evaluation metrics

We computed three metrics for the evaluation of the recognition model. First, we considered the accuracy, computed as the percentage of correctly classified cases among the total test cases. This metric is known to be dependent on the distribution of the amount of samples in each class within the dataset. Consequently, when dealing with classification problems with unbalanced datasets (as in our case), the accuracy might not be the most suitable indicator of performance, especially for comparisons with other studies involving a different dataset. So, to account for this problem, as a second metric, we considered the F1 score. Third, we considered the reliability metric Cohen's Kappa (K) [27] to account for how much agreement between the actual and predicted values of a class could be occurred by chance. The value of K ranges from 0 to 1. The closer K to 1, the more reliable the classifier is.

5.4 Hyperparameters and Loss Function

The values of hyperparameters in machine learning are used to control the learning process. In this study, the values of hyperparameters were set using the training and validation sets of the ISI dataset (see Table 1). It is worth noting that this dataset contained only numerals and basic characters. We took the decision of employing only this dataset to reduce the risk of overfitting from one side (without specializing on the entire set of characters), and on the other side, to reduce the computational time required for their setting. The set of hyperparameters used in the proposed model were: i) the number of epochs n, ii) batch size b, iii) learning rate α, and v) optimization and loss function.

To set the values of these hyperparameters, we initialized the number of epochs $n = 200$ with batch size $b = 128$. An early stopping callback on the validation loss with the patience of 30 epochs was applied to halt the training process when no improvements were detected. Initially, we used the learning rate $\alpha = 0.001$ and then it was updated to 75% of its previous value, if our model validation accuracy did not improve for six consecutive epochs. Finally, the values of learning rate α and number of epochs n for which the model provided maximum accuracy on the validation set of ISI dataset were 0.0000422 and 142, respectively. The Adam optimizer [25] was used for the optimization of model parameters with categorical cross entropy as loss function.

6 Results

We have developed a dataset for handwritten Bengali modifiers and compound characters. There are 107 distinct compound characters found in our study. They appeared with different frequencies. Some of them have been used very rarely ($< 3\%$), and some are used with high frequency ($\geq 97\%$). There are 87 frequently used compound characters found in our study (selected when its frequency was $\geq 5\%$). The frequency of each character is reported in Fig. 5.

The performance of the proposed model has been studied at two different resolutions: 32×32, and 64×64 pixels. The model achieved slightly more accuracy of 97.74% at resolution of 64×64 than 97.64% at resolution of 32×32 when considering the ISI dataset for basic character recognition. Given the very limited improvement in using a higher resolution and considering that most of the works reported in the literature (*e.g.*, [5, 36]) have performed their experiments at resolution of 32×32, to facilitate the comparisons,

CC	IF(%)	CC	IF(%)	CC	IF(%)	CC	IF(%)	CC	IF(%)	CC	IF(%)
ক্ষ	97.02	ঙ্ক	43.06	ব্দ	24.02	ম্ম	18.00	হ্ন	9.91	হ্ব	4.01
ন্দ	96.01	ন্ড	41.70	শ্ব	24.00	প্ন	17.10	হ্ল	9.80	ঙ্ক্ষ	3.89
চ্ছ	85.04	ন্ধ	41.00	প্ত	22.90	ন্ট	17.10	ল্ক	9.56	জ্ম	3.89
ন্ত	81.02	স্প	40.90	ক্ল	22.50	স্ন	16.20	ত্ন	8.95	ক্ন	3.50
ন্ধ	70.02	স্ক	40.20	চ্চ	22.40	ন্ম	16.16	স্ল	8.51	দগ	3.00
স্থ	66.08	ক্ট	40.00	দব	21.75	ঙ্খ	16.03	শ্ল	8.00	ঙ্গ	2.96
স্ত	63.50	ত্ব	39.91	প্ল	21.16	ত্ত্ব	16.00	ব্ধ	7.98	ম্ব	2.45
স্ট	61.03	ঞ্জ	39.17	ষ্ঠ	21.00	ঙক্ষ	15.90	ল্ড	7.76	জ্জ্ব	2.30
স্ব	60.33	শ্চ	31.50	হ্ল	20.68	জ্জ	15.50	শ্ম	7.75	ট্ট	2.21
ত্ত	60.13	ত্থ	30.03	ল্ট	20.12	ঞ্চ	15.08	গ্ধ	7.50	ঙ্গ	2.16
ম্ল	54.20	ঞ্চ	28.56	ষ্প	20.10	ক্ক	14.17	ন্দ্ব	7.00	জ্ঞ	2.08
ন্ট	51.60	ল্ব	28.50	ব্ল	20.09	নব	12.36	ষ্ম	6.96	ক্র	2.00
ষ্ট	49.00	স্প	28.01	ষ্ক	20.04	ডড	12.14	ম্ন	6.91	ঙন	1.19
ঙ্গ	45.50	ন্স	27.04	জ্ম	19.50	গ্ন	11.78	ষ্ফ	5.98	ল্ল	1.15
ন্ন	45.10	ঙ্ক্ষ	26.00	দ্দ	19.50	ব্ব	11.04	ঙঘ	5.80	শছ	1.10
ক্ষ্ম	44.80	শ্ন	25.00	ম্ফ	19.00	ন্ঠ	10.80	ক্ষন	4.45	ন্স	1.10
ক্ত	44.00	প্ন	24.33	জ্ব	18.90	ল্ন	10.32	ধ্ব	4.20	ট্র	0.90
দ্ধ	43.20	ন্ত	24.22	ত্ম	19.60	প্ট	10.05	ফ্ল	4.17		
CC: Compound Character, IF: Individual Frequency											

Fig. 5: Compound characters and their individual frequency of occurrence found in the newspapers.

Table 2: Recognition performance of different models on the ISI dataset. The bold faced values denote the best performance.

Model	Metrics (%)	Digit	Basic
LeNet-5	Accuracy	98.50	94.33
	F1-score	98.46	93.03
	Cohan's Kappa	98.35	93.13
VGG-16	Accuracy	99.62	95.87
	F1-score	99.60	95.86
	Cohan's Kappa	99.51	94.90
ResNet-50	Accuracy	99.13	95.93
	F1-score	99.09	95.85
	Cohan's Kappa	98.00	95.75
DenseNet-121	Accuracy	99.55	95.80
	F1-score	99.55	95.75
	Cohan's Kappa	99.41	95.70
RATNet-No-SAM-RB	Accuracy	99.02	94.94
	F1-score	99.02	94.87
	Cohan's Kappa	98.90	94.64
RATNet	Accuracy	**99.80**	**97.64**
	F1-score	**99.80**	**97.63**
	Cohan's Kappa	**99.70**	**97.55**

Table 3: Results on the BanglaLekha-Isolated dataset for different models. The bold faced values denote the best performance.

Model	Metrics (%)	Digit	Basic	Compound
LeNet-5	Accuracy	98.38	93.14	90.58
	F1-score	98.36	93.12	90.31
	Cohen's Kappa	98.20	93.05	90.29
VGG-16	Accuracy	98.65	94.94	93.18
	F1-score	98.64	94.93	**93.86**
	Cohen's Kappa	98.42	94.70	93.05
ResNet-50	Accuracy	98.71	95.36	93.41
	F1-score	98.70	**95.94**	93.39
	Cohen's Kappa	98.55	95.08	93.29
DenseNet-121	Accuracy	98.72	95.60	93.52
	F1-score	98.71	95.58	93.75
	Cohen's Kappa	98.61	95.42	**93.70**
RATNet-No-SAM-RB	Accuracy	98.24	93.80	92.24
	F1-score	98.24	93.88	92.22
	Cohen's Kappa	98.05	93.74	91.90
RATNet	Accuracy	**98.92**	**95.73**	**93.74**
	F1-score	**98.92**	95.60	93.72
	Cohen's Kappa	**98.81**	**95.64**	93.47

we report the results of the rest of the study considering only images at a resolution of 32×32.

We now describe the results obtained on the selected datasets. Table 2 reports the results achieved by all selected models on the ISI dataset. Our model performed similarly to the other architectures for digit recognition (all accuracies were $> 99\%$ but LeNet-5's) while it outperformed for basic character identification of approximately 2%. The individual performance of the model for recognition of numerals, basic and compound characters on the BanglaLekha-Isolated dataset is reported in Table 3. All models achieved

Table 4: Results on the CMATERdb dataset for different models. The bold faced values denote the best performance.

Model	Metrics (%)	Digit	Basic	Modifier	Compound
LeNet-5	Accuracy	98.72	97.10	98.40	91.80
	F1-score	98.70	97.08	98.37	91.77
	Cohen's kappa	98.20	97.00	98.20	91.25
VGG-16	Accuracy	99.16	97.56	98.63	93.10
	F1-score	99.15	97.50	98.60	93.09
	Cohen's Kappa	99.05	96.62	98.50	92.85
ResNet-50	Accuracy	99.27	97.40	98.37	92.28
	F1-score	99.24	97.38	98.36	92.27
	Cohen's Kappa	99.10	96.42	98.15	92.11
DenseNet-121	Accuracy	98.90	98.83	98.63	96.31
	F1-score	98.88	98.81	98.62	96.29
	Cohen's Kappa	98.61	98.67	98.50	95.70
RATNet-No-SAM-RB	Accuracy	98.94	97.20	98.01	95.75
	F1-score	98.91	97.15	97.75	95.05
	Cohen's Kappa	98.82	97.00	97.85	95.72
RATNet	Accuracy	**99.66**	**99.27**	**98.78**	**97.70**
	F1-score	**99.61**	**98.94**	**98.71**	**97.67**
	Cohen's Kappa	**99.60**	**98.86**	**98.50**	**97.61**

Table 5: Results on the IUBMCdb and IUBCCdb dataset for different models. The bold faced values denote the best performance.

Model	Metrics (%)	Modifier	Compound
LeNet-5	Accuracy	96.21	81.47
	F1-score	96.19	81.44
	Cohen's kappa	96.13	81.26
VGG-16	Accuracy	96.15	93.27
	F1-score	96.12	93.25
	Cohen's Kappa	96.02	90.10
ResNet-50	Accuracy	97.10	90.83
	F1-score	97.06	90.72
	Cohen's Kappa	96.95	90.52
DenseNet-121	Accuracy	96.63	92.91
	F1-score	96.61	92.90
	Cohen's Kappa	96.54	92.72
RATNet-No-SAM-RB	Accuracy	96.89	90.83
	F1-score	96.95	89.85
	Cohen's Kappa	96.60	90.05
RATNet	Accuracy	**97.41**	**93.42**
	F1-score	**97.41**	**93.41**
	Cohen's Kappa	**97.29**	**93.33**

comparable performance for the recognition of numerals (accuracies $> 98\%$). The worst performance across all models was observed for the recognition of compound characters with Accuracies ranging from 90% (LeNet-5) to 93% (DenseNet-121 and our model). In particular, our model achieved an Accuracy of 93.74%, F1-score of 93.72%, and Cohen's Kappa of 93.47%. The closest performance was achieved by the DenseNet-121, which is a more complex model with a very large set of parameters, as reported in Table 7. Similar results were observed in the CMATERdb dataset and in the newly created IUBMCdb and IUBCCdb, as reported in Table 4 and 5. The highest and lowest performances were achieved for digit and compound character recognition, respectively, across all models.

The role of the spatial attention module and residual block on the classification performance was assessed by re-

Table 6: Results on the combined performance achieved by our model on each dataset.

Dataset	Metrics		
	Accuracy	F1-score	Cohen's Kappa
ISI (Digit, Basic)	97.79	97.17	95.63
CMATERdb (Digit, Basic, Modifiers, Compound)	96.94	96.74	94.70
BanglaLekha-Isolated (Digit, Basic, Compound)	95.10	94.90	93.10
IUBMCdb and IUBCCdb (Modifier, Compound)	92.91	92.49	91.17

Table 7: Number of parameters of each model.

Model	Trainable parameters	Non-trainable parameters	Total parameters
LeNet-5	2,645,791	0	2,645,791
VGG-16	40,588,779	640	40,589,419
ResNet-50	4,145,579	14,464	4,160,043
DenseNet-121	7,122,859	83,648	7,206,507
RATNet-No-SAM-RB	844,011	1,728	845,739
RATNet	2,889,203	5,056	2,894,259

moving these two components and repeating the training phase. The removal of these two blocks was sensible on basic character, modifier and compound character recognition, with a performance consistently lower than $2-3\%$ with respect to the original model, across all datasets.

Finally, the performances achieved by our model using the combined setting are reported in 6. As expected, the lowest performance was obtained on the proposed datasets, *i.e.*, IUBMCdb and IUBCCdb, containing only modifiers and compound characters, while the best ones were achieved on the ISI dataset.

7 Discussions

The close observation of the performance achieved by different models presented from Tables 2 to 5 on different datasets reveals that every model provided similar performance for the recognition of numerals and modifiers on each dataset (Table 8). The range of maximum difference in accuracy across all datasets and models was $0.5-1.3\%$ for numerals and $0.4-1.3\%$ for modifiers, suggesting that all models were equally good in these recognition tasks. On the other hand, such ranges varied for the recognition of basic and compound characters. In particular, ranges were $2.2-3.3\%$ and $3.2-12\%$ for basic character and compound character recognition (Table 8), respectively. In this case, models did not perform equally for these two recognition tasks. For example, on the proposed dataset (Table 5), the lowest accuracy was 81.47% and achieved by LeNet-5, while the best accuracy was 93.42% by our model, obtaining a gap of 12%.

Table 8: Range of accuracy in percentage for each dataset and character type. Non-available information is marked with $-$.

Dataset	Digit	Basic	Modifier	Compound
ISI	1.3	3.3	–	–
CMATERdb	0.9	2.2	0.4	5.9
BanglaLekha-Isolated	0.5	2.6	–	3.2
IUBMCdb/IUBCCdb	–	–	1.3	12

The model that consistently reported the lowest accuracy across all datasets was LeNet-5. The lower performance obtained with respect to the other models was visible for both basic and compound character recognition. Two possible explanations might be possible. First, the number of parameters of the model was the lowest among all models (excluding RATNet-No-SAM-RB). Second, the heterogeneity of the layers composing the network was also very scarce in this model. These two factors might have compromised LeNet-5 in achieving similar model performance, lacking in capturing the variability of the characters within these two sets. On the other hand, the proposed model ranked first in all datasets and character type, with a few exceptions depending on the metric considered. Given the fact that the model had a low number of parameters (second-last), we believe that the good performance was due, on a certain extent, to the high heterogeneity of the layers within the model. In other words, the opposite reason behind the compromised performance of LeNet-5 was likely the one that made achieved the best. In fact, RATNet is composed by a CNN, spatial attention module, residual block and hypercolumn technique. It is therefore reasonable that building a model with combinations of different mathematical functions, rather than repeating the same layers in cascade, may allow a better feature representations while keeping the number of parameters relatively low.

Another crucial point for Bengali HCR is the number of compound characters. The authors of [12] reported on BanglaLekha-Isolated have considered only a set of 24 compound characters. On the other hand, the authors of [19] proposed an alphabet of 171 compound characters, which is about two times the class of compound characters we proposed in this study. Differently from their study, we have neglected some cases where the same symbol is written in different forms, *e.g.*, (Anurup) and (rudro). In the first word, r+u is written in – form, but the same character is written differently in the second letter. The authors [19] have considered them as distinct classes. Besides this, the size of the compound characters of CMATERdb is also larger than our proposed dataset. This is because they have considered the combination of a basic character with one or more basic characters or modifiers as the combined characters. But in our definition, even in other works [12], we did not consider

Table 9: Comparative accuracy of the proposed method with the existing best methods on the same dataset.

Study	Methodology	Data set	Accuracy (%)			
			Digit	Basic	Modifier	Compound
Shopon *et al.* [42]	DCNN	ISI	99.35	—	—	—
Bhattacharya *et al.* [10]	MQDF + MLP	ISI	—	95.84	—	—
Fardous *et al.* [20]	CNN	CMATERdb	—	—	—	95.50
Alom *et al.* [5]	DCNN	CMATERdb	99.13	98.31	98.18	—
Rabby *et al.* [33]	CNN	CMATERdb	—	98.00	—	—
		ISI	—	96.81	—	—
		BanglaLekha-Isolated	—	95.71	—	—
Proposed model	**RATNet**	**CMATERdb**	**99.66**	**99.27**	**98.78**	**97.70**
		ISI	**99.80**	**97.64**	—	—
		BanglaLekha-Isolated	**98.92**	**95.73**	—	**93.74**

the combination of modifiers with basic characters in the definition of combined characters.

In real life, the characters and numerals sometimes are used in a mixed form, *e.g.*, the number plate of a car, postal address, bank check, *etc*. Thus, the performance of the proposed model was also evaluated when both characters and numerals were combined together of each dataset. The best combined performance was obtained by our model in the ISI dataset (Accuracy: 97.79%, F1-score: 97.17%; Table 6). This result is reasonable because the ISI dataset contains only the basic characters and numerals only, and the individual performance of the model for these symbols is also better than those in other datasets. The worst-case was reported for the proposed dataset, which also reasonable because the dataset contains only modifiers and compound characters, and the individual performances of the model for these modifiers and compound characters were also worse for this dataset than other others.

Finally, the performance of the proposed RATNet model and those of other exiting methods, determined on the same datasets used in this study, has been summarized in Table 9. These methods reported the highest accuracy currently available in the scientific literature for Bengali HCR. The proposed RATNet has outperformed the accuracy of these methods on each dataset. To the best of our knowledge, the highest accuracy (95.50%) has been reported by Fardous *et al.* [20] in their work for recognition of compound characters in the CMATERdb dataset. Our proposed method has improved its recognition accuracy by more than 2%. However, it is worth acknowledging that without having access to the trained models, it is difficult to derive conclusions on the performance reported, although the performance achieved by our model looks promising.

8 Conclusions

We have proposed a DL architecture of convolutional neural network which includes both residual unit and spatial attention mechanism. The recognition performance of the proposed model and some already existing models have been evaluated on the existing and proposed datasets of handwritten Bengali compound characters and modifiers. The proposed recognition model outperformed all existing models for the recognition of handwritten Bengali characters (numerals, basic characters, modifiers, and compound characters) with relatively low number of parameters. However, it is worth mentioning that transfer learning may be applied too, thus likely reducing substantially the number of parameters of the model while keeping the same level of performance. We leave the comparison of the proposed algorithm and transfer learning of other DL architectures for future investigations.

In this study, in addition, we have collected a set of compound characters of 107 distinct classes, and only 87 characters out of them have been selected as the most frequently (frequency $\geq 5\%$) compound characters. The performance of the proposed model has been evaluated for recognition of isolated characters only. To develop a complete OCR, we need an appropriate segmentation algorithm for segmenting characters from unconstrained handwritten script. The major challenge of dealing with unconstrained characters recognition is the segmentation of characters with overlapping. In future, we want to propose a method for segmenting handwritten characters from unconstrained script, and the performance of the proposed RATNet model will be evaluated on these characters.

Acknowledgements We thankfully acknowledge the contribution of those volunteers, who participated in writing the characters. We are grateful to Prof. Dr. Ujjal Bhattacharya, Indian Statistical Institute for providing the ISI handwritten numeral dataset. We are also grateful to the University Grants Commission (UGC) of Bangladesh and the authority of Islamic University, Kushtia-7003, Bangldesh for providing partial financial support to develop this dataset.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Akhand M, Rahman MM, Shill P, Islam S, Rahman MH (2015) Bangla handwritten numeral recognition using convolutional neural network. In: Proceeding of the International Conference on Electrical Engineering and Information Communication Technology, Dhaka, Bangladesh, pp 1–5
2. Akhand MAH, Ahmed M, Rahman MMH (2016) Multiple convolutional neural network training for Bangla handwritten numeral recognition. In: Procedding of the International Conference on Computer and Communication Engineering, San Francisco, Canada, pp 311–315
3. Ali AAA, Suresha M (2019) A novel features and classifiers fusion technique for recognition of arabic handwritten character script. SN Applied Sciences 1(10):1286
4. Alom MZ, Sidike P, Taha TM, Asari VK (2017) Handwritten Bangla digit recognition using deep learning. arXiv:170502680
5. Alom MZ, Sidike P, Hasan M, Taha TM, Asari VK (2018) Handwritten Bangla character recognition using the state-of-the-art deep convolutional neural networks. Comput Intel Neurosc abs/1712.09872:315–320
6. Amin MS, Yasir SM, Ahn H (2020) Recognition of pashto handwritten characters based on deep learning. Sensors 20(20):5884
7. Ba J, Mnih V, Kavukcuoglu K (2014) Multiple object recognition with visual attention. arXiv:14127755
8. Basu S, Das N, Sarkar R, Kundu M, Nasipuri M, Basu DK (2009) A hierarchical approach to recognition of handwritten Bangla characters. Pattern Recognit 42(7):1467–1484
9. Bhattacharya U, Chaudhuri BB (2009) Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals. IEEE Trans Pattern Anal Mach Intell 31(3):444–457
10. Bhattacharya U, Shridhar M, Parui SK, Sen P, Chaudhuri B (2012) Offline recognition of handwritten Bangla characters: An efficient two-stage approach. Pattern Anal Appl 15(4):445–458
11. Bhowmik TK, Ghanty P, Roy A, Parui SK (2009) SVM-based hierarchical architectures for handwritten Bangla character recognition. Int J Doc Anal Recogn 12(2):97–108
12. Biswas M, Islam R, Shom GK, Shopon M, Mohammed N, Momen S, Abedin MA (2017) Banglalekha-isolated: A comprehensive Bangla handwritten character dataset. arXiv:170310661
13. Breuel TM, Ul-Hasan A, Al-Azawi MA, Shafait F (2013) High-performance ocr for printed english and fraktur using lstm networks. In: 2013 12th International Conference on Document Analysis and Recognition, pp 683–687
14. Chatterji SK (2002) The origin and development of the Bengali language. Indogermanische Forschungen 47(1):370–380
15. Chaudhuri B, Pal U (1998) A complete printed Bangla OCR system. Pattern Recognit 31(5):531–549
16. Das, Nibaran, Sarkar, Ram, Basu, Subhadip, Kundu, Mahantapas, Nasipuri, Mita, Basu, Kumar D (2012) A genetic algorithm based region sampling for selection of local features in handwritten digit recognition application. Appl Soft Comput 12(5):1592–1606
17. Das N, Basu S, Sarkar R, Kundu M, Nasipuri M, Basu D (2009) Handwritten bangla compound character recognition: Potential challenges and probable solution. In: Procedding of the International Joint Conference on Artificial Intelligence, pp 1901–1913
18. Das N, Das B, Sarkar R, Basu S, Kundu M, Nasipuri M (2010) Handwritten Bangla basic and compund characters recogntion using MLP and SVM classifiers. Journal of computing 2(2):109–115
19. Das N, Acharya K, Sarkar R, Basu S, Kundu M, Nasipuri M (2014) A benchmark image database of isolated Bangla handwritten compound characters. Int J Doc Anal Recogn 17:413–431, DOI 10.1007/s10032-014-0222-y
20. Fardous A, Afroge S (2019) Handwritten isolated Bangla compound character recognition. In: Procedding of the International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox'sBazar, Bangladesh, pp 1–5
21. Gruber I, Hlaváč M, Železný M, Karpov A (2017) Facing face recognition with Resnet: round one. In: Procedding of the International Conference on Interactive Collaborative Robotics, Hatfield, Hertfordshire, UK, pp 67–74
22. Hariharan B, Arbeláez P, Girshick R, Malik J (2015) Hypercolumns for object segmentation and fine-grained localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Boston, Massachusetts, pp 447–456
23. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. CoRR abs/1512.03385, URL `http://arxiv.org/abs/1512.03385`
24. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, HI, USA, pp 4700–4708
25. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv:14126980
26. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324

27. McHugh ML (2012) Interrater reliability: The kappa statistic. Biochemia medica 22(3):276–282
28. Mnih V, Heess N, Graves A, et al. (2014) Recurrent models of visual attention. In: Procedding of the Advances in neural information processing systems, Montreal Convention Centre, Canada, pp 2204–2212
29. Modhej N, Bastanfard A, Teshnehlab M, Raiesdana S (2020) Pattern separation network based on the hippocampus activity for handwritten recognition. IEEE Access 8:212803–212817
30. Pal U, Wakabayashi T, Kimura F (2007) Handwritten Bangla compound character recognition using gradient feature. In: Procedding of the International conference on information technology (ICIT), Orissa, India, IEEE, pp 208–213
31. Park J, Woo S, Lee JY, Kweon IS (2018) BAM: Bottleneck attention module. arXiv:180706514
32. Purnamawati S, Rachmawati D, Lumanauw G, Rahmat R, Taqyuddin R (2018) Korean letter handwritten recognition using deep convolutional neural network on android platform. In: Journal of Physics: Conference Series, vol 978
33. Rabby ASA, Haque S, Islam S, Abujar S, Hossain SA (2018) Bornonet: Bangla handwritten characters recognition using convolutional neural network. Procedia Comput Sci 143:528–535
34. Racki D, Tomazevic D, Skocaj D (2018) A compact convolutional neural network for textured surface anomaly detection. In: Procedding of the IEEE Winter Conference on Applications of Computer Vision, Lake Tahoe, NV, USA, pp 1331–1339
35. Rahman MM, Akhand MAH, Islam S, Shill PC, Rahman MMH, et al. (2015) Bangla handwritten character recognition using convolutional neural network. International Journal of Image, Graphics and Signal Processing 7(8):42–49
36. Rahman MM, Islam MS, Sassi R, Aktaruzzaman M (2019) Convolutional neural networks performance comparison for handwritten Bengali numerals recognition. SN Appl Sci 1(12):1660
37. Rownak AF, Rabby MF, Ismail S, Islam MS (2016) An efficient way for segmentation of bangla characters in printed document using curved scanning. In: 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV), pp 938–943, DOI 10.1109/ICIEV.2016.7760138
38. Roy S, Das N, Kundu M, Nasipuri M (2017) Handwritten isolated Bangla compound character recognition: A new benchmark using a novel deep learning approach. Pattern Recognit Lett 90:15–21
39. Saufi MM, Zamanhuri MA, Mohammad N, Ibrahim Z (2018) Deep learning for roman handwritten character recognition. International Journal of Electrical Engineering and Computer Science 12(2):455–460
40. Sazal MMR, Biswas SK, Amin MF, Murase K (2014) Bangla handwritten character recognition using deep belief network. In: Procedding of the International Conference on Electrical Information and Communication Technology, Dhaka, Bangladesh, pp 1–5
41. Shopon M, Mohammed N, Abedin MA (2016) Bangla handwritten digit recognition using autoencoder and deep convolutional neural network. In: Procedding of the International Workshop on Computational Intelligence (IWCI), pp 64–68
42. Shopon M, Mohammed N, Abedin MA (2017) Image augmentation by blocky artifact in deep convolutional neural network for handwritten digit recognition. In: Procedding of the IEEE International Conference on Imaging, Vision & Pattern Recognition, Himeji, Japan, pp 1–6
43. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:14091556
44. Tsai C (2016) Recognizing handwritten japanese characters using deep convolutional neural networks. University of Stanford in Stanford, California pp 405–410
45. Wang F, Jiang M, Qian C, Yang S, Li C, Zhang H, Wang X, Tang X (2017) Residual attention network for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, pp 3156–3164
46. Wikipedia contributors (2020) List of languages by number of native speakers — Wikipedia, the free encyclopedia. URL `https://en.wikipedia.org/w/index.php?title=List_of_languages_by_number_of_native_speakers&oldid=978856845`, [Online; accessed 9-October-2020]
47. Woo S, Park J, Lee JY, So Kweon I (2018) CBAM: Convolutional block attention module. In: Proceedings of the European Conference on Computer Vision, Munich, Germany., pp 3–19
48. Zhang Y (2015) Deep convolutional network for handwritten chinese character recognition. Computer Science Department, Stanford University