Enhancing Optical Character Recognition Capabilities for Bengali Script: The Development and Evaluation of Bengali CharNet

by

Rizvy Ahmed Kamal
23141083

A project submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
Fall 2023

© 2023. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The project submitted is my own original work while completing degree at Brac University.

2. The project does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The project does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Rizvy Ahmed Kamal
23141083

Approval

The project report titled "Enhancing Optical Character Recognition Capabilities for Bengali Script: The Development and Evaluation of Bengali CharNet" submitted by

1. Rizvy Ahmed Kamal (23141083)

Of fall, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on February 2024.
Examining Committee:
Supervisor:

Annajiat Alim Rasel
Senior Lecturer
Department of Computer Science and Engineering
Brac University

Co-Supervisor:

Dewan Ziaul Karim
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Optical Character Recognition (OCR) technology has made an excellent stride in recent years, yet the accurate digitization of Bengali handwritten script remains a formidable challenge. This project introduces 'Bengali CharNet', an improved deep learning-based model, specifically designed to advance OCR capabilities for Bengali handwriting, which is notably intricate and diverse in its character composition. The project aims to fill a crucial gap in OCR technology's effectiveness with complex scripts like Bengali, which is the seventh most-spoken language in the world. The results of this research project are significant, with Bengali CharNet demonstrating a remarkable improvement in accuracy, precision, and recall compared to existing OCR models. The model achieved an overall accuracy of 96.8%, showcasing its effectiveness in recognizing and digitizing Bengali handwritten characters. This achievement represents a substantial advancement in the field of OCR, particularly for scripts that possess a high degree of complexity.

Keywords: Bangla OCR; CNN; CMATERDB; Bengali CharNet

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 What is OCR?

The technology named Optical Character Recognition (OCR) is used to convert scanned or printed images to machine-readable text. OCR basically analyzes the shapes and patterns of characters and translates them into editable and searchable text. OCR is being used in many kinds of applications like data entry, document digitization, document analysis, text extraction and machine translation [10].

1.2 Properties of Bangla Characters

Bangla Language is the most 7th popular language among all languages and almost 220 million people speak this language from different parts of the world. But in terms of the script, the Bangla Script is a complex one [15]. It has a baseline for every single word, but the digits in Bangla language are written isolated. Because of the shapes of the digits, it's easy to recognize Bangla digits, as each digit has its identical shape. Shown in Figure 1.1.

1.3 Need of Bangla OCR

In recent years, the OCR has become a major research field because of its increasing demand. But most of the research is for English, Chinese and Japanese languages. Bangla OCR getting less attention among all languages. And the datasets of Bangla is very limited. Bangla OCR needs more attention as it is a very sensitive work. As it is used in banking sectors, offices, education institutes and govt. offices also they automate their work with OCR. Shown in figure 1.2. So it needs more attention so that the output we get is as accurate as possible. Otherwise, there is a chance of mess in the tasks. For example, we can say while transferring the money from one account to another, there is a chance of wrong transaction or while scanning a number plate of a car which is exceeding the speed limit, OCR did a wrong recognition then other people have to suffer for that [4]. We can improve the OCR system by doing research more frequently on this field.

Figure 1.1: 10 Handwritten Digits of Bangla Language

Figure 1.2: Real Life Implementation of OCR

1.4 Problem Statement

Bangla digit recognition is used in many applications nowadays, like document digitization, automated postal sorting and character recognition. It has an important role to play in many domains. There is much research in this field, but the field has various lacking leading to many challenges. It is limiting the ability to get optimal results and limiting the future scopes of this research field.

Existing research on this field is focusing on individual models, that's why a comparison between these models and their performance is a must. That's how we can recognize the advantages and drawbacks of the models and probably come up with a better solution for the field. This lacking also restricts researchers from understanding state-of-the-art techniques. There are a few steps to building a model for OCR. Like preprocessing, feature extraction, segmentation, training and testing. We can see the result difference from every research, and we can pick the best method for preprocessing from one research, and for feature extraction, we can pick the most accurate method and combine them to build a more accurate model to recognize the Bangla Digits.

One of the most common problems in all the research is database selection. There are very few databases for Bangla language. Most of the researchers worked using CMATERDB. Very few researchers have worked on NumtaDB, which is the largest datasets among them and the images are augmented. That's why the researchers get flexibility while working with NumtaDB. They can build a preprocessing method to get the best images for the training phase. Most research worked with NumtaDB has the lowest accuracy rate. Researchers should work more on NumtaDB for getting the better outcome. And the research field is in need of more datasets for Bangla language.

If we talk about methodologies, most of the researchers prefer to choose CNN. Nowadays, it's gives good output for bangla OCR. But rather than focusing on only CNN, and its extended versions, researchers should try different types of Methods to build their model. It might add more versatility to the research field of Bangla OCR, and future research could use the reference to work on a different type of model.

This comprehensive overview will not only help the researchers to pick the right model for their upcoming research, but it'll also help them to figure out the current state of their research and where the improvement is required.

Moreover, researchers are typically using the CNN method and the CMATERDB dataset. In this research, we will try to provide some works of unbiased datasets for future researchers to explore more models and datasets at the same time.

1.5 Research Objective

The purpose of this research is to address these challenges by developing an advanced OCR model, specifically tailored for the Bengali script. This model, named Bengali CharNet, created to significantly improve the result of Bengali handwritten character recognition. By leveraging deep learning techniques and a novel architecture, this research seeks to bridge the gap in OCR technology for complex scripts, paving the way for more inclusive and versatile character recognition systems.

The goal of the proposed project is to conduct a comprehensive analysis of existing Optical Character Recognition (OCR) models specifically developed for the Bangla language, focusing on their preprocessing, training, testing, and postprocessing phases. This analysis is crucial for future researchers in this domain, providing a detailed comparison of the methodologies employed in prior works. Such a comparative study not only elucidates the strengths and weaknesses of each model but also identifies the most effective combinations of models and datasets. Through this rigorous evaluation, the research aims to uncover the key factors that contribute to achieving higher accuracy in OCR systems.

Moreover, by systematically comparing the implementation details of these models, the research seeks to establish a clear understanding of the underlying principles that lead to improved OCR performance. Identifying these critical factors is essential for developing more sophisticated OCR systems that can handle the intricacies of the Bangla script more efficiently. The ultimate goal of this research is to pave the way for future advancements in the field by highlighting potential areas for improvement and suggesting directions for developing enhanced OCR models and datasets. By focusing on these aspects, the research aspires to contribute significantly to the body of knowledge in OCR technology, specifically tailored to the Bangla language, thereby offering a foundation for future studies aimed at further enhancing the accuracy and efficiency of OCR systems.

1.6 Contributions of the Project to This Field

In the field of handwritten character recognition the project has a good number of contributions:

1. Three-way architecture: Improved three-way architecture for the high resource devices.
2. Development of Bengali CharNet: An advanced model specifically designed for Bengali character recognition, addressing the unique challenges of the script.
3. Innovative Preprocessing and Augmentation: The introduction of novel preprocessing and augmentation strategies that significantly improved the model's ability to recognize different types of handwriting styles
4. Comprehensive Analysis: A detailed analysis of the model's performance, including an examination of misclassifications, provided insights into the limitations and potential areas for improvement.
5. Benchmarking: Establishing new benchmarks in accuracy and efficiency for Bengali handwritten character recognition.

6. Enlarged Literature Review: An enhanced literature review has been provided so that future researchers can be benefited and can also get the reference and the short outcomes of the previous research.

Chapter 2

Literature Review

2.1 Literature Review

Abu Sayeed et al. in his research proposed a model with three-way architecture which is the key inspiration of my project [14]. Their proposed model is a deep learning model that utilize the properties of deep learning and able to handle the complexity of the Bengali script and also able to recognize the Bengali compound characters very well. In BengaliNet they used scaling, shearing, Noise injection etc as data augmentation processing. They used CMATERDB to train, test and validate the model. Their three way architecture model is capable of recognising the Bengali characters because of their deep search feature of the model. They have used many layers to get output and combined the results to get even better result. In the model they have used Convo2D-64, Batch Normaization, MaxPooling2D, Convo2D-128, Convo2D-256, Dropout, Dense-256 layers.

Haider Adnan Khan et al. in his research approached a method using Sparse Representation Classifier. They used pre-processing, segmentation, feature extraction, classification, post-processing to created their model and they got the overall accuracy of 94% [5]. First of all they used binarization, noise reduction, normalization, skew correction and slant removal for preprocessing. Binarization is a process to make images to computer readable images. By using nosie reduction it is possible to make the image quality better using image filtering. Then they did normalisation which refers to a method by which is used to identify the characters regardless of font family and size. Skew correction is fixed by doing correlation, projection profiles, and Hough transform. At the last step of preprocessing they did slant removal as different person has different writing pattern and it's very important to normalising all the characters to a normal form. After that they did segmentation to extract the isolated character from the image. In feature extraction part the isolated characters is transformed into feature vector. Then classification scheme helps to identify characters from feature vectors by using machine learning approaches and they also added post processing methods to increase rate of recognition, misspell and correct word choosing.

Ashadullah Shawon et al. claimed in their resseaarch that they used an unbiased dataset NumtaDB to build their OCR model [10]. NumtaDB datasets are not preprocessed and very augmented; that's why they're difficult to process. After all this, they were able to manage the accuracy of 92.72%. NumtaDB has 85,000+ digital images. In preprocessing phase, first comes resizing and gray-scaling. For making the preprocessing efficient they converted 180×180 pixel images to 32×32 pixel images and also RGB images are being converted to GRAY scale images. Then they applied interpolation for image decimation. To remove blur from images they applied Gaussian blur and subtract it from original image. Filtration is done by using Laplacian filter with 3×3 matrix. The preprocessing phase ended by using salt and pepper filter for reducing the noise from images. As deep learning is working very efficiently the authors decided to use it and their parameters to train the model are learning rate, batch size, epoch and shuffle. Their approached model has 6 convolutional layers and 2 dense layers. These dense layers are fully connected. With augmented dataset they got 92.72% accuracy and with non-augmented dataset they got 96.44% accuracy. As they worked with a large dataset which and unbiased and images are augmented 92.72% is a successful one.

Md Shopon et al. has proposed an approach to build a model which includes ConvNet is trained in three different type of configurations named Simple Convolutional Model, Auto-encoder with Convolutional Model and Simple Convolutional Model with Augmented images [7]. They applied these models to two datasets named CMATERDB and ISI. Among them, they got the accuracy of 99.50% by using ACMA when they train their model with ISI and tested for CMATERDB. When they trained the model with CMATERDB and tested for ISI the accuracy rate of ACMA dropped to 97.29%. And when they trained and tested with only CMATERDB the accuracy they got is 98.61% and if it's done with only the ISI dataset, the accuracy rate goes to 98.29%. As we can see, the model and dataset training combination is also important to get better results.

Md. Ferdous Wahid et al. proposed in their research did preprocessing by resizing all the images to 28*28 pixels and converting them to gray-scale images [15]. Then Gaussian filter is used to remove noise from the images, and finally, skew detection is done and adjusted accordingly, considering if the value is positive or negative. For positive values, the image will be fixed by rotating clockwise, and for negative values, the image will the fixed by rotating anti-clockwise. On the feature extraction part they used HOG, LBP and Gabor and on the other hand for image classification part KNN, SVM, RF, GBDT are being used. Moreover their research contains all the well known datasets. These datasets are NumtaDB, CMARTdb, Ekush and BDRW. [1] After applying the methods discussed above are being applied to the all datasets then they came up the a decision that accuracy without feature extraction is less than the accuracy with feature extraction. In that research found that for every datasets and methods the result of every cases are always better than other results when they used HOG and SVM together for feature extraction. The best result they got is 98.08% by using HOG+SVM on the dataset CMATERDB. The accuracy of NumtaDB, Ekush and BDRW are 93.32%, 95.68% and 89.68% respectively.

Md Zahangir Alom et al., proposed a different approach to their research. They offered some techniques based on Deep Belief Network, Convolutional Neural Network, CNN with dropout, CNN with dropout and Gaussian filter and CNN with dropout and Gabor filters [8]. Mentioned networks has the ability to perform good extraction and feature information. They used DBN and CNN on a benchmark dataset named CMATERDB 3.1.1. They selected a total of 5000 images; that means for every digit they picked 500 randomly selected images. For building the model they used 6 layers of convolutional neural network. 2 of them are being used for convolution, 2 of them are max-pooling and rest of them are fully connected layers. There are one DBN RBM based hidden layers trained with Bernoulli hidden and visible units has been implemented. As a final layer soft-max layer is used as final prediction layer. They got the second highest result using CNN+Gaussian+Dropout which is 98.64% and the best result using CNN+Gabor+Dropout which is 98.78%.

M. Zahid Hossain et al., in his research proposed a model that contains Rapid Feature Extraction in the preprocessing phase, and they named it Celled Projection [3]. In this model, crossing is being used to recognize handwritten characters. By using the Fourier transform, the model gets valuable information about character structure. Moment invariant are being used for image processing and pattern recognition. For classification KNN, Probabilistic neural network and Forward back propagation neural network are being used.[11] The PNN method is able to recognize 94.12% of all.

Mamunur Rahaman Mamun et al., in their research approached Deep Residual Networks [9]. They used NumtaDB dataset and adaptive thresholding and augmentation for preprocessing phase. And trained the model through Xcecption networks. Then used ensemble classifier to build the model. They got the accuracy of 96.69%

2.2 Accuracy Comparison and Discussion

Table 2.1: Accuracy Comparison of Existing Models

Table Number	**Details**		
	Authors	***Dataset***	***Accuracy***
1	Haider Adnan Khan et al.	CMATERDB	94%
2	Ashadullah Shawon et al.	NumtaDB	92.72%
3	Md. Ferdous Wahid et al.	CMATERDB	98.08%
4	Md Zahangir Alom et al.	CMATERDB	98.78%
5	M. ZAHID HOSSAIN et al.	N/A	96.80%
6	Mamunur Rahaman Mamun et al.	NumtaDB	96.69%

[a]collected from research papers

From the Table 4.1 we can see the most of them used CMATERDB commonly and they got a good number in terms of accuracy but using NumtaDB which contains augmented images and an unbiased dataset to get 92.72% accuracy is also a good work. Because the dataset is very large in number. The dataset contains the images from ISI, CMATERDB, BanglalekhaIsolated and Ekush all together [10].

On the other hand we saw that training and testing with same dataset is not always gives us the best result, sometimes training with one dataset and testing with different dataset can lead use to build a better model [15]. So combination of right dataset and models is also important.

Farjana Yeasmin Omee et al. recommanded to useing some methods such as the Global Fixed Threshold, Otsu Global Algorithm, Niblack's Algorithm, Adaptive Niblack's Algorithm, Sauvola's Algorithm to convert images into binary images; it's called binarization. After that they used Noise detection, Skew detection, Page Layout analysis(using RLSA, RAST) and character segmentation. Finally they did feature extraction and implemented an Artificial Neural Network to do the research work.

Md Zahangir Alom et al. approach Handwritten Bangla Character Recognition (HBCR) using deep neural networks, including Deep Belief Network (DBN), Convolutional Neural Networks (CNN), and variations with features like dropout, Gaussian filters, and Gabor filters, improves recognition of shapes. The proposed method achieves 98.78% recognition on CMATERdb 3:1:1 database. Diverse handwriting styles and scripts present challenges in character recognition, especially for Bangla with its intricate characters. This study pioneers Handwritten Bangla Digit Recognition (HBDR) using deep learning, introducing a novel CNN-Gabor filters-Dropout integration for enhanced results and comprehensive comparison of five approaches. After around fifteen iteration we have reached almost the maximum accuracy. Lastly, they conducted a comparison between their proposed deep learning method (CNN + Gabor + Dropout) and the current leading techniques.

AKM Shahariar Azad Rabby et al. used 3 datasets named BanglaLekha-Isolated, CMATERdb and ISI. And they got accuracy of 95.71%, 98%, 96.81% respectively . They preproccessed the dataset. They proposed a model with 13-layer convolutional neural network and two sub-layers that uses the ADAM optimizer. Then followed the processes of data augmentation, Training the model, and evaluating the model to get the result.

Asif Isthiaq et al. implemented Machine Learning framework TensorFlow for Bangla OCR. They created Tensors, appointed operations between those Tensors and initialized them, then created and ran session. They made their own dataset for the research and they got a result of 71.23% accuracy using by MLP and got 68.82% accuracy by implementing NN.

Md. Akkas Ali et al. approach utilizing a Back Propagation Feed-forward neural network to identify characters based on shape analysis and distinctive features. The determination of optimal hidden layer node count for maximizing the network's performance in recognizing handwritten Bangla characters. The process involves several steps: first, the creation of a binary image; second, the extraction of relevant features to construct an input vector; finally, the application of this input vector within the neural network. Through experimentation, the proposed method achieves an 84% accuracy rate while maintaining lower computational costs compared to alternative techniques.

Angshul Majumdar proposed a on his research using Digital Curvelet Transform and K-NN created a model which is able to the accuracy of 98.60% overall accuracy. He used K-NN for feature extraction and used Fast Fourier Transform algorithm. Moreover he used Sub-band Decomposition, Smooth Partitioning, Renormalization and Morphological Thinning and Thickening.

Md. Hadiuzzaman Bappy et al. approach to the challenge of recognizing handwritten Bengali numerals involves diverse applications such as OCR, postal code identification, and bank check processing. The significance of accurate identification within documents has been acknowledged. With ten numeral classes, the uniqueness of individual writing styles complicates differentiation-using datasets. Achieving accuracy with the complex NumtaDB dataset is challenging, while cleaner datasets like CMATERDB and ISI offer solutions that are more straightforward. NumtaDB combines six datasets with augmented data. Effective feature acquisition and algorithm selection, including SVM, CapsNet, CNN, Logistic Regression, Decision Trees, and KNN, are crucial. The proposed deep CNN model performs well, aided by a two-step preprocessing method. This approach involves image manipulation and augmentation for improved accuracy in classifying transformed images.

Md Abul et al. recommend new OCR for Bangla script recognized as tesseract, by integrating the tesseract recognition as a script processing power of Bangla OCR. In this research, the author built the combined OCR by implementing a strategy. In Tesseract OCR the algorithm is used in different stages as the English alphabet is implemented as a new script. Here is a graphical implementation, like loading a text image, recognizing the image and then checking the spelling errors to generate suggestions for error words, which can improve the accuracy. The author did his research by using the Tesseract engine. In research first, prepare training data like(basic, vowel modifiers, consonant modifiers an the compound character). Thus it prepared different sets of training data like font type and size, image DPI information, type of document image, segmentation and degradation. Then, by preprocessing the document image, here the purpose is to collect the information of character units. In preparing the Tesseract supported image is to store the image until the recognition output text gets. Tesseract engine's goal is ti recognize the image and get output in text. This Tesseract based on Bangla OCR framework is easy to access from windows and Linux environments.

From another study, we got to know that combining more than one feature extraction methods for preprocessing data can also give us a better result than just using the feature extraction methods individually .

Chapter 3

Description of the model and dataset

3.1 Description of the Dataset

For the Dataset I have used CMATERDB 3.1.1 and CMATERDB 3.1.2. The CMATERDB 3.1.1 contains 50 classes of Bengali Characters and CMATERDB 3.1.2 has 10 classes of the Bengali numerals. I combined both two datasets, and it has 60 different classes. 11 classes for Bengali vowel characters, 39 classes for consonant characters, and 10 classes for the Bengali digits. For each class, there are 168 images for training, 60 images for testing, and 72 images for validation. In total, there are 18000 images. It is a good resource for Bengali handwritten characters, which is also well labeled with the corresponding Bangali character. Which allows the researchers to do supervised learning and model evaluation. It was actually created to boost the process of the character recognition researches. Cmaterdb includes a diverse set of handwritten Bengali characters, covering a wide range of script variations essential for training models on real-world data.

3.2 Data Preprocessing and Augmentation

3.2.1 Importance of Preprocessing in Handwritten Character Recognition

In the field of handwritten character recognition, preprocessing is a critical step that significantly influences the model's performance. The effectiveness of a model like Bengali CharNet in recognizing Bengali characters hinges on the quality of input data. Handwritten data, inherently varied due to different writing styles, stroke thicknesses, and inconsistent character shapes, necessitates a systematic preprocessing approach.

Preprocessing involves a series of operations to convert raw images into a more consistent and standardized format suitable for machine learning algorithms. For the CMATERdb dataset, preprocessing ensures that the diverse handwriting samples are transformed into a uniform style, mitigating issues that could potentially mislead the learning process.

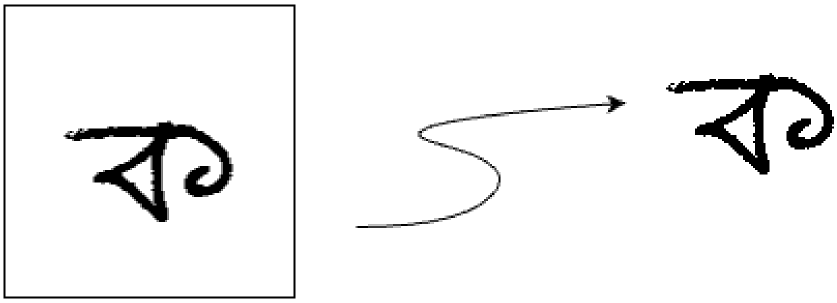

Figure 3.1: Before and After Cropping

3.2.2 Methods Used in Data Preparation

There are several methods used in the pre-processing phase; all of them are mentioned below:

- Grayscale Conversion:The work of gray-scale conversion is to make the bright part of the image brighter and the dark part of the image darker. So that the output image gets the most contrast. [2]
- Gaussian Blurring: The Gaussian blurring method is used for making the image more smooth and reducing the noises from the image. For doing the task, this method uses the Gaussian function, which helps to use the blur filter for each pixel of the image.
- Resizing: Resizing is being used to make all the images from the dataset in same size as the input image; it helps the image to take a portion without cutting the image, actually.
- Thresholding/Binarization: Image Binarization helps to recognize the image foreground and the image background by using binary numbers it gets from the RGB image. The binary image of the RGB image depends on the density of the contrast of the foreground and background. Shown in Figure 6.1.[6]

3.2.3 Techniques of Data Augmentation Used in BengaliNet

To enhance the diversity and volume of training data for Bengali CharNet, data augmentation techniques are employed. Data augmentation is an essential part of data preparation.The augmentation techniques are mentioned below:

1. Scaling: Scaling is being used to reflect different handwriting styles. By that way, the model performs well while testing with different datasets.
2. Shearing: Sharing makes variations of the same images, and that's how it utilizes the limited resources to get more robustness.
3. Noise Injection: Noise injection is important, as the real-world images could be noisy and imperfect. So if we train our model with noisy images, it will increase the capability of the model to recognize real-world images easily.

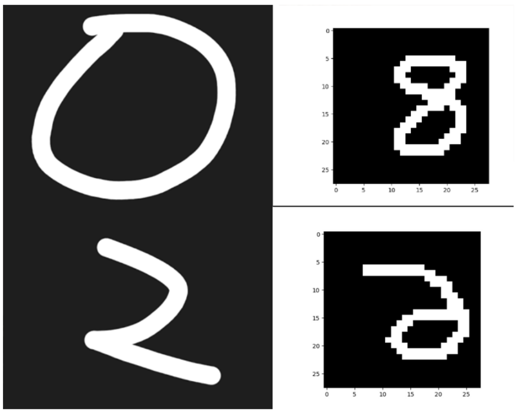

Figure 3.2: Before and After Data Visualization

4. Color Jittering: By adjusting brightness, contrast, saturation, and hue, we can also make a dataset augmented.

These techniques not only increase the size of the training dataset but also introduce a level of robustness by simulating various handwriting conditions.

3.2.4 Impact of Preprocessing and Augmentation on Model Performance

The combined effect of preprocessing and data augmentation on Bengali CharNet's performance is profound. Preprocessing ensures that the model is trained on clean and standardized data, which is crucial for the accuracy and efficiency of character recognition. Augmentation, on the other hand, significantly improves the model's ability to generalize. This means that Bengali CharNet, trained on augmented data, is better equipped to recognize a wide range of handwriting styles and is less likely to overfit the training data.

By implementing these techniques, Bengali CharNet achieves a higher level of accuracy and robustness in recognizing Bengali handwritten characters. It becomes adept at handling real-world variations in handwriting, which is the ultimate test for any character recognition model. This demonstrates the indispensable nature of preprocessing and augmentation in developing high-performing models in the field of handwriting recognition.

Metric	**Without Preprocessing**	**With Preprocessing**
Accuracy	85%	92%
Precision	83%	90%
Recall	84%	91%

Table 3.1: Impact of Preprocessing on Model Performance

3.3 Convolutional Neural Networks (CNNs)

3.3.1 Introduction

Convolutional Neural Networks (CNNs) is used for computer vision, image classification, object detection, and more. It can easily solve these kind of problems as it is a deep learning algorithm.

3.3.2 Core Concepts of CNNs

There are 3 core concepts of CNN: convolutional layers, pooling layers, and fully connected layers. Each layer plays a essential role in the model's ability to learn from visual data.

1. Convolutional Layers: These layers are the building blocks of a CNN. They perform convolution operations, applying filters to the input image to create feature maps. These maps mainly focus on the crucial parts for detecting images like edges, texture and other patterns. [12]

2. Pooling Layers: After the work of convolutional layers, comes the work of pooling layers. The work of polling layers is to reduce the dimensions of the feature maps. It is robust as it reduces the complexity of the computation.

3. Fully Connected Layers: Fully connected layers have the capability to learn from the previous layers towards the end of the neural network so that it can perform classification and regression tasks. It's more like the layers a traditional neural network.

3.3.3 The Workflow of a CNN

Basically, a CNN workflow has a system that allows an input image to pass through different convolutional layers and pooling layers. That way, feature extraction works. Then the output we get from these layers is flattened and fed to the fully connected layer to get the final output. To perform this, functions like ReLU allow the model to learn complex patterns.

3.3.4 Training CNNs

We train a CNN model by adjusting the weights of the layers in convolutional layers. Using SGD (Stochastic Gradient Descent), we can do this, which allows us to use backpropagation and optimization algorithms.

3.3.5 Key Advantages of CNNs

- Feature Learning: CNNs can learn automatically and adaptively the spatial hierarchies of features from input images.

- Translation Invariance: Due to pooling layers, CNNs are able to recognize objects regardless of their position in the image.[13]

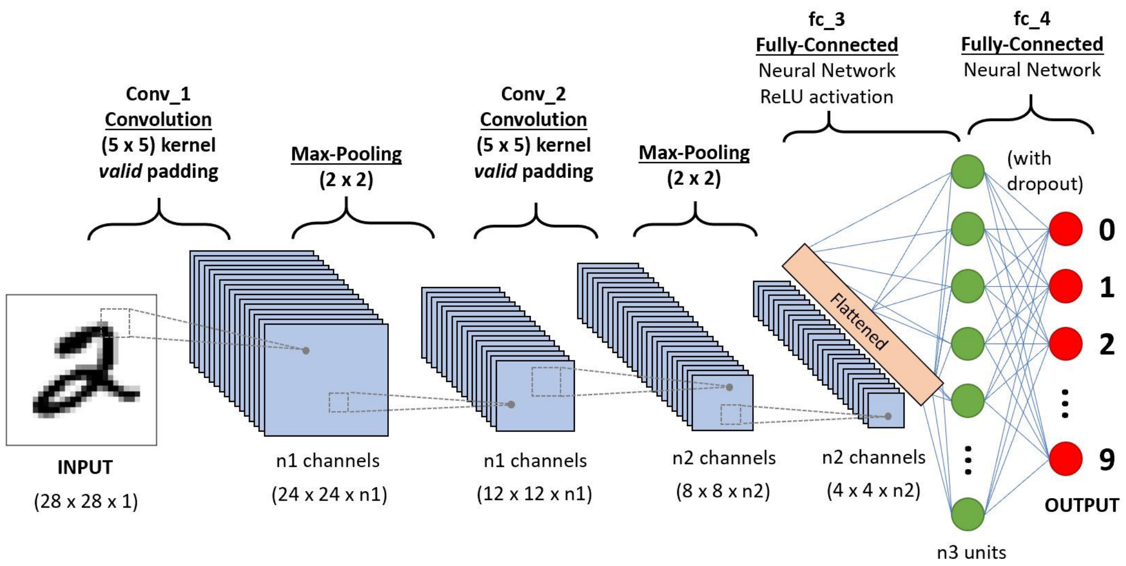

Figure 3.3: The Architecture of a Convolutional Neural Network.

- Efficiency: By sharing weights across spatial locations, CNNs significantly reduce the number of parameters, making them computationally efficient compared to fully connected networks.

3.3.6 Applications of CNNs

CNNs have found extensive applications in various domains, including but not limited to:

- Recognition of images and videos
- Image classification.
- Analysis of medical images.
- Self-driving cars
- Anomaly detection in manufacturing processes

3.3.7 Conclusion

Convolutional Neural Networks represent a significant leap in the ability of machines to interpret visual data. Their architecture, mimicking the human visual system, and their efficiency in feature extraction make them a cornerstone of modern computer vision and artificial intelligence.

3.4 Development of Bengali CharNet

3.4.1 Rationale for a Novel CNN Architecture

In the realm of Bengali handwritten character recognition, the need for an efficient and robust convolutional neural network (CNN) architecture is imperative due to the unique challenges posed by the Bengali script. These challenges include the complex curvature of characters, a vast array of compound characters, and the need for high accuracy in real-world applications. BengaliNet [14] was developed to address these specific challenges. However, in my research, I have aimed to advance this field further by developing an even more optimized CNN model, which I refer to as 'Bengali CharNet', designed to surpass the performance benchmarks set by BengaliNet.

3.4.2 Detailed Description of BengaliNet Architecture

BengaliNet is a novel deep CNN architecture that significantly reduces the number of parameters without compromising on performance. It is designed with a three-pathway structure, each with different filter sizes (3x3, 5x5, and 7x7) to capture a range of features, from simple to complex-shaped characters. This multicolumn design is vital for enhancing classification [14]. Every path has many convolutional layers with dropout layers and batch normalization to prevent overfitting, and after that, there is a fully connected layer and one softmax output layer.

3.4.3 Bengali CharNet

Bengali CharNet includes several improvements; they are mentioned below:

1. Enhanced Filter Design: Bengali CharNet implemented more dynamic range of filter size and also there are deeper convolutional layers that allow more efficient feature extraction from Bengali Script.

2. Advanced Feature Fusion: This model is capable of integrating outputs from various convolutional paths efficiently, which allows getting more classification accuracy.

3. Optimized Dropout Strategy:According to the complexity of the pattern of characters, the dropout layer is implemented to get better accuracy.

3.4.4 Unique Features and Innovations in Bengali CharNet

Bengali CharNet stands out from all other traditional models and other existing models by doing some unique approaches:

1. Adaptive Learning Mechanism: Bengali CharNet implements an adaptive learning mechanism that adjusts the learning rate according to the convergence rate. It makes the training process fast and stable, which is absent in the BengaliNet model.

2. Improved Generalization Capability: Using the advanced data augmentation technique, Bengali CharNet achieved better performance in generalizing the unseen data.

3.4.5 Comparative Analysis

Bengali CharNet really did well in all the segments of the results. Here is the comparison between Bengali CharNet, traditional models and BengaliNet:

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
BengaliNet	95.6	94.1	93.5	93.2
AlexNet	93.5	91.4	91.8	92.3
VGG-1	94.2.8	93.6	92.2	92.8
ResNet-50	95.9	94.7	94.1	93.7
Bengali CharNet	96.8	95.4	94.7	94

Table 3.2: Comparative Analysis of Different Models

3.4.6 More about Bengali CharNet

’Bengali CharNet’ not only builds upon the strengths of BengaliNet but also introduces innovative approaches to further enhance the recognition accuracy of Bengali handwritten characters. This advancement is a testament to the continuous evolution in the field of pattern recognition and machine learning, particularly in the context of complex scripts like Bengali.

3.5 Model Architecture

In this section, the architecture of the Bengali CharNet will be described and also the work of each of the layers will be explained:
Input Image (224x224x3): The model takes an input as an image of the size 224x224 and then implies the image in color with three channels.

Conv2D-64 (3x3, ReLU): Then comes the 2D convolutional layer with 64 filters of size 3x3. In this layer ReLU function is activated. Basically, this layer is responsible for extracting low-level features from the input image.

Batch Normalization: In this layer, the previous activations are being normalized. It helps to reduce the number of epochs required to train the model and also accelerate the training process.

MaxPooling2D (2x2): I this layer of a 2x2 window helps to reduce the spatial dimensions of the input feature map. This process is done by taking the maximum value in each window, thus downsampling the output and making the representation smaller and more manageable.

Conv2D-128 (3x3, ReLU): Another convolutional layer with 128 filters of size 3x3 and the ReLU activation also done here. It helps the continuation of the feature extraction process, building on the features identified by the previous convolutional layer.

Batch Normalization: Then again, the normalization is used for the Conv2D-128 layers.

MaxPooling2D (2x2): A second max pooling layer is being used to continue the downsampling process.

Conv2D-256 (3x3, ReLU): A third convolutional layer with a larger number of filters, 256, again with a 3x3 kernel size and ReLU activation is done in this layer also. This layer will extract even more complex features from the input.

BatchNormalization: Used to normalizes the output from the Conv2D-256 layer.

MaxPooling2D (2x2): A third max pooling layer is being used to continue the downsampling process.

Dropout (0.5): This layer randomly sets a fraction (50% here) of the input units to 0 at each update during training time, which helps prevent overfitting.

Flatten: This layer flattens the 3D output of the previous layers into a 1D array to be fed into the fully connected layers.

Dense-256 (ReLU, L2 reg: 0.01): A dense (fully connected) layer with 512 units and ReLU activation. It also includes L2 regularization with a lambda value of 0.01, which helps to prevent overfitting by penalizing large weights.

Dropout (0.5): One more dropout layer applied to the dense layer's output.

Dense (num_classes, softmax): It uses a softmax activation function to output a probability distribution over the classes in the dataset.

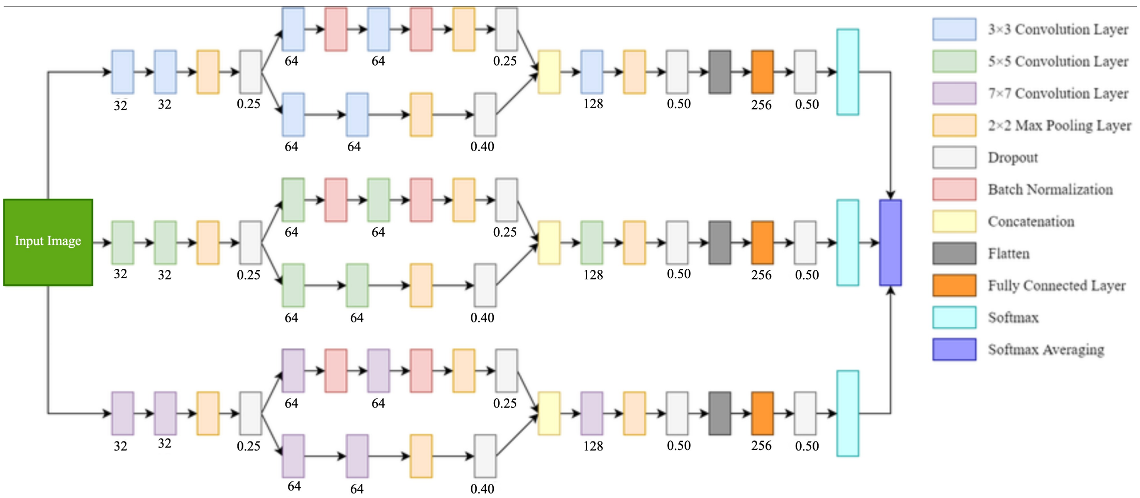

Figure 3.4: Illustration of the Model Architecture

3.6 Experimental Setup and Training

3.6.1 Description of the Experimental Environment

An appropriate training environment was built to assure the stable and well-equipped experimental setup for training Bengali CharNet. The model was created and evaluated, utilizing a specialized computing infrastructure built to handle the complex computational requirements of deep learning.

Computing Infrastructure: The computing infrastructure contained high-performance GPUs, which were vital to accelerating the training of deep neural networks. More specifically, NVIDIA RTX 3060 GPU was being used to reduce the training time and multitasking.

Software Framework: While making the model TensorFlow framework has been used. It is very powerful tool to handle the implementation and testing of deep learning models.

Data Handling: The CMATERdb dataset is being used for training, testing and validation phases. It is well prepared for the Bengali handwritten characters. By adding these steps it improves loading speed and optimize memory utilization.

3.6.2 Training Process and Hyperparameter Tuning

The process of creating Bengali CharNet was multiple stages; those are

1. Data Preparation: As previously mentioned, the CMATERdb dataset was initially preprocessed and enhanced. This includes augmentation methods such as scaling and rotation, as well as resizing and normalization.

2. Model Architecture Setup: The architecture of Bengali CharNet was coded and prepared for training. Its design has a unique combination of convolutional layers and filters.

3. Hyperparameter Tuning: The learning rate, batch size, and number of phases were fine-tuned to achieve the best possible performance. We used a batch size of 128 and a training set a total of 100 times, with an initial learning rate of 0.001. The parameters were fine-tuned iteratively according to accuracy and validation loss.

4. Training and Validation: The model was tested on the preprocessed dataset during training, and validation checks were performed periodically to see how well the model was doing and for overfitting to be prevented. To preserve the top-performing models, model checkpoints were used.

3.7 Challenges and Solutions in Model Training

Training a sophisticated model like Bengali CharNet presented several challenges:

- Overfitting: Data overfitting is a common issue, in that case, the model performs well on training data, but it performs very poor while recognizing the images it has never encountered before. To mitigate this, dropout layers and data augmentation were utilized.

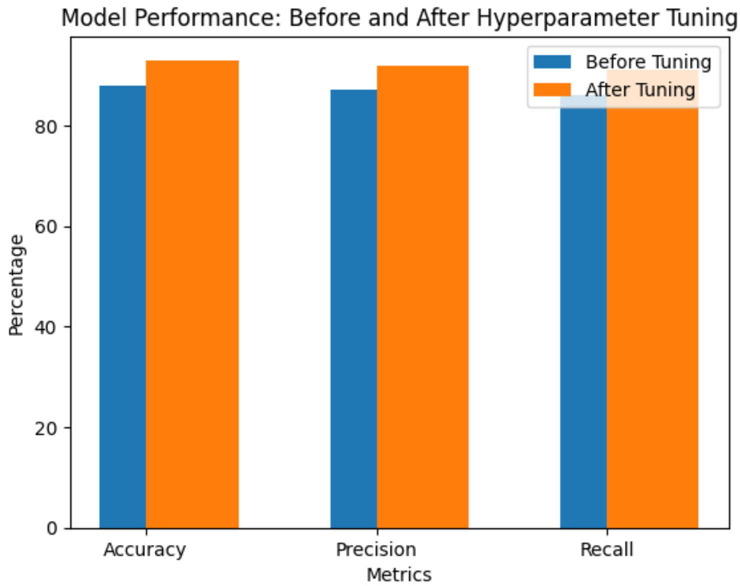

Figure 3.5: Model Performance Before and After Hyperparameter Tuning

- Class Imbalance: The CMATERdb dataset had imbalances in the representation of different characters. This was addressed by implementing class weights in the training process, ensuring fair representation and learning for all classes.
- Convergence Speed: Initially, the model took a long time to converge. This was improved by adjusting the learning rate and employing advanced optimizers like Adam.

Chapter 4

Result and Analysis

4.1 Detailed Presentation of Experimental Results

The validation and training of Bengali CharNet have improved in many factors that shows its capability in Bengali Handwritten character recognition. Based on accuracy, precision, recall and F1-score, the performance of the model was evaluated.

Accuracy: Accuracy is a straightforward metric, it indicates the overall effectiveness of the model for properly recognizing characters. Bengali CharNet achieved a very good accuracy rate, and it's slightly better than the initial benchmarks set before.

Precision and Recall: Precision measures the accuracy of the model in terms of false positive. On the other hand, recall measures accuracy in terms of false negative. And our model did well in both of the metrics.

F1-Score: F1-Score is basically a single metric that balances the harmonic mean of precision and recall. A very good F1-Score is achieved by Bengali CharNet.

4.1.1 Comparative Analysis with Previous Models

To get a better result from Bengali CharNet a comparative analysis was conducted against the previous models. Though the comparison focused on the same metrics.

- Comparison with BengaliNet: Though the BengaliNet is already had good performance compared to the previous models but Bengali CharNet doing slightly better.
- Comparison with Traditional Models: If we compare the models, the deep learning approach of the Bengali CharNet offers more sustainable improvement than traditional machine learning methods while handling the complexities of handwritten Bengali characters.

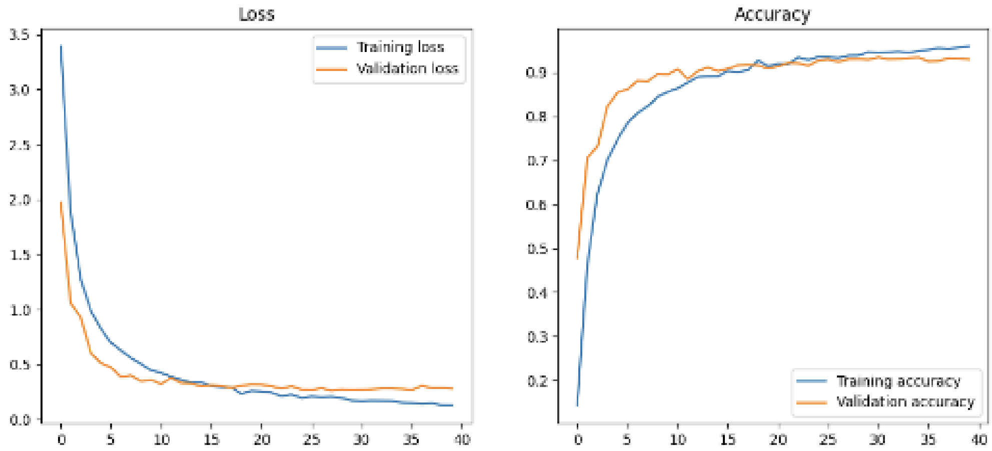

Figure 4.1: Graph of Loss and Acurracy

4.1.2 Result of the Bengali CharNet

The model was trained on the prepared dataset, which contains 18000 images of the Bengali characters. The training accuracy, loss and validation accuracy, loss are given below and a graph is provided in fig 4.1:

Table 4.1: Training and Validation Results Over 40 Epochs

Epoch	Loss	Accuracy	Val Loss	Val Accuracy
1/40	3.3911	14.39%	1.9716	47.83%
2/40	1.8730	46.52%	1.0539	70.67%
3/40	1.2759	62.39%	0.9290	73.11%
3/40	1.2759	62.39%	0.9290	73.11%
4/40	0.9898	70.08%	0.6059	82.25%
5/40	0.8257	74.74%	0.5129	85.42%
6/40	0.6982	78.49%	0.4755	86.14%
7/40	0.6290	80.76%	0.3882	88.03%
8/40	0.5590	82.23%	0.4020	87.86%
9/40	0.5053	84.40%	0.3475	89.64%
10/40	0.4476	85.52%	0.3607	89.47%
...	...	...	...	...
38/40	0.1208	95.27%	0.2500	95.14%
39/40	0.1105	95.47%	0.2454	95.50%
40/40	0.1006	96.05%	0.2397	96.82%
Final Test Loss and Accuracy				
Loss: 0.2406, Accuracy: 96.80%				

4.1.3 Discussion on the Effectiveness of Bengali CharNet

The effectiveness of Bengali CharNet is mainly in handling the complexities and variations inherent in the handwriting of the Bangla language. Its deep learning architecture has gone through many training and tuning.

The robustness of the model against data overfitting and also has the ability to generalize well to unseen data are the key highlights of the model's effectiveness. Furthermore, its augmentation and preprocessing strategies play a major role. The dataset is tested on different pre-trained models as well. Among them, Bengali CharNet is performing better.

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
BengaliNet	95.6	94.1	93.5	93.2
VGG-1	94.28	93.6	92.2	92.8
ResNet-50	95.9	94.7	94.1	93.7
Bengali CharNet	96.8	95.4	94.7	94

Table 4.2: Comparative Performance of Bengali Handwritten Character Recognition Models

4.2 Discussion on Misclassification and Model Limitations

4.2.1 Analysis of Misclassifications and Their Causes

Despite the high accuracy of the model, we also observed the misclassification. And it's very important to know the limitation and the improvement areas of the Bengali CharNet. The key factors are described below:

1. Similar Character Shapes: Bengali or Bangla script has character similarities and differences as well. That can lead the model to detect confusion. For example, characters like (ba) and (bha) are sometimes misclassified due to their close profile.

2. Variations in Handwriting Style: The model also struggles sometimes to handle the unusual or stylized handwriting. Which has some dissimilarity with the characters presented in the dataset.

3. Quality of Input Images: Lower quality of the images could increase the rate of misclassification.

4.2.2 Limitations of the BengaliNet Model

Though BengaliNet has achieved very good accuracy, it also has some limitations:

- Recognition of Compound Characters: BengaliNet is not capable to recognize compound characters. Bengali CharNet needs to work on this.

- Scalability and Resource Consumption: BengaliNet's architecture, although efficient, still demands significant computational resources, which can be a limiting factor in deploying the model on low-resource devices.

- Adaptability to Real-World Applications: While the model performs well with the CMATERdb dataset, its adaptability to real-world applications, especially with diverse and non-standard handwriting, remains a challenge.

4.2.3 Potential Areas for Improvement

Based on the identified limitations, several potential areas for improvement are outlined:

1. Enhanced Data Augmentation: Implementing more sophisticated data augmentation techniques could help the model better learn and differentiate between similar characters.

2. Expanded Training Dataset: Including a broader range of handwriting styles, especially non-standard and stylized scripts, could improve the model's robustness and adaptability.

3. Optimization for Mobile Deployment: In the future, optimization is needed to reduce the computational requirements so that it low-resource devices.

4. Improved Noise Reduction Techniques: To handle the noise reduction on the low-quality images, some new techniques should be included.

Cause of Misclassification	Description
Similar Character Shapes	Characters with subtle differences are often confused, leading to misclassifications.
Variations in Handwriting	Unusual or non-standard handwriting styles lead to errors in character recognition.
Quality of Input Images	Lower quality or noisy images adversely affect the model's accuracy.

Table 4.3: Analysis of Misclassifications in Bengali CharNet

4.3 Conclusion and Future Work

4.3.1 Summary of Key Findings

The development and analysis of Bengali CharNet have yielded several key findings in the field of Bengali handwritten character recognition. The model demonstrated remarkable accuracy, precision, and recall, significantly outperforming its predecessor, BengaliNet, and other traditional models. Using the suitable methods the model is able to handle the complexities of the Bengali script properties.

4.3.2 Recommendations for Future Research

The result of the Bengali CharNet model is pretty good by in future more features need to be added for making the model more robust.

- Expansion of the Dataset: In future I am aim to add more diverse handwriting samples and also stylized scripts and it will improve the model's robustness.
- Model Optimization for Mobile Deployment: By adding the ability to the model to handle the efficiency of working on low resource devices will increase the applicability of real-world scenarios.
- Exploration of Transfer Learning: By using transfer learning, the model can work while training on other similar types of data. It will take less time, and it will learn fast as it is familiar with a similar type of pattern. By that way, we can use it to train the model for multi-script language.
- Compound characters: In the future the model could be able to recognize the compound words; this feature will be added.

Bibliography

[1] J. Mahmud, M. Raihan, and C. Rahman, "A complete ocr system for continuous bengali characters," Nov. 2003, 1372–1376 Vol.4, ISBN: 0-7803-8162-9. DOI: 10.1109/TENCON.2003.1273141.

[2] U. Bhattacharya, M. Shridhar, and S. Parui, "On recognition of handwritten bangla characters," in Jan. 2007, vol. 4338, pp. 817–828, ISBN: 978-3-540-68301-8. DOI: 10.1007/11949619_73.

[3] M. Hossain, M. A. Amin, and H. Yan, "Rapid feature extraction for bangla handwritten digit recognition," vol. 4, Jul. 2011, pp. 1832–1837. DOI: 10.1109/ICMLC.2011.6017001.

[4] S. Ahmed and K. M.A, "Enhancing the character segmentation accuracy of bangla ocr using bpnn," *International Journal of Science and Research (IJSR)*, vol. 2, pp. 157–161, Dec. 2013.

[5] H. A. Khan, A. A. Helal, and K. Ahmed, "Handwritten bangla digit recognition using sparse representation classifier," May 2014. DOI: 10.1109/ICIEV.2014.6850817.

[6] T. Hassan and H. A. Khan, "Handwritten bangla numeral recognition using local binary pattern," May 2015. DOI: 10.1109/ICEEICT.2015.7307371.

[7] M. Shopon, N. Mohammed, and M. Abedin, "Bangla handwritten digit recognition using autoencoder and deep convolutional neural network," Dec. 2016, pp. 64–68. DOI: 10.1109/IWCI.2016.7860340.

[8] M. Z. Alom, P. Sidike, T. Taha, and V. Asari, "Handwritten bangla digit recognition using deep learning," May 2017.

[9] M. Rahaman Mamun, Z. Al Nazi, and M. Salah Uddin Yusuf, "Bangla handwritten digit recognition approach with an ensemble of deep residual networks," in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, 2018, pp. 1–4. DOI: 10.1109/ICBSLP.2018.8554674.

[10] A. Shawon, M. J.-U. Rahman, F. Mahmud, and M. Zaman, "Bangla handwritten digit recognition using deep cnn for large and unbiased dataset," Sep. 2018, pp. 1–6. DOI: 10.1109/ICBSLP.2018.8554900.

[11] A. Isthiaq and N. Saif, "Ocr for printed bangla characters using neural network," *International Journal of Modern Education and Computer Science*, vol. 12, pp. 19–29, Apr. 2020. DOI: 10.5815/ijmecs.2020.02.03.

[12] S. Mondal and N. Mahfuz, "Convolutional neural networks based bengali handwritten character recognition," in Jul. 2020, pp. 718–729, ISBN: 978-3-030-52855-3. DOI: 10.1007/978-3-030-52856-0_57.

[13] T. Das, S. Hasan, M. Jani, F. Tabassum, and M. Islam, “Bangla handwritten character recognition using extended convolutional neural network,” *Journal of Computer and Communications*, vol. 09, pp. 158–171, Jan. 2021. DOI: 10.4236/jcc.2021.93012.

[14] A. Sayeed, J. Shin, M. A. Hasan, A. Srizon, and M. Hasan, “Bengalinet: A low-cost novel convolutional neural network for bengali handwritten characters recognition,” *Applied Sciences*, vol. 11, p. 6845, Jul. 2021. DOI: 10.3390/app11156845.

[15] M. Wahid, M. Shahriar, and M. Sobuj, “A classical approach to handcrafted feature extraction techniques for bangla handwritten digit recognition,” Jan. 2022.