Shahjalal University of Science and Technology
Department of Computer Science and Engineering

A Deep Convolutional Neural Network and Transfer Learning based approach for recognition of handwritten Bangla characters (Basic, Compound and Numeric)

REYAD HASAN RINKON
Reg. No.: 2017331034
4^{th} year, 1^{st} Semester

UMME HABIBA
Reg. No.: 2017331055
4^{th} year, 1^{st} Semester

Department of Computer Science and Engineering

Supervisor
ENAMUL HASSAN
Assistant Professor
Department of Computer Science and Engineering

12th November, 2022

A Deep Convolutional Neural Network and Transfer Learning based approach for recognition of handwritten Bangla characters (Basic, Compound and Numeric)

A Thesis submitted to the Department of Computer Science and Engineering, Shahjalal University of Science and Technology, in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering.

By

Reyad Hasan Rinkon
Reg. No.: 2017331034
4^{th} year, 1^{st} Semester

Umme Habiba
Reg. No.: 2017331055
4^{th} year, 1^{st} Semester

Department of Computer Science and Engineering

Supervisor
ENAMUL HASSAN
Assistant Professor
Department of Computer Science and Engineering

12th November, 2022

Recommendation Letter from Thesis/Project Supervisor

The thesis/project entitled *A Deep Convolutional Neural Network and Transfer Learning based approach for recognition of handwritten Bangla characters (Basic, Compound and Numeric)* submitted by the students

1. Reyad Hasan Rinkon
2. Umme Habiba

is under my supervision. I, hereby, agree that the thesis/project can be submitted for examination.

Signature of the Supervisor:
Name of the Supervisor: Enamul Hassan
Date: 12th November, 2022

Certificate of Acceptance of the Thesis/Project

The thesis/project entitled *A Deep Convolutional Neural Network and Transfer Learning based approach for recognition of handwritten Bangla characters (Basic, Compound and Numeric)* submitted by the students

1. Reyad Hasan Rinkon
2. Umme Habiba

on 12th November, 2022, hereby, accepted as the partial fulfillment of the requirements for the award of their Bachelor Degrees.

Head of the Dept. Mohammad Abdullah Al Mumin,PhD Professor Department of Computer Science and Engineering	Chairman, Exam Committee M. Jahirul Islam,PhD Professor Department of Computer Science and Engineering	Supervisor Enamul Hassan Assistant Professor Department of Computer Science and Engineering

Contents

List of Tables

List of Figures

Abstract

Handwritten characters are more difficult to recognize than printed ones. The size and form of a handwritten character written by a variety of individuals vary. Numerous variations in writing styles complicate the character recognition mechanism. The recognition of Bangla handwritten characters has previously been the subject of several scholars. Recent advancements in the domains of image-based identification, computer vision, and natural language processing have been made possible by the convolutional neural network's (CNN) unique ability to extract and classify features. The outcome of this study is a deep convolutional neural network (DCNN)-based method for detecting Bengali handwritten characters. In the area of pattern recognition, a deep, efficient architecture that can analyze a vast quantity of data is one of the most effective approaches to achieving better accuracy or a less error rate. Therefore, DCNN was employed for both feature extraction and classification in this work. This research evaluates their accuracy by applying the DCNN model to three datasets: BanglaLekha-Isolated, CMATERdb, and Ekush data. By using DCNN , 98.07%accuracy has been attained In addition, CNN, DCNN, and more complex VGG-16 models for classification have been considered. Finally, the resulting precisions from each dataset have been assessed.

Keywords: Character recognition, Handwritten character recognition, Bangla Handwriten character recognition, Neural Network, Deep Convolutional Neural Network

Acknowledgements

We would like to express our deepest gratitude and admiration to our thesis advisor, Enamul Hassan, for his constant support, assistance, inspiration, and skilled guidance during the duration of his supervision. His abundant resources and practical ideas in the field of computer vision have enabled me to work with exhaustive datasets and state-of-the-art theories and approaches.I am indebted to him for the opportunity to do research under his supervision.

Chapter 1

Introduction

The way we communicate with one another is via language. Language includes sentences, which are built from words and grammar rules. A word is made up of separate characters. A character is the smallest unit of any language. In today's world of rapid digitisation, it's more important than ever to devote effort into enabling computers to understand human languages. It's possible that this will make our lives easier by automating a number of activities that now demand a significant amount of human effort, labor, and resources. To achieve this objective, we must begin by training our computer with characters. Several languages are enriched in this kind of study. This thesis will include a study of character recognition techniques for the Bangla language.

1.1 Motivation

We have a lot of tools and gadgets to help us take notes these days. But back then, a pen and a piece of paper were the only ways to write down important information. Some great books and documents can only be found offline. We need to keep some few important documents safe. If we want digital versions of these assets now, we need people to type the documents, which takes a significant amount of time and effort. This would be a tough job that would cost a great deal of money. But if a computer can read writing, we can just scan photos to make it simpler to turn things that need to be kept safe into digital files.

Our nation is overpopulated, and the population-to-physician ratio is much below the norm. As a consequence, our physicians must see a large number of patients. Their handwritten prescription revealed their heavy schedule. Occasionally, it is so challenging to comprehend a doctor's prescription that patients probably end up with the incorrect medication. They might simply misinterpret instructions provided by physicians on prescriptions. This is a frightening stuff. We must automate doctor's prescriptions in digital form in order to minimize errors. This would greatly benefit the healthcare industry.

Even now, we still write checks by hand. As there are no methods that can verify individuals based on their handwriting, fraud does occur. We believe that computer science is capable of handlingÂ this fraud incidents. Research on handwritten character recognition may eventually be beneficial to the creation of tools for validating identity by handwriting.

We sometimes have ideas or thoughts and feel comfortable putting them on paper. Using a system that can read handwritten characters, we may be able to exchange the ideas that are written down. This will make it much easier for us to communicate with others about what we think.

Research and applications of bangla handwritten character recognition could be useful in many more fields than the ones mentioned above. Our work on recognizing handwritten Bangla characters is inspired on and driven by this goal.

1.2 Problem Defination

Bangla is the seventh most used language in the world. About 260 million people around the world speak bangla. Bangla language has a rich and interesting history. The sacrifices and movement to keep bangla language as mother language, that led to the international mother language event, happened in 1952. Even though research on handwritten character recognition has come a long way for many other languages, it hasn't come as far for Bangla. There are studies that show that handwritten character recognition does not work as well as printed character recognition. Research on recognizing handwritten characters in Bangla is progressing slowly.

The character list for the Bangla language is quite diverse. Bangla language contains 49 alphabets. There are 11 vowels, and each one of them may take on one of two distinct forms, depending on whether it is used alone or in combination with a consonant in a word. There are also some letters known as compound characters, that are actually combinations of two or more alphabets. In addition, there are a few characters that seem to be duplicates of one another. Because of these characteristics, it is challenging to figure out the features that are used to recognize bangla characters.

When it comes to handwriting, recognition becomes more challenging. What someone writes depends on the time or place. Different individuals have their own ways of writing. Even writings by the same person can be drastically different. As well as writing by hand makes it harder to identify the difference between characters that appear alike. Due to this, each of these behavioursÂ makes it harder to figure out handwritten characters.

1.3 Objectives

Many machine learning approaches, including ANN, MLP, SVM are employed to address these handwritten character recognition challenges. Convolutional neural networks are now commonly employed in this field of study. Because this approach extracts the characteristics from the provided input on its own. To get the best possible results from this strategy, we must appropriately pre-process the training data. We intend to employ our deep convolution neural network on a significant public dataset (Ekush)[1] that contains 122 classes and of 367,018 isolated characters.Another two public datasets named CMATERdb 3.1.2 [2] and Bangla Lekha Isolated [3] were also used.

1.4 Outline

The following sections are organized as follows: This thesis consists of six chapters, with the following themes given for each:

Chapter 1: The majority of the text focuses on stating the research's rationale, objects, issues, and limitations.

Chapter 2: This chapterÂ provides a comprehensive literature review that addressed the field of handwritten Bangla character recognition, and it summarizes the findings of previous research that was conducted in this area.

Chapter 3:The purpose of this section is to provide an overview of the theoretical underpinning of our investigation by introducing the relevant concepts and terminologies.

Chapter 4:Provides an overview of the two frameworks, the three evaluation datasets, the transfer learning models, and the experimental design and implementation design.

Chapter 5:illustrates research findings and provides an analysis of the datasets. Provides a synopsis of the findings to address the research questions and make recommendations for the way forward.

Chapter 2

Related Work

The Bangla Handwritten Character Recognition (BHCR) system is comprised of image capture, preprocessing, feature extraction, classification, and postprocessing. Image preprocessing includes image separation, noise reduction, and labeling. The image is then delivered as input to various feature extractors. The image is then classified using various classifiers, such as CNN, MLP, SVM, etc.

2.1 ANN , SVM and MLP Based Researches

Nibaran Das et al. (2014) came up with a method for classifying 50 basic characters and 10 numbers using 125 features based on a convex hull. For classification, an MLP with one hidden layer was chosen. This method correctly recognized handwritten basic characters 76.86% of the time and numbers 99.45% of the time[4].

Subhadip Basu et al. (2009) proposed a method where characters were taken from word segmentation. A new feature descriptor with 76 features was also made [5].

Nibaran Das et al. (2009) also suggested using another set of features with 132 features. The feature set had modified shadow features, a quad tree-based longest run feature, distance-based features, and octant and centroid features. In both cases, MLP was used to sort things into groups. The accuracy of 50 basic character classes went up from 75.05% to 85.40% [6].

In 2009, T.K.Bhowmik et al. came up with a method based on SVM. They used RBF, MLP, and SVM to divide characters into 45 basic groups. In this method, the image is put into a group first, and then the actual class is found.This three-stage, two-level, hierarchical architecture worked better than SVM [7].

U. Bhattacharya et al. (2012) suggested a two - stage technique with a 95.8% accuracy on 50 Bangla basic characters[8].

Different projection-based features, such as left projection features, right projection features, quadratic feature algorithms, etc. were tested using Quad tree-based features, longest run features, and Octant centroid features before being classified with ANN, according to a method presented by K. L. Kabir et al. (2015) [9].

2.2 CNN based techniques

A CNN-based technique for BHCR was proposed by Md. Mahbubar Rahman et al. (2015). 50 handwritten Bangla basic characters were classified using two convolutional layers with a 5*5 kernel, two subsampling layers with a 2*2 average area, input and output layers. Each class had 50 neurons in the output layer and 784 neurons in the input layer. Test accuracy was reported to be 85.36% and training accuracy to be 94.55%. [10]

M. A. R. Alif and colleagues (2017) utilized a Resnet architecture for handwritten Bangla recognition characters. Adam optimizer and dropout layers were employed to improve the efficiency of the current Resnet architecture.[11]

A layer-wise training deep neural network method was put out by Saikot Roy et al. (2017) for classifying complex Bangla handwritten characters. They also suggested using the faster-converging RMSProp optimization technique.[12]

To classify 171 composite character classes, A. Ashiquzzaman et al. (2017) suggested a CNN technique that uses dropouts to reduce overfitting andÂ ELU filter to reduce gradient vanishing.CMATERdb 3.1.3.3 served as their database. They recognized 171 different compound characters with 93.68% accuracy.[13]

For the recognition of 122 Bangla handwritten character classes, Akm Shahariar Azad Rabby et al. (2018) developed a CNN architecture with 22 layers. They employed CMATERdb for cross validation and Ekushdb for testing and training. Reports of accuracyÂ 95.01% and 97.73% forÂ CMATERdb and Ekushdb respectively.[1]

A 13-layered CNN with dropout layers was proposed by Akm Shahariar Azad Rabby et al.

(2018). Layers with dropouts were utilized to minimize overfitting. This model also makes use of the Adam optimizer. They used three separate databases to test this model.[14]

In order to categorize 171 compound characters, A. Fardous et al. (2019) developed a CNN architecture made up of 8 convolutional layers, 4 pooling layers, 2 fully connected layers, and Relu activation function. Dropout reduced overfitting and relu introduced non linearity. This technique has a reported accuracy of 95.5%.[15]

2.3 Summary

2.4 Research Gap

2.4.1 Transfer Learning

Transfer learning has various advantages, but the primary ones are reduced training time, improving neural network performance (in most cases), and not requiring a great deal of data.If the original model was trained using an open-source library like as TensorFlow, any user can easily recover it and retrain the necessary layers for his purpose.But trabsfer learning has not been explored in this particular Bangla character recognition problem.

2.4.2 Combination of Datasets

There are several datsets for bangla character recognition. Ekush, Bangla Lekha Isolated,NumtaDB, CMATERdb 3.1.2, CMATER 3.1.3.3, ISI datasets are the prominінet example among them.But virtually no was done to combine them into one and use it for training models.

Authors	Method	Dataset and Classes	Accuracy
Nibaran Das(2009)][4]	MLP	1000 sample of 50 class	Train:85.40%
T.K Bhowmik[7]	MLP and SVM	27000 sample and 45 class	Train:89.22
Nibaran Das [6]	MLP and SVM	19776 and 50 class	MLP:79% SVM:80%
K. L. Kabir[9]	ANN	ISI Dataset , 60 Characters	Train:84.14%
R. Sarkhel[16]	SVM	CMATERdb, 231 characters	Basic:86.53%, compunt:78.38%,Mixed:72.85%
Mahbubar Rahman[10]	CNN	20000, 50 basic	Train: 94.55% Test:85.36%
M. A. R. Alif[11]	CNN(Modified ResNet)	Isolated Bangla(84 classes), CMATERdb(231 Classes)	Proposed ResNet1895.10% ResNet18:4.52 ResNet34:94.59%% Test:85.36%
A. Ashiquzzaman[13]	CNN	CMATERdb, 171 class	Testing 93.68%
saikot Roy [17]	DCNN	CMATERdb , 171 Classes	Train:90.68%
A K M Shahriar[1]	CNN	Ekush(122 Class)CMATERdb(171 Class)	EkushDb:97.73% CMATERdb:95.01%
A K M Shahriar[14]	CNN	CMATERdb and ISI(50 Basic)	ISI:95.7% CMATERdb:98%
Md Zahangir[18]	DCNN	CMATERdb ,171 Classes	Basic:98.31% Numerals:99.13%
Sourajit Saha[19]	DCNN	Bangla Lekha Isolated(84 Classes)	Test:97.21%
A. Fardous[15]	CNN	CMATERdb(171 character)	Test:95.5%
Md. Zahidul Islam[20]	CNN	Ekush ,122 Class	Train:90%
Mr Moynuddin [21]	CNN	Ekush ,122 Class	Train:98.68%
Mr Moynuddin [21]	CNN	Bangla Lekha Isolated(84 Classes)	Train:92.67%

Table 2.1: Summary of Literature Review.
[22]

Chapter 3

Background Study

3.1 Convolutional Neural Network

3.1.1 Basic Structure

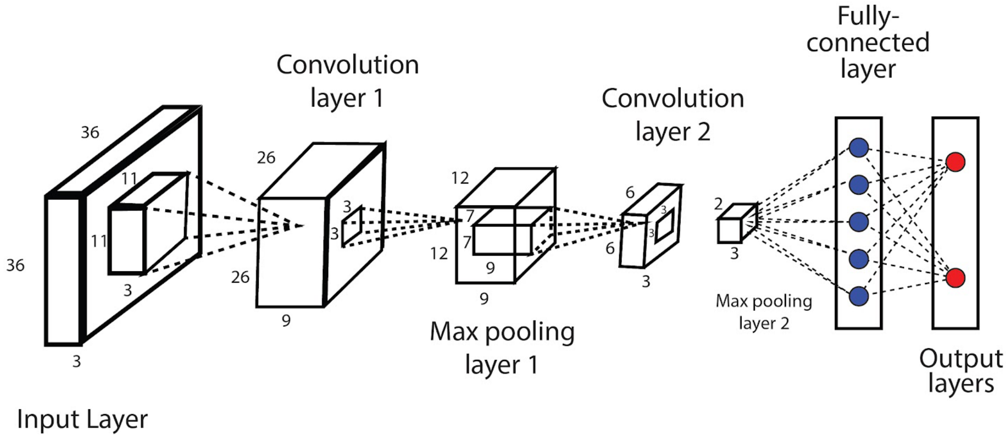

Figure 3.1: Basic Structure of CNN

In general, convolutional neural networks are a type of hierarchical model whose input comprises of raw datasets like RGB ,Grayscale or Binary pictures, raw audio data, and so forth[23]. Comparing Convolutional Neural Networks to other classification techniques, the amount of pre-processing needed is significantly less. Convolutional Neural Networks have the ability to learn many filters and characteristics, whereas in primitive approaches filters are hand-engineered, with sufficient training. The structure of a convolutional neural network was influenced by the way the visual cortex is organized and is similar to the connectivity pattern of neurons in the human brain.The Receptive Field, a constrained area of the visual field, is the unique area in which individual neurons are stimulated. In order to fill the complete visual area, a number of these fields overlap.[24].

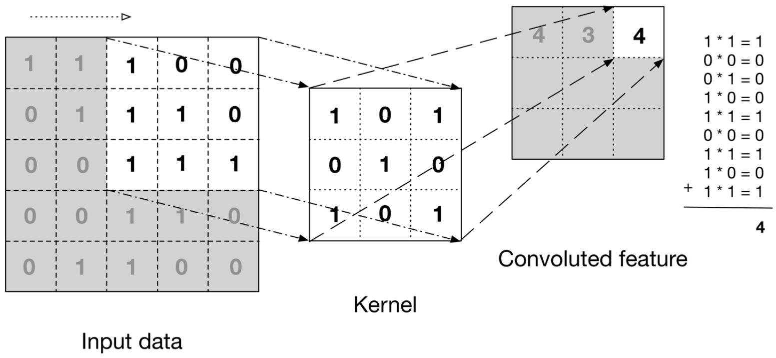

Figure 3.2: Convolutional Layer

3.1.2 Convolutional Layer

The core aspect of a CNN is the convolutional layer, which is also where the majority of processing takes place. It needs input data, a filter, and a feature map, among other things.

Convolution is a mathematical function formed by integrating two input functions. It demonstrates how another function can alter the shape of a function.[25]

$$s(t) = \int x(a)w(t-a)\partial a \tag{3.1}$$

An asterisk is often used to indicate the convolution process:

$$s(t) = (x * w)(t) \tag{3.2}$$

In equation 4.2, the variables x and w stand for the input of a convolutional layer and the kernel, respectively. The created feature map is the result[26].

Assume that the input will be a 3D pixel matrix containing a color image. As a result, the input will have three dimensions corresponding to RGB in a picture: height, width, and depth. In addition, we have a feature detector, also known as a kernel or filter, which traverses the image's receptive fields to determine whether the feature is there.

Typically, the feature detector is a weighted two-dimensional (2-D) array[26] that represents a portion of the picture. The filter size, which also controls the size of the receptive field, is normally a 3x3 or occasionally a 2*2 matrix, however they can vary in size. Following the application of the filter to a portion of the picture, the dot product between the input pixels and the filter is determined. The output array is then supplied with this dot product. Once the kernel has swept through the whole picture, the filter moves by a stride and repeats the operation. A feature map, activation map, or convolved feature is the ultimate result of the series of dot products from the input and the filter.

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(m,n) K(i-m, j-n) \tag{3.3}$$

Convolution is commutative so,

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n) K(m,n) \tag{3.4}$$

It should be noted that weights of the feature detector remains constant as it moves over the images,it is also known as parameter sharing.Through the processes of backpropagation and gradient descent, some parameters, such as the weight values, modify during training. But before the neural network starts being trained, there are three hyperparameters that must be adjusted since they have an impact on the output volume size.These are

1. **The number of filters :** The depth of the output is determined by the number of filters. For instance, three distinct filters would result in three different feature maps.[27]
2. **Stride :** The kernel's stride is how many pixels or how far it travels across the input matrix. Despite the rarity of stride values of two or higher, a longer stride results in a smaller

output.[28]

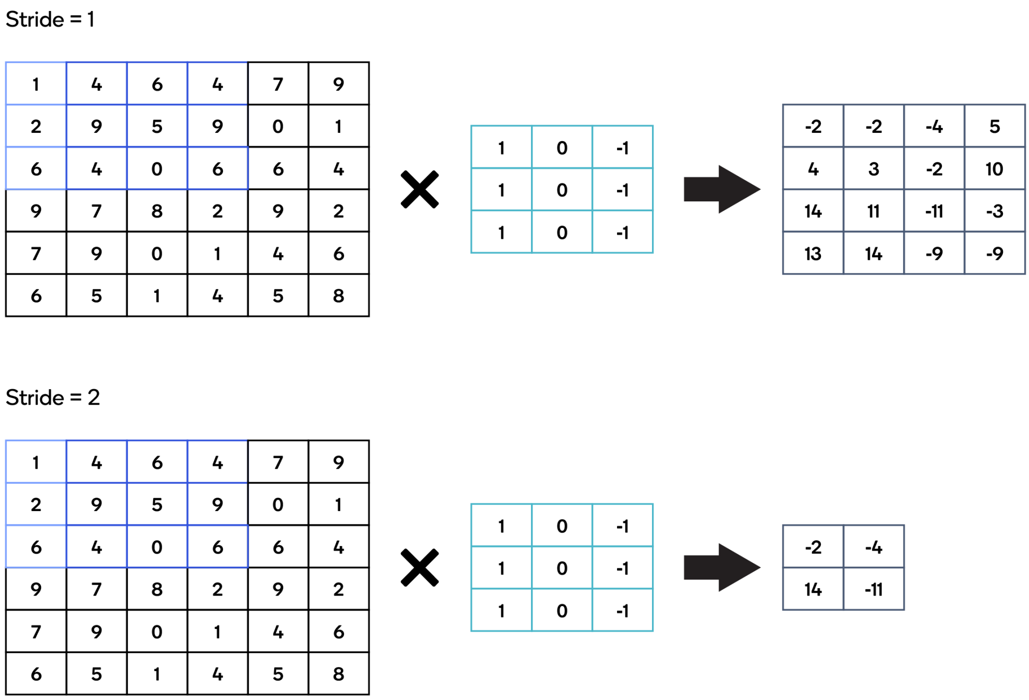

Figure 3.3: Effect of Stride in CNN

3. **padding :**A convolutional neural network uses padding to expand the region of an image it can analyze. Padding is added to the image's frame to provide the kernel additional room to process the image. This helps the kernel process the image more quickly.[3]

   (a) **Same Padding :** To allow the filter we're employing to cover the edge of the matrix and perform the inference with it as well, the padding layers in this type of padding append zero values to the outside frame of the images or data.

   (b) **Valid Padding :** Also referred to as no padding, this If the dimensions are out of alignment in this scenario, the last convolution is dropped.

   (c) **Full Padding :** By extending the input's border with zeros, this type of padding makes the output larger.

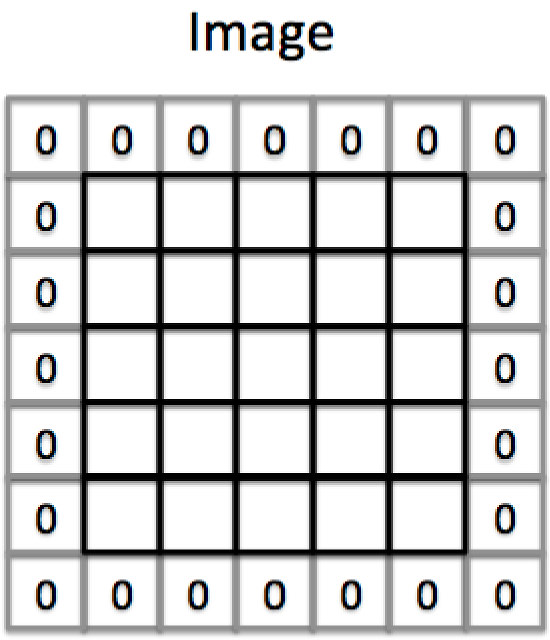

Figure 3.4: Example of Padding

3.1.3 Pooling Layer

Pooling layers, also known as downsampling, reduces the amount of parameters in the input by performing dimensionality reduction.The pooling operation sweeps a filter across the entire input similarly to the convolutional layer, with the exception that this filter lacks weights.Instead, the kernel populates the output array by applying an aggregation function to the values in the receptive field. There are mainly two forms of pooling:

1. **Max pooling :** The maximum, or biggest, value in each patch of each feature map is decided by the maximum pooling process, often known as max pooling.As a side note, this method is applied more frequently than average pooling.

$$f(x) = \max(x_{[i,i+k],[j,j+k]}) \tag{3.5}$$

2. **Average pooling :** Average pooling requires calculating the average for each patch of the feature map. As the filter moves across the input it finds the average value in the receptive

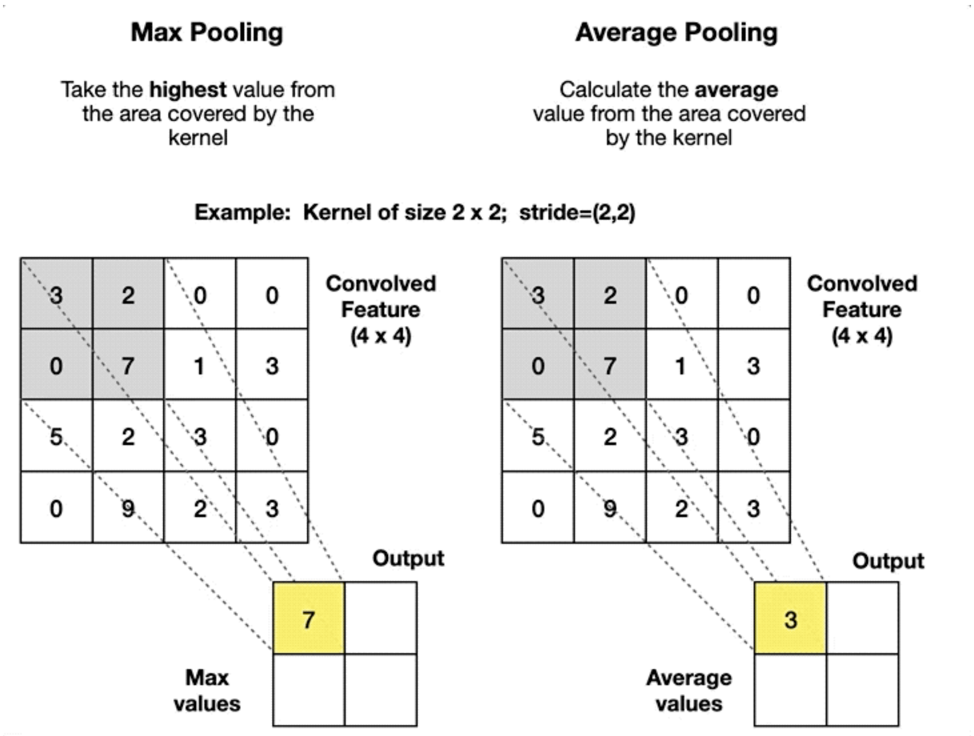

Figure 3.5: Max Pooling and Avarage Pooling

field and sends it to the output array.

$$f(x) = \frac{1}{k \times k} \sum_{i=i_l \ldots i_{l+k}} \sum_{j=j_l \ldots j_{l+k}} x_{i,j} \tag{3.6}$$

3.1.4 Activation Function :

An activation function in a neural network describes how the weighted sum of the input is turned into an output from a node or nodes in a network layer. The purpose of the activation function is to infuse the neural network with nonlinearity.During forward propagation, activation functions require an additional step at each layer, but their computation is well worth it.

1. **Binary Step :** The binary step activation function is basic and always comes to mind when we attempt to bind output. Essentially, it is a threshold-based classifier in which we choose a threshold value to determine whether a neuron's output should be triggered or silenced.

$$BinaryStep(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.7}$$

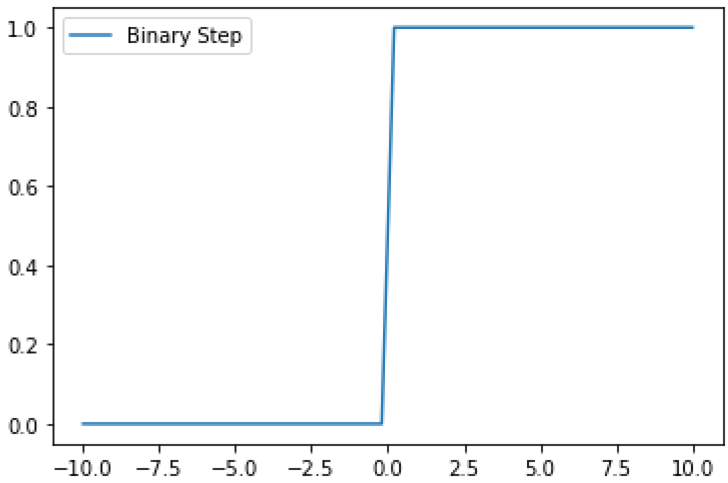

Figure 3.6: Binary Step Activation Function

2. **Rectified Linear Activation (ReLU) :** The rectified linear activation function, or ReLU, is a piecewise linear function that outputs the input directly if it is positive and 0 otherwise. In many applications of neural networks, it has replaced the more complex activation functions as the default since models trained with it are more straightforward and generally yield better results.[29]

$$ReLU(x) = max(0, x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.8}$$

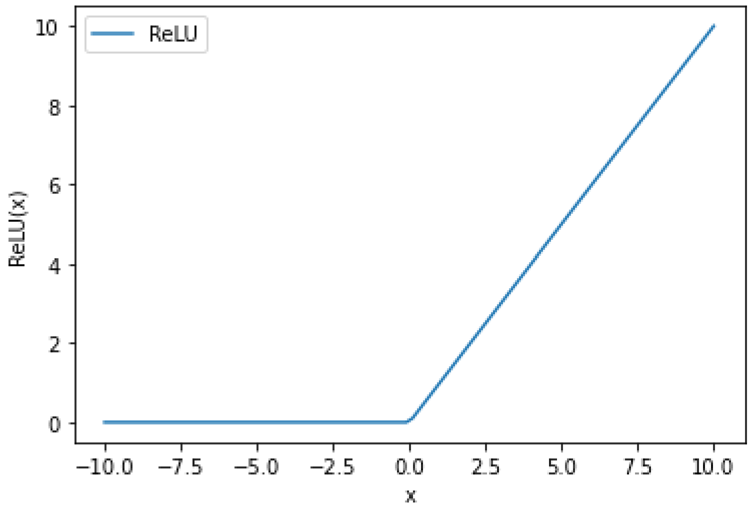

Figure 3.7: ReLU Activation Function

3. **Sigmoid Function :** Sigmoid accepts a real value as input and returns a value between 0 and 1 as output. It is simple to handle and possesses all the desirable qualities of activation functions: non-linearity, continuous differentiation, fixity, and a set output range. [30]

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.9}$$

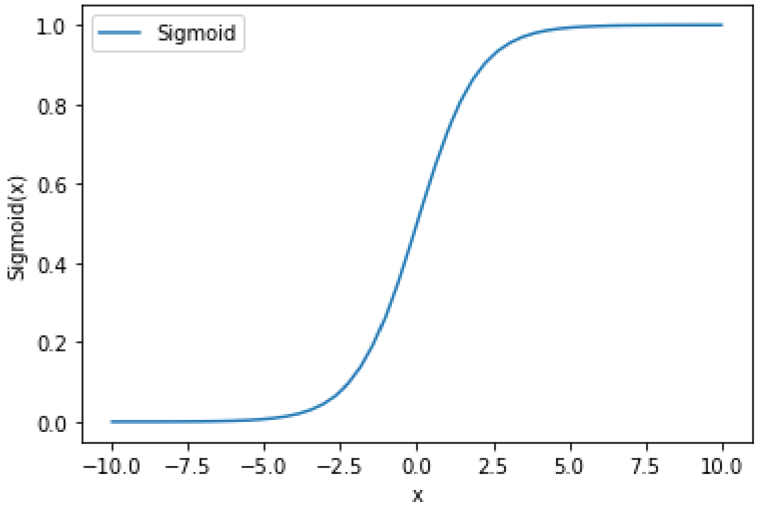

Figure 3.8: Sigmoid Activation Function

4. **Tanh Function :** Tanh activation function is marginally superior to the sigmoid function; like the sigmoid function, it is used to estimate or discriminate between two classes, but it maps only negative inputs to negative quantities and ranges from -1 to 1.

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.10}$$

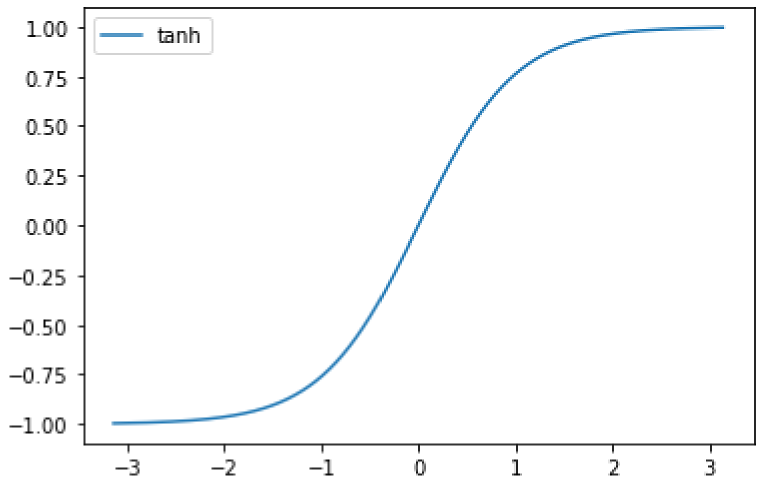

Figure 3.9: Tanh Activation Function

5. **Softmax activation function :** Softmax is mostly utilized in the output layer for decision

making, similar to how sigmoid activation functions.Softmax assigns a value proportional to the input variable's weight, and the sum of these weights is one.

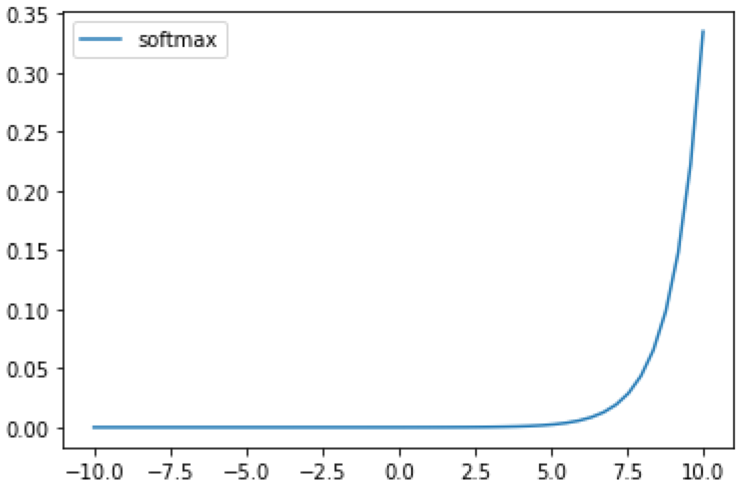

Figure 3.10: Softmax Activation Function

3.1.5 Objective Function :

The objective function's aim is to calculate the difference between the predicted and actual values.The catagorical-cross-entropy cost function and the L2 loss function are the most often used objective functions in classification and regression problems in the current convolutional neural network. **Binary Cross Entropy:** Cross-entropy loss, often known as log loss, is a performance metric for classifiers whose output is a probability value between 0 and 1. As the projected probability varies from the actual label, cross-entropy loss grows.For example, predicting a chance of 0.012 when the actual observation value is 1 would be disadvantageous and result in a significant loss value. The log loss of a perfect model would be 0.

$$f(x) = \frac{1}{N} \sum_{i}^{N} \sum_{j}^{M} y_{ij} \log(p_{ij}) \tag{3.11}$$

3.1.6 Optimizer Function :

We must utilize the loss obtained by the loss function to train our network to perform better. We must attempt to minimize the loss, because a lower loss indicates that our model will perform better. Optimization is the process of minimizing (or maximizing) any mathematical statement. **Adaptive Moment Estimation(Adam) :** Adam optimization is a stochastic gradient descent technique that utilizes adaptive estimate of first-order and second-order moments.Adam is comparable to RMSprop and stochastic gradient descent with momentum. It uses squared gradients to adjust the learning rate, similar to RMSprop, and it takes advantage of momentum by utilizing a moving gradient average instead of the gradient itself, similar to SGD with momentum.[31]

3.2 Tools :

3.2.1 TensorFlow :

TensorFlow is a popular, free and open-source software framework. It is widely used for machine learning and artificial intelligence. Within Googleâs Machine Intelligence research organization, Google Brain Team developed this tool and it was first released in 2015. There is an updated version of tensorflow released in 2019. With TensorFlow, programmers can build dataflow graphsâstructures that depict the flow of data via a graph or a collection of processing nodes. Tensolflow encompasses the whole procedure from beginning to end. It provides an integrated suite of resources for developers, organizations, and academics who are interested in advancing the state of the art in machine learning and create highly scalable applications that are driven by ML. Programming languages as diverse as Python, JavaScript, C++, and Java can all use TensorFlow.

3.2.2 Keras :

The Keras API was created with people in mind, not with computers. Keras conforms to established practices for minimizing cognitive load. It provides uniform and straightforward APIs, reduces the amount of user interactions needed for typical use cases, and gives clear and responsive error signals. Additionally, it contains a wealth of development instructions and documentation.

3.2.3 Jupyter Notebook :

The Jupyter Notebook is a free, open-source online software for creating and sharing documents with embedded code, mathematical expressions, graphical representations of data, and markdown text. Some of Jupyter's most notable characteristics are listed here. The benefits of using a coding environment that highlights syntax errors, tabs in automatically, and completes tabs. Simply execute the code in the browser and show the output in the box below the code. Whether you're making a remark or a statement in the code, you may do it using the Markdown syntax and modify it right in your browser. the capability to quickly incorporate LaTeX-based mathematical notation into markdown blocks. Rendering the computation results in a visual medium such as HTML, latex, PNG, SVG, etc.

3.2.4 Libraries :

We have used Python for this research . There are a bunch of libraries in python. Here is a overview of some of them that we have used in our proposed model.

1. **Numpy** Numerical Python is referred to as NumPy. It is an array handling Python library. Additionally, it includes matrices, fourier transform, and functions for operating in the area of linear algebra. In 2005, Travis Oliphant built it. It is an open source project. Anyone can use it freely.

2. **Pandas** For handling and analyzing data, the Python programming language includes the pandas software package. It consists of certain data structures and processes for dealing with time-series data and computational tables. It is free software distributed under the BSD license's three clauses.

3. **Matplotlib** There is a graphing library called Matplotlib, for the Python programming language and its NumPy module for numerical mathematics. With the help of all-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK, it offers an object-oriented API for embedding plots into programs.

4. **Scikit-learn** Python has a machine learning package called Scikit-learn. It includes a number of methods for clustering, classification, and regression, including SVMs, gradient boosting, k-means, random forests, and DBSCAN. It is made to operate with SciPy, Numpy, and Python. David Cournapeau started the scikit-learn project as a Google Summer of Code (commonly known as GSoC) project under the name scikits.learn. Its name comes from "Scikit," a different third-party addition to SciPy.

5. **Open CV** OpenCV is a significant open-source library for image processing, machine learning, and computer vision. It currently plays a significant part in real-time operation, which is crucial in modern systems. Using it, one may analyze pictures and videos to find a person, objects, and even human handwriting. Python is able to deal with the OpenCV array structure for study when it is combined with other libraries, such as NumPy. We employ vector space and apply mathematical operations to these characteristics to identify visual patterns and their different features.

Chapter 4

Method

4.1 Datasets

1. Ekush:Ekush dataset of isolated Bangla handwritten charactersÂ data was gathered from 3086 people representing university, school, and college students, with approximately 50% 1510 male and 50% 1576 female participants.After writing the handwriting characters on a form, it was scanned to obtain image data from raw data.

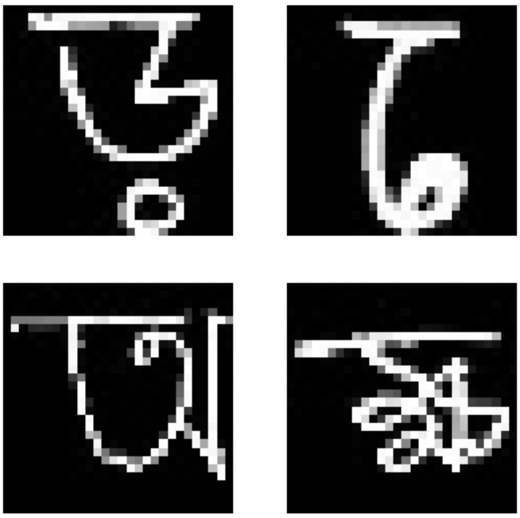

Figure 4.1: Ekush Dataset Example

2. Banla Lekha Isolated:Handwriting examples of all Bangla basic characters and numbers (50 basic characters and 10 numerals) are included in the data. It also includes 24 carefully

chosen compound characters. As a result, the dataset contains 84 distinct Bangla characters. Following the collection of raw data on forms, the samples are digitized and pre-processed.

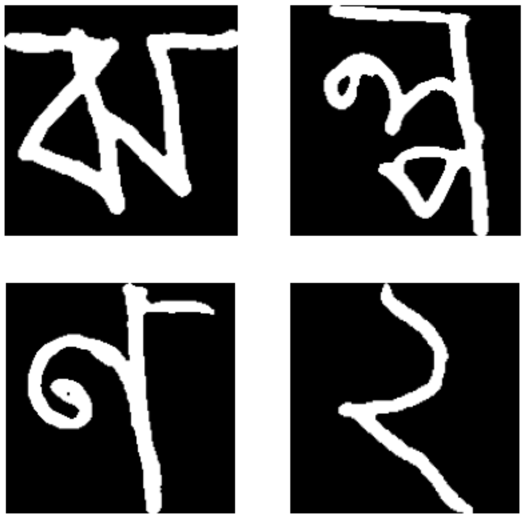

Figure 4.2: Bangla Lekha Isolated Dataset Example

3. CMATERdb 3.1.2: It is a dataset containing 15000 samples of 50 basic bangla character.It is created by CMATER research labratory of Jadavpur University.All the samples here are handwritten images.

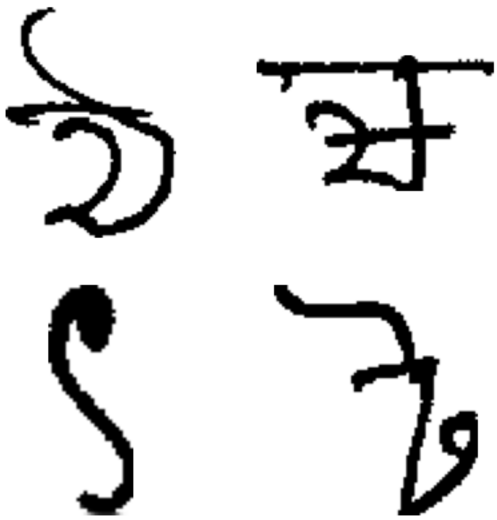

Figure 4.3: CMARTdb 3.1.2 Dataset Example

4. Combined Dataset:All the three dataset is combined together and produced a new dataset.

We hava ran experiment using all the data here mentioned

4.2 Models

4.2.1 Proposed Model

In this paper we propose 2 models of convolutional neural network.One is 11 Layer deep and another is 26 layers deep. Total number of Parameters varies very much.

Layer(Type)	Output Shape	Total Number of parameters
conv2d (Conv2D)	(None, 32, 32, 32)	320
max pooling2d (MaxPooling2D)	(None, 16, 16, 32)	0
conv2d 1 (Conv2D)	(None, 14, 14, 64)	18496
max pooling2d 1 (MaxPooling2D)	(None, 7, 7, 64)	0
conv2d 2 (Conv2D)	(None, 5, 5, 128)	73856
max pooling2d 2 (MaxPooling 2D)	(None, 2, 2, 128)	0
dropout (Dropout)	(None, 2, 2, 128)	0
flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 128)	65664
dropout 1 (Dropout)	(None, 128)	0
dense 1 (Dense)	(None, 122)	15609

Table 4.1: Architecture of Proposed Model1.

4.2.2 Classic vision models and Transfer Learning

Transfer learning involves utilizing feature representations from a previously-trained model so that a new model does not need to be trained from start.

Typically, the pre-trained models are trained on enormous datasets that serve as a benchmark at the cutting edge of computer vision. The weights derived from the models can be reused for additional computer vision applications. we have tested the datasets using classical computer vision models through transfer learning namely Resnet-50, VGG16, MobilNet.

- Resnet-50
- VGG16
- MobileNet

Layer(Type)	Output Shape	Total Number of parameters
input 1 (InputLayer)	[(None, 32, 32, 3)]	0
conv2d (Conv2D)	(None, 32, 32, 32)	896
batch normalization (BatchNormalization)	(None, 32, 32, 32)	128
conv2d 1 (Conv2D)	(None, 32, 32, 32)	9248
batch normalization 1 (BatchNormalization)	(None, 32, 32, 32)	128
max pooling2d (MaxPooling2D)	(None, 16, 16, 32)	0
conv2d 2 (Conv2D)	(None, 16, 16, 64)	18496
batch normalization 2 (BatchNormalization)	(None, 16, 16, 64)	256
conv2d 3 (Conv2D)	(None, 16, 16, 64)	36928
batch normalization 3 (BatchNormalization)	(None, 16, 16, 64)	256
max pooling2d 1 (MaxPooling2D)	(None, 8, 8, 64)	0
conv2d 4 (Conv2D)	(None, 8, 8, 128)	73856
batch normalization 4 (BatchNormalization)	(None, 8, 8, 128)	512
conv2d 5 (Conv2D)	(None, 8, 8, 128)	147584
batch normalization 5 (BatchNormalization)	(None, 8, 8, 128)	512
max pooling2d 2 (MaxPooling2D)	(None, 4, 4, 128)	0
conv2d 6 (Conv2D)	(None, 4, 4, 256)	295168
batch normalization 6 (BatchNormalization)	(None, 4, 4, 256)	1024
conv2d 7 (Conv2D)	(None, 4, 4, 256)	590080
batch normalization 7 (BatchNormalization)	(None, 4, 4, 256)	1024
max pooling2d 3 (MaxPooling2D)	(None, 2, 2, 256)	0
flatten (Flatten)	(None, 1024)	0
dropout (Dropout)	(None, 1024)	0
dense (Dense)	(None, 2048)	2099200
dropout 1 (Dropout)	(None, 2048)	0
dense 1 (Dense)	(None, 122)	249978

Table 4.2: Architecture of Proposed Model 2.

Parameter Name	Parameters for Model 1	Parameters for Model 2
Non-trainable params	0	1,920
Trainable params	173,945	3,523,354
Total params	173,945	3,525,274

Table 4.3: Parameter Information of Proposed Model

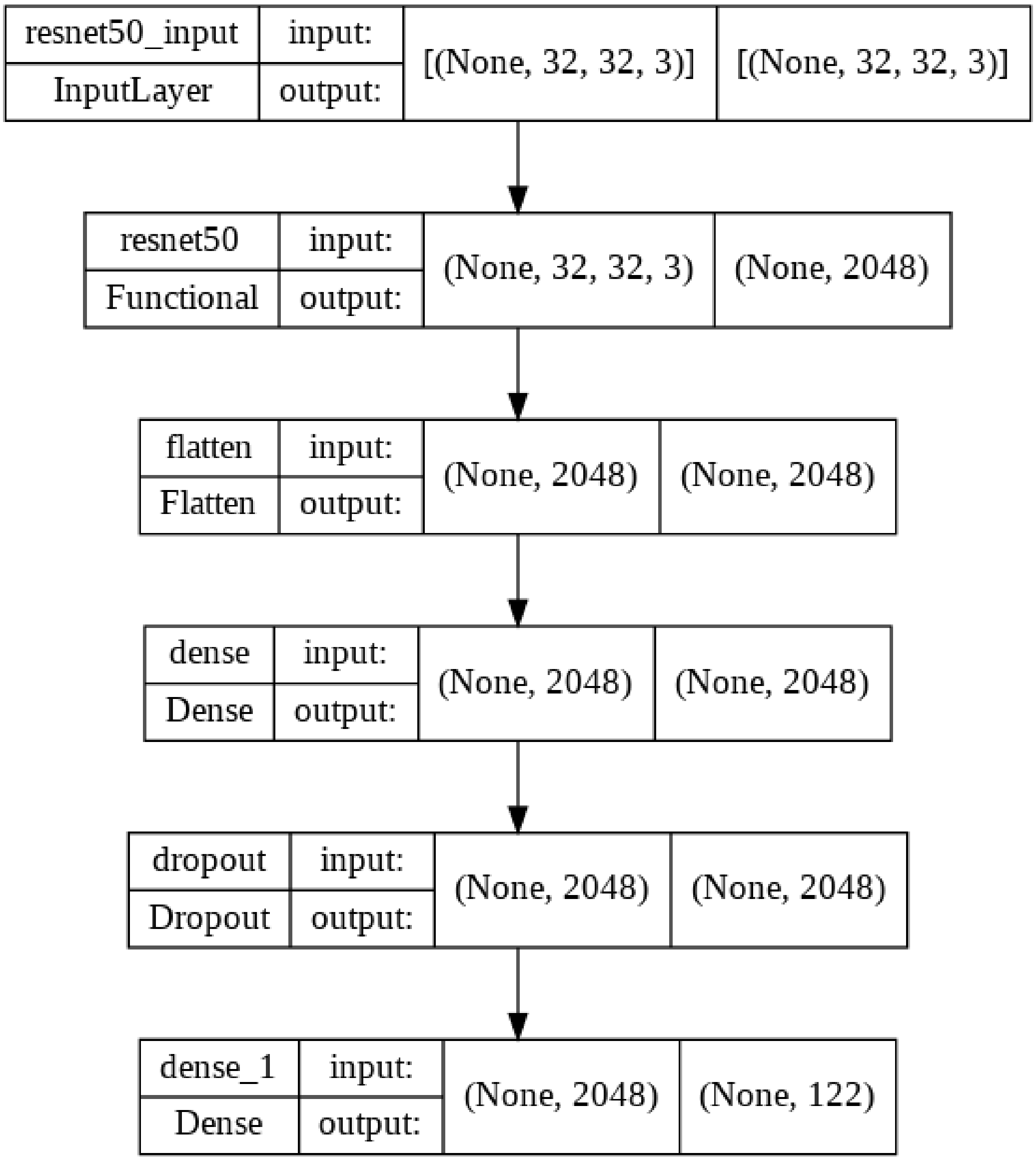

Figure 4.4: Resnet50 Architechture Using Transfer Learning

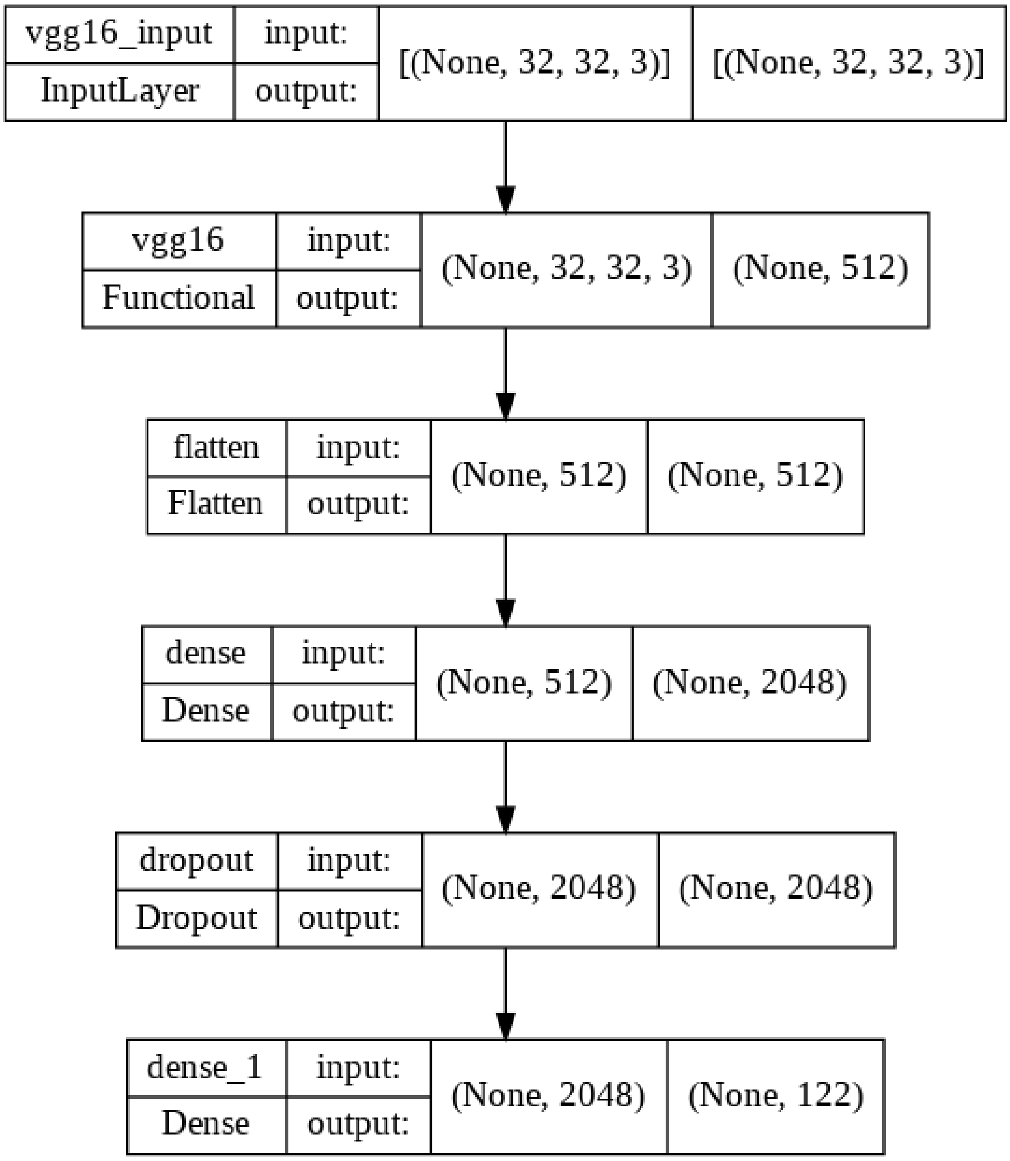

Figure 4.5: VGG-16 Architechture Using Transfer Learning

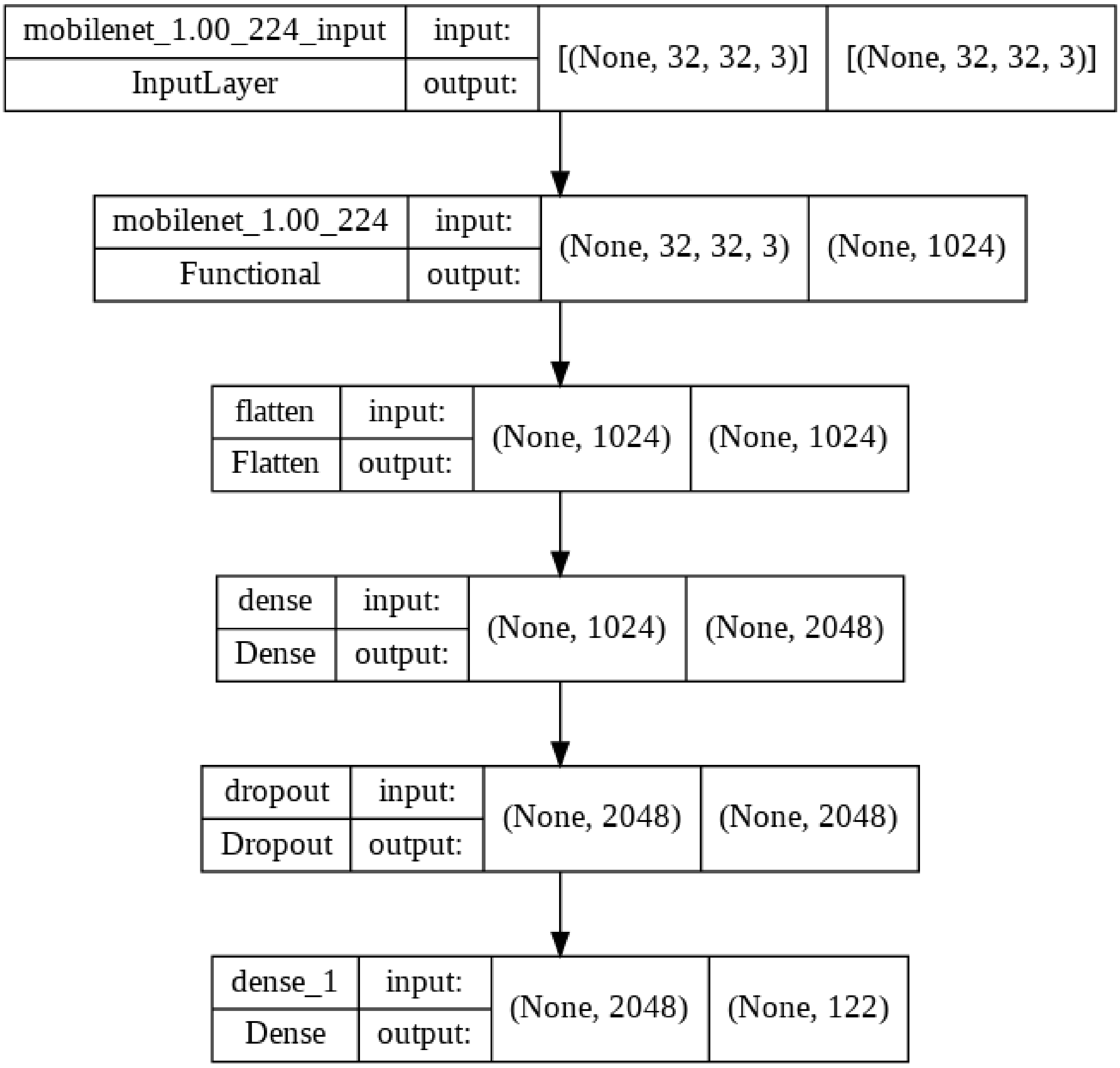

Figure 4.6: MobileNet Architechture Using Transfer Learning

4.3 Experiment Design

4.3.1 Experiment A

In this section we have decided that we are going to experiment only with transfer learning models.

- Resnet-50 + Ekush Dataset, Bangla Lekha Isolated , CMATERdb, Bangla Lekha Isolated
- VGG16 + Ekush Dataset, Bangla Lekha Isolated , CMATERdb, Bangla Lekha Isolated
- MobileNet + Ekush Dataset, Bangla Lekha Isolated , CMATERdb, Bangla Lekha Isolated

4.3.2 Experiment B

In this section we have decided that we are going to experiment only with our proposed models.

- Proposed Model 1 + Ekush Dataset, Bangla Lekha Isolated , CMATERdb, Bangla Lekha Isolated
- Proposed Model 2 + Ekush Dataset, Bangla Lekha Isolated , CMATERdb, Bangla Lekha Isolated

Model	Weights	Total params	Trainable params	Non-trainable params
resnet50	imagenet	28,034,042	4,446,330	23,587,712
vgg16	imagenet	16,015,290	1,300,602	14,714,688
mobilenet1.00	imagenet	5,578,042	2,349,178	3,228,864

Table 4.4: Comparison of transfer learning models

4.4 Implementation

All the experiment were done on a Remote desktop. The hardware and software configuration is listed below

Tool	Version
Python	3.9.7
Tensorflow-gpu	2.8.2
Keras	2.8.0
CUDA	11.32
CuDnn	10.31

Table 4.5: Environment Configuration

4.5 Model Training

In this task of classifying handwritten characters, 122 classes are taken, which correspond to 50 basic characters, 10 numerals, 10 modifiers, and 52 compound characters. At the time of integrating datasets, all labels were matched, and two data classes from the Bangla Lekha Isolated dataset that were in conflict were set aside.

If not already in binary format, each image was converted to that format. Each image in our training datasets was scaled to 32 pixels by 32 pixels.

Batch Size	32
Number of epochs	150(Tranfer Learning) and 50(Our propose models)
Cost function	binary cross entropy
Optimizer	adam

Table 4.6: Training information

â¢ Cost Function : SoftMax Loss function.

4.6 Evaluation Matrics

Chapter 5

Result Analysis

Model Name	Ekush	Isolated	CMATERdb
ResNet50	93.51%	91.34%	88.32%
VGG-16	96.35%	95.34%	87.34%
MobileNet	71.32%	77.82%	81.75%

Table 5.1: Result of Experiment A

Model Name	Ekush	Isolated	CMATERdb	Combined
Model 1	88.36%	88.8%	82.63%	89.32%
Model 2	98.77%	98.82%	95.78%	98.78%

Table 5.2: Result of Experiment B

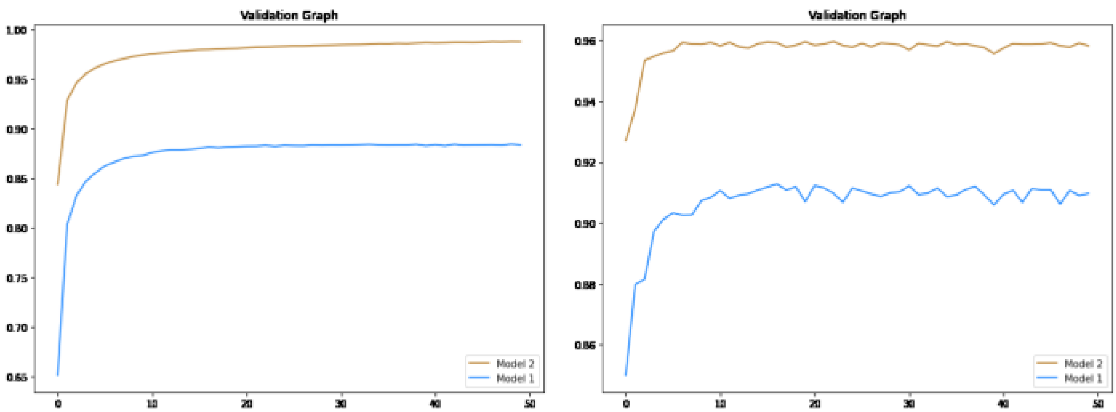

Figure 5.1: Validation Accuracy and Training Accuracy on Ekush Dataset of Model 1 and Model 2

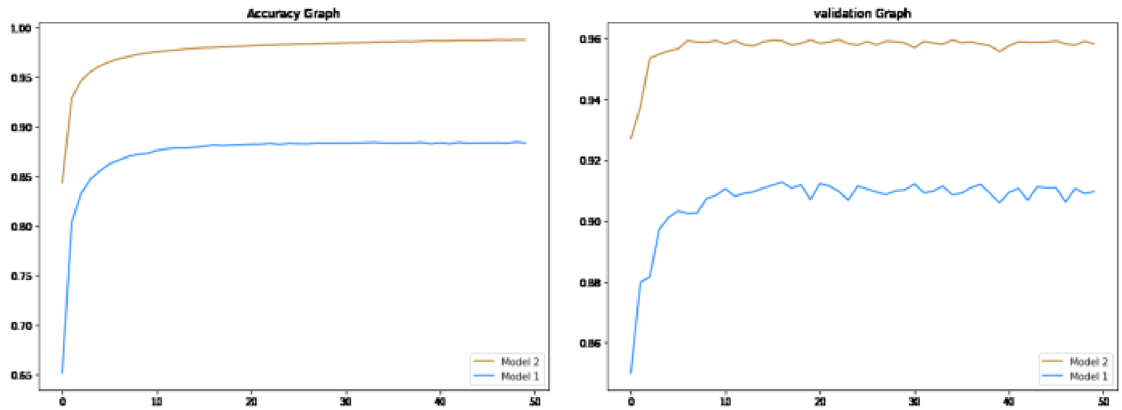

Figure 5.2: Validation accuracy and Training Accuracy on Bangla Lekha Isolated Dataset of Model 1 and Model 2

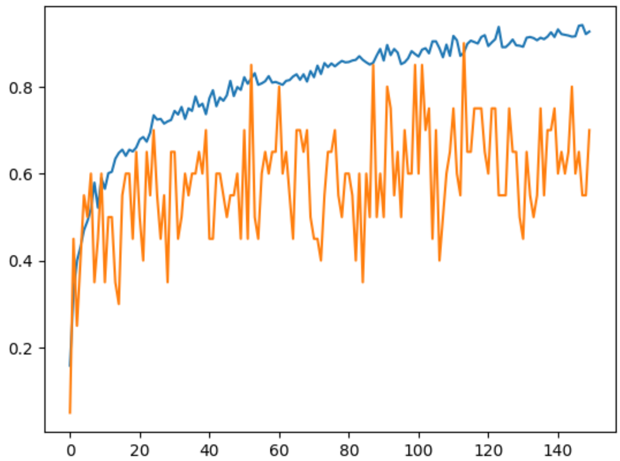

Figure 5.3: Validation and Accuracy on Bangla Lekha Isolated Dataset of vgg-16

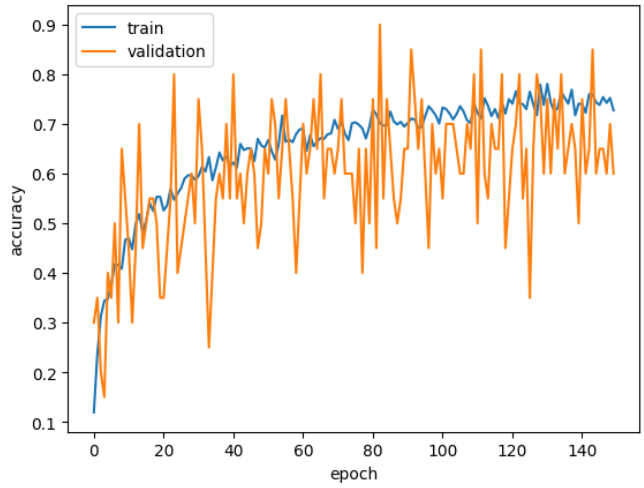

Figure 5.4: Validation and Training Accuracy on Bangla Lekha Isolated Dataset of Mobilenet

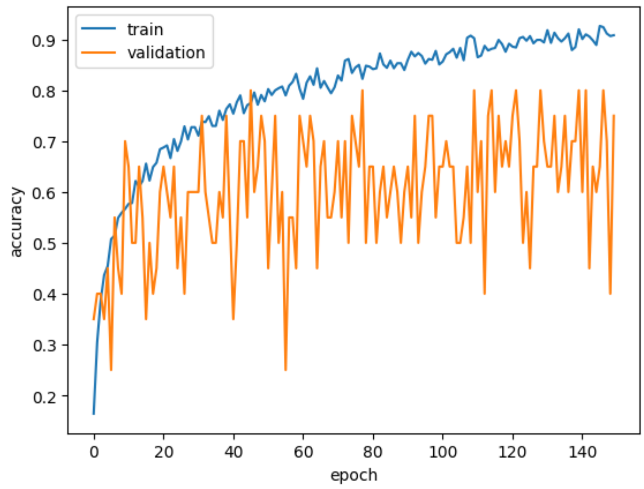

Figure 5.5: Validation and Training Accuracy on Bangla Lekha Isolated Dataset of Resnet50

5.0.1 Analysis

Our Proposed Model 2 gives better output in terms of same epoch.If we run 50 epoch for transfer learning model and our model, our model 2 gives better accuracy.But transfer learning model gradually picks up after running more than 100 epochs.And transfer leaning models perform better when added more trainable parameter to their side.

5.1 Future work

The purpose of this study was to identify handwritten bangla characters. We utilized three datasets containing bangla alphabetic characters, bangla compound characters, and bangla numeric characters. In the near future, we want to incorporate bangla handwritten word recognition. The databases do not include all compound bangla characters. More data about compound characters might improve the identification of all bangla characters. More small-scale datasets for bangla handwritten characters are available. That need cleaning and preparation for use in research. In the study area of handwritten character identification in the Bangla language, a merger of all accessible datasets might be a significant step.

References

[1] A. Rabby, S. Haque, M. Islam, S. Abujar, S. A. Hossain *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*. Springer, 2018, pp. 149–158.

[2] R. Sarkar, N. Das, and S. Basu, “Google code archive - long-term storage for google code project hosting.” [Online]. Available: https://code.google.com/archive/p/cmaterdb/

[3] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *2017 international conference on engineering and technology (ICET)*. Ieee, 2017, pp. 1–6.

[4] N. Das, S. Pramanik, S. Basu, P. K. Saha, R. Sarkar, M. Kundu, and M. Nasipuri, “Recognition of handwritten bangla basic characters and digits using convex hull based feature set,” *arXiv preprint arXiv:1410.0478*, 2014.

[5] S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri, and D. K. Basu, “A hierarchical approach to recognition of handwritten bangla characters,” *Pattern Recognition*, vol. 42, no. 7, pp. 1467–1484, 2009.

[6] N. Das, S. Basu, R. Sarkar, M. Kundu, M. Nasipuri, and D. kumar Basu, “An improved feature descriptor for recognition of handwritten bangla alphabet,” in *Proceedings of 2nd International conference on signal and Image Processing ICSIP*, 2009, pp. 451–454.

[7] T. K. Bhowmik, P. Ghanty, A. Roy, and S. K. Parui, “Svm-based hierarchical architectures for handwritten bangla character recognition,” *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 12, no. 2, pp. 97–108, 2009.

[8] U. Bhattacharya, M. Shridhar, S. K. Parui, P. Sen, and B. Chaudhuri, "Offline recognition of handwritten bangla characters: an efficient two-stage approach," *Pattern Analysis and Applications*, vol. 15, no. 4, pp. 445–458, 2012.

[9] K. L. Kabir, M. K. Shafin, T. T. Anannya, D. Debnath, M. R. Kabir, M. A. Islam, and H. Sarwar, "Projection-based features: A superior domain for handwritten bangla basic characters recognition," in *2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO)*. IEEE, 2015, pp. 1–7.

[10] M. M. Rahman, M. Akhand, S. Islam, P. C. Shill, M. Rahman *et al.*, "Bangla handwritten character recognition using convolutional neural network," *International Journal of Image, Graphics and Signal Processing (IJIGSP)*, vol. 7, no. 8, pp. 42–49, 2015.

[11] M. A. R. Alif, 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)*. IEEE, 2017, pp. 1–6.

[12] M. Biswas, R. Islam, G. K. Shom, M. Shopon, N. Mohammed, S. Momen, and A. Abedin, "Banglalekha-isolated: A multi-purpose comprehensive dataset of handwritten bangla isolated characters," *Data in brief*, vol. 12, pp. 103–107, 2017.

[13] A. Ashiquzzaman, A. K. Tushar, S. Dutta, and F. Mohsin, "An efficient method for improving classification accuracy of handwritten bangla compound characters using dcnn with dropout and elu," in *2017 Third International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*. IEEE, 2017, pp. 147–152.

[14] A. S. A. Rabby, S. Haque, S. Abujar, and S. A. Hossain, "Ekushnet: Using convolutional neural network for bangla handwritten recognition," *Procedia computer science*, vol. 143, pp. 603–610, 2018.

[15] A. Fardous and S. Afroge, "Handwritten isolated bangla compound character recognition," in *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*. IEEE, 2019, pp. 1–5.

[16] R. Sarkhel, A. K. Saha, and N. Das, "An enhanced harmony search method for bangla handwritten character recognition using region sampling," in *2015 IEEE 2nd international conference on recent trends in information systems (ReTIS).* IEEE, 2015, pp. 325–330.

[17] S. Roy, "Nibaran das, mahantapas kundu, and mita nasipuri. handwritten isolated bangla compound character recognition: A new benchmark using a novel deep learning approach," *Pattern Recognition Letters*, vol. 90, no. 15, pp. 15–21, 2017.

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

[19] S. Saha and N. Saha, "A lightning fast approach to classify bangla handwritten characters and numerals using newly structured deep neural network," *Procedia computer science*, vol. 132, pp. 1760–1770, 2018.

[20] M. Z. Islam, M. A. Based, and M. M. Rahman, "Offline bangla handwritten character recognition with convolutional neural network (cnn)," *International Journal of Scientific Research and Engineering Development*, 2021.

[21] M. M. A. Shibly, T. A. Tisha, T. A. Tani, and S. Ripon, "Convolutional neural network-based ensemble methods to recognize bangla handwritten character," *PeerJ Computer Science*, vol. 7, p. e565, 2021.

[22] T. Ghosh, S. M. Chowdhury, M. A. Yousuf *et al.*, "A comprehensive review on recognition techniques for bangla handwritten characters," in *2019 International Conference on Bangla Speech and Language Processing (ICBSLP).* IEEE, 2019, pp. 1–6.

[23] X. Changzhen, W. Cong, M. Weixin, and S. Yanmei, "A traffic sign detection algorithm based on deep convolutional neural network," in *2016 IEEE International Conference on Signal and Image Processing (ICSIP).* IEEE, 2016, pp. 676–679.

[24] Y. LeCun, Y. Bengio *et al.*, "The handbook of brain theory and neural networks," 1998.

[25] D. M. , "Convolutional neural networks (cnn) step 1 - convolu- tion operation." [Online; accessed 2022-09-17].

[26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[27] W. S. Ahmed *et al.*, "The impact of filter size and number of filters on classification accuracy in cnn," in *2020 International conference on computer science and software engineering (CSASE)*. IEEE, 2020, pp. 88–93.

[28] J. Yepez and S.-B. Ko, "Stride 2 1-d, 2-d, and 3-d winograd for convolutional neural networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 4, pp. 853–863, 2020.

[29] K. Eckle and J. Schmidt-Hieber, "A comparison of deep networks with relu activation function and linear spline-type methods," *Neural Networks*, vol. 110, pp. 232–242, 2019.

[30] X. Yin, J. Goudriaan, E. A. Lantinga, J. Vos, and H. J. Spiertz, "A flexible sigmoid function of determinate growth," *Annals of botany*, vol. 91, no. 3, pp. 361–371, 2003.

[31] Z. Zhang, "Improved adam optimizer for deep neural networks," in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. Ieee, 2018, pp. 1–2.

Appendices

Appendix A

List of Abbreviations

Layer(Type)	Output Shape
BBCD	Bangla Basic Character Database
BHCR	Bangla Hand-written Character Recognition
CMATER	Center for Microprocessor Applications for Training Education and Research
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
DNN	Deep Neural Network
GPU	Graphic Processing Unit
HCR	Hand-written Character Recognition
HMM	Hidden Markov Model
MLP	Multilayer Perceptron
NN	Neural Network
OCR	Optical Character Recognition
ReLU	Rectified Linear Unit
ResNet	Residual Network
SVM	Support Vector Machine

Table A.1: List of Abbreviations