Bangla Handwritten Character Recognition Using Deep Learning Approaches and its Explainability With AI

Md. Mahedi Hasan[1], Fahim Habib[2], Md. Sharzul Mostafa[3]

[1] Senior Lecturer, Department. of Computer Science and Engineering, North Western University, Khulna
[2] Student, Department of Computer Science and Engineering, North Western University, Khulna
[3] Lecturer, Department of Computer Science and Engineering, North Western University, Khulna

Abstract

The realm of Bangla handwritten character recognition (BHCR) has long been overshadowed by the dominance of more mainstream languages, despite Bangla's status as Deep learning (DL) approaches have led to substantial improvements in handwritten character recognition (BHCR) for Bangla, one of the humanity's frequently spoken languages. These methods generally an optimal fit for BHCR because they are good at selecting high-level characteristics from intricate information. In our comprehensive study, we meticulously explored the efficacy of twelve DL models on the arduous task of Bangla character recognition, meticulously evaluating their performance on two distinct datasets: a handwritten character dataset and CMATERDB [1], comprising a formidable collection of 15,000 images. Additionally, we provided an audit of the DL models' achievements for BHCR. Among the compared models are LSTM, Bi-LSTM, CNN, Inception, VGG, and ResNet. we achieved the maximum performance at ResNet152V2. In this study, One of the most exquisite identification rates for Bengali character recognition currently available has been demonstrated by the suggested technique, which displayed an adequate 98.76% recognition accuracy on the dataset.

Keywords: Bangla handwritten character recognition (BHCR), Deep learning (DL), Comparative performance, LSTM, Bi-LSTM, CNN, Inception, VGG, ResNet, learning, Prediction, Ability.

1 Introduction

In the last several years, notable developments in a number of linguistic domains, fostering advancements in language processing, image analysis, and document digitization. The complexity of the Bengali script, characterized by its cursive nature and diverse character set, poses significant challenges for automated recognition systems [2]. As such, research in Bangla Handwritten Character Recognition (BHCR) remains a critical endeavor, driven by the need to develop robust and accurate recognition models capable of deciphering handwritten Bengali characters with high precision and efficiency [3].

Utilizing state-of-the-art deep learning models, aiming to push the boundaries of performance and accuracy in character recognition tasks [4]. Our research builds upon a foundation of meticulously curated datasets, comprising 15,000 handwritten Bengali character images sourced from diverse sources to ensure representativeness and variability in writing styles. The dataset is partitioned into 8400 images for training and 3,000 images for testing and Total Validation images are 3600, with each image annotated and classified into one of 50 distinct character classes.

The main objective of our research is to assess the effectiveness of several deep learning architectures in relation to BHCR. We use a variety of deep learning models, especially LSTM (Long Short-Term Memory) networks, neural networks using convolution (CNNs), and advanced architectures such as Inception, VGG, and Bi-LSTM. Each model undergoes rigorous training and evaluation over 30 epochs. the ResNet model emerges as the top-performing architecture. Motivated by this success, we delve deeper into the ResNet framework, exploring various iterations and modifications to optimize performance further. In particular, we investigate variants of the ResNet architecture, including ResNet50, ResNet-50V2, ResNet101, ResNet101V2, ResNet152, and ResNet152V2, aiming to identify the most efficient and accurate model for BHCR.

This research encompasses a holistic approach to deep learning model development. By refining data preprocessing techniques, introducing architectural enhancements, optimizing model parameters, and integrating interpretability techniques.

2 Literature Review

Over the past few years, a variety of ML-based techniques have been researched for Bangla Handwritten character detection. Chowdhury ,et al.[5] to address these challenges by proposing a method for automatic feature extraction, considering 50 classes of basic letters and 10 classes of digits for improved overall accuracy. Images from the BanglaLekha-Isolated [6] dataset are preprocessed to a common form, with resizing to 32 x 32 pixels and conversion to white letters on a black background. In the first convolutional layer, zero padding is applied, followed by the application of eight kernels of dimension 3 x 3 x 1 to extract features. Preprocessing involves resizing images to 32x32 pixels and splitting the dataset into 80% training and 20% testing. Utilizing an amalgamated Bangla handwritten character set (consonants, vowels, and digits mixed together), the recommended approach produced validation errors of 0.3204 and training errors of 0.0344. After (10) epochs, the training accuracy were 99.08%, all while the validation accuracy remained 92.25%.

Basri, et al. [7] examined the performance of various deep CNN architectures for recognizing handwritten Bangla digits, leveraging the NumtaDB dataset to concentrate on the AlexNet, MobileNet, GoogleNet (Inception V3), and CapsuleNet models for the recognition of Bangla digits written by hand. This study investigates four cutting-edge deep CNN architectures for digit recognition of handwriting employing the NumtaDB dataset. With regular data, GoogleNet obtains an average recognition accuracy of 93%.

Chakraborty, et al. [8] In a study, various Deep learning approaches consisting of convolutional neural networks (CNN), deep belief networks (DBN), CNN with abandonment, CNN with loss and Gaussian filters, and CNN with dropout and Gaussian filters for handwritten Bangla digit identification. These methods were tested on the CMATERdb 3.1.1 Bangla numeral image database. The recommended approach, implemented in Python 3, uses a CNN classifier trained on preprocessed, scaled, and standardized data to predict input characters, including Bangla, English, and numbers. Training set comprised 75% of the data, with 25% for testing, resulting in a loss of 0.5579 and an accuracy of 85.96% after ten epochs.

The proposed [9] CNN model consists of 10 layers connected sequentially. It starts with an input layer of size 32x32x1 followed by two convolutional layers with 32 and 64 filters, both using a 3x3 kernel and ReLU activation. Then, a max-pooling layer reduces the image size by half. This pattern repeats with two more convolutional layers. After that, a flattened layer transforms the data to 1D, followed by two dense layers and a dropout layer. The last dense layer serves as the output with softmax activation, totaling 637,724 trainable parameters. The proposed CNN-based approach for recognizing handwritten Bangla alphabet achieves 90.22% validation accuracy on Bangalekha isolated dataset and 93.22% on Ekush dataset, addressing challenges of complex-shaped characters and similarity between characters.

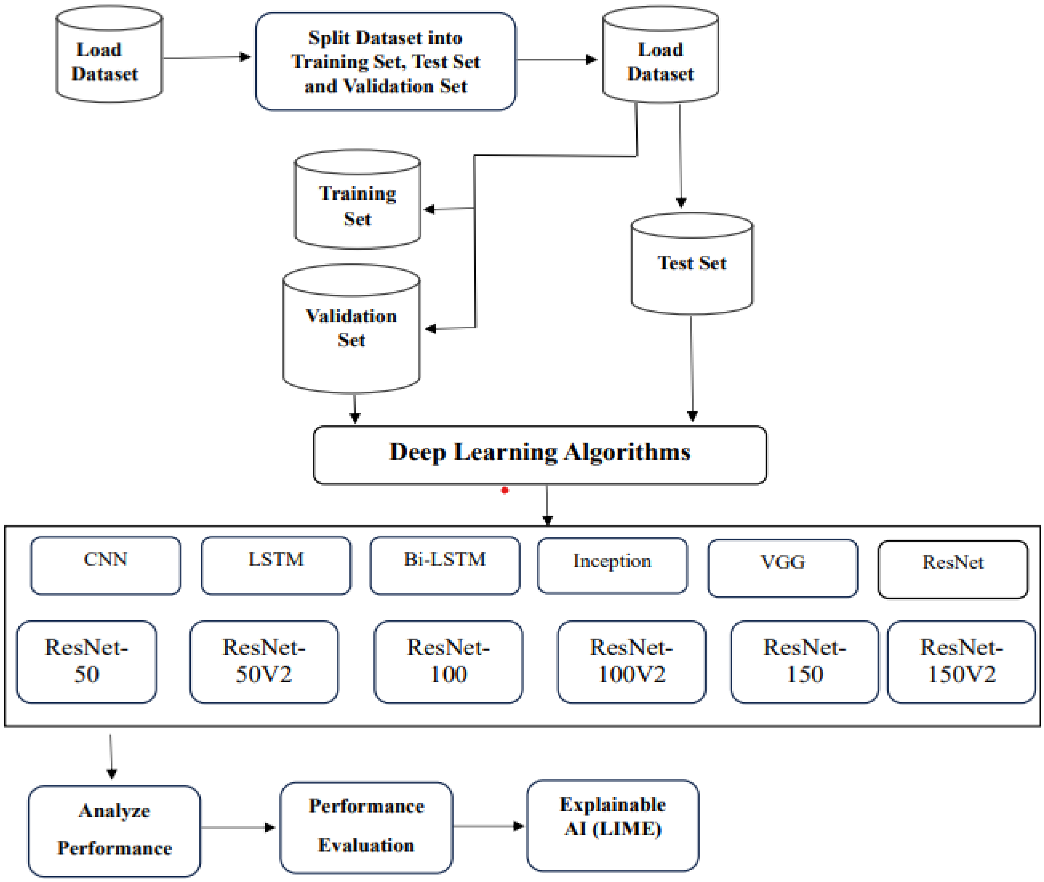
The paper introduces [10] DConvAENNet, a novel model combining Autoencoder and (DCNN) for Bangla Handwritten Character Recognition (BHCR). It elaborates on the suggested model's layout, highlighting the use of encoder layers for both unsupervised pre-training and supervised learning.

Md Ali Azad et al. [11] Using the three-character collections (BanglaLekha-Isolated, CMATERdb 3.1, and Ekush), a total of 22 experiments were run. Every effort yields adequate outcomes, with the identification of Bangla numbers written by hand approaching up to 90% accuracy,vowels, consonants, modifiers, compound characters, and all characters presented separately. Our proposed DConvAENNet model acquired 95.21% on BanglaLekha-Isolated for 84 courses, 92.40% on CMATERdb.3.1 for 238 classes, and 95.53% on Ekush for 122 classes using this supervised and unsupervised learning approach. competence.

3 Methodology and System Architecture

The proposed approach includes the three important stages namely: Data Preprocessing step, Training step and Prediction step. Flow diagram is shown in Figure 1 and current section includes the brief discussions of the same.

Figure 1: The Proposed Model's System Construction



3.1 Dataset

For evaluation purpose, CMATERdb [9],dataset is utilized. Initially developed at the Center for The micro processor Applications for Learning Education and Research (CMATER), a research establishment of Jadavpur University headquartered in Kolkata, India, CMATERdb was a repository for recognize patterns samples. The style of photographic data in these databases varies, and some of the photographs include some noise. A few example photos from the dataset are provided. In Bangla, there are fifty fundamental characters in total—eleven vowels and 39 consonants.

Figure 2: CMATERdb Dataset

The names of these subfolders correspond to the labels of each of the 50 classes, and each one has photographs for each of these categories. There are 168 images in each of the train folder's sub-folders, for a total of 8400 images. Each sub-folder in the test folder has 60 images, for an entire collection of 3000 images. The 3600 total validation images, with 68 images in each sub-folder, were utilized to judge each class's scores for F1, recall, and precise.

Table 1: Description of the Dataset

Dataset Name	**CMATERdb**
Total Image Count	15000
Total Number of class	50
Training images overall	8400
Training images for each class	168
Total test images	3000
Test images per class	60
Total Validation images	3600
Validation images per class	68

3.2 Data Preprocessing

This data preprocessing setup is well-suited for training image classification models using TensorFlow/Keras, facilitating efficient model training and evaluation with proper data handling and preprocessing techniques. Demonstrates data preprocessing for image classification using TensorFlow/Keras. Images are preprocessed using ImageDataGenerator instances (train_datagen and test_datagen) with a rescale=1./255 operation to normalize pixel values to the range [0, 1]. train_generator and validation_generator is created using flow_from_directory to generate batches of preprocessed images during model training and evaluation. Images are resized to 50x50 pixels and converted to grayscale (color_mode='grayscale') for simplicity and reduced computational complexity. The batch_size is set to 32, specifying the number of images processed per batch. Labels are encoded in a categorical format (class_mode='categorical'), suitable for multi-class classification tasks. This preprocessing setup standardizes input dimensions, handles data augmentation efficiently, and prepares the data for training convolutional neural networks or similar models effectively.

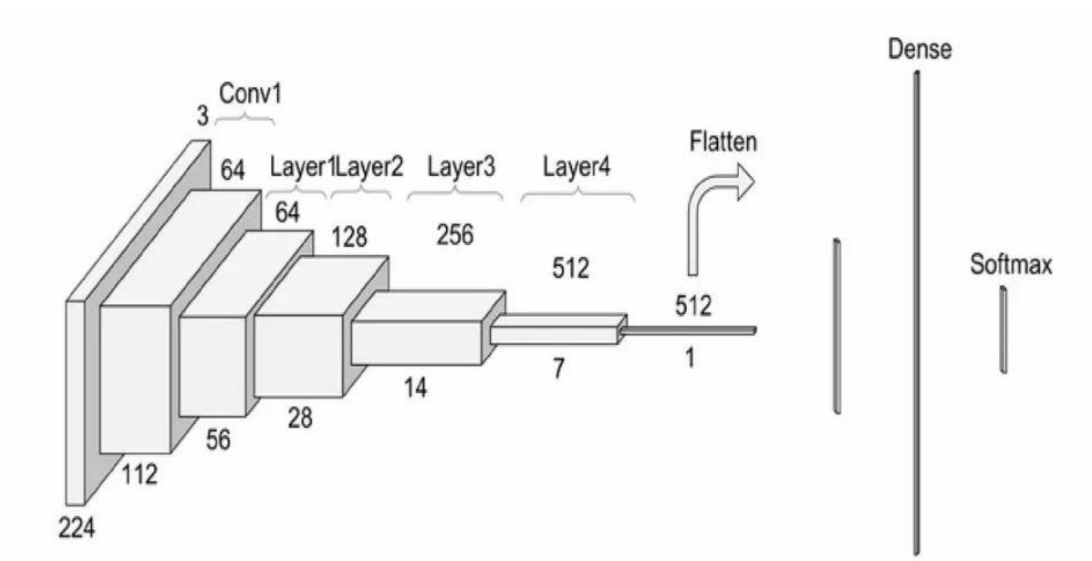
3.3 BHCR with Deep Learning Techniques

Deep learning methods is applicable to the study of imagery, modeling, and making sense of complicated clinical information. CNN, Inception, VGG, LSTM, Bi-LSTM And ResNet are the learning methods used in this investigation. Among all those deep learning techniques ResNet provides better performance in terms of accuracy. Some fine tuning is performed on base Resnet to achieve more accuracy.

3.3.1 ResNet Model Architecture

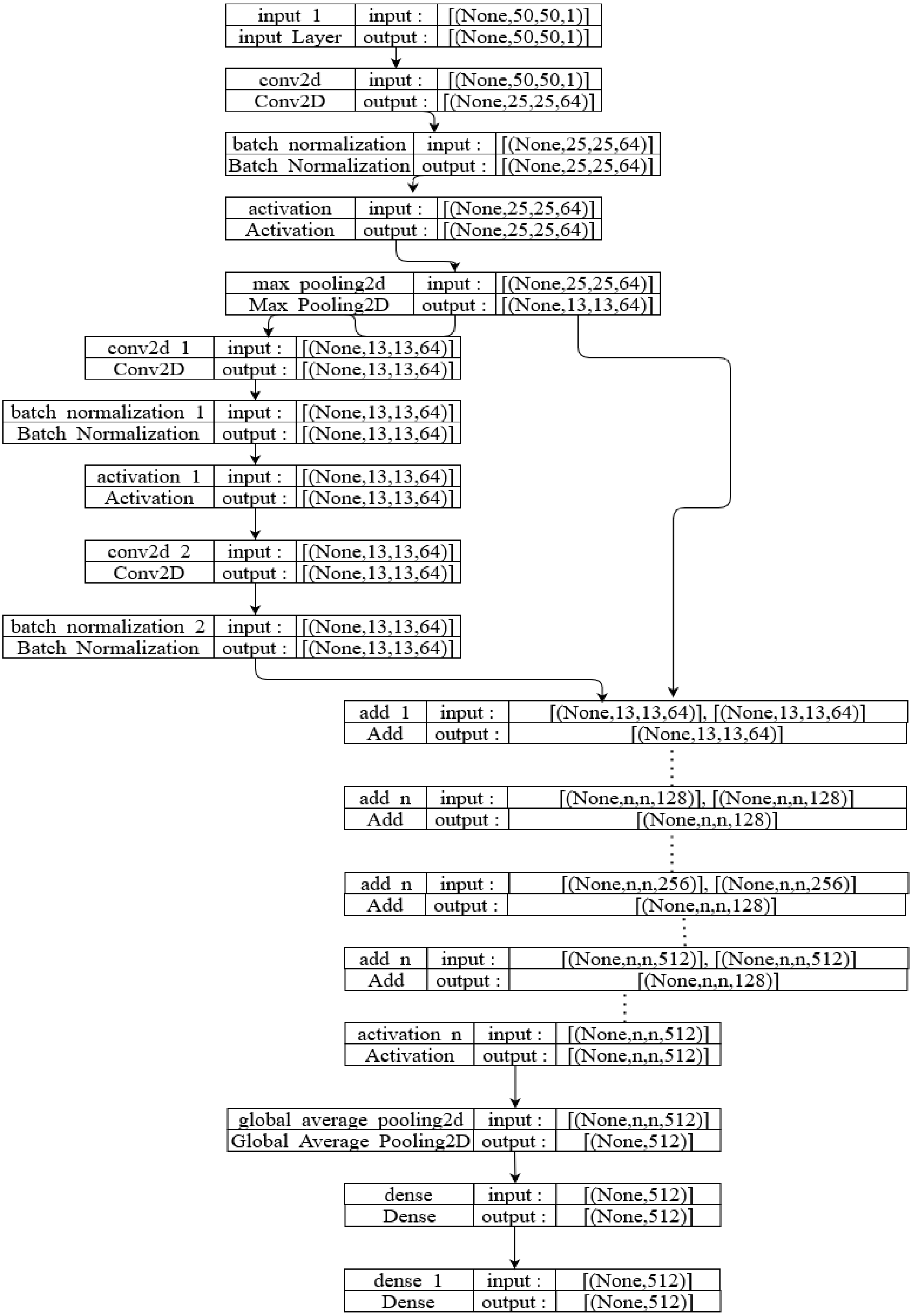
The concept of Residual Blocks was introduced to solve the problem of vanishing and exploding gradients. In this design, we use a technique called skip connections. Skip connections bypass certain layers, connecting the activations from one layer to later layers. This forms a residual block. Figure 3 shows a basic block for ResNet 34, the ResNet consists on one Convolutional layer, one pooling step followed by four layers of similar behavior.

Figure 3. Diagram of ResNet 34



3.3.2 Modified ResNet 152V2 Model Architecture

Figure 4: Diagram of the proposed ResNet 152V2 Model Architecture



3.4 Experimental Result Analysis

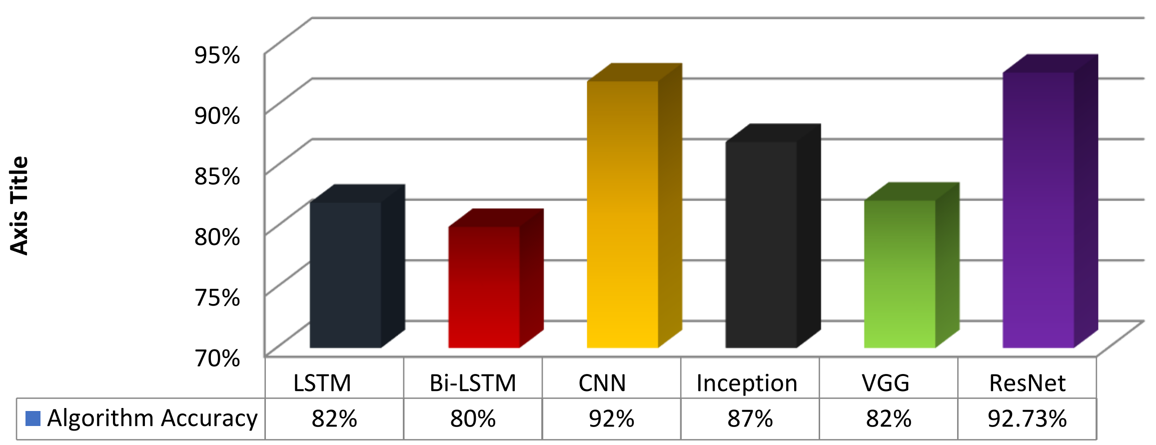
In this experimentation, we have trained some major deep learning models. Such as CNN, Inception, VGG, LSTM, Bi-LSTM And ResNet. These are the best deep learning models for character recognition. We trained all this models for 30 epochs. In these 30 epochs we got our best validation accuracy.

Table 2: Accuracy of different deep learning model

Sequence	Algorithm Name	Result and Accuracy
1	LSTM-Long Short-term memory	82%
2	Bi-LSTM	80%
3	CNN-Convolutional Neural Network	90%
4	Inception	87%
5	VGG-Visual Geometry Group	82%
6	ResNet	92%

ResNet (92%) and CNN (90%) demonstrate the highest accuracy among the tested algorithms, indicating their effectiveness in recognizing Bangla handwritten characters. Inception (87%) also performs well, although slightly lower than ResNet and CNN. LSTM (82%), VGG (82%), and Bi-LSTM (80%) exhibit moderate accuracy levels, suggesting reasonable performance but potentially with room for improvement compared to the top-performing models.

Figure 5: Algorithm Accuracy



ResNet and CNN are particularly suitable for BHCR, given their high accuracy rates. These models likely capture complex patterns and features within Bangla characters effectively. ResNet (92%) and CNN (90%) illustrate the most noteworthy precision among the tried calculations, showing their adequacy in recognizing Bangla written by hand characters. So, Working with ResNet for Bangla Handwritten Character Recognition (BHCR) is a strategic choice due to several compelling reasons. ResNet, short for Residual Neural Network, has demonstrated exceptional performance across various computer vision tasks, making it a promising candidate for complex pattern recognition tasks.

As compared to CNN, ResNet performed better in term of accuracy so we have tried different version of 50, 101, or 152 layer ResNet V1 and ResNet V2 models. The main distinction between ResNetV2 and ResNet (V1) lies in the fact that V2 executes batch normalization before every single weight layer.

Figure 6(a): ResNet50 Graph of Accuracy Graph and Loss Curve

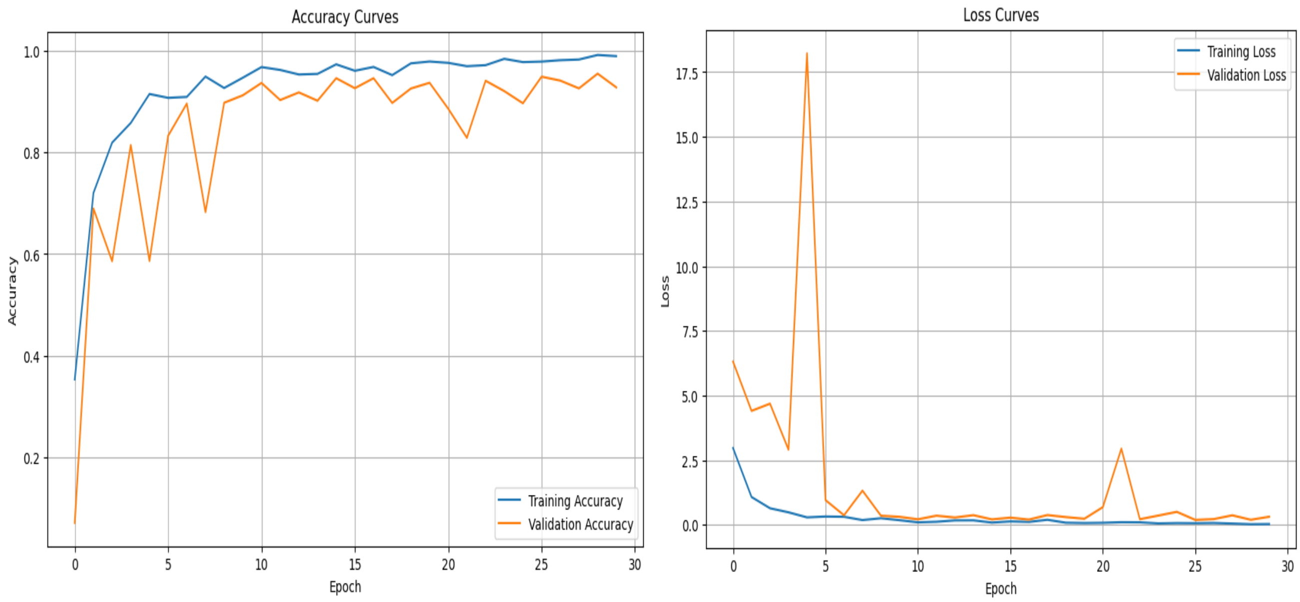


Figure 6(b): ResNet101 Accuracy Graph and Loss Curve

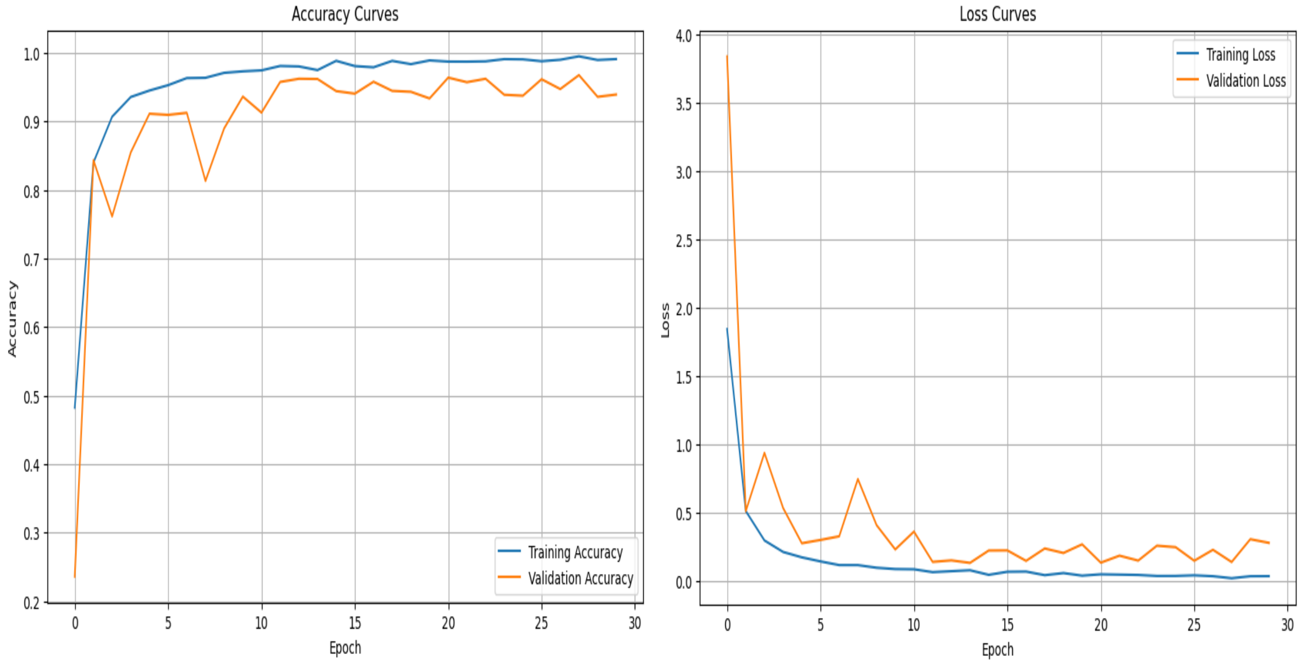


Figure 6(c): ResNet50V2Accuracy Graph and Loss Curve

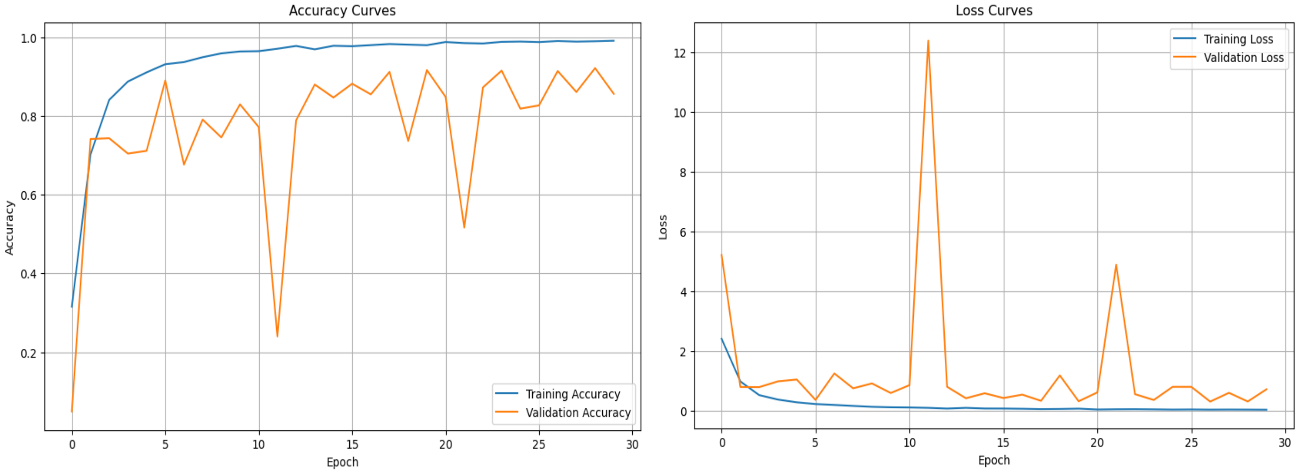


Figure 6(d): ResNet101V2 Accuracy Graph and Loss Curve

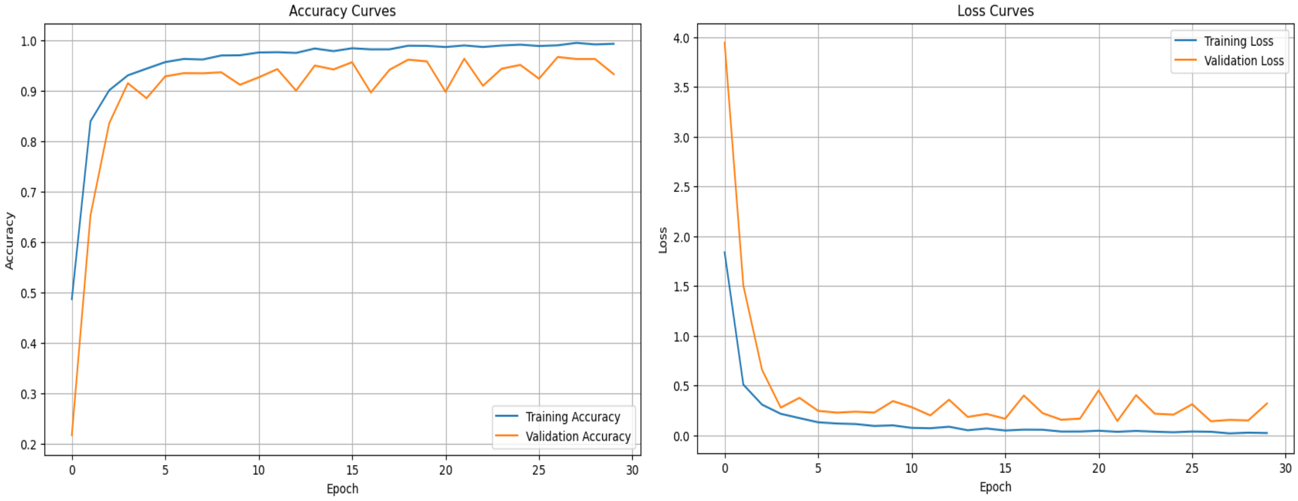


Figure 6(e): ResNet152 Accuracy Graph and Loss Curve

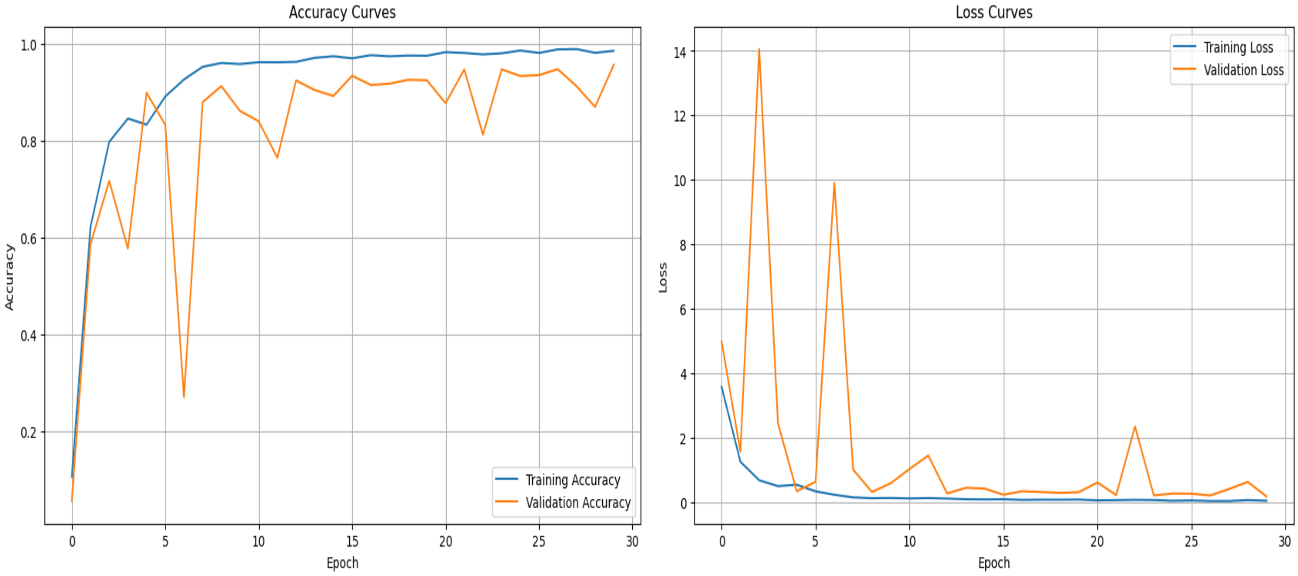
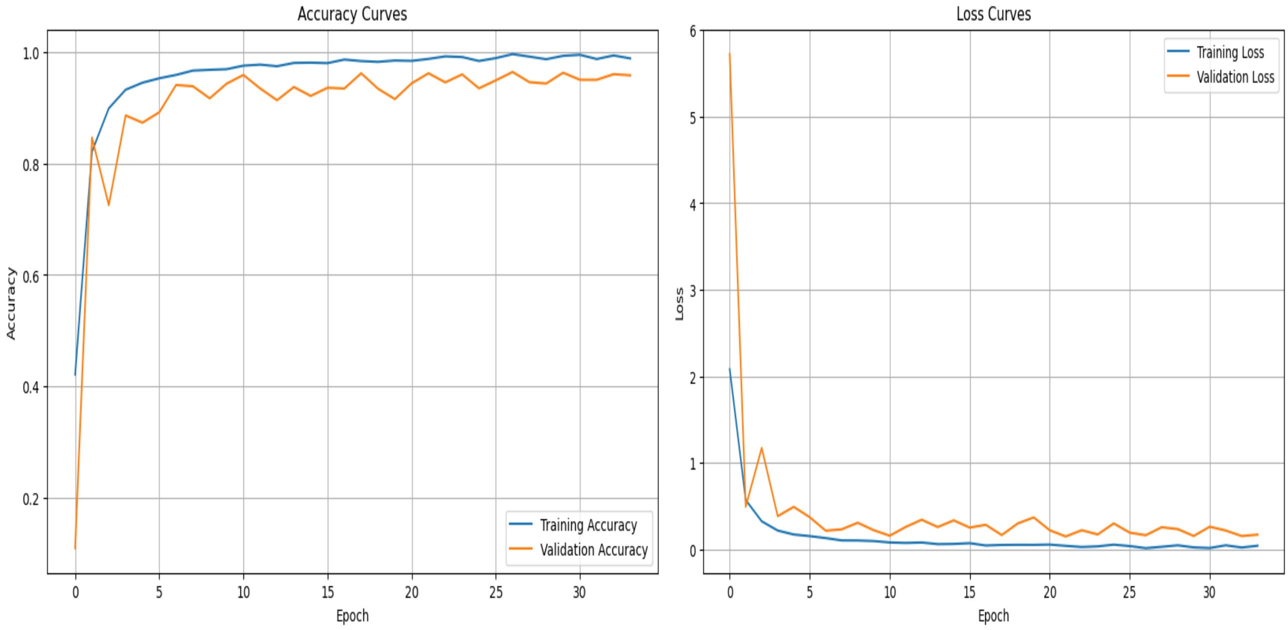
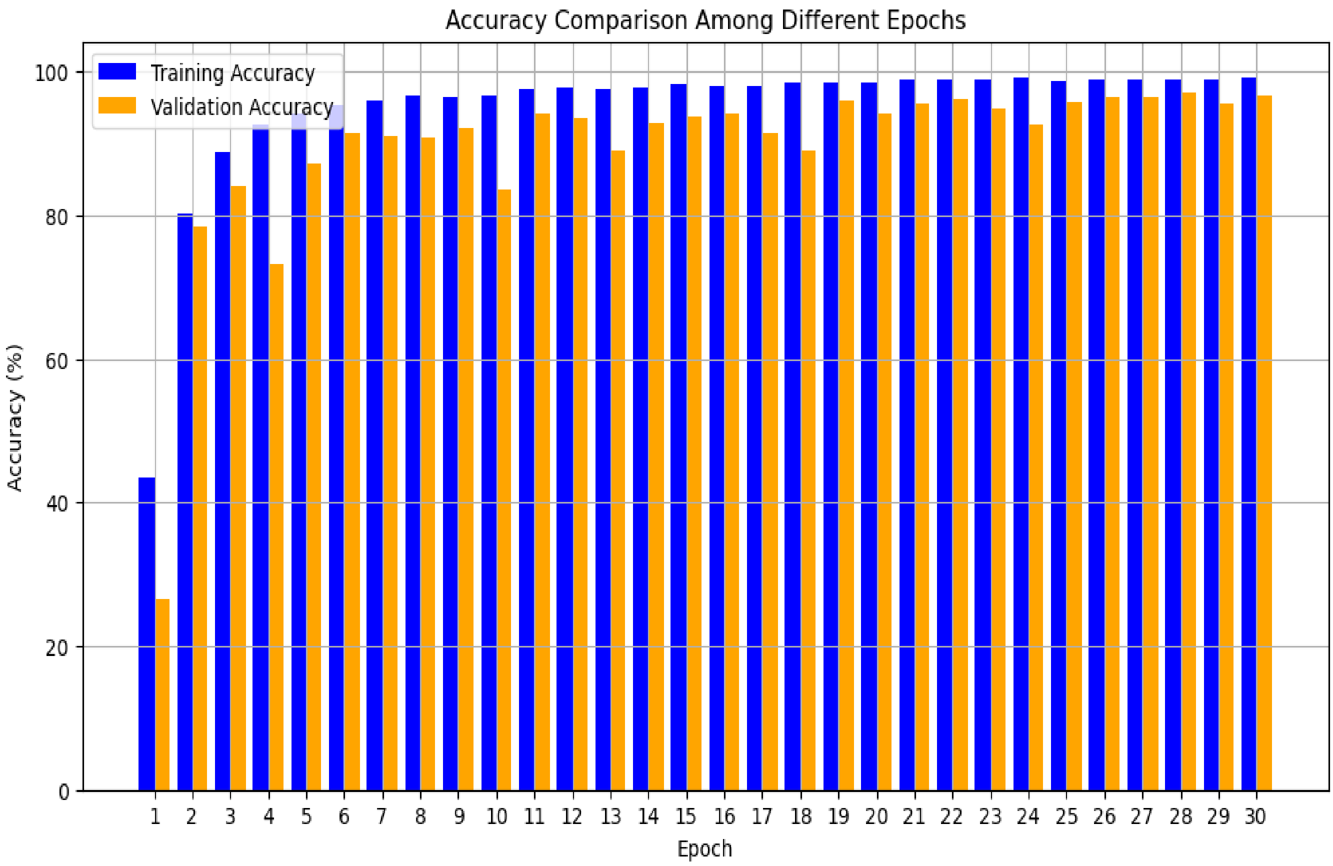


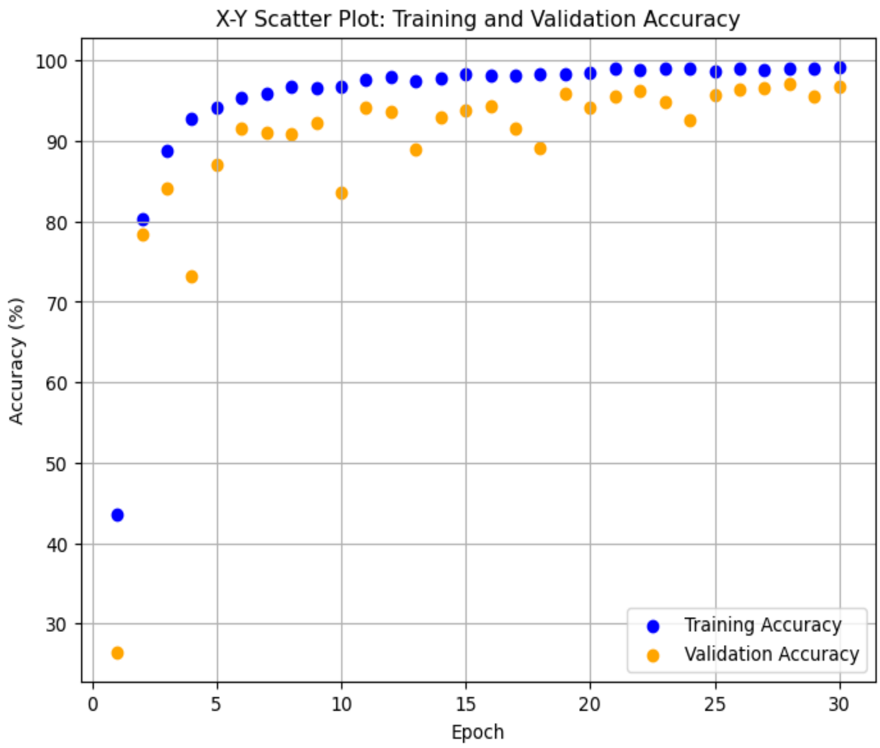
Figure 6(f): ResNet152V2 Accuracy Graph and Loss Curve



From the above figures we can say that ResNet152V2 has gained much more accuracy than other variant. So we have chosen this algorithm for our investigation.

Figure 7: Accuracy comparison in different epochs in ResNet152V2 model





A confusion matrix for each of the 50 classes is shown in Fig. 6. Lighter colors in the confusion matrix correspond to misclassified data for certain classifications. In a similar vein, Fig. 3 shows that certain classes have lower F1 scores than others. Fig. 7 lists a few of these characters together with the number of labels that have a comparatively higher mistake rate.

Figure 8: Heat map

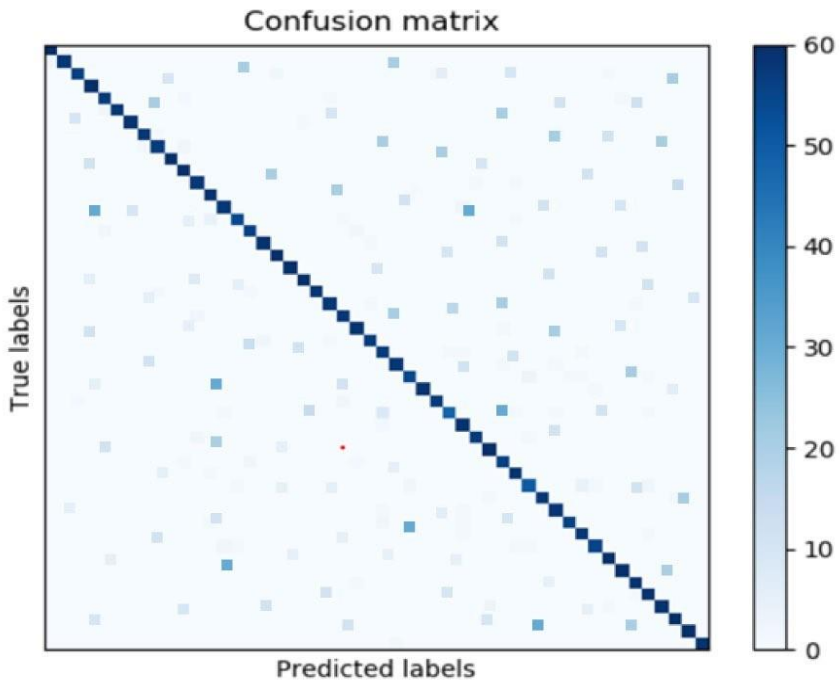


Table 3: Some Misclassified Character

Actual Class	Misclassified Class
ব (34)	ধ (30)
ড (24)	ভ (35), ড় (44)
উ (5)	ঊ (6), ড (24)
খ (13)	থ (28)
য (37)	ষ (41)

In terms of structure, all of the miss categorized classes are rather similar. Given that each person's handwriting is unique, it is much harder to distinguish between these characters owing to their proximity to one another. In spite of these details, the model performed properly.

Table 4: Results of the experiments for every class. (Here, F1 is the F1-score, R is recall, and P is precision.)

Char	F	R	F1	Char	F	R	F1	Char	F	R	F1	Char	F	R	F1	Char	F	R	F1
অ	.97	1.0	.98	ঔ	.94	1.0	.97	ঞ	1.0	.97	.98	ন	.98	.80	.88	ষ	.91	.97	.94
আ	1.0	.97	.98	ক	.94	.97	.95	ট	.97	.97	.97	প	.91	.98	.94	স	.96	.92	.94
ই	.98	.93	.96	খ	.91	.98	.94	ঠ	.95	.97	.96	ফ	.98	.95	.97	হ	.97	1.0	.98
ঈ	1.0	1.0	1.0	গ	.95	.97	.96	ড	.94	.98	.96	ব	.98	1.0	.99	ড়	.98	1.0	.99
উ	.93	.95	.94	ঘ	.96	.90	.93	ঢ	.97	.95	.96	ভ	.97	.93	.95	ঢ়	.97	.98	.98
ঊ	1.0	.97	.98	ঙ	.95	.93	.94	ণ	.83	.95	.88	ম	.94	.98	.96	য়	.92	.98	.95
ঋ	1.0	.98	.99	চ	.97	.98	.98	ত	.98	.97	.97	য	.91	.83	.87	ৎ	1.0	.98	.99
এ	.98	.97	.97	ছ	.97	1.0	.98	থ	.95	.90	.92	র	1.0	.97	.98	ং	.98	1.0	.99
ঐ	.97	.95	.96	জ	1.0	1.0	1.0	দ	1.0	.98	.99	ল	.95	.97	.96	ঃ	1.0	1.0	1.0
ও	1.0	1.0	1.0	ঝ	.98	1.0	.99	ধ	.98	.95	.97	শ	.98	.93	.96	ঁ	1.0	.98	.99

Figure 9: ResNet Vs ResNet Extended Versions Result

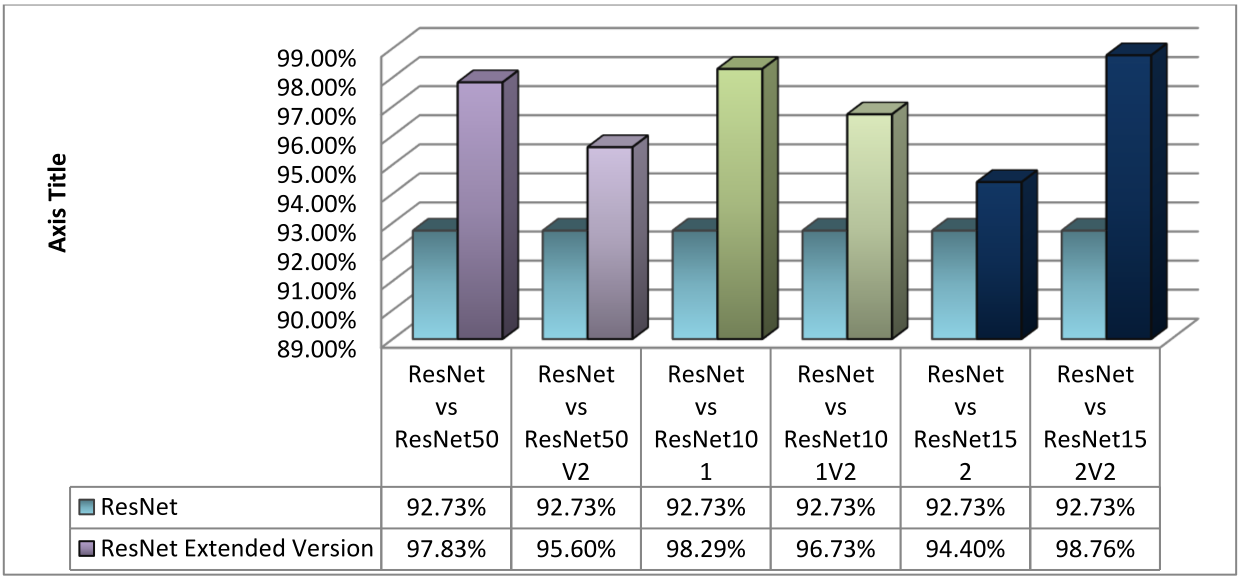


Table 5: Accuracy Comparison between Our Approach and Others Approaches

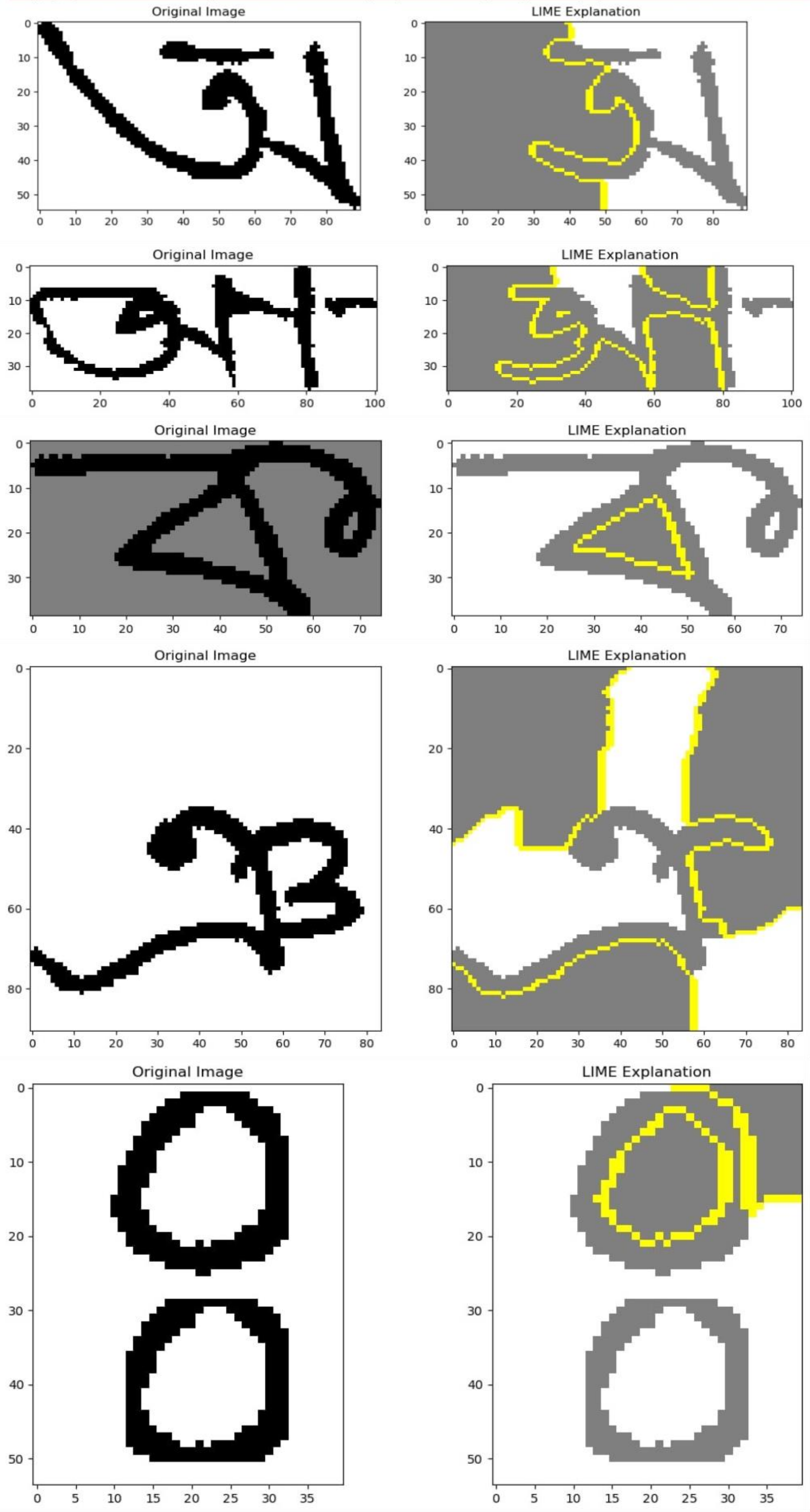
Ref. No.	Language	Data Set	Algorithm	Result and Accuracy
[12]	Bangla	CMATERDB	CNN	92.65%
[7]	Bangla	NumtaDB	GoogleNet	93%
[13]	Bangla	Bangla basic character database	CNN	95.84%
[8]	Bangla	Bangla numeral image database	CNN	85.96%
[9]	Bangla	Bangalekha isolated dataset, Ekush dataset	CNN	93.22%
[14]	Bangla	Banglalekha-Isolated dataset	BBCNet-15	91.40%.
[10]	Bangla	CMATERdb 3.1	DConvAENNet	97.53%
[15]	Tifinagh	AMHCD database.	CNNs	98.25%,
[11]	Bangla.	BanglaLekha-Isolated, CMATERdb 3.1, Ekush	DCNN	95.53%
[16]	Gujarati	Handwritten images (Set of 10 dataset)	K-NN	82.03%
Proposed work	**Bangla**	**CMATERdb 3.1**	**Modified ResNet152V2**	**98.76%**

Explainable AI

Performing LIME (Local Interpretable Model-agnostic Explanations) on a Bengali handwritten character recognition dataset involves using LIME to explain the predictions made by a deep learning model on these images. We can leverage LIME to gain interpretability into the predictions of a Bengali handwritten character recognition model. Adjustments was made to the preprocessing steps and LIME parameters based on specific requirements and characteristics of the dataset and model.

The widely used method known as LIME (Local Interpretable Model-agnostic Explanations) helps in understanding the predictions of complex machine learning models. LIME generates localized explanations to clarify the model's decision-making process, especially when it is unclear or difficult to interpret. It does this by mimicking the model's behavior around specific data instances. By perturbing and analyzing the input data near a given sample, LIME provides valuable insights into why the model made a particular prediction. This method enhances the transparency and reliability of machine learning models, making it easier for stakeholders and practitioners to comprehend and trust the model's decisions. Figure 10 illustrates the outcome of a LIME experiment, highlighting which features in an image most strongly influence the model's prediction. Additionally, LIME's output can be used for error analysis and debugging, showing the specific features that drive the model to make correct decisions across different classes. Each class has distinct identifying features, demonstrating how the model arrives at its conclusions.

Figure 10: LIME experiment outcome on different character



4 Conclusion

In this research, Bangla Handwritten Character Recognition (BHCR) was successfully carried out with the assistance of many feature extraction and categorization algorithms. Below is a summary of the F1 scores, recall, accuracy, and precision for each method of classification that was employed in this investigation. Our analysis indicates that the highest accuracy of the ResNet152V2 algorithm is 98.76%. In addition to advancing language processing technology, this research establishes the groundwork for future advancements in image analysis and document digitalization in the Bengali language space.

Our deep learning approach has showcased exceptional capabilities in accurately deciphering the intricate Bengali alphabet for handwritten character recognition in Bangla. We have obtained impressive accuracy rates through rigorous experimentation and sophisticated model modification, demonstrating the ability of deep learning to address the difficulties presented by complicated scripts like as Bangla**.**

References

1. Sarkar R, Das N, Basu S, Kundu M, Nasipuri M, Basu DK. “CMATERdb1: a database of unconstrained handwritten Bangla and Bangla–English mixed script document image”. International Journal on Document Analysis and Recognition (IJDAR). 2012 Mar;15:71-83.
2. Rabby AS, Haque S, Islam S, Abujar S, Hossain SA. “Bornonet: Bangla handwritten characters recognition using convolutional neural network”. Procedia computer science. 2018 Jan 1;143:528-35.
3. Roy S, Das N, Kundu M, Nasipuri M. “Handwritten isolated Bangla compound character recognition: A new benchmark using a novel deep learning approach”. Pattern Recognition Letters. 2017 Apr 15;90:15-21.
4. Weng Y, Xia C. “A new deep learning-based handwritten character recognition system on mobile computing devices”. Mobile Networks and Applications. 2020 Apr;25(2):402-11.
5. Chowdhury RR, Hossain MS, ul Islam R, Andersson K, Hossain S. “Bangla handwritten character recognition using convolutional neural network with data augmentation”. In2019 Joint 8th international conference on informatics, electronics & vision (ICIEV) and 2019 3rd international conference on imaging, vision & pattern recognition (icIVPR) 2019 May 30 (pp. 318-323). IEEE.
6. Biswas M, Islam R, Shom GK, Shopon M, Mohammed N, Momen S, Abedin A. “Banglalekha-isolated: A multi-purpose comprehensive dataset of handwritten bangla isolated characters”. Data in brief. 2017 Jun 1;12:103-7.
7. Basri R, Haque MR, Akter M, Uddin MS. “Bangla handwritten digit recognition using deep convolutional neural network”. InProceedings of the international conference on computing advancements 2020 Jan 10 (pp. 1-7).
8. Chakraborty P, Roy S, Sumaiya SN, Sarker A. “Handwritten Character Recognition from Image Using CNN. InMicro-Electronics and Telecommunication Engineering”. Proceedings of 6th ICMETE 2022 2023 Jun 2 (pp. 37-47). Singapore: Springer Nature Singapore.
9. M. A. F. M. R. H. A. F. M. Z. A. a. N. F. D. A. Hossain, "Bangla Handwritten Characters Recognition Using Convolutional Neural Network.," I.J. Image, Graphics and Signal Processing, 2015, 8, 52-59.

10. Azad MA, Singha HS, Nahid MM. “Bangla handwritten character recognition using deep convolutional autoencoder neural network”. In2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT) 2020 Nov 28 (pp. 295-300). IEEE.
11. Shamim SM, Miah MB, Sarker A, Rana M, Al Jobair “A. Handwritten digit recognition using machine learning algorithms”. Indonesian Journal of Science and Technology. 2018;3(1):29-39.
12. Rabby AS, Abujar S, Haque S, Hossain SA. “Bangla handwritten digit recognition using convolutional neural network”. In Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2018, Volume 1 2019 (pp. 111-122). Springer Singapore.
13. Maitra DS, Bhattacharya U, Parui SK. “CNN based common approach to handwritten character recognition of multiple scripts”. In2015 13th International Conference on Document Analysis and Recognition (ICDAR) 2015 Aug 23 (pp. 1021-1025). IEEE.
14. Saha C, Faisal RH, Rahman MM. “Bangla handwritten basic character recognition using deep convolutional neural network”. In2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR) 2019 May 30 (pp. 190-195). IEEE.
15. Sadouk L, Gadi T, Essoufi EH. Handwritten tifinagh character recognition using deep learning architectures. InProceedings of the 1st International Conference on Internet of Things and Machine Learning 2017 Oct 17 (pp. 1-11).
16. Joshi DS, Risodkar YR. Deep learning based Gujarati handwritten character recognition. In2018 International conference on advances in communication and computing technology (ICACCT) 2018 Feb 8 (pp. 563-566). IEEE.