State-of-the-Art Bangla Handwritten Character Recognition Using a Modified Resnet-34 Architecture

Mujadded Al Rabbani Alif
Department of Computer Science
Huddersfield University
Huddersfield, HD1 3DH, United Kingdom

Abstract:- Bangla Handwritten Character Recognition (HCR) remains a persistent challenge within the domain of Optical Character Recognition (OCR) systems. Despite extensive research efforts spanning several decades, achieving satisfactory success in this field has proven to be complicated. Bangla, being one of the most widely spoken languages worldwide, consists of 50 primary characters, including 11 vowels and 39 consonants. Unlike Latin languages, Bangla characters exhibit complex patterns, diverse sizes, significant variations, intricate letter shapes, and intricate edges. These characteristics further differ based on factors such as the writer's age and birthplace. In this paper, we propose a modified ResNet-34 architecture, a convolutional neural network (CNN) model, to identify Bangla handwritten characters accurately. The proposed approach is validated using a merged subset of two popular Bangla handwritten datasets. Through our technique, we achieve state-of-the-art recognition performance. Experimental results demonstrate that the suggested model attains an average accuracy of 98.70% for Bangla handwritten vowels, 97.34% for consonants, and 99.02% for numeric characters. Additionally, when applied to a mixed dataset comprising vowels, consonants, and numeric characters, the proposed model achieves an overall accuracy of 97%. This research contributes to advancing digital manufacturing systems by addressing the challenge of Bangla Handwritten Character Recognition, offering a high-performing solution based on a modified ResNet-34 architecture. The achieved recognition accuracy signifies significant progress in this field, potentially paving the way for enhanced automation and efficiency in various applications that involve processing Bangla handwritten text.

Keywords:- *Handwritten Character Recognition; ResNet; Optical Character Recognition; Computer Vision; Convolutional Neural Networks.*

I. INTRODUCTION

In this modern era of digitisation, the importance of Handwritten Character Recognition (HCR) is rapidly increasing. From the recognition of text on paper-based documents to the automatic manufacturing recite reading, much of the automation of tasks depends on using HCR. Moreover, with the improvement of technologies, researchers are trying to implement HCR algorithms that recognise characters of important languages in any medium. Bangla is one such language. It is the mother tongue and the official language of Bangladesh. Moreover, in a vast subcontinent such as India, it is the second most popular language [1]. Bangla also ranks fifth among the world's most popular languages, with about 234 million speakers. The language comprises 11 vowels, 39 consonants, and over 334 compound characters, formed mainly by combining two to three primary characters [2]. With this massive usage of the Bangle language, the need for and uses of Bangla HCR are considerably more significant in the digital era. However, to achieve fully automated Bangla HCR, we need advanced algorithms and methodologies where researchers face challenges because of the unique pattern of the Bangla alphabet and insufficient research on this subject.

Computer vision has become increasingly prevalent across various industries, including healthcare [3, 4], renewable energy [5], and quality inspection [6]. In recent years, the advancement of modern machine learning algorithms and deep learning recognition algorithms, such as image recognition, natural language processing, etc., has flourished and paved the way for new research, resulting in advancement in handwritten image recognition. Deep learning is a set of machine learning methods that work with multi-level representations where the inputs are transformed and learned through different complex functions [7]. It has gained incredible success in computer vision with the evolution of graphical processing units (GPU) and large-scale neural network models called convolutional networks (CNN). For example, Deep CNNs (DCNNs) like AlexNet [8], GoogLeNet [9], ResNet [10], and DenseNet [11] had tremendous success in the ImageNet Large Scale Visual Recognition Challenge for detecting and classifying an object in photographs from 2012 [12]. Additionally, CNN, like YOLOv2 and YOLOv3, succeeded in object detection and is also a remarkable achievement [13] in the sector of deep learning. If we dive into the core mechanism, the CNN model is a class of artificial neural networks (ANN) made of a series of filters that try to detect image features. Therefore, it does not require manual engineering feature detectors like previous algorithms such as the "Histogram of Oriented Gradients" [14], SIFT [15], or Bag of Visual Words(BoVW) [16]. Nevertheless, the success rate of these models requires a high volume of labelled training datasets to learn to extract the feature and classify the image simultaneously. As a result, although DCNNs are phenomenally successful in diverse types of image recognition when applied to Bangla HCR, they need more

ISSN No:-2456-2165

success. In recent years, a few significantly large Bangla handwritten characters datasets, such as the BanglaLekha-Isolated dataset [17] and the Ekush dataset [18], have been published, with an open new opportunity and scope of research on Bangla HCR with DCNNs methodology.

However, automatic recognition of Bangla characters with high accuracy, even with DCNN technology, still needs to be improved. One of the reasons for this is that most of the Bangla characters have complex and convoluted edges. For example, even primary vowel characters like "ঈ" and "ঔ" can become a pattern with convoluted edges and geometric structures. Moreover, many variations repeat the same pattern in several characters, like the letters "ও" and "ঔ", which have similar shapes but are entirely different.

Fig 1 Different Patterns in Same Handwritten Bangla Character

Another difficult challenge for the researchers is the various patterns of the same character that different people write, which vary depending on the place, age, and technique [19]. For example, in Figure 1, the same letter "ঋ" has different patterns de-pending on the writers. To overcome these challenges most efficiently in this paper, we have proposed a modified ResNet-34 architecture that can achieve a state-of-the-art recognition performance for Bangla characters. We have also provided a relative analysis of the performance of different well-recognized deep-learning architectures and compared them with our proposed model to provide a performance baseline for further future research.

The remainder of the paper is structured as follows: Section II describes the relevant efforts of the Bangla HCR. Section III discusses the suggested technique. Section IV contains the experimental results and associated analyses. Finally, Section V has the conclusion. The remainder of the paper is structured as follows: Section II describes the relevant efforts of the Bangla HCR. Section III discusses the suggested technique. Section IV contains the experimental results and associated analyses. Finally, Section V has the conclusion.

II. RELATED WORK

We have found various exciting research related to Handwriting Character Recognition, and we will briefly discuss them in this section. In [20], researchers have used CNN to classify Chinese characters that have 3,755 classes and have exemplary achievements. [21] is another non-English HCR research paper that greatly classified Arabic languages. Other high-achieving non-English HCR papers for Japanese, Ro-man, Urdu, Gujarati, etc. [22-25] has proposed different CNN architectures for automat-ic character recognition. In terms of the Bangla language, khan et al., in their paper [26], have done very inspiring work where they used a combination of different net-works and created a DCNN model that holds the characteristic of squeeze and excitation ResNet called SE-ResNet that achieved one of the highest accuracy in terms of Bangla HCR. Their architecture has achieved these results with the help of a combination of an SE computational layer on top of the ResNet model, a special kind of DCNN that can learn complex patterns by deciding channel-wise feature interdependencies. Additionally, Das et al., in their paper [19] proposed a solution to the explicit feature extraction problem by proposing a CNN model that can classify 50 classes of the introductory letter in Bangla. They achieved 92.25% accuracy in the 50 classes, another step toward the Bangla HCR. They have also verified that their proposed CNN model's performance is better than the previous methods, like MLP (multi-layer perceptron) and SVM (Support Vector Machine). On the other hand, [27], in their paper, investigated a methodology using MLP to examine primary shape decomposition for Bangla HCR.

Although there are examples of MLP methodologies, as mentioned earlier, and SVM methods that do well in Bangla HCR, as researched by [2] in their paper, these methodologies use either handcrafted feature extraction or their acquired accuracy is not up to the level needed for a practical HCR implementation. Recent studies like [28-30] have displayed the Bangla HCR using DCNN for characters and digits, which got outstanding accuracy on large-scale datasets. Inspired by these results, we have experimented with DCNNs and proposed an architecture that overcomes the limitations of the previously investigated models.

III. PROPOSED METHOD

Bangla letters are written from left to right in a single letter case, with minor spiral units rather than cylindrical ones. Furthermore, some characters have highly convoluted borders and similar patterns with a small change, such as a different pattern spiral line known as a mātrā running along the tops of the letters. These qualities make it difficult to detect even with the human eye. Previously, we presented a slightly modified Resnet architecture that demonstrated state-of-the-art accuracy in identifying Bangla-isolated handwritten characters to address these issues and detect these minor differences in the Bangla handwriting [31]. After more experimentation and research on the previously proposed architecture, we are now presenting a version of modified ResNet-34 that is remarkably resilient in identifying Bangla-isolated handwritten letters and has higher ac-curacy than its predecessor.

➢ *Modified Resnet-34 Architecture*

Deep Residual Network (Resnet) Architecture solves the problems of gradient vanishing explosion and performance degradation caused by depth increase. After winning first place in ILSVRC 2015 [10], it has been used widely for its image classification capabilities. ResNet generally consists of

ISSN No:-2456-2165

building blocks stacked together. These residual blocks have an input parameter x and a target output H(x), as shown in Equation 1.

$$F(x) = H(x) - x \qquad (1)$$

Which makes the output as Equation 2:

$$H(x) = F(x) + x \qquad (2)$$

In the previous paper [31], we explored the ResNet-18 capabilities by adding additional dropout layers, which provide generalised output with a higher degree of regularisation inspired by [32]. In this paper, we propose an improvement to the previously modified ResNet resulting in a more efficient gaining of the representation of structural characteris-tics of the Handwritten character statistical information and thus improving the overall classification accuracy. Increasing the depth of layers for ResNet is a popular approach in image classification used by several research fields [33, 34]. We experimented with ResNet by adding additional layers to our previously proposed model. We found good results with this approach. The optimal result was achieved with a layer count like ResNet 34.

Moreover, we also experimented with different combinations of dropout layer numbers in the proposed model; interestingly, we saw with our new layer count that the previous approach of adding dropouts in every resnet block needed to give better accuracy. However, we achieved the desired outcome by applying the dropout layers in a sequence of 1 after every two layers. As a result, our proposed model is a modified ResNet-34 with a sequence of dropouts after every two layers of a ResNet block per filter group. Also,

We are keeping the max pooling to 3×3 with a 2×2 stride, the same as our previously proposed ResNet-18 architecture, and after fully connected layers, we are still using SoftMax. The proposed modified RestNet-34 is illustrated in Figure 2.

Additionally, we have experimented with several optimisers to determine the global minima of the categorical cross-entropy loss function. Among them, the top performers are Root Mean Square Propagation (RMSProp), Stochastic Gradient Descent (SGD) with momentum [35] and Adam [36]. The Adam optimiser is a computational optimisation approach that is memory and time-efficient. Furthermore, rather than modifying the parameter learning rates based on the average first moment (the mean), as in RMSProp, Adam uses the average of the gradients' second moments (the uncentered variance), which gives it a performance boost when it comes to classifying patterns. In our experiment, Adam outperformed other optimisers and was thus selected as the optimiser for our proposed architecture.

In Figure 2, conv stands for convolutional layer, pool stands for maxpool layer, norm stands for batch normalisation, relu stands for rectified linear unit activation layer, and fc stands for fully connected hidden layers. Sixteen ResNet modules were modified by adding a dropout layer after every second filter group of 64,128,256 and 512.

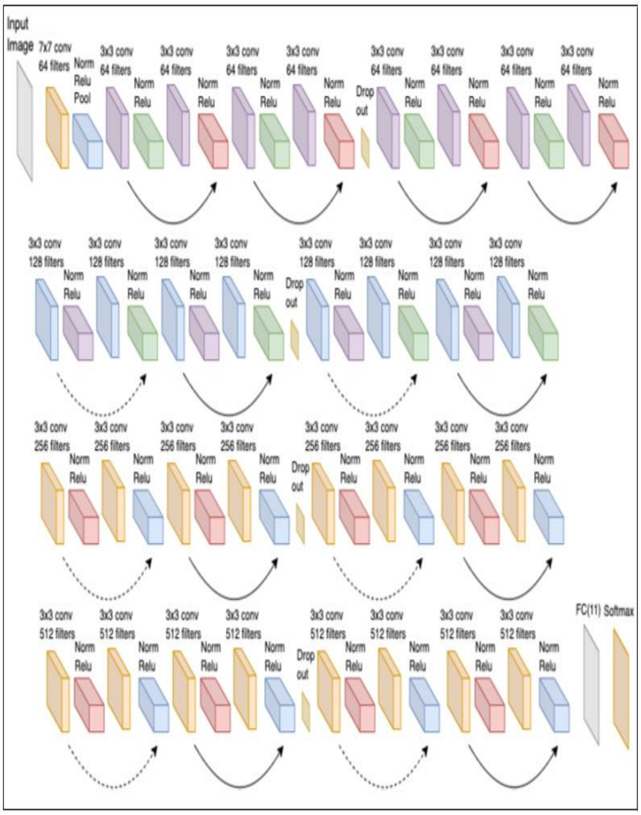

Fig 2 Proposed Modified RESNET-34 Architecture for Bangla Handwritten Character Recognition

➢ *Input Processing*

To get the maximum result in the most time-efficient way, we have used pre-processed datasets with inverted background, foreground, and noise-removed images [17]. Moreover, Gaussian blur was also used to smooth the images with padding to preserve the aspect ratio. The inputs were diversified by augmentation, oversampling, and data warping to increase the size of the original training dataset. The primary augmentation in use is height and width shifting, encouraged by different research results [37, 38]. The shift range was kept to 0.5 to ensure proper data augmentation and maximum data could be provided to the architecture to avoid overfitting.

IV. EXPERIMENTAL RESULTS

This part will discuss the results of various experiments and the performance analysis of our proposed modified ResNet-34 architecture. The experiments were prepared using the Anaconda development environment on a Windows 11 operating machine with an AMD Ryzen 9 5900HX (3.30 GHz) CPU and 16 GB RAM. The graphics card used in the experiment is NVIDIA GeForce RTX 3070 Laptop GPU with 8 GB GDDR6 graphics memory. The proposed model code was implemented using Keras v.2.6.0 [39] using the TensorFlow [40] backend.

ISSN No:-2456-2165

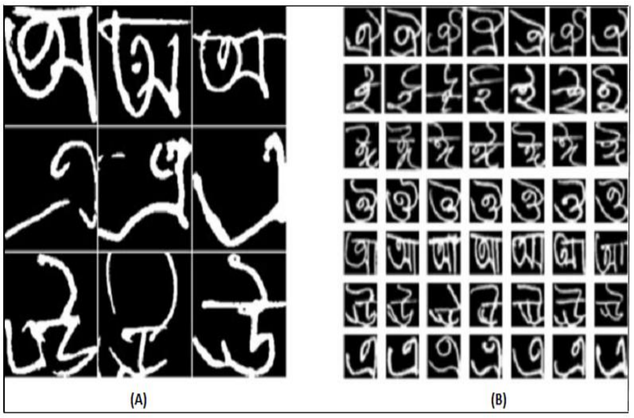

Fig 3 Example of Bangla Vowel Characters From A) BanglaLekha0Isolated Dataset, B) Ekush Dataset

➢ *Dataset*

We have used a mixture of two large datasets called the BanglaLekha-Isolated dataset from [17] and the Ekush dataset by [18] for our experiments. The BanglaLekha-Isolated dataset contains 84 classes of Bangla handwritten characters, 50 of which are vowels and consonants, 10 are numerals, and 24 are commonly used conjunct characters. It has 98,950 basic vowels and consonants, 19,748 digits, and 47,407 often-occurring conjunct consonants. This dataset's image resolution ranges from110×110 to 220 ×220 pixels. The handwriting images were gathered from people aged 4 to 27, with a tiny fraction of the samples coming from physically challenged people. The example of vowel characters taken from the Bangla Isolated Dataset is shown in Figure 3(A). The Ekush dataset is another large dataset that has 122 classes of characters with a resolution of 28× 28, which includes 367,018 images consisting of 154,824 vowels, 150,840 compound characters, and 30,687 numerical characters. These images were collected from school, college, and university students with a mixture of males and females. Figure 3(B) shows a few examples from the Ekush Dataset. This dataset is relatively more wide-ranging than the Bangla-Lekha-Isolated dataset. For our experiments, we have merged the characters from these two datasets and a hybrid dataset, resulting in a total of 53,321 images.

➢ *Hyperparameter Selection and Training*

At the start of our experiment, we distributed the images of the hybrid dataset to train, validate, and test with a ratio of 70% for training (37,319 images), 20% for validations (10,658 images), and 10% for testing (5,344 images). To expedite the experimental results, we initially focused on conducting experiments specifically on vowel characters to determine the optimal hyperparameters. This approach allowed us to streamline the research process and gain insights into the best configuration settings. By concentrating on vowel characters, which constitute a distinct and identifiable subset of the dataset, we aimed to minimize computational time while still achieving meaningful findings. Once the optimal hyperparameters were determined through these initial experiments, we extended our analysis to the entire dataset, encompassing all characters present in the hybrid dataset. This sequential approach ensured efficient experimentation while maintaining the generalizability of our results across the broader dataset.

Table 1 Different Hyperparameters and Related Accuracy of the Model on Vowel Character Classification

Hyperparameters		**Accuracy**
Type	***Value***	
Image Size	32x32	98.01
	64x64	98.31
	112x112	**98.70**
	118x118	98.25
Dropout	0.15	97.90
	0.20	**98.70**
	0.30	97.94
	0.50	97.80

The image size in the datasets varied across different files, and thus, we conducted various experiments to determine the optimal input size. Table 1 displays the results of these experiments, indicating that our modified ResNet-34 achieved the highest classification accuracy of 98.7% with an image size of 112× 112. Therefore, we selected this image size for the subsequent stages of the experiments. Furthermore, we have also experimented with two different hyperparameters to fine-tune our proposed architecture. At first, as we modified the ResNet-34 by adding layers of dropout after every 2nd ResNet module in every filter group, we experimented and investigated its performance using different Dropout Rates. As shown in Table 1, it is observed that a drop-out of 0.2 gives the best performance concerning classification accuracy. Secondly, we have also

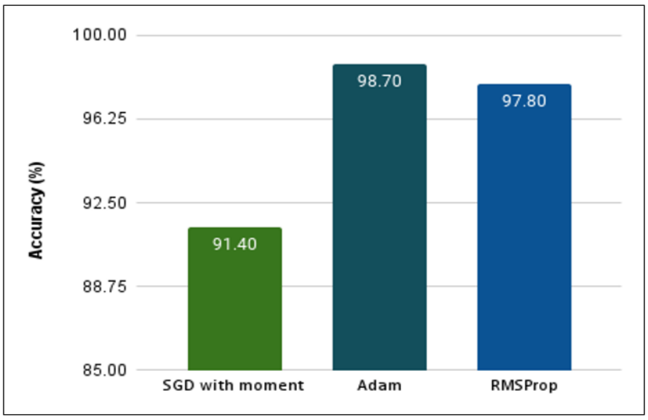

Fig 4 Different Optimizers Selection for Fine-tuning

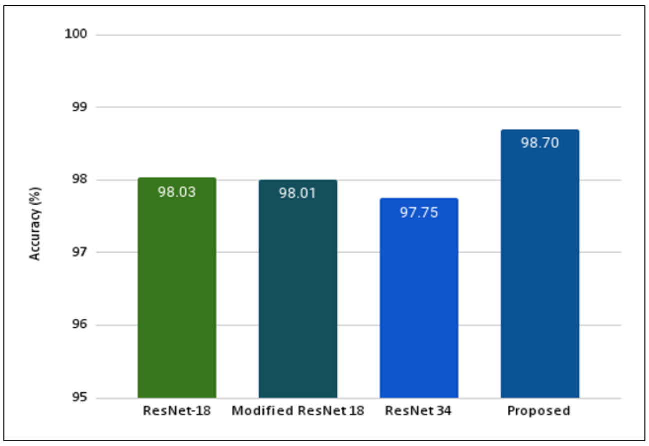

Fig 5 Performance of Classifications for Different Architectures

ISSN No:-2456-2165

Investigated the classification performance of the proposed architecture using various optimisers. We have mainly used three state-of-the-art optimisers known as SGD with momentum, Adam, and RMS props. The result of these different experiments can be found in Figure 4, where it can be seen that, as mentioned earlier, using the Adam optimiser, we have gained approximately a 7% performance boost to SGD with momentum and a 0.9% performance boost to RMSProp optimiser. Based on this experiment's results, we decided to use a dropout rate of 0.2 and Adam optimisers for the rest of the experiments.

We have also experimented with various state-of-the-art CNN models, and our previously introduced modified ResNet-18 [31] with the same hybrid dataset and measured their performance against our proposed model. This experiment included unmodified Res-Net-18, unmodified ResNet-34, previously modified ResNet-18, and the proposed modified ResNet-34 architecture on the mixture of the two BanglaLekha-Isolated and Ekush dataset's 11 vowel classes. Figure 5 illustrates a bar chart for the classification performance of these experiments. The bar chart shows that ResNet 18 and modified Res-Net-18 with dropouts are performing very closely to each other regarding vowel character classification by achieving 98.03% and 98.01%. On the other hand, when it came to un-modified ResNet-34, the performance dropped by 0.28% and achieved 97.75% accuracy. Although ResNet-18 and modified ResNet-18 performed well, it can be seen that our modified ResNet-34 achieved a higher accuracy of 98.70 %, giving a significant 0.96% boost from ResNet34 and consecutively 0.70% better than our previously proposed model.

Finally, we employed our proposed ResNet-34 architecture to assess its performance on the hybrid dataset encompassing vowel, consonant, numeric, and mixed characters. We aimed to evaluate the model's ability to handle the diverse range of characters present in real-world scenarios. To this end, we trained and validated our architecture on the entire dataset, including all character types. The performance of our model was evaluated by examining the training and validation accuracy as a function of the number of epochs. Figure 6 illustrates the training and validation accuracy curves, providing insights into the model's learning progress and generalisation capabilities. Similarly, Figure 7 displays the corresponding training and validation loss curves, allowing us to analyse the convergence and optimisation of the model during the training process. These visualizations serve as valuable tools for understanding the behavior of our proposed architecture and assessing its overall performance on the hybrid dataset, encompassing a wide range of character types.

- *Results and Discussions*

When collecting and analyzing the results, we defined accuracy as the percentage of total samples classified correctly by the model. We followed Equation 3 from [26] to calculate this accuracy.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \text{ x } 100 \tag{3}$$

Where, TP is True positive, TN is True negative, FP is False positive, and FN is False Negative.

We also analyzed the precision, defined as the percentage of total predicted positive examples by the model. In short, we determined the precision with Equation 4:

$$Precision = \frac{Tp}{TP+FP} \times 100 \tag{4}$$

Furthermore, we also calculated the recall, which is the percentage of total correctly predicted positive samples by the model with Equation 5:

$$Recall = \frac{TruePositive}{TruePositive+FalseNegative} \times 100 \tag{5}$$

Finally, we have analyzed the F1-score with the help of Equation 6, which shows the relation between precision and Recall:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \times 100 \tag{6}$$

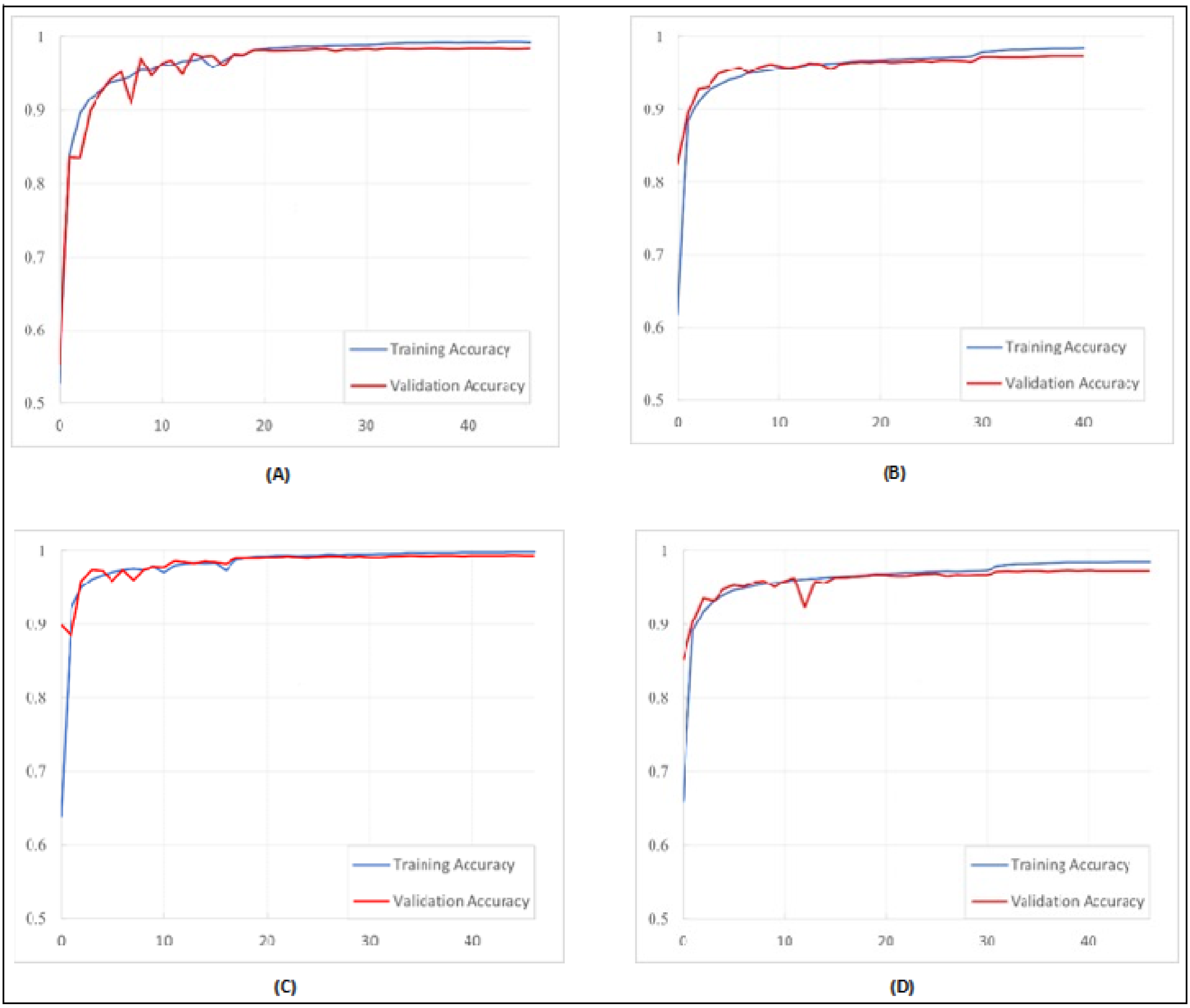

Fig 6 Training And Validation Accuracy With Respect To Epochs Of The Proposed Model For (A) Vowel Characters, (B) Consonant Characters, (C) Numeric Characters And (D) Mixed Characters

The results of these calculations we got from our model are shown in Table 2 and can be defined as an incredibly decent performance. The True Positive, True Negative, False Positive and False Negative are computed using Equation (7) – (10) inspired by [41] such as:

$$True\ Positive_i = C_{ii} \tag{7}$$

$$True\ Negative_i = \sum_{k=1,k\neq i}^{n} \sum_{j=1,j\neq i}^{n} C_{jk} \tag{8}$$

$$False\ Positive_i = \sum_{j=1,j\neq i}^{n} C_{ji} \tag{9}$$

$$False\ Negative_i = \sum_{j=1,j\neq i}^{n} C_{ij} \tag{10}$$

Here the class is "i", and the total number of character classes is n (n=11). "C_jk " is the j-th row and k-th column item in the confusion matrix.

Table 2 presents the performance metrics of our proposed ResNet-34 architecture on different character types within the hybrid dataset. We evaluated the model's accuracy, precision, recall, and F1-score to measure its effectiveness in accurately classifying the various character categories. For vowel characters, the average accuracy, precision, recall, and F1-score were consistently high at 98%. This indicates that the model achieved a high level of accuracy in correctly classifying vowel characters, demonstrating its proficiency in recognizing this specific character type. Similarly, the model performed well on consonant characters, with an average accuracy, precision, recall, and F1-score of 97%. These results indicate a robust ability to identify and classify consonant characters effectively. Numeric characters exhibited even higher performance, with an average accuracy, precision, recall, and F1-score of 99%. This suggests that the model excelled in accurately recognizing and distinguishing numeric characters within the dataset. The mixed character category also demonstrated a strong performance, with an average accuracy, precision, recall, and F1-score of 97%. This suggests that the model was capable of handling the complexity of mixed characters, effectively differentiating them from other categories. Overall, the results indicate the effectiveness of our proposed ResNet-34 architecture in accurately classifying characters across various types within the hybrid dataset. The high-performance metrics obtained validate the model's ability to generalize well and provide reliable character recognition across diverse categories.

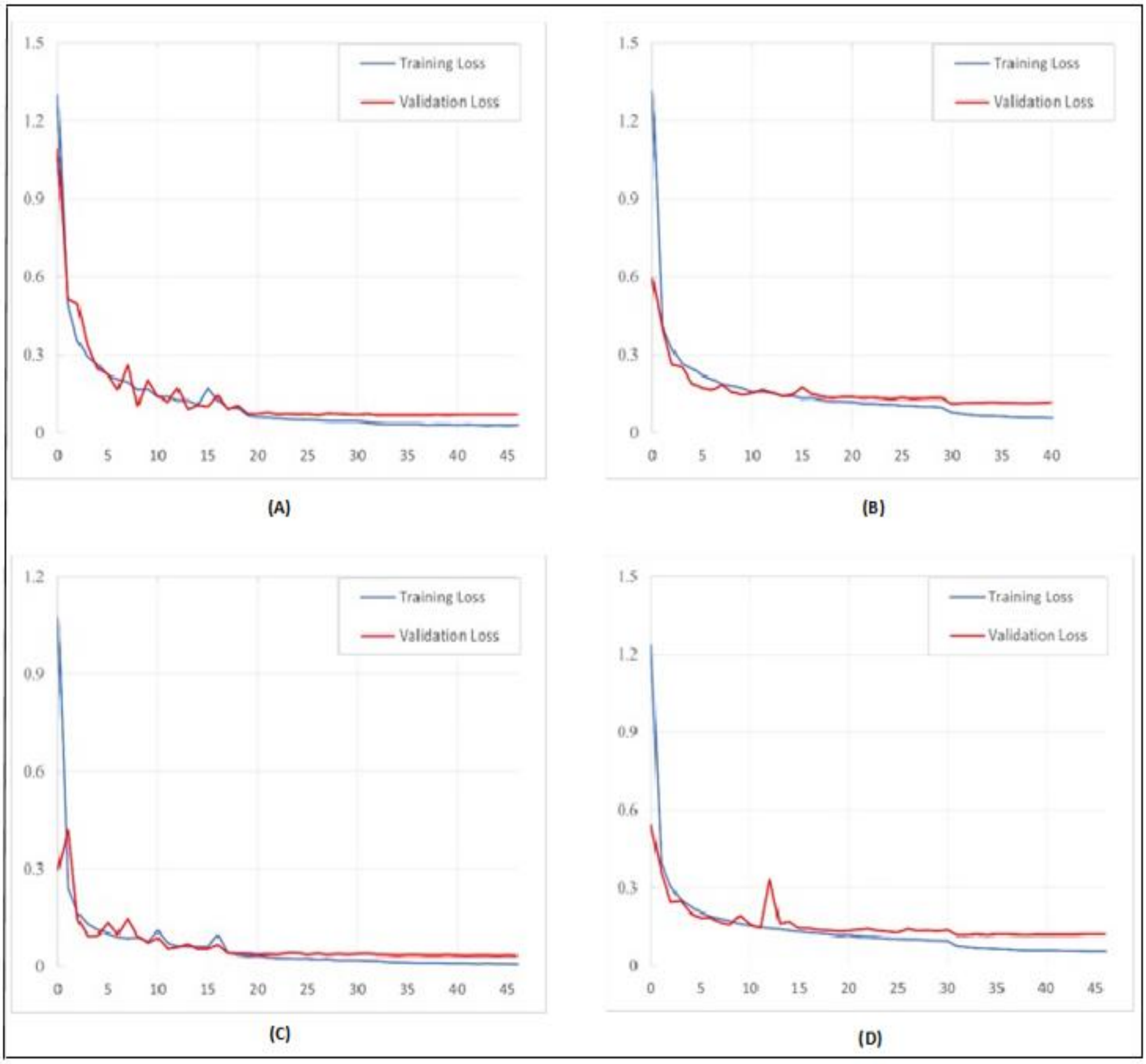

Fig 7 Raining And Validation Loss With Respect To Epochs Of The Proposed Model For (A) Vowel Characters, (B) Conso-Nant Characters, (C) Numeric Characters And (D) Mixed Characters

Finally, in Table 3, we can see the confusion matrix report of the performance of our proposed modified ResNet-34 model for mixed characters. It can be observed from the confusion matrix that there are 29 instances (5.71% interclass confusion) of the char-acter "ঋ" that is mislabeled as character “৬” out of 507 test samples which is the highest mislabeled instance in the confusion matrix as result in the confusion matrix report the precision is 0.93 and recall is 0.92. This happens because when writing both characters, there is a similarity in the pattern, especially the vertical lines at the top of the charac-ters. Moreover, the writing pattern of these images is similar, and the written charac-ters are tough to distinguish, even by human eyes. Additionally, the second highest confusion is found in the 20 instances where the character classification of "চ " is misclas-sified (3.94% inter-class confusion) out of 507 test samples and in the report the precision is 0.90 and recall is 0.94. This also happens because of the similarity in appearance be-tween the similar patter characters, as explained earlier. Furthermore, similar mis-matches also happened between the true label vs the predicted label for characters like "ধ ","জ"," ন "and " ৬ " which is already reflected in the confusion matrix report.

ISSN No:-2456-2165

Table 2 The Performance Metrics of our Proposed ResNet-34 Architecture on Different Character Types

Characters	**Average**			
	Accuracy	**Precision**	**Recall**	**F1-Score**
Vowel	98 %	98 %	98%	98%
Consonant	97 %	97 %	97%	97%
Numeric	99%	99%	99%	99%
Mixed	97%	97%	97%	97%

Table 3 Confusion Matrix Report of our Proposed Restnet-34 Architecture on Mixed Character Type Dataset

Class	**Precision**	**Recall**	**F1-score**	**Support**
অ	0.98	0.99	0.98	476
আ	0.98	0.97	0.98	508
ই	0.99	0.98	0.99	507
ঈ	0.98	0.97	0.98	507
উ	0.97	0.94	0.95	506
ঊ	0.98	0.97	0.98	506
ঋ	0.93	0.92	0.93	505
এ	0.96	0.97	0.97	507
ঐ	0.96	0.97	0.97	507
ও	0.98	0.97	0.98	506
ঔ	0.99	0.98	0.99	506
ক	0.99	0.98	0.99	500
খ	0.98	0.97	0.98	507
গ	0.98	0.98	0.98	510
ঘ	0.99	0.98	0.99	506
ঙ	0.95	0.97	0.96	501
চ	0.90	0.94	0.92	505
ছ	0.95	0.96	0.95	503
জ	0.93	0.94	0.94	501
ঝ	0.98	0.93	0.95	507
ঞ	0.96	0.97	0.96	606
ট	0.96	0.99	0.97	510
ঠ	0.99	0.98	0.98	466
ড	0.98	0.96	0.97	506
ঢ	0.94	0.93	0.93	509
ণ	0.98	0.95	0.96	505
ত	0.98	0.98	0.98	507
থ	0.98	0.98	0.98	508
দ	0.95	0.92	0.94	506
ধ	0.93	0.96	0.94	508
ন	0.91	0.9	0.9	497
প	0.99	0.98	0.99	508
ফ	0.98	0.98	0.98	504
ব	0.98	0.99	0.99	407
ভ	0.99	0.99	0.99	506
ম	0.95	0.98	0.96	515
য	0.97	0.95	0.96	505
র	0.95	0.94	0.95	504
ল	1.00	0.98	0.99	507
শ	0.99	0.98	0.99	506
ষ	0.99	0.97	0.98	514
স	0.99	0.99	0.99	505

হ	0.99	0.99	0.99	508
ড়	0.99	0.99	0.99	507
ঢ়	0.95	0.96	0.96	478
য়	0.98	0.99	0.99	505
ৎ	0.99	0.99	0.99	506
ং	0.99	0.98	0.99	506
ঃ	0.96	0.97	0.96	502
ঁ	0.93	0.98	0.96	505
০	0.97	0.98	0.98	506
১	0.96	0.94	0.95	507
২	0.95	0.98	0.97	506
৩	0.99	0.99	0.99	503
৪	0.99	0.98	0.98	506
৫	0.98	0.98	0.98	480
৬	0.96	0.97	0.96	503
৭	0.98	0.97	0.98	506
৮	0.98	0.98	0.98	508
৯	0.99	0.99	0.99	508
Accuracy			0.97	
Macro avg	0.97	0.97	0.97	504
Weighted avg	0.97	0.97	0.97	504

Class	Precision	Recall	F1-score	Support
ঁ	0.93	0.98	0.96	505
০	0.97	0.98	0.98	506
১	0.96	0.94	0.95	507
২	0.95	0.98	0.97	506
৩	0.99	0.99	0.99	503
৪	0.99	0.98	0.98	506
৫	0.98	0.98	0.98	480
৬	0.96	0.97	0.96	503
৭	0.98	0.97	0.98	506
৮	0.98	0.98	0.98	508
৯	0.99	0.99	0.99	508
Accuracy			0.97	
Macro avg	0.97	0.97	0.97	504
Weighted avg	0.97	0.97	0.97	504
০	0.97	0.98	0.98	506
১	0.96	0.94	0.95	507
২	0.95	0.98	0.97	506
৩	0.99	0.99	0.99	503
৪	0.99	0.98	0.98	506
৫	0.98	0.98	0.98	480
৬	0.96	0.97	0.96	503
৭	0.98	0.97	0.98	506
৮	0.98	0.98	0.98	508
৯	0.99	0.99	0.99	508
Accuracy			0.97	
Macro avg	0.97	0.97	0.97	504
Weighted avg	0.97	0.97	0.97	504

V. CONCLUSIONS

In this article for Bangla character classification, we have proposed a modified Res-Net-34, which can provide state-of-the-art accuracies on the two large Bangla handwritten character datasets. The proposed model achieved a rate of 98.70% accuracy on vowel characters, 97.34% on consonant characters, 99.02% on numeric characters, and overall, on the mix of all these characters it achieved 97% with an F1-Score of 97%. Furthermore, the obtained experimental results of this model demonstrated the effectiveness and robustness of classifying Bangla characters. We also compared the classification abilities of the proposed model with our previously proposed modified ResNet-18 model and observed the extended capabilities of the modified ResNet-34 architecture. The suggested methodology will lead to developing high-performance OCR for Bangla digital documentation based on all these results. In this paper, we ignored the Bangla numerals, consonants, and compound characters, which will be included with modifications to the proposed ResNet architecture in our future research. Additionally, as we succeed in character pattern recognition, we can extend this research to other fields like facial expression identification, fingerprinting and other biometric item recognition in images and videos. Furthermore, there is scope for integration into CNN-based industrial defect detection systems, i.e., in the food monitoring [42, 43], renewable energy [44] and edge-constrained devices for security monitoring such as [45].

REFERENCES

[1]. Pereltsvaig, A., Languages of the World. 2020: Cambridge University Press.

[2]. Kibria, M.R., et al. Bangla compound character recognition using support vector machine (SVM) on advanced feature sets. in 2020 IEEE Region 10 Symposium (TENSYMP). 2020. IEEE.

[3]. Hussain, M., et al., Exudate Regeneration for Automated Exudate Detection in Retinal Fundus Images. IEEE Access, 2022.

[4]. Aydin, B.A., et al. Domain modelling for a lightweight convolutional network focused on automated exudate detection in retinal fundus images. in 2023 9th International Conference on Information Technology Trends (ITT). 2023. IEEE.

[5]. Zahid, A., et al. Lightweight convolutional network for automated photovoltaic defect detection. in 2023 9th International Conference on Information Technology Trends (ITT). 2023. IEEE.

[6]. Hussain, M., et al., Feature Mapping for Rice Leaf Defect Detection Based on a Custom Convolutional Architecture. Foods, 2022. 11(23): p. 3914.

[7]. LeCun, Y., Y. Bengio, and G. Hinton, Deep learning. nature, 2015. 521(7553): p. 436-444.

[8]. Krizhevsky, A., I. Sutskever, and G.E. Hinton, Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 2012. 25.

[9]. Ballester, P. and R. Araujo. On the performance of GoogLeNet and AlexNet applied to sketches. in Proceedings of the AAAI conference on artificial intelligence. 2016.

[10]. He, K., et al. Deep residual learning for image recognition. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[11]. Huang, G., et al. Densely connected convolutional networks. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[12]. Deng, J., et al., ImageNet large scale visual recognition challenge (ILSVRC) 2012. 2012.

[13]. Fang, W., et al., Computer vision for behaviour-based safety in construction: A review and future directions. Advanced Engineering Informatics, 2020. 43: p. 100980.

[14]. Dalal, N. and B. Triggs. Histograms of oriented gradients for human detection. in 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). 2005. Ieee.

[15]. Lowe, D.G., Distinctive image features from scale-invariant keypoints. International journal of computer vision, 2004. 60(2): p. 91-110.

[16]. Yang, Y. and S. Newsam. Bag-of-visual-words and spatial extensions for land-use classification. in Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems. 2010.

[17]. Biswas, M., et al., Banglalekha-isolated: a comprehensive bangla handwritten character dataset. arXiv preprint arXiv:1703.10661, 2017.

[18]. Rabby, A., et al. 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. 2018. Springer.

[19]. Das, T.R., et al., Bangla Handwritten Character Recognition Using Extended Convolutional Neural Network. Journal of Computer and Communications, 2021. 9(3): p. 158-171.

[20]. Li, Z., et al., Deep matching network for handwritten Chinese character recognition. Pattern Recognition, 2020. 107: p. 107471.

[21]. Alheraki, M., R. Al-Matham, and H. Al-Khalifa, Handwritten Arabic Character Recognition for Children Writ-ing Using Convolutional Neural Network and Stroke Identification. arXiv preprint arXiv:2211.02119, 2022.

[22]. Ly, N.-T., et al. Deep convolutional recurrent network for segmentation-free offline handwritten Japanese text recognition. in 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). 2017. IEEE.

[23]. Abdulhussain, S.H., et al., A robust handwritten numeral recognition using hybrid orthogonal polynomials and moments. Sensors, 2021. 21(6): p. 1999.

[24]. KO, M.A. and S. Poruran, OCR-nets: variants of pre-trained CNN for Urdu handwritten character recognition via transfer learning. Procedia Computer Science, 2020. 171: p. 2294-2301.

[25]. Doshi, P.D. and P.A. Vanjara, A Comprehensive Survey on Handwritten Gujarati Character and Its Modifier Recognition Methods. Information and Communication Technology for Competitive Strategies (ICTCS 2020), 2022: p. 841-850.
[26]. Khan, M.M., et al., A squeeze and excitation ResNeXt-based deep learning model for Bangla handwritten compound character recognition. Journal of King Saud University-Computer and Information Sciences, 2022. 34(6): p. 3356-3364.
[27]. Pramanik, R. and S. Bag, Shape decomposition-based handwritten compound character recognition for Bangla OCR. Journal of Visual Communication and Image Representation, 2018. 50: p. 123-134.
[28]. Majid, N. and E.H.B. Smith, Character spotting and autonomous tagging: offline handwriting recognition for Bangla, Korean and other alphabetic scripts. International Journal on Document Analysis and Recognition (IJDAR), 2022: p. 1-19.
[29]. Hossain, M.T., M.W. Hasan, and A.K. Das. Bangla Handwritten Word Recognition System Using Convolutional Neural Network. in 2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM). 2021. IEEE.
[30]. Huda, H., et al. Bangla Handwritten Character and Digit Recognition Using Deep Convolutional Neural Network on Augmented Dataset and Its Applications. in 2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM). 2022. IEEE.
[31]. Alif, M.A.R., S. Ahmed, and M.A. Hasan. Isolated Bangla handwritten character recognition with convolutional neural network. in 2017 20th International conference of computer and information technology (ICCIT). 2017. IEEE.
[32]. Zagoruyko, S. and N. Komodakis, Wide residual networks. arXiv preprint arXiv:1605.07146, 2016.
[33]. Jing, E., et al., ECG heartbeat classification based on an improved ResNet-18 model. Computational and Mathematical Methods in Medicine, 2021. 2021.
[34]. Zhou, Y., et al. Facial sentiment classification based on resnet-18 model. in 2019 International Conference on electronic engineering and informatics (EEI). 2019. IEEE.
[35]. Liu, Y., Y. Gao, and W. Yin, An improved analysis of stochastic gradient descent with momentum. Advances in Neural Information Processing Systems, 2020. 33: p. 18261-18271.
[36]. Kingma, D.P. and J. Ba, Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
[37]. Khosla, C. and B.S. Saini. Enhancing performance of deep learning models with different data augmentation techniques: A survey. in 2020 International Conference on Intelligent Engineering and Management (ICIEM). 2020. IEEE.
[38]. Sánchez-Peralta, L.F., et al., Unravelling the effect of data augmentation transformations in polyp segmentation. International journal of computer assisted radiology and surgery, 2020. 15(12): p. 1975-1988.
[39]. Ketkar, N., Introduction to keras, in Deep learning with Python. 2017, Springer. p. 97-111.
[40]. Dillon, J.V., et al., Tensorflow distributions. arXiv preprint arXiv:1711.10604, 2017.
[41]. Habib, M.T., et al., Machine vision based papaya disease recognition. Journal of King Saud University-Computer and Information Sciences, 2020. 32(3): p. 300-309.
[42]. Hussain, M. and R. Hill, Custom Lightweight Convolutional Neural Network Architecture For Automated Detection of Damaged Pallet Racking In Warehousing & Distribution Centers. IEEE Access, 2023.
[43]. Hussain, M., YOLO-v5 Variant Selection Algorithm Coupled with Representative Augmentations for Modelling Production-Based Variance in Automated Lightweight Pallet Racking Inspection. Big Data and Cognitive Computing, 2023. 7(2): p. 120.
[44]. Animashaun, D. and M. Hussain, Automated Micro-Crack Detection within Photovoltaic Manufacturing Facility via Ground Modelling for a Regularized Convolutional Network. Sensors, 2023. 23(13): p. 6235.
[45]. Hussain, M. and H. Al-Aqrabi, Child Emotion Recognition via Custom Lightweight CNN Architecture, in Kids Cybersecurity Using Computational Intelligence Techniques. 2023, Springer. p. 165-174.