

Premier University, Chittagong
Department of Computer Science & Engineering



Final year thesis report on
Handwritten Bangla District Name Recognition using Deep Learning Techniques

*A thesis submitted in partial fulfillment of the requirements
for the degree of Bachelor of Science in the
Department of Computer Science and Engineering
under the guidance of*

Mohammad Hasan

Lecturer

Department of Computer Science and Engineering
Premier University, Chittagong.

Submitted By:

Sultan Md. Ayman (1603110201307)

Md. Ibrahim Siddiquee (1603110201308)

Md. Hosen Zisad (1603110201311)

Declaration

We, *Sultan Md. Ayman, Id:1603110201307; Md. Ibrahim Siddiquee, Id: 1603110201308; Md. Hosen Zisad, Id: 1603110201311* of 31th batch declare that the thesis work entitled **“Handwritten Bangla District Name Recognition using Deep Learning Techniques”** submitted to the Premier University, is a record of an original work done by us under the guidance of **Mohammad Hasan**, Lecturer, Department of Computer Science & Engineering and this work is submitted for fulfillment of the degree of Bachelor of Science in Computer Science & Engineering. No portion of the work contained in this thesis has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Signed By Sultan Md. Ayman: _____

Signed By Md. Ibrahim Siddiquee: _____

Signed By Md. Hosen Zisad: _____

Approval

This thesis entitled “**Handwritten Bangla District Name Recognition using Deep Learning Techniques**” prepared and submitted by *Sultan Md. Ayman, Id:1603110201307; Md. Ibrahim Siddiquee, Id: 1603110201308; Md. Hosen Zisad, Id: 1603110201311* of 31th Batch in partial fulfillment of the requirements for the degree of Bachelor of Science in the Department of Computer Science and Engineering has been examined and is recommended for approval and acceptance.

Prof. Dr. Taufique Sayeed

Dean, Faculty of Science and Engineering

&

Chairman, Department of Computer Science and Engineering, Premier University

Mohammad Hasan

Lecturer

Department of Computer Science and Engineering, Premier University
(Thesis Supervisor)

Acknowledgment

At first, we want to express gratitude to the Almighty for his endless kindness for keeping us mentally and physically fit to complete this sophisticated task.

This study would not get its own shape without the general support of the Premier University, Chittagong which provided us the chance for our Bachelor's Program. We wish to acknowledge the help provided by the technical and support staff in the CSE department of our university. The completion of this thesis is not a result of our individual effort, but is an aggregate of co-operation of many other people.

Foremost, we would like to express our deepest thanks to our supervisor, **Mohammad Hasan**. His vast knowledge in Deep Learning, Machine learning, and curiosity to disseminate that knowledge, understanding capability, patience, humbleness, tolerance, dealing, attitude, behavior and wisdom are really appreciable. His directions and guidelines of preparing manuscripts, reports, presentations make us more dynamic and well-organized. We do not hesitate to certify him as the best academic scholar so far we have experienced. We would also like to thank our friends and family who supported us and offered deep insight into the study.

We would like to thank those who have helped us a lot for collecting the handwritten text for our dataset. We also wish to thank all of my seniors for continuous supports. Finally, we would like to show our gratitude to all of my teachers who helped us a lot during our Bachelor's program, without their teaching it was impossible for us to learn even a bit of what we've learned all these years.

Abstract

Automatic handwritten word recognition seems to have a variety of academic and commercial interests. Analysis of Bengali handwritten word recognition with good sensitivity has been one of the most difficult tasks. automation in Bangla handwritten word recognition is becoming important day by day because it is used in various organization including digitize official documents and in development of intelligent traffic systems. Recognition of handwritten words are significantly difficult compared to printed characters as handwritings vary in size and shape from person to person. As Bangla script is curvy and contains compound characters, recognizing Bangla language is more complex task compared to English. In this study we introduce a novel dataset of 7040 samples which contains handwritten district names of Bangladesh. Here, we offer an overview of the fast and reliable Bangla OCR to extract words from images. We propose a Convolutional Neural Network based classification system. The model was able to achieve 97 percent accuracy in both training and validation set with good sensitivity. We have used a benchmark dataset to analyze our model where the proposed model achieved 80 percent accuracy. We have compared our proposed model with three other CNN architecture and our proposed CNN model outperformed other models with high sensitivity.

Table of Contents

Declaration.....	2
Approval	3
Acknowledgment.....	4
Abstract.....	5
Chapter 1:.....	9
Introduction.....	9
1.1 Introduction:.....	9
1.2 Related Works:.....	10
1.3 Motivation:.....	12
1.4 Problem Statement:	12
1.5 Objective:.....	12
1.6 Applications:	12
1.7 Our Contribution:.....	13
Chapter 2	14
Background study	14
2.1 Definition of Data:	14
2.2 Definition of Dataset:.....	15
2.3 Annotation of Data:.....	18
2.4 Define OCR:	21
2.5 Steps of OCR:	21
2.6 Applications of OCR:	22
2.7 The advantage of OCR:	23
2.8 Challenges in Bengali OCR:	23
2.9 Architecture of OCR:.....	24
2.10 Computer Vision:.....	25
2.11 Overview of Convolutional Neural Network:.....	25
2.12 Classification of Neural Network:	26
Chapter 3	35
Research Methodology	35
3.1 Overall working process of District name recognition:	35
3.2 Environment setup:	37
Chapter 4	38
Dataset Collection, Preprocessing and Segmentation.....	38
4.1 BN-HW-DSNd: Data Annotation Process	38

4.2 Automatic Word Segmentation:.....	40
4.3 Observation of Datasets:	45
4.4 The manual segmentation process is given below:	46
4.5 Structure of our dataset:	49
Chapter 5	50
Proposed CNN Model	50
5.1 Working Procedure of proposed model:	50
5.2 Image Preprocessing:	50
5.3 Dataset Augmentation:.....	50
5.4 Architecture of proposed CNN Model:.....	51
5.5 Importing all of the keras packages:	52
5.6 Model creation:	53
5.7 Model Compile:	55
5.8 Fitting the model:	56
Chapter 6	57
Experiments & Result	57
6.1 Training accuracy vs Validation accuracy:	57
6.2 Training Loss vs Validation Loss:	58
6.3 Comparison among classifier, AlexNet, InceptionV3 and our CNN model:	59
6.4 Model comparison in terms of test accuracy:	62
6.5 List of Hyperparameters:	63
6.6 Effects of the activation function on CNN Model:	63
6.7 Comparison of models on proposed and benchmark datasets:	63
6.8 Classification of District name recognition:	64
6.9 Miss Classification of District name recognition:.....	65
6.10 Comparison among Classifier, Alexnet and proposed CNN models in terms of Recall and F1 score:	65
6.11 Confusion Matrix:	66
6.12 Representation of sensitivity for some selected classes:.....	67
6.13 Calculating TP, FP, TN, FN for some selected class from Confusion Matrix of our proposed model:	67
6.14 Classification report based on True Positive Rate:	69
6.15 Classification Report:.....	70
6.16 Calculation of average F1 score:.....	71
Chapter 7	72
Future Work and Conclusion	72

7.1 Future Work:	72
7.2 Conclusion:	72
References:	73

Chapter 1:

Introduction

1.1 Introduction:

Bangla is the national and official language of Bangladesh [1]. There are 11 vowels, 39 consonants, ten digits in the Bangla script. Apart from this, there are some compound characters. Compound characters are the typical type of at least two consonant characters [2]. Bangla script is based on Sanskrit and has certain characters that are identical to one another since some of them can only be distinguished by a dot or a line (matra) [3]. Other than that, it is spoken in India's West Bengal, Sierra Leone, Assam, Tripura, and immigrant populations in Great Britain, the United States, and the Middle East [4]. There are 228 million people who speak it around the world. It is now the fifth most widely spoken language.

The native language of Bangladesh is Bangla. It is used in various organization including postal offices, vehicle number plate, courier services national id card, birth certificates and other govt. or organizations to maintain daily official records [5]. So, this is why automation in Bangla handwritten word recognition is becoming important day by day and Bangla handwritten document analysis could be useful.

There are several academic and commercial interests in automatic handwritten word recognition. The most challenging part of handwritten word recognition is dealing with the wide range of handwriting styles used by various writers [6]. Recognition of handwritten words are significantly more difficult than recognition of printed characters. Because handwritten characters vary in size and shape from person to person. Furthermore, writing style and inclination are not the same. But recognizing Bangla language is more complex task compared to English. Because Bangla scripts are curvy in nature and there are some compound characters which are the typical type of at least two consonant characters.

Hence, recently researchers are paying attention to handwritten word recognition of Bangla scripts. There are some related works has been done by a few researchers. The author in [7] published a system towards Indian postal document automation based on pin-code and city name recognition. The author in [7] published an Offline Handwritten Recognition of Malayalam District Name. The proposed work can be used for the recognition of district in the address written in Malayalam. The author in [8] published a real-time Bangla License Plate Recognition System for Low Resource Video-based Applications. The proposed system aims to provide a solution for detecting, localizing, and recognizing license plate characters from vehicles appearing in video frames. The author in [9] published a Real-Time Computer Vision-Based system using Contour Analysis and Prediction Algorithm which recognizes the license plate from a vehicle written in Bangla. The author in [10] introduced A Benchmark Dataset for Document Level Offline Bangla Handwritten Text Recognition (HTR). The system proposed the most extensive dataset named BN-HTRd, for Bangla handwritten images to support the advancement of end-to-end recognition of documents and texts.

Many intensive research studies have been conducted in other countries in this area. To the best of our knowledge very small amounts of research studies are done in Bangladesh in handwritten word recognition and there is no other system that recognizes the name of districts of Bangladesh from handwritten texts. It can be beneficial to digitize official

documents and create a real-time system, which can save time. It can also be helpful in development of intelligent traffic systems and finding stolen vehicles [11].

Therefore, this system introduces a novel dataset which consists of the handwritten district names of Bangladesh. The proposed system will generate an effective model and will be compared with a benchmark dataset to improve sensitivity. Here, we offer an overview of the fast and reliable Bangla OCR to extract words from Bangla handwritten images, as well as the procedure and obstacles we had while developing Bangla OCR. In this study, Deep Learning algorithms, such as the Convolutional Neural Network will be used to train the dataset to achieve the most accurate results.

1.2 Related Works: In recent years, there are few publications on the analysis of Bengali handwritten documents. Although several studies on handwritten character and digit recognition have been published in recent years, there are a few remarkable works on handwritten character and text recognition.

Deep learning approaches

Alif Ashrafee et al. [12] has proposed Real-time Bangla License Plate Recognition System for Low Resource Video-based Applications to provide solution for recognizing license plate characters from vehicles appearing in video frames. They have created a dataset of 1000 images and 79 videos of license plate and used YoloV3 tiny model along with Mobile Net SSDv2 model. They achieved a detection rate of 75.5% and 98 % respectively. MobileNet SSDv2 provides a pipeline with a very high detection rate but at the cost of a slower inference speed.

To convert handwritten text into the digital format, Batuhan Balci et al. [13] presented a deep learning based handwritten text recognition system. For classifying words directly, they used Convolutional Neural Network (CNN) with various architectures to train the model. For character segmentation they used Long Short-Term Memory networks (LSTM) with convolution to construct bounding boxes for each character. They claim that their approach achieved a training accuracy of 35 percent and validation of accuracy of 27 percent using the RESNET-34 model.

In [14] Guttula Bhargav Mani Deep et al. proposed a character recognition system using deep learning. They have used OCR and provided a CNN (Convolutional Neural Network) -based character recognition method. They extended the EMNIST dataset by adding characters in other languages. Their datasets were scaled to 28 x 28 pixels. So, their input image is pre-processed, standardized, and provided with a categorizer to predict characters. Their technique yields an accuracy of 95 percent on validation data.

Nikitha A et al. [15] suggested the approach to build a Handwritten Text Recognition system using Deep Learning. To improve accuracy, they used the strategy to recognize in terms of words rather than the characters. In their presented model, the inception architecture and four stacked bidirectional LSTMs (BLSTMs) have been used for all the CNN layers and LSTM layers respectively. Character Error Rate (CER) and Word Error Rate (WER) obtained by the CNN-1DLSTM CTC is 6.2 percent and 20.5 percent.

For Bangladeshi Vehicles, Ashim Kumar Ghosh et al. [16] has proposed an Automatic License Plate Recognition (ALPR) system using neural network which consists of three steps: license plate locating, character segmentation and character recognition. They used feed forward neural network for the classification and recognition Bangla characters. Different sized 300 JPEG colored images are used in their experiment. Their proposed model achieved accuracy of 84% for license plate extraction and 80% for character recognition.

In [17] Sukhdeep Singh et al. suggested a fine-tuned Deep Convolutional Neural Network based system for Online handwritten Gurmukhi word recognition using on offline features. Their proposed model achieved 87.9% accuracy using SVM classifier while using logistic regression they got 84.5% accuracy in offline data. Using the VGG16-DNN architecture they got accuracy of 97.23% and using InceptionV3-DNN they got 97.06% accuracy in offline data.

Machine learning approaches

Shilpi Barua et al. [18] has created a dataset of 20 city names of west Bengal having 10000 samples and proposed a machine learning algorithm for the classification of city names, written in Bangla script employing well known classifiers namely SMO, Bayes Network, Random Forest, Bagging, and Multilayer Perceptron. They used WEKA to perform the actual experiment. Using Sequential Minimal Optimization (SMO) classifier they were able to obtain 90.65% accuracy in best case using 5-fold cross validation.

In [19] S M Shamim et al. presented an approach to off-line handwritten digit recognition based on different machine learning techniques such as SVM, ANN, Bayes Net, Nave Bayes, Random Forest, J48, and Random Tree. While the majority of these algorithms perform well in terms of accuracy, they fall short in terms of evaluation metrics and visual representation. They concluded that the highest accuracy was obtained by Multilayer Perceptron with the value of 90.37%.

For Automation of Indian Postal Documents written in Bangla and English, Szilárd Vajda et al. [20] offered a system based on pin-code and city name recognition using CNN. They used RLSA and DAB for segmentation. They proposed a water reservoir concept-based feature to identify the script and NSHP-HMM (Non-Symmetric Half Plane-Hidden Markov Model) based technique for recognition of city names. They used a database of 10,342 handwritten words and 1250 printed words which were collected from postal documents. Their model has accuracy of 98.42% on printed text and their handwriting identification result was 93.27%.

Kannan Balakrishnan et al. [21] has proposed A Holistic approach to Offline Handwritten Recognition of Malayalam District Name using Machine learning. Classifiers used in their proposed system are Neural Network, SVM and RandomForest. Their Data has been collected from 56 different writers of 784 samples. In their system, they employed PCA, GRP and SRP with 50 and 100 features, where PCA outperformed the rest methods with all the classifiers. Using SVM they got comparatively good accuracy of 97 % with 100 Features within 50 epochs.

An approach for stroke segmentation and recognition from Bangla online handwritten text is proposed here by Nilanjana Bhattacharya et al. [22]. In their proposed system, combination of online and offline information has been used for better segmentation. They collected a set of 2000 Bangla words written by 50 writers using Wacom tablet. Their proposed model

achieved 97.89% accuracy using SVM classifier in 10,896 strokes. Using point-float feature in HMM they got accuracy of 81.55% and using chain-code feature in Nearest Neighbor classifier they got 91.01% accuracy.

1.3 Motivation:

Automatic handwritten word recognition has a number of academic and commercial purposes. Many intensive research studies have been conducted in other countries in this area. Bangla is used in various government and non-government organizations such as Post Office, banks, and those involved in legal decision making, vehicle number plate, courier services national id card, birth certificates to maintain daily records. Despite the fact that Bangla is one of the most widely spoken languages, little attention has been paid to the task of handwritten word recognition from documents containing Bangla scripts. The most difficult problem in the aspect of handwritten document image recognition is segmenting document images into words and text lines as the scripts are curvilinear. Due to a lack of handwritten datasets, we are unable to use the capabilities of modern ML algorithms in this field.

Therefore, we present an unsupervised segmentation methodology of hand-written documents into corresponding district names along with our dataset. The motivation for this task has come from the recent growing interest in the Bengali dataset creation and developing an automated word recognition system for Bangla handwritten texts.

1.4 Problem Statement:

Since there is not any existing dataset available of district names of Bangladesh, so we will create a novel dataset which will contain 64 handwritten district names of Bangladesh. There is no state of the art model to recognize handwritten district names so we will develop a system to recognize handwritten district names. Then, our proposed model will be compared with a benchmark dataset. So, it is important to design an effective model to improve sensitivity which will be able to recognize Bangla handwritten district name more accurately.

1.5 Objective:

The primary goal of this study is to design an efficient deep learning-based model to recognize handwritten district name.

Sub goal-1: To create a novel dataset which will contain 64 district names.

Sub goal 2: To generate an effective model to improve true positive rate which will recognize handwritten district names.

Sub goal-3: To compare the proposed model with another benchmark dataset.

1.6 Applications:

- Postal Automation
- Address identification
- Image to text conversion
- Bengali number plate identification.
- Bank check automation

1.7 Our Contribution:

We have created a novel dataset BN-HW-DSNd: A Bangla HW District Name Dataset [23] that can be used in automation of Bengali handwritten text. And we have collected different handwritten district names from various ages, genders, and backgrounds with varying writing styles.

The rest of the report is organized as follows: Chapter 2 explains the background study and overview of CNN and related works, chapter 3 explains the details about research methodology, chapter 4 describes the process of Dataset Collection, Preprocessing and Segmentation, chapter 5 describes the Proposed CNN Model, chapter 6 discusses experiments & results and chapter 7 concludes with future work and conclusion.

Chapter 2

Background study

2.1 Definition of Data:

A collection of information, such as numbers, texts, measurements, observations, or simple descriptions of things, is referred to as data. It is a collection of facts that may be studied or used to gather information or make decisions. It endures and serves no purpose other than to exist. It can take any form, be it useful or not. It looks to be meaningless in and of itself. In computer terms, a spreadsheet usually starts with data collection. Data is a term that represents a point or remark about an occurrence that is distinct from other items [24].

There are two types of data:

- Qualitative: Qualitative data is information that is descriptive
- Quantitative: Quantitative data is information that is numerical in nature.

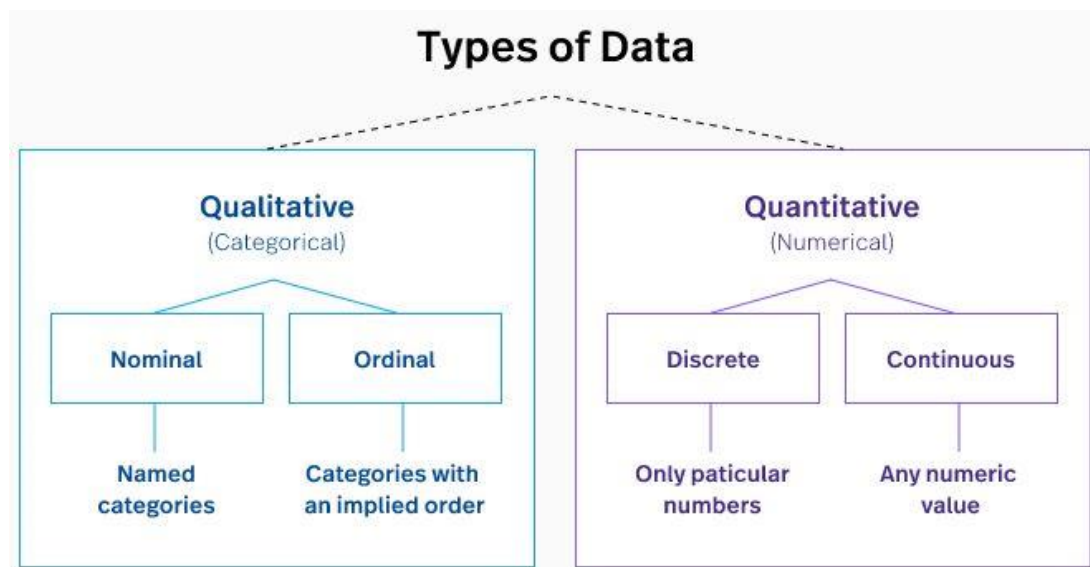


Figure 2.1: Types of Data

2.1.1 Importance of Data:

Data is, at its most basic level, a collection of distinct facts, such as statistics, measurements, and observations, that have been converted into a format that computers can understand. This may appear to be a simple task, but data is fundamentally altering the society we live in and the way we work. If you own a business and want to expand, you probably already know that data is critical to taking the next step.

Here's a summary of why data collection is so important:

- People can make insightful decisions with the help of data.
- People can identify problems with the help of data.
- The use of data allows us to create accurate theories.
- Our views will be backed up by data.
- Our approach becomes strategic with the help of data.

- Data assists you in getting hand-on funding.
- Data tells us how well we're doing.
- People can save time by data.

2.2 Definition of Dataset:

An orderly collection of data is referred to as a data set. The data set can be a collection of tables, schema, and other objects that are used to manipulate the data. The information is simply structured into a model that aids in the processing of the data. Any permanently recorded collection of information that often contains case-level, obtained data or statistical guideline level data is referred to as a set of data [25].

2.2.1 Purpose of dataset:

It can be used to remotely process data before updating the database with the new information. As a result, it is possible to work with data in a disconnected manner. This boosts performance by lowering the number of times a database is consulted for data manipulation.

Some examples of dataset:

- International Students/Faculty dataset
- Human Resource dataset
- Research dataset
- Intellectual Property dataset
- Financial dataset
- Legal dataset
- Law Enforcement dataset
- Health dataset
- Email dataset
- Audit dataset
- Security dataset

2.2.2 Types of Datasets:

2.2.2.1 Numerical dataset:

A numerical data set is one in which the information is expressed in numbers rather than natural language. Quantitative data is a term used to describe numerical data. The numerical data set is the collection of all quantitative/numerical data. Numerical data is always in the form of numbers, allowing us to execute arithmetic operations on it. Such as: A person's weight and height, in a medical report, the number of RBCs is counted, the number of pages in a book.

#Dataset	# Instances	# Attributes	# Numeric	# Nominal	# Classes	Base Rate [%]	Rnd. Choice [%]
1 Appendicitis	106	8	7	1	2	80.2	50.0
2 Breast C.W.	286	10	0	10	2	70.3	50.0
3 Horse Colic	368	23	7	16	2	63.0	50.0
4 Credit rating	690	16	6	10	2	55.5	50.0
5 German credit	1000	21	8	13	2	70.0	50.0
6 Pima I.D.	768	9	8	1	2	65.1	50.0
7 Glass	214	10	9	1	6	35.5	16.7
8 Cleveland heart	303	14	6	8	2	54.4	50.0
9 Hungarian heart	294	14	6	8	2	63.9	50.0
10 Heart Statlog	270	14	13	1	2	55.6	50.0
11 Hepatitis	155	20	2	18	2	79.4	50.0
12 Labor	57	17	8	9	2	64.7	50.0
13 Lymphography	148	19	0	19	4	54.8	25.0
14 Primary Tumor	339	18	0	18	21	24.8	4.8
15 Sonar	208	61	60	1	2	53.4	50.0
16 Voting	435	17	0	17	2	61.4	50.0
17 Zoo	101	18	0	18	7	40.6	14.3
Average	337.8	18.2	8.2	9.9	3.8	58.4	41.8

Figure 2.2: Example of Numerical Dataset

2.2.2.2 Bivariate dataset:

A bivariate data set is one that contains two variables. It is concerned with the relationship that exists between the two parameters. Typically, a bivariate dataset has two categories of connected data.

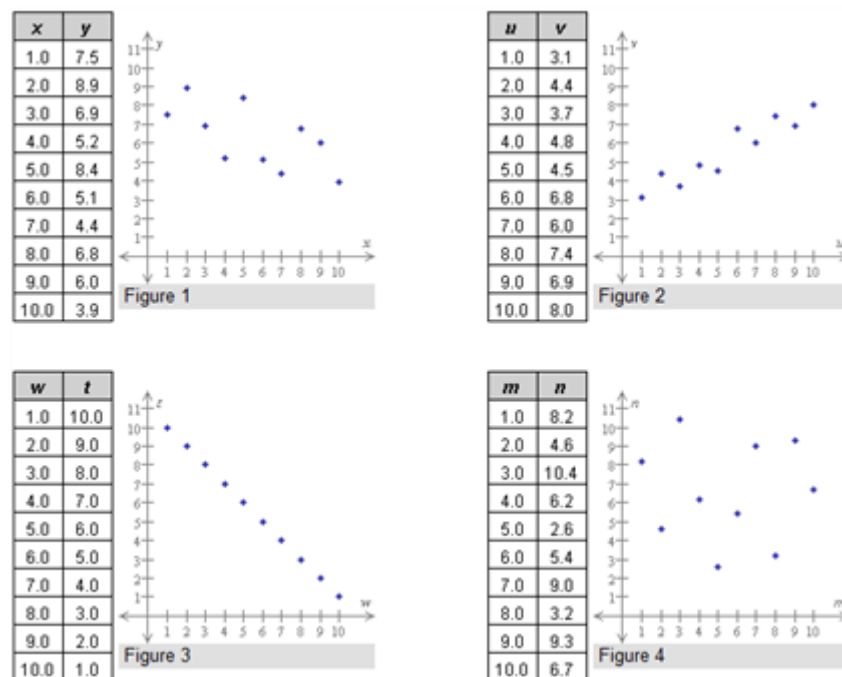


Figure 2.3: Example of Bivariate Dataset

2.2.2.3 Multivariate dataset:

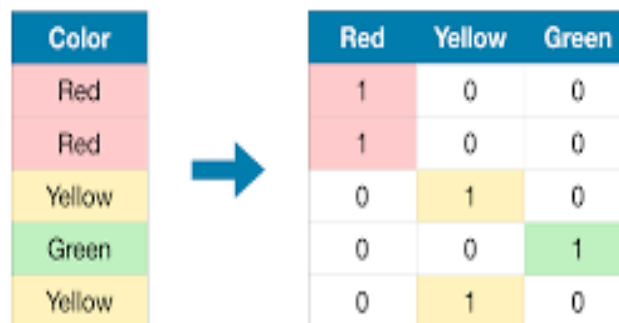
A data set having a large number of variables. A multivariate dataset is defined as one that has three or more data kinds (variables). To put it another way, the multivariate dataset is made up of individual measurements taken as a function of three or more factors.

# Dataset	# Instances	# Attributes	# Numeric	# Nominal	# Classes	Base Rate [%]	Rnd. Choice [%]
1 Appendicitis	106	8	7	1	2	80.2	50.0
2 Breast C.W.	286	10	0	10	2	70.3	50.0
3 Horse Colic	368	23	7	16	2	63.0	50.0
4 Credit rating	690	16	6	10	2	55.5	50.0
5 German credit	1000	21	8	13	2	70.0	50.0
6 Pima I.D.	768	9	8	1	2	65.1	50.0
7 Glass	214	10	9	1	6	35.5	16.7
8 Cleveland heart	303	14	6	8	2	54.4	50.0
9 Hungarian heart	294	14	6	8	2	63.9	50.0
10 Heart Statlog	270	14	13	1	2	55.6	50.0
11 Hepatitis	155	20	2	18	2	79.4	50.0
12 Labor	57	17	8	9	2	64.7	50.0
13 Lymphography	148	19	0	19	4	54.8	25.0
14 Primary Tumor	339	18	0	18	21	24.8	4.8
15 Sonar	208	61	60	1	2	53.4	50.0
16 Voting	435	17	0	17	2	61.4	50.0
17 Zoo	101	18	0	18	7	40.6	14.3
Average	337.8	18.2	8.2	9.9	3.8	58.4	41.8

Figure 2.4: Example of Multivariate Dataset

2.2.2.4 Categorical dataset:

Categorical data sets describe a person's or an object's attributes or qualities. A categorical variable, also known as a qualitative variable, exists in the categorical dataset and can take just two values. As a result, it's known as a dichotomous variable.



Color	Red	Yellow	Green
Red	1	0	0
Red	1	0	0
Yellow	0	1	0
Green	0	0	1
Yellow	0	1	0

Figure 2.5: Example of Categorical Dataset

2.2.2.5 Correlation dataset:

Correlation data sets are a collection of variables that have some sort of relationship with one another. The parameters are obtained to be interdependent in this case.

(a) A Market Basket Database		(b) Item Pairs with Upper Bounds and Correlation Coefficients		
TID	Items	Pair	Upper Bound	Correlation
1	a, b, c	{a, b}	0.667	0.667
2	a, b, c	{a, c}	0.333	-0.333
3	a, c	{a, d}	0.218	0.218
4	a, b	{a, e}	0.167	0.167
5	a, b	{a, f}	0.111	0.111
6	a, b	{b, c}	0.5	-0.5
7	a, b, c, d, e, f	{b, d}	0.327	0.327
8	a, b, d, e	{b, e}	0.25	0.25
9	a, b, d	{b, f}	0.167	0.167
10	c	{c, d}	0.655	-0.218
		{c, e}	0.5	0
		{c, f}	0.333	0.333
		{d, e}	0.764	0.764
		{d, f}	0.509	0.509
		{e, f}	0.667	0.667

Figure 2.6: Example of Correlation Dataset

2.2.3 Characteristics of Dataset:

Identifying the nature of the data is critical before performing any statistical study. Several Exploratory Data Analysis (EDA) techniques can be used to help uncover data features so that relevant statistical procedures can be applied to the data. The following aspects of the dataset can be checked using EDA techniques:

- Centre of data
- Skewness of data
- Spread among the data members
- Presence of outliers
- Correlation among the data
- Type of probability distribution that the data follows

2.3 Annotation of Data:

2.3.1 Annotation: Annotation is a bit of extra information associated with a specific location in a document or a piece of data. Annotations are sometimes depicted near the edge of text pages. We annotate a variety of digital media, including web annotation and text annotation. Annotations can be applied in a number of ways. They can be used to add information to a website about a certain word or phrase. They can also be used to provide additional information at the end of a publication. This latter example could contain an annotated bibliography that provides extra information about the origin used to the reader.

2.3.1.1 Approaches of Annotation:

- Summarizing
- Questioning
- Predicting
- Making connections
- Finding the main idea and key details § Outlining text structure

- Identifying and defining new words

2.3.1.2 Data Annotation:

The technique of labeling data so that machines may utilize it is known as data annotation. It's notably valuable in supervised machine learning (ML), where the system processes, understands, and learns from input patterns to produce desired outputs. Before data is delivered to a system in machine learning, it is annotated. The method is comparable to teaching toddlers with flashcards. A flashcard containing an apple picture and the word "cat" would show the kids how a cat looks and how to spell the word. The label in that case is the word "cat".

2.3.1.3 Types of Data Annotation:

- **Text Annotation:** This type of annotation includes adding any additional notes to a text, whether it's your own or someone else's. Text annotation can be utilized for personal usage, as well as to inform and collaborate with others. It is frequently inserted after the original text is written and can be used for a variety of purposes.

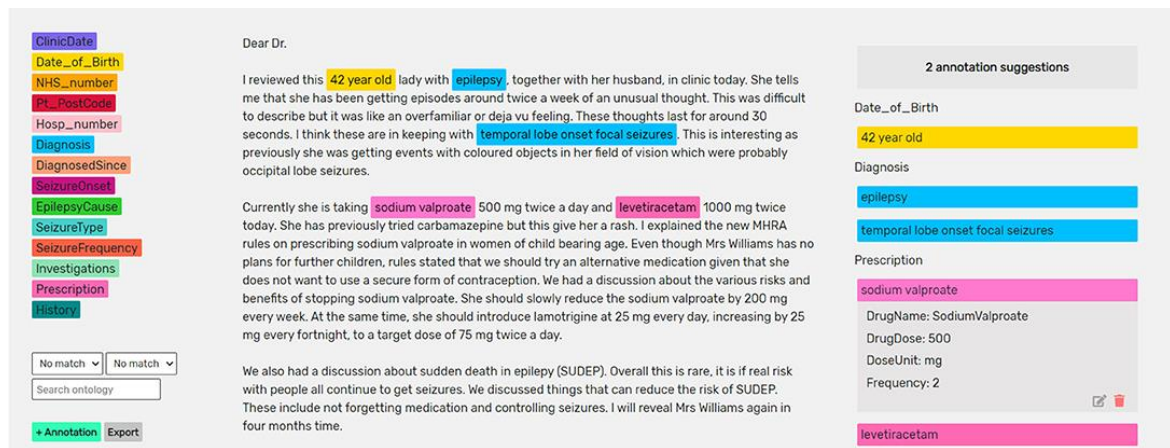


Figure 2.7: Annotation of Text [26]

- **Image Annotation:** The process of categorizing photos in a dataset in order to train a machine learning model is known as image annotation. As a result, picture annotation is utilized to indicate the aspects your system needs to recognize. Supervised Learning is the process of training an ML model given labeled data. The annotation task usually involves manual work, sometimes with computer-assisted help. A Machine Learning engineer predetermines the labels, known as “classes”, and provides the image-specific information to the computer vision model. After the model is trained and deployed, it will predict and recognize those predetermined features in new images that have not been annotated yet.

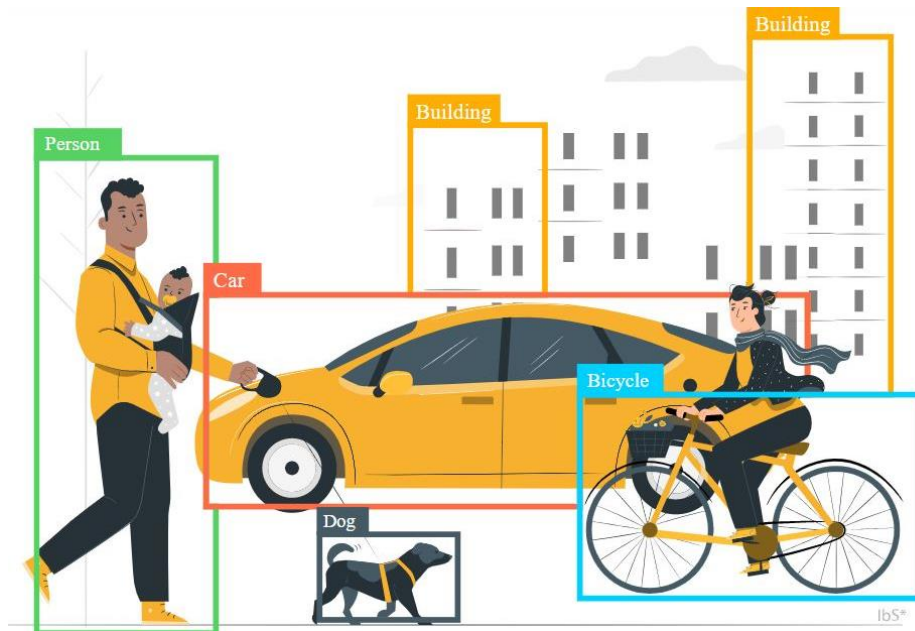


Figure 2.8: Annotation of Image [27]

- Audio Annotation:** Audio annotation is a kind of data annotation that entails classifying audio components from persons, animals, the surroundings, instruments, and other sources. Engineers employ data formats including MP3, FLAC, and AAC for the annotating process. Audio annotation, like all other types of annotation (such as image and text annotation), necessitates physical labor and annotation software designed specifically for the task. When it comes to audio annotation, data scientists use software to specify the labels or "tags" and then pass the audio-specific data to the NLP model being trained [28].

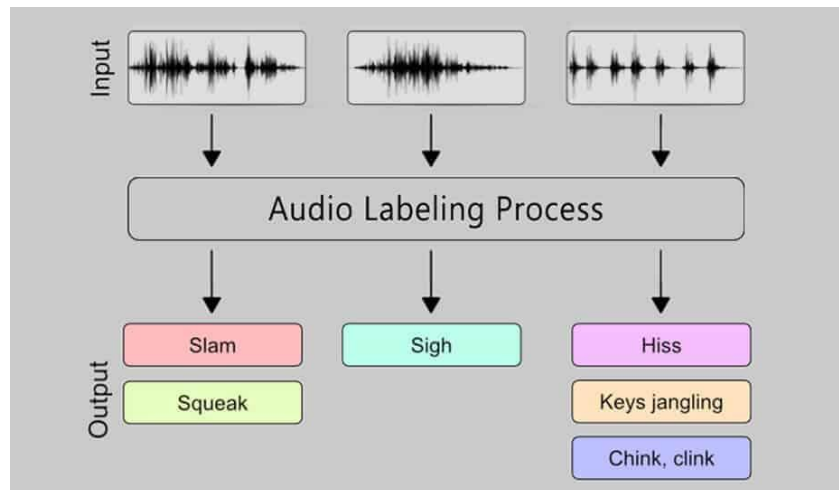


Figure 2.9: Audio Annotation [29]

- Video Annotation:**
 The technique of labeling or tagging video clips in order to train computer vision models to detect or identify things is known as video annotation. Video annotation differs from image annotation in that it includes labeling objects frame by frame to make them recognizable to machine learning models. For optimal machine learning

capabilities, high-quality video annotation creates ground truth datasets. Many fields, like self-driving cars, medical AI, and geospatial technology, have deep learning applications for video annotation [30].



Figure 2.10: Annotation from a Video [31].

2.4 Define OCR:

The acronym OCR stands for optical character recognition. It entails converting scanned images of handwritten and typewritten text into machine text via mechanical and electrical means. It's a typical way of digitizing printed texts so that they can be electronically searched, stored more compactly, displayed on line, and used in machine processes such as machine translation, text to speech and text mining. In recent years, OCR (Optical Character Recognition) technology has revolutionized the document management process across a wide range of sectors. OCR has made scanned documents more than just image files, allowing them to become fully searchable documents with text content that computers can recognize [32]. With the help of OCR, people no longer need to manually retype important documents when entering them into electronic databases. Whereas, OCR extracts relevant information and enters it automatically. As a result, information is processed more accurately and efficiently in less time. For simple languages like English, the recognition system works well. However, for complicated languages such as Bangla, the OCR system is still in its infancy.

2.5 Steps of OCR:

The steps of optical character recognition (OCR) [33] are:

2.5.1 Grayscale: A grayscale image is generated by converting a normal image to a grayscale image. The colors in this image are equally intense in Red, Green, and Blue. Luminosity Method algorithm used.

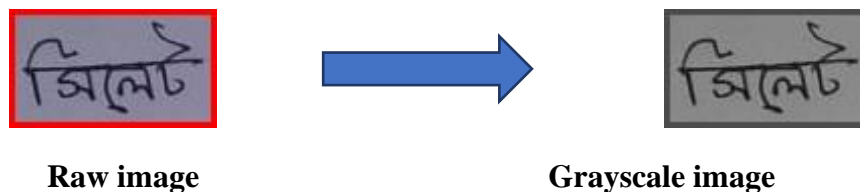


Figure 2.11: Raw image to grayscale image.

2.5.2 Binarization: Binarization is the process of converting the image into black and white representation. In this process intensity information of the picture will be reduced to two values respectively 0 and 1. For binarization Otsu's Algorithm is used.



Figure 2.12: Original Image to Binary Image of “চট্টগ্রাম”

2.5.3 Noise Removing & Image Sharpening: Random fluctuations in brightness, color, and intensity that typically make image processing more difficult. This will result in dark pixels in bright areas (pixels) and bright pixels in dark areas. Erosion and Dilation Algorithm was used for Image Sharpening. Erosion reduces image size by removing small extrusions. Dilation broadens the region by filling in small intrusions.

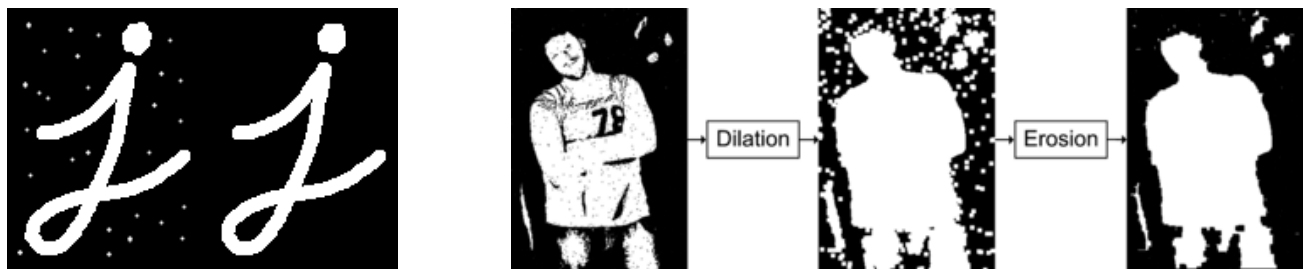


Figure 2.13: Removing Noise and Sharpening of an image

2.5.4 Segmentation: Segmentation is the process of dividing an image into subparts for further processing. Image segmentation is performed in the following order:

- Line segmentation
- Word segmentation
- Character segmentation.

2.6 Applications of OCR:

There are many applications of OCR [34], such as:

- **Postal Automation:** OCR technology can be used for inbound and outbound international mail sorting. OCR locates and reads a country name, which may be written in Bangla as well as in the language of the destination country.
- **Number Plate Recognition:** OCR technology is used to identify the number of license plates in automatic number-plate recognition. Today, number-plate recognition is widely used in a variety of commercial applications, including locating stolen vehicles, calculating parking costs, invoicing tolls, and controlling access to safety zones, among others.
- **Bank Cheque Automation:** The application of OCR technology in the banking industry has been hugely beneficial in that it has made banking processes and business transactions much faster. When OCR technology was first applied to the banking

industry, its capability of reading cheque numbers immensely helped in the processing of documents. The improvements made to the OCR technology since then have only served to improve the banking industry. Today, with the help of the OCR technology and a scanner, passbooks can be updated and scanned to the last entry. Once the transaction is completed, passbook printers can print the entries of the account. As the adoption of OCR technology has increased gradually with the passage of time, the need for human intervention has decreased gradually and with that, there have been fewer errors caused by humans as well.

- **National ID recognition system:** OCR technology extracts all information from ID cards. All the information pulled from the captured ID card will be in a simple text/numerical format. This helps to maintain data in an organized fashion and facilitates any sort of verification or registration process.
- **Parking validation:** Cities and towns are using mobile OCR to automatically validate if cars are parked according to city regulations. Parking inspectors can use a mobile device with OCR to scan license plates of vehicles and check with an online database to see if they are permitted to park.
- **Mobile document scanning:** A variety of mobile applications allow users to take a photo of a document and convert it to text. This OCR task is more challenging than traditional document scanners, because photos have unpredictable image angles, lighting conditions, and text quality.

2.7 The advantage of OCR:

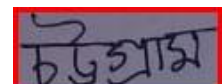
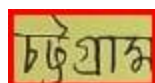
Optical Character Recognition offers a wide range of benefits. Such as:

- **Improve accuracy:** Software based character recognition eliminates human errors, resulting in improved accuracy.
- **Speed-up the processes:** The technology converts unstructured data into searchable information, providing the required data available at faster rates and subsequently speeding up business processes.
- **Cost-effective:** OCR technology does not require a lot of resources which reduces the processing costs and subsequently reduces the overall costs of a business. 2
- **Enhance Customer Satisfaction:** The accessibility of searchable data by the customers ensures a good experience, assuring better customer satisfaction.
- **Improve productivity:** The easy accessibility of searchable data makes a stress-free environment for the employees, allowing them to focus on the main goals, boosting the productivity of a business [35].

2.8 Challenges in Bengali OCR:

Bengali OCR is more challenging than other OCR due to the following reason:

- **Shape of Character:** In Bangla language, some characters are same in shape. They are quite similar which make Bangla OCR more challenging. Ex: ঢ & ড, ষ & য়, ব & র etc.
- **Size of Character:** Handwritten characters written by different writers are not only nonidentical but also vary in different aspects such as size and shape. Example:



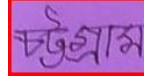
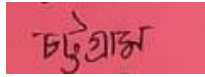


Figure 2.14: Different handwritten representation of ‘চট্টগ্রাম’

- **Overlap Issue:** Another challenge of Bangla OCR is overlap with one character to another character or diacritics. When our system is going to classify individual Characters, they conflict with this overlap issue. For example:

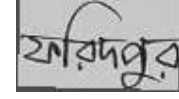
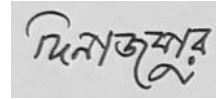
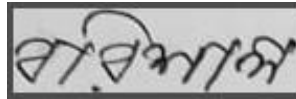
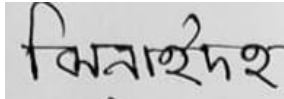


Figure 2.15: ি is overlapping with the next following character in each

- **Error in Segmentation:** When the value of intensity gradient and no. of iteration is comparatively lower than regular color image intensity gradient then the segmentation process doesn't work properly.

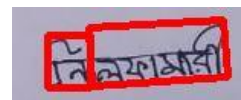
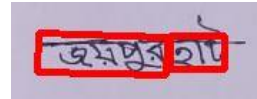
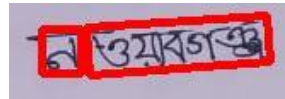
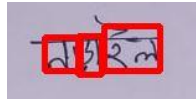


Figure 2.16: Error in segmenting words

2.9 Architecture of OCR:

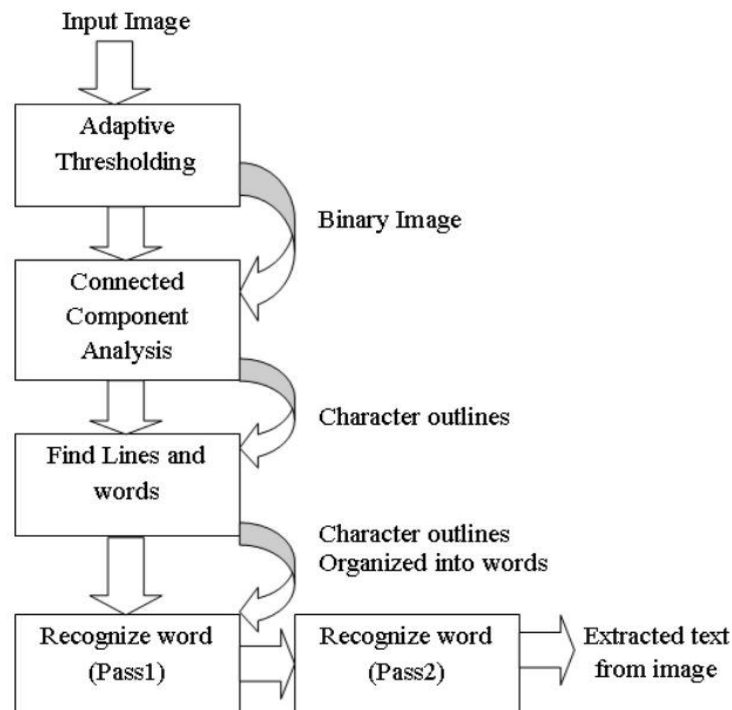


Figure 2.17: Architecture of OCR [36]

2.10 Computer Vision:

One of the most promising fields of research in artificial intelligence and computer science is computer vision technology. In today's world, it has a lot of benefits for businesses. It is a multidisciplinary area that can be categorized as a subfield of artificial intelligence and machine learning, and it may employ specialized approaches as well as general learning algorithms.[37]

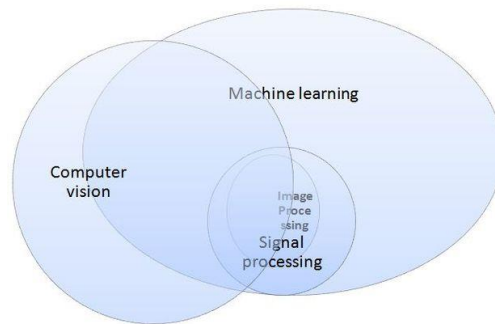


Figure 2.18: An Illustration of the Artificial Intelligence and Computer Vision Relationship

Computer vision helps to clarify the content of digital images. Typically, this leads to the creation of methods that seek to replicate human eyesight. Extracting a description from a digital image, which could be an item, a text description, a three-dimensional model, and so on, is one way to understand the content of the image. The primary goal of many computer vision applications is to recognize objects in images. For example:

- Object Classification: What broad category of object is in this image?
- Object Verification: Is the object in the image?
- Object Detection: Where are the objects in the image?
- Object Recognition: What objects are in this photograph and where are they?

2.11 Overview of Convolutional Neural Network:

2.11.1 Neural Network: A neural network is a computational learning system that uses a network of functions to understand and translate a data input of one form into a desired output, usually in another form. The concept of the artificial neural network was inspired by human biology and the way neurons of the human brain function together to understand inputs from human senses. Neural networks are just one of many tools and approaches used in machine learning algorithms. The neural network itself may be used as a piece in many different machine learning algorithms to process complex data inputs into a space that computers can understand. "The neural network is this kind of technology that is not an algorithm, it is a network that has weights on it, and you can adjust the weights so that it learns," Howard Rheingold explained. You teach it through trials." We can use neural networks to cluster and classify data. Deep neural networks can extract features from neural networks for clustering and classification. Today, neural networks are being used to solve a variety of real-world problems, including speech and image recognition [38].

2.11.2 Working process of Neural Networks: There are many layers in a neural network. Each layer serves a specific purpose, and the more layers there are, the more complex the network. As a result, a neural network is also known as a multi-layer perceptron. A neural network in its purest form has three layers: input, hidden, and output. Each of these layers

serves a specific purpose, as their names simply. These layers are comprised of nodes. A neural network can have multiple hidden layers depending on the requirements. The input layer receives input signals and forwards them to the next layer. It collects information from the outside world. The hidden layer handles all of the calculation's back-end tasks. A network can have no hidden layers at all. A neural network, on the other hand, has at least one hidden layer. The final result of the hidden layer's calculation is transmitted by the output layer.

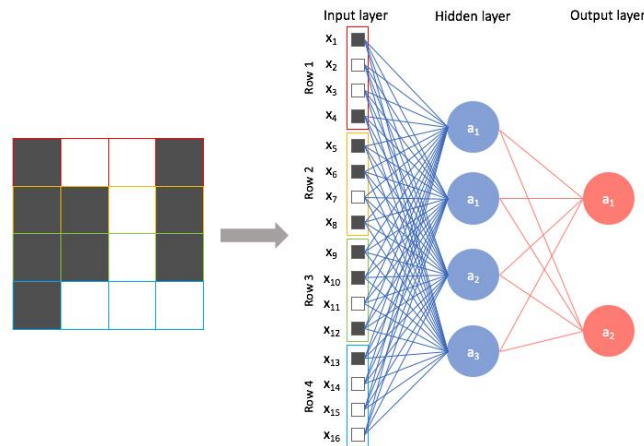


Figure 2.19: Working process of neural network.

2.12 Classification of Neural Network:

Different types of neural networks use different concepts to determine their own rules. There are numerous types of artificial neural networks, each with its own set of advantages.

2.12.1 Feed-Forward Neural Network:

This is the most basic type of artificial neural network. Data flows in only one direction in this network, from the input layer to the output layer. The output layer in this network receives the sum of the products of the inputs and their weights. This neural network does not use back-propagation. These networks could have a large number of hidden layers or none at all. These are less difficult to maintain and have applications in face recognition.

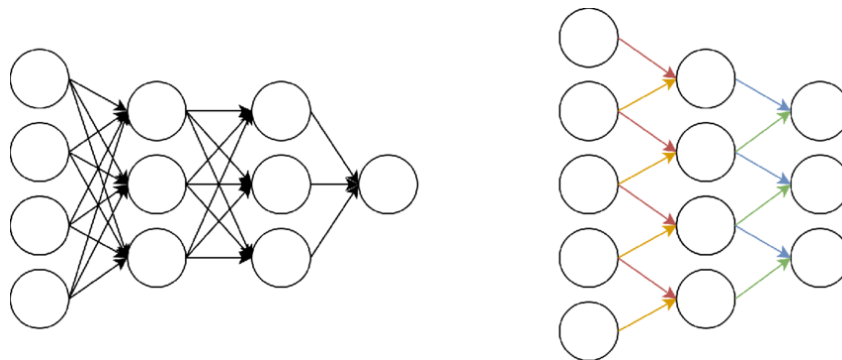


Figure 2.20: Feed-Forward Neural Network

2.12.2 Radial Basis Function Neural Network: A radial basis function is used in this neural network. This function takes into account a point's distance from the center. These networks are made up of two layers. The features are combined with the radial basis function in the hidden layer, and the output is transferred to the next layer. The following layer repeats the

previous layer's actions while using the output of the previous layer. In power systems, radial basis function neural networks are used.

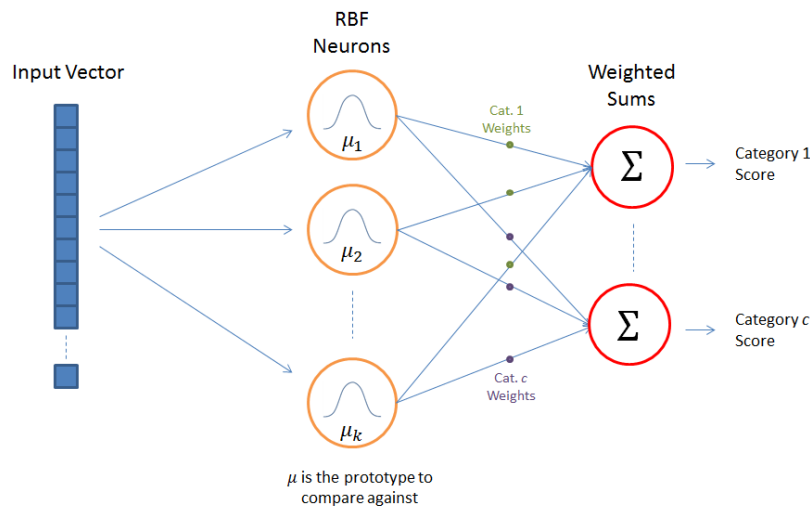


Figure 2.21: Radial Basis Function Neural Network

2.12.3 Modular Neural Network: This network contains several networks that operate independently. They all have specific tasks to complete, but they do not interact with one another during the computation process. A modular neural network can thus perform a highly complex task much more efficiently. These networks are more difficult to maintain than simpler networks (such as FNN), but they produce faster results for complex tasks.

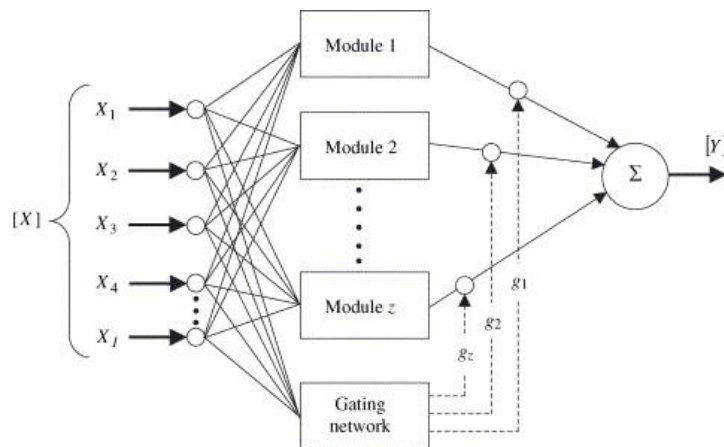


Figure 2.22: Modular Neural Network

2.12.4 Recurrent Neural Network (RNN):

The output of a layer is saved and transferred back to the input in this network. In this manner, the nodes of a specific layer remember some information about previous steps. The sum of weights and features is the product of the input layer's combination. In the hidden layers, the recurrent neural network process begins. In this case, each node remembers some of the information of its predecessor step. The model saves some information from each iteration for future use. When the system's outcome is incorrect, it self-learns. It then uses that information to improve the accuracy of its prediction in back-propagation. The most common application of RNN is in text-to-speech technology.

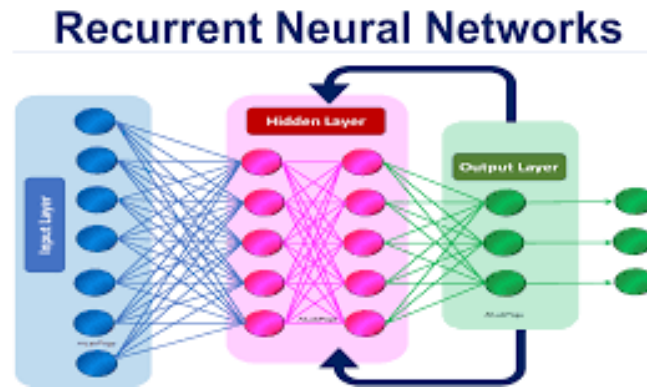


Figure 2.23: Recurrent Neural Network

2.12.5 Convolutional Neural Network (CNN):

CNN, also known as ConvNet (convolutional neural network), is a type of multilayer perceptron. A convolutional neural network is composed of many layers, including convolution layers, pooling layers, and fully connected layers, and it employs a backpropagation algorithm to automatically and adaptively learn spatial hierarchies of features. A convolutional neural network (CNN/ConvNet) is a type of deep neural network that is commonly used to analyze visual imagery⁴ in deep learning. A CNN contains one or more convolutional layers. These layers may be completely connected or grouped together. Before passing the output to the next layer, the convolutional layer performs a convolutional operation on the input. Because of the convolutional technique, the network can be much deeper while having far fewer parameters. Each input image will be passed through a sequence of convolution layers using filters (Kernels), Pooling, fully connected layers (FC), and the SoftMax function to categorize an object with probabilistic values ranging from 0 to 1. The diagram below depicts the flow of CNN to process an input image and classify objects based on values.

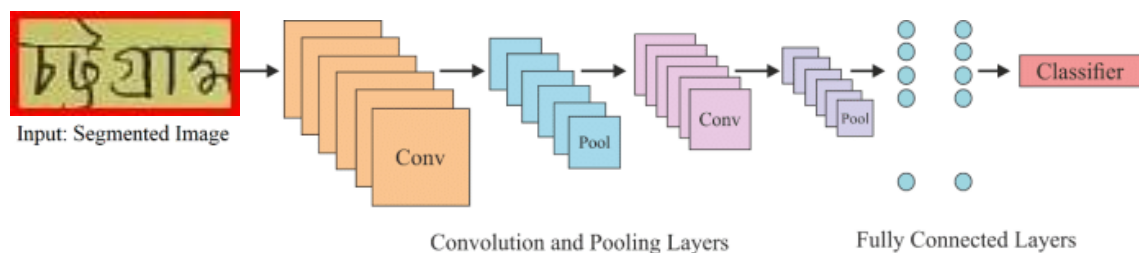


Figure 2.24: Convolutional Neural Network

2.12.5.1 Padding: Padding is a term that is relevant to convolutional neural networks because it refers to the number of pixels that are added to an image when it is processed by the CNN kernel. For example, if the padding in a CNN is set to zero, then every pixel value added will be zero. If the zero padding is set to one, a one-pixel border with a pixel value of zero will be added to the image. Padding works by increasing the size of the area over which a convolutional neural network processes an image. The kernel is the neural network filter that moves across the image, scanning each pixel and converting the data into a smaller, or occasionally larger, format. Padding is added to the image frame to allow for more space for the kernel to cover the image while it is being processed by the kernel. Adding padding to a CNN-processed image allows for more accurate image analysis.

Types of Padding:

- **Valid Padding:** Padding that is valid means that there is no padding. That is, the input image is not padded, and it is assumed that all dimensions are valid so that the filter and the stride specified by our needs fully cover the input image. This means that the filter window will always be contained within the 19-input image. This padding is referred to as valid padding because it only takes into account the valid and original elements of the input image.
- **Same Padding:** Same Padding adds padding to the input image so that it is completely covered by the filter and the specified stride. It's called SAME because the output for stride 1 will be the same as the input.
- **Zero Padding:** Zero padding is a technique for adding zeroes to an input matrix in a systematic manner. It's a suitable technique that allows us to change the size of the input to meet our needs. If zero padding = 1, the original image will be surrounded by one pixel of padding with pixel value = 0.

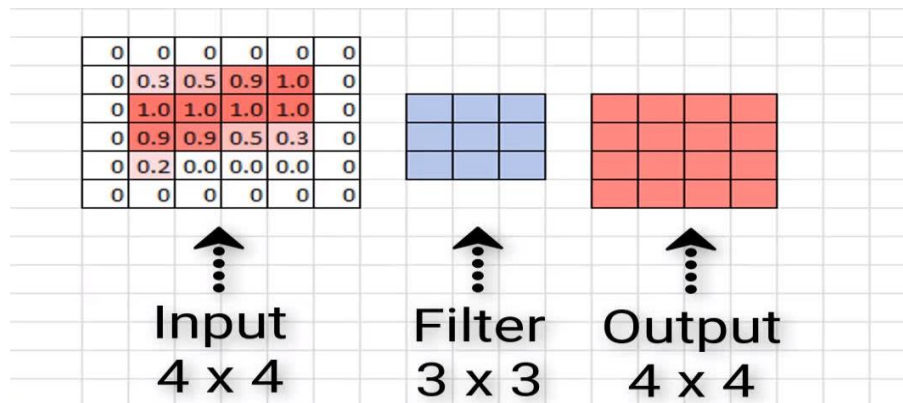


Figure 2.25: Zero Padding

2.12.5.2 RELU (Rectified linear units): In deep learning models, the Rectified Linear Unit is the most commonly used activation function. If it receives any negative input, it returns 0, but if it receives any positive input, it returns that value. As a result, it can be written as $f(x) = \max(0, x)$.

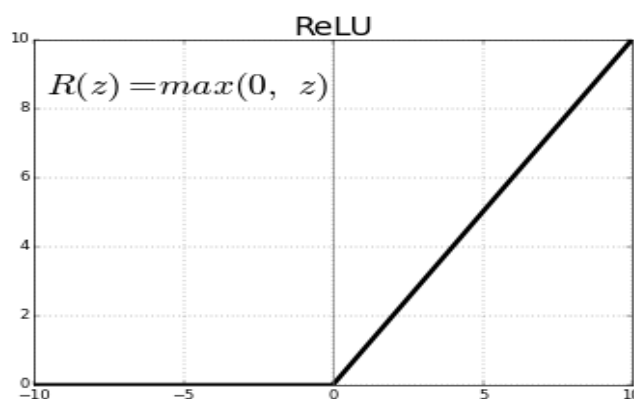


Figure 2.26: ReLU activation function

2.12.5.3 Pooling Layers: A Convolutional Layer is usually followed by a Pooling Layer. This layer's primary goal is to reduce the size of the convolved feature map in order to reduce computational costs. This is accomplished by reducing the connections between layers and operating independently on each feature map. Pooling operations are classified into several

types based on the method used. The largest element from the feature map is used in Max Pooling.

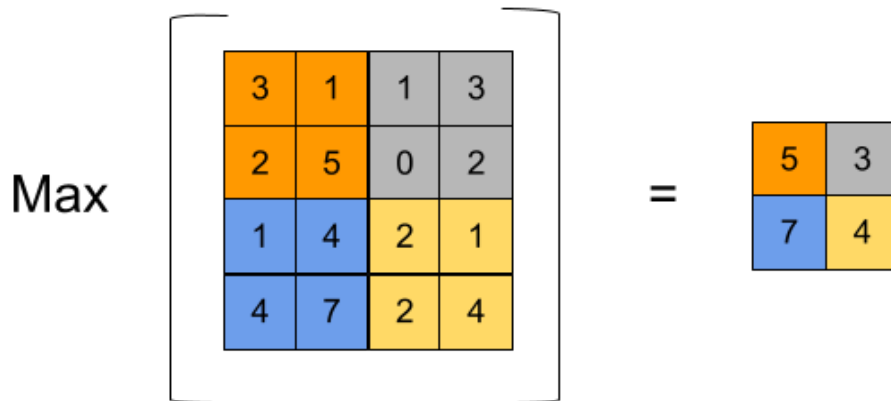


Figure 2.27: Pooling Layers

Average Pooling computes the average of the elements in a predefined image section size. The minimum element is taken from the feature map when using Min Pooling. Sum Pooling computes the total sum of the elements in the predefined section. Typically, the Pooling Layer acts as a link between the Convolutional Layer and the FC Layer.

2.12.5.4 Flatten layer: Flattening is the process of converting data into a one-dimensional array for input into the next layer. The output of the convolutional layers is flattened to create a single long feature vector. It is also linked to the final classification model, which is referred to as a fully-connected layer.

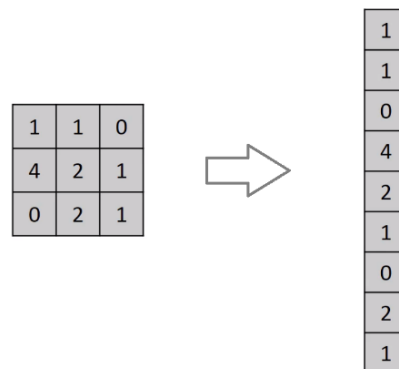
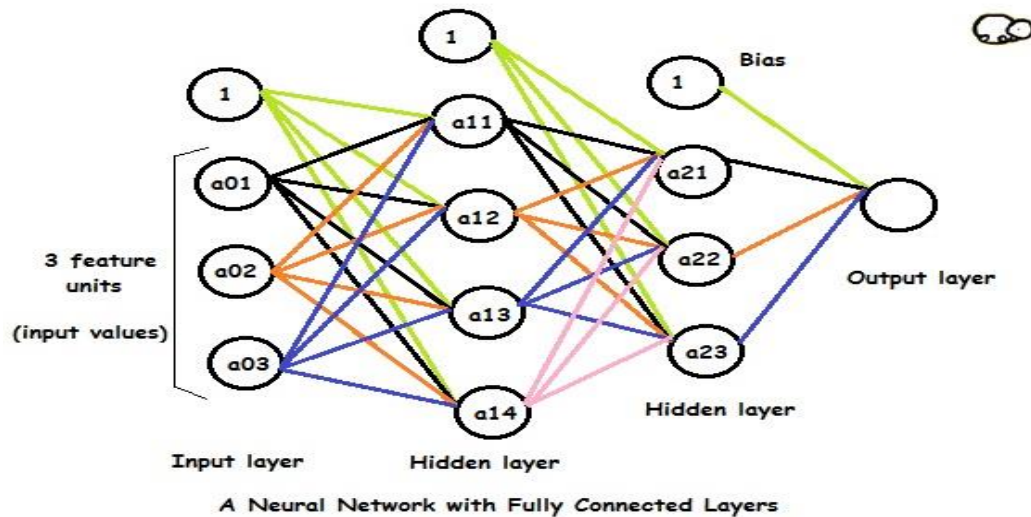


Figure 2.28: Flatten Layer

2.12.5.5 Fully Connected Layer: Basically, fully connected layers are feed forward neural networks. Fully connected layers in neural networks are those in which all of the inputs from one layer are connected to every activation unit in the following layer. The final few layers of most popular deep learning models are full connected layers that compile the data extracted by previous layers to form the final output. The Fully Connected (FC) layer, which includes weights and biases as well as neurons, is used to connect neurons from different layers. These layers are typically placed prior to the output layer and constitute the final few layers of a CNN Architecture. The previous layers' input images are flattened and fed to the FC layer in this step. The flattened vector is then passed through a few more FC layers, where the

mathematical function operations are typically performed. At this point, the classification procedure is initiated.

Figure 2.29: Fully Connected Layers



2.12.5.6 Batch normalization:

Batch normalization is a method for automatically standardizing the inputs to a deep learning neural network layer. Batch normalization is a method for normalizing the inputs of each layer in order to combat the internal covariate shift problem. Batch normalization normalizes the output of a previous activation layer by subtracting the batch mean and dividing by the batch standard deviation⁵ to improve the stability of a neural network. However, the weights in the next layer are no longer optimal after this shift/scale of activation outputs by some randomly initialized parameters. Batch normalization adds two trainable parameters to each layer, so the normalized output is multiplied by a "standard deviation" (gamma) parameter and a "mean" parameter is added (beta).

During training time, a batch normalization layer does the following:

1. Calculate the mean and variance of the layers input.

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i \quad \text{Batch mean}$$

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \quad \text{Batch Variance}$$

1. Normalize the layer inputs using the previously calculated batch statistics.

$$\bar{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2}}$$

2. Scale and shift in order to obtain the output of the layer.

$$y_i = \sqrt{\gamma} \bar{x}_i + \beta$$

2.12.5.7 Dropout: Dropout is a training technique in which randomly selected neurons are ignored. They're "dropped out" at random. This means that their contribution to downstream

neuron activation is removed temporally on the forward pass, and any weight updates are not applied to the neuron on the backward pass. Dropout is a method of regularization. When all of the features are connected to the FC layer, the training dataset is prone to overfitting. Overfitting occurs when a model performs so well on training data that it has a negative impact on the model's performance when applied to new data. To address this issue, a dropout layer is used, in which a few neurons are removed from the neural network during the training process, resulting in a smaller model⁶. When a dropout of 0.3 is reached, 30% of the nodes in the neural network are dropped out at random.

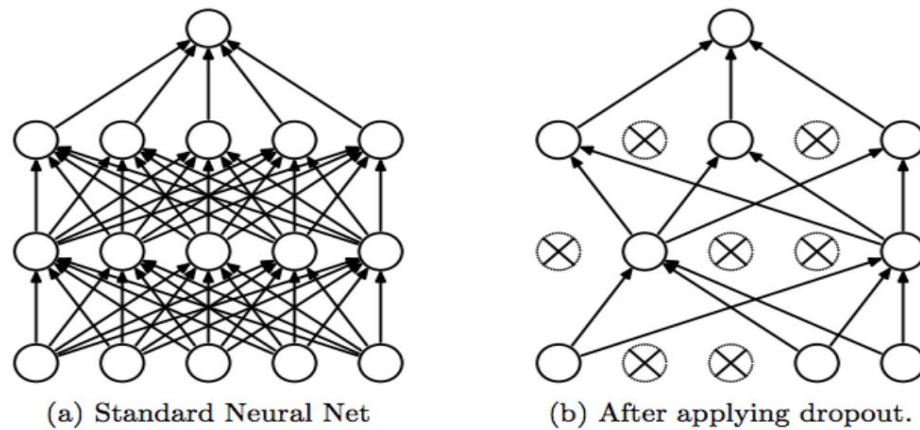


Figure 2.30: Dropout

2.12.5.8 Backpropagation: Backpropagation is the primary mechanism used by neural networks to learn. It is the messenger's job to inform the network whether or not the network made a mistake when making a prediction. To propagate is to send something (light, sound, motion, or information) in a specific direction or through a specific medium. When we talk about deep learning backpropagation, we're talking about the transmission of information, and that information is related to the error produced by the neural network when it makes a guess about data. After the data has been processed by the neural network and the prediction error has been computed, we back propagate the error and compute error terms for each layer, which are then used to compute the gradients.

$$\partial^l = (W^{l+1})^T \partial^{l+1} \odot f'(W^l X^{l+1} + b^l)$$

needed for gradient descent parameter update⁷. These are the equation by which we can compute the error and update weights for each node:

$$\frac{\partial E}{\partial W^{(1)}} = X^{(l-1)} (\partial^1)^T$$

$$\Delta W^{(l)} = -\eta \frac{\delta E}{\delta E^{(l)}}$$

2.12.5.9 Overfitting: One of the issues that arises during neural network training is overfitting. Overfitting is defined as a model that perfectly models the training data. Instead of learning the overall distribution of the data, the model learns the expected output for each data point. On the training set, the error is driven to a very small value; however, the error is high when new data is introduced into the network.

The network has learned to remember the training examples, but it has not learned to generalize to new situations. As previously stated, overfitting is characterized by the model's inability to generalize. To test this ability, a simple method of dividing the dataset into two parts, the training set and the test set, is used. We may want to split the dataset when selecting models. With this split, we can examine the model's performance on each set to gain insight into how the training process is progressing and detect overfitting when it occurs.

2.12.5.10 Categorical Cross Entropy Loss: Categorical determine the efficiency is a loss function used in multi-class classification problems. These are tasks in which an example can only belong to one of many possible categories, and the model must determine which one it is. Its formal purpose is to quantify the difference between two probability distributions. Also known as Softmax Loss. It consists of a Softmax activation and a Cross-Entropy loss. We will train a CNN to output a probability over the C classes for each image if we use this loss. It is used to classify data into multiple categories. Cross entropy measures the difference between what the model predicts the output distribution should be and what the original distribution actually is. According to its definition, the cross-entropy measure is a popular alternative to squared error. It is used when node activations can be interpreted as representing the probability that each hypothesis is true when the output is a probability distribution. As a result, it is used as a loss function in neural networks with Softmax activations in the output layer.

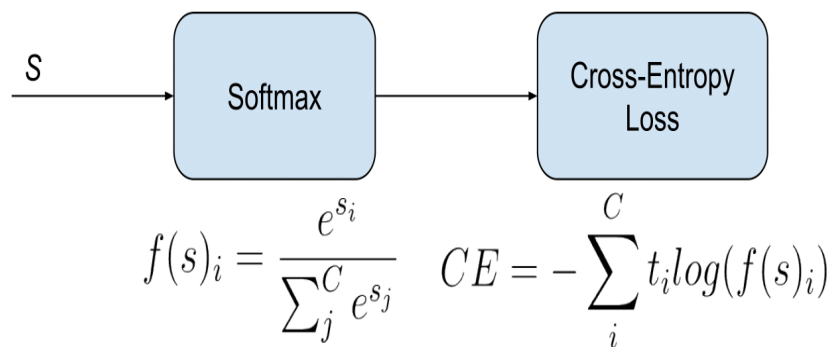


Figure 2.31: Categorical Cross Entropy Loss

Because the labels in the specific (and common) case of multi-Class classification are one-hot, only the positive class C_p retains its term in the loss. There is only one non-zero element in the Target vector t , $t_i = t_p$. So, after removing the summation elements that are zero due to target labels, we can write:

$$CE = -\log\left(\frac{e^{s_p}}{\sum_j^C e^{s_j}}\right) \text{ ,Where } S_p \text{ is the CNN score for the positive class.}$$

2.12.5.11 Performance Measure of the models: The confusion matrix is a useful technique for assessing the accuracy of binary and multi-class classification. We used accuracy, precision, recall, and the F1-score to evaluate performance. We chose the above- mentioned performance evaluation criteria because the simple confusion matrix could be misleading. However, the basic confusion matrix as well as the aforementioned performance were both evaluated.

	Predicted Class	
Actual Class	True Positive	False Negative
	False Positive	True Negative

Table 2.1 Confusion Matrix

True Positive (TP): True Positive is the number of truly classify as a positive.

False Positive (FP): False Positive is the number of falsely classify as a positive.

True Negative (TN): True Negative is the number of truly classify as a negative.

False Negative (FN): False Negative is the number of falsely classified as negative.

Accuracy: The simplest evaluation method is accuracy(A), which is simply the ratio of correctly predicted observations to all observations. The proportion of correctly classified predictions is calculated as follows:

$$A = \frac{TP+TN}{TP+TN+FP+FN}$$

Note that, True positive, true negative, false positive, and false negative are denoted by the symbols TP, TN, FP, and FN, respectively.

Precision: Precision is defined as the ratio of correctly predicted positive observations (True Positives) to all predicted positive observations, both correct (True Positives) and incorrect (False Positives), multiplied by 25. It is calculated in this manner.

$$P = \frac{TP}{TP+FP}$$

Recall: The proportion of relevant results correctly identified by data models is referred to as recall. Sensitivity is another term for recall. Recall is defined as the ratio of True Positives (TP) to the total of True Positives (TP) and False Negatives (FN), and it is calculated as follows:

$$R = \frac{TP}{TP+FN}$$

F1 score: The F1 Score is the weighted average of Precision and Recall. As a result, in order to establish a ratio between precision and recall, this score takes into account both False Positives and False Negatives, and it is calculated as follows:

$$F = 2 \frac{P \cdot R}{P+R}$$

Chapter 3

Research Methodology

3.1 Overall working process of District name recognition:

In this section, overall working process will be described in details. Figure 3.1 illustrates the process of BN-HW-DSNd dataset creation(left) and the overall system architecture of the word segmentation (middle) and steps of proposed model(right).

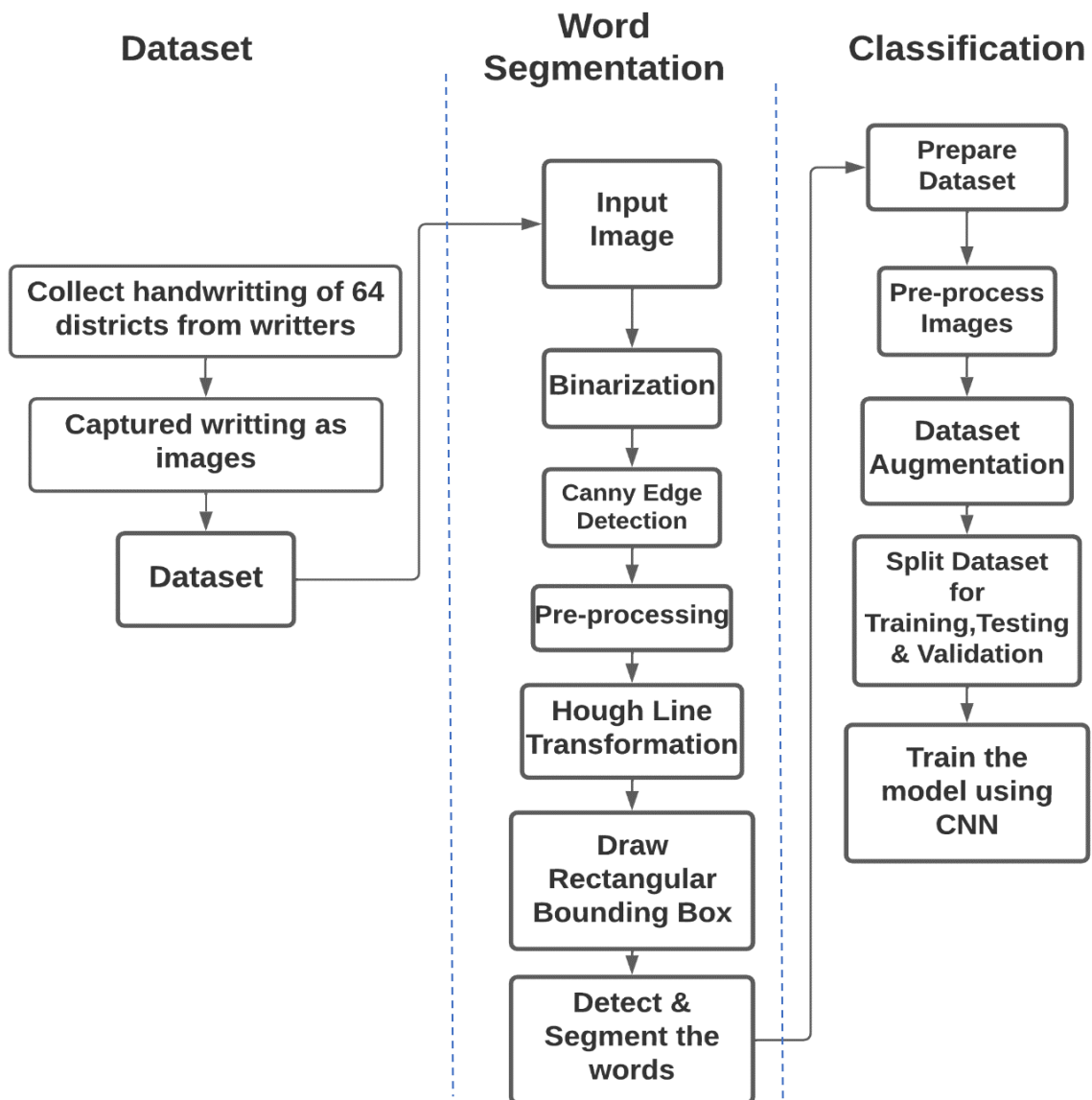


Figure 3.1: Overall working process of District name recognition

The proposed methodology for Bangla handwritten district name recognition is a three-step process including the followings:

- Dataset
- Word Segmentation
- Classification

3.1.1 Dataset: In the first step, we have created a new dataset for Bangla handwritten district name which is called BN-HW-DSNd dataset. To create the dataset, we have collected the district names in a text file as sources. Then we distributed the blank pages and the source file to people from various ages. The blank pages included color page and white page. Then we have collected the hard copies that contains handwritten district names of Bangladesh from the annotators. We have collected some postal envelope for our dataset. These collected pages are included in dataset which is ready to be segmented in next step.

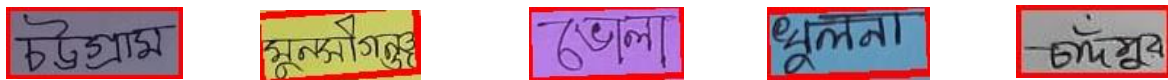


Figure 3.2: Dataset samples

3.1.2 Word Segmentation: In the second step, after the acquiring the handwritten images, we followed some preprocessing steps for the word segmentation of an image. To binarize the input image we have used a technique called OTSU's method. Then we have used the Canny Edge Detection technique to find edges. After that we have applied some morphological operation and noise removal techniques. Then we have used Hough line Transform to identify the straight line. To draw the bounding boxes, we have used Connected Component (CC) analysis for our next step. Finally, we segmented words by drawing a rectangle over each word through the entire width of the image. Then we have segmented those words manually which were not segmented properly. Following the above steps, we have created a new dataset which is used in our proposed CNN model.

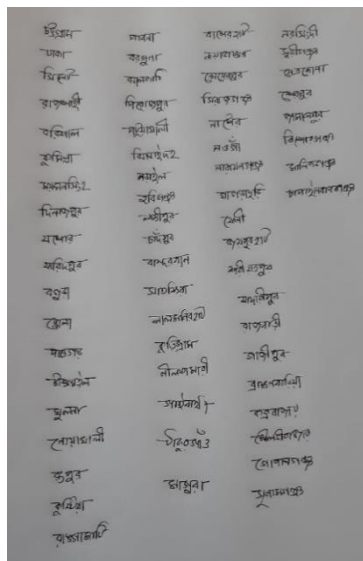


Figure 3.3: Input Image

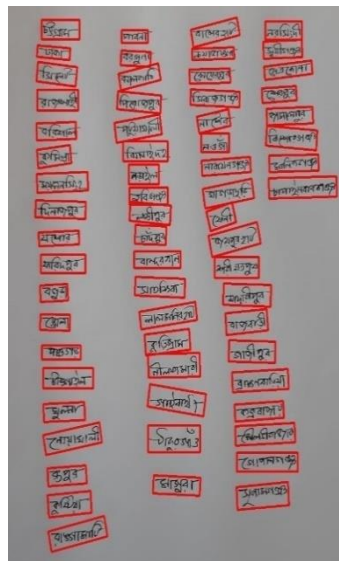


Figure 3.4: Detected Words

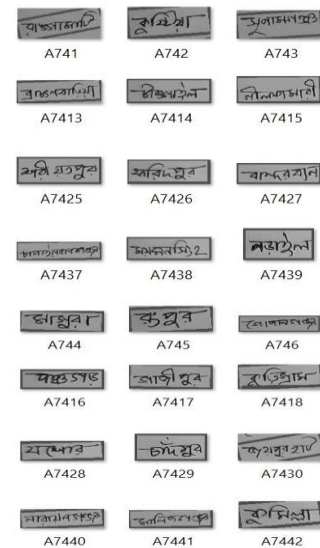


Figure 3.5: Segmented images

Dataset creation and word segmentation process will be described in next chapter Dataset Collection, Preprocessing and Segmentation.

3.1.3 Classification: In the third step, after acquiring the dataset we have resized all the images into a fixed size. The reason behind resizing was the difference of size in the images. Then, we have used image augmentation to expand the size of the dataset artificially. Augmentation includes verity of techniques but we have used shifting, zooming, rotation, brightness for the image augmentation process. Then, we have built a Convolutional Neural Network model that will be trained on 7040 images of district name. The RMSprop optimizer is used during the classification to decrease training time and reduce the loss. Then we have used dropout to overcome the overfitting problem. In hidden layers, Relu activation function is used. In output layer, Softmax activation function is used because of multiclass classification. Then, we have fit the data to our model. The test image variable contains the image to be tested on the CNN.

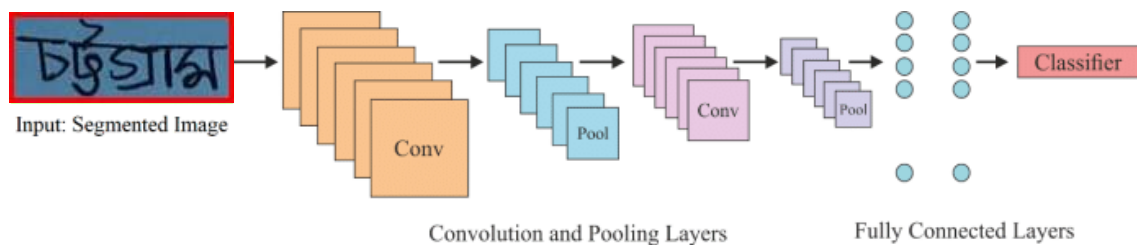


Figure 3.6: Classification process

Working procedure of Convolutional Neural Network model, we have used in this study will be described in Chapter 5 Proposed CNN Model

3.2 Environment setup:

3.2.1 Operating System:

- Windows 10 pro

3.2.2 IDE:

- PyCharm
- Google Collaboratory
- Jupyter notebook
- VS Code

3.2.3 Tools Requirement:

- | | |
|--------------|----------------|
| • Pytorch | • Scikit-learn |
| • Numpy | • Skimage |
| • Matplotlib | • Scipy |
| • Tensorflow | • io |
| • Pandas | • OS |
| • Keras | • OpenCV |

Chapter 4

Dataset Collection, Preprocessing and Segmentation

Appropriate datasets are required at all stages of object recognition and classification research from the training phase to the evaluation of recognition algorithm's performance. The dataset that has been used here are prepared by us. The dataset is available on Kaggle and can be downloaded from the following:

<https://www.kaggle.com/datasets/mdhosenzisad/bnhwdsnd>

4.1 BN-HW-DSNd: Data Annotation Process

Our data annotation approach will be described in details in this part. The process of BN-HW-DSNd dataset annotation followed by the writers and annotators is represented in Figure

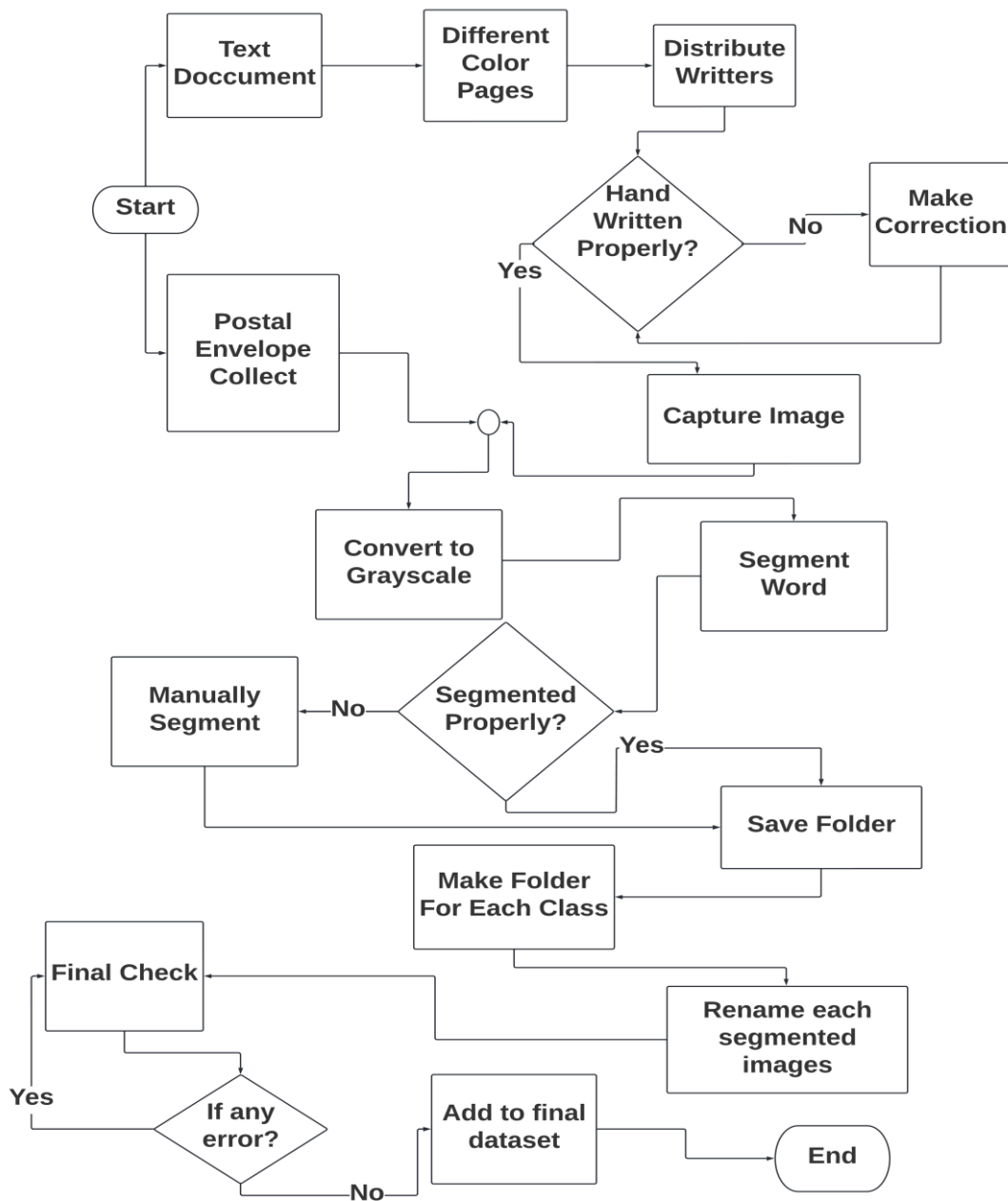


Figure 4.1: Flow Chart of the Annotation Process.

4.1.1 Data Source and Collection

Data annotation is the process of labeling information so that machines can use it. It is especially useful for supervised machine learning (ML), in which the system uses labeled datasets to process, understand, and learn from input patterns in order to get desired outputs. As a first step, we have created a text file which contains 64 district names of Bangladesh. Then we wrote the district names in blank pages. We have used color images and white images both.

4.1.2 Postal envelope collection: As a part of our dataset, we have collected postal envelope and captured image of these. From the envelope we have segmented the district name only. We have segmented the image manually. Postal envelope images were collected for testing purpose. The manual segmentation process will be discussed in later.

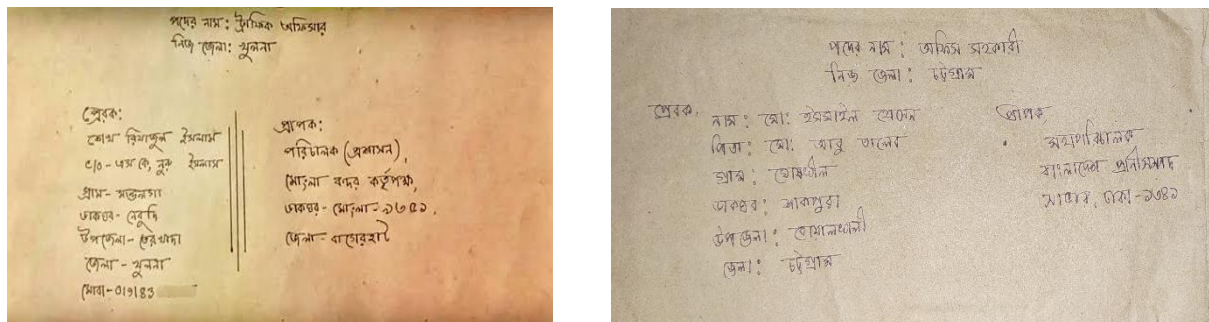


Figure 4.2: Sample image of postal envelope

4.1.3 Data Distribution: We have to distribute data among individuals for data annotation. So, we provided the blank pages to people of different ages and professions. To be specific, 60 of those data were prepared by students, and the rest of the dataset are prepared by teachers and officials. We also sent them a sample. In return, they wrote the content in page and gave us hard copies of the handwriting pages.

[illegible]

Figure

4.3: Sample page of a handwritten dataset.

Then, we captured images of the hardcopies that has been collected from different writers. Then we stored the images in a folder. Then we converted each image into grayscale. Then, we have used OCR to automatically crop the images. Then we stored the segmented images in a folder. Then we created 64 folders according to the district name. Then we sorted the segmented images according to each class where the folders were annotated manually. After that we have renamed all the images in each class automatically according to the respected class label.

4.2 Automatic Word Segmentation: We used an unsupervised approach to segment the words. The word segmentation mechanism is composed of the following steps:

Pre-processing:

1. RGB to Grayscale conversion
2. Thresholding
3. Edge Detection
4. Morphological Operation and Noise Removal.

Segmentation process:

5. Hough line detection
6. Connected Component Analysis(CCA)

4.2.1 Pre-processing: Pre-processing basically tends to simplify the input so that it can be processed by the system. Figure 4.2 illustrates the pre-processing steps that have been used.

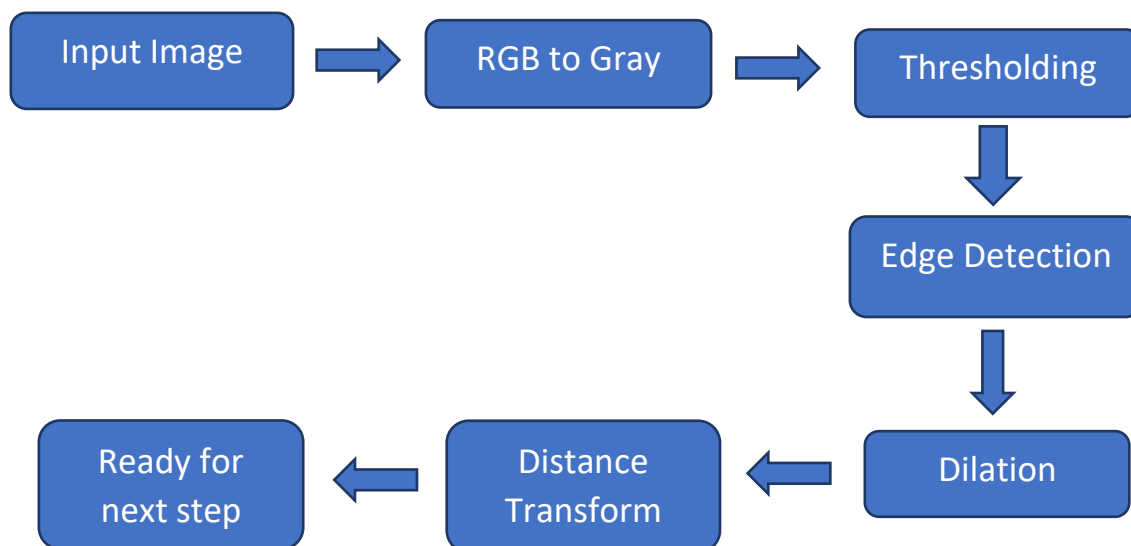


Figure 4.4: Pre-processing steps

4.2.1.1 RGB to Grayscale conversion: We read the input image as an RGB image which has 3 channels. These are Red, Green and Blue. There are a number of commonly used methods to convert an RGB image to a grayscale image such as average method and weighted method. The Average method takes the average value of R, G, and B as the grayscale value.

$$\text{Grayscale} = (R + G + B) / 3$$

The weighted method, also called luminosity method, weighs red, green and blue according to their wavelengths. The improved formula is as follows:

$$\text{Grayscale} = 0.299R + 0.587G + 0.114B$$

We used OpenCV to process images. OpenCV reads the image as BGR instead of RGB. Then, using the cvtColor function we can convert BGR image to grayscale image

```
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

4.2.1.2 Thresholding: Thresholding is a method of image segmentation, in general it is used to create binary images. Thresholding, also known as image binarization, is a non-linear method that converts a grayscale (or colored) image into a binary image with only two levels (0 or 1) for representing each pixel. If the pixel value exceeds the threshold, it is allocated one value (white), otherwise it is allocated another value (i.e., black). For this, we used a technique which is called local OTSU's method. The result of the binarization process over a segment of the original image is shown in Figure4.3.

Otsu's Technique: Thresholding is the process of separating the foreground (1) pixels from the background (0) pixels. There are a variety of approaches to solve optimal thresholding, one of which is known as Nobuyuki Otsu's method. The Otsu Method is a variance-based technique for determining the weighted variance threshold between foreground and background pixels. Iterating through all possible threshold values and measuring the spread of background and foreground pixels is the fundamental concept here. Then measuring the point at which the spread is the shortest.

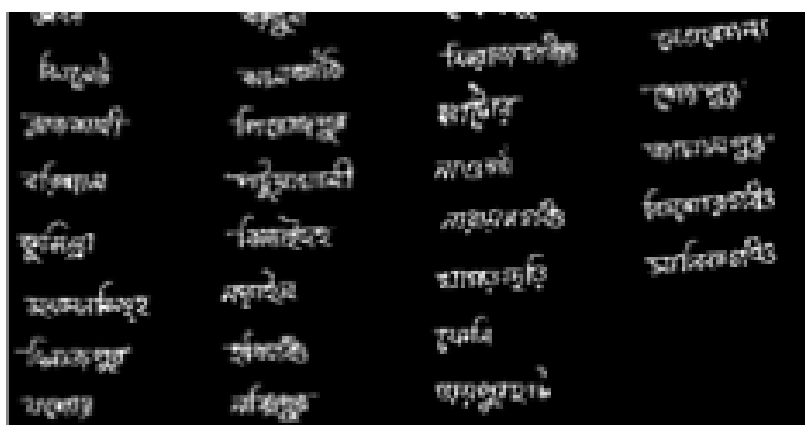


Figure 4.5: Thresholded Images

4.2.1.3 Edge detection: Edge detection is an image processing approach that identifies points in a digital image that have discontinuities, or variation in image brightness. The edges (or boundaries) of an image are the points where the image brightness varies sharply. Canny edge detection is a technique for extracting relevant structural information from various vision objects while reducing the amount of data to be processed considerably.

The following are some general criteria for edge detection:

1. Edge detection has a low error rate, which indicates that the detection should catch as many of the image's edges as possible.

2. The edge point recognized by the operator should be accurate in locating the edge's center.
3. Image noise should not cause incorrect edges, and a specific edge in the image should only be marked once.

4.2.1.4 Morphological Operation and Noise Removal: Dilation and Erosion are the most significant morphological operation. To remove noise from the image, we employed the 'Dilation' technique. The operation is classified as a 'Dilation' or an 'Erosion' depending on the rule used to process the pixels. Dilation combines or merges pixels to the boundaries of objects in an image, while erosion transfers or removes, or shifts pixels on object boundaries. We have used morphological opening followed by dilation to separate the sure foreground (Figure 4.5) noise from the background to remove the small salt and paper type noise.

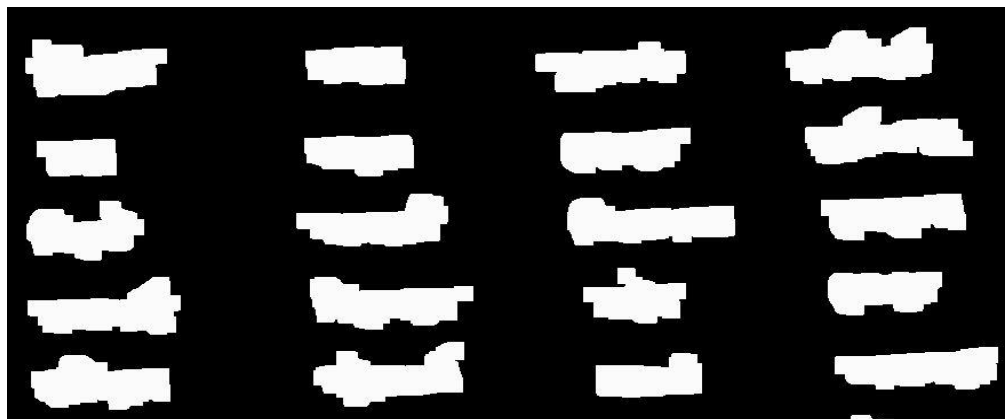


Figure 4.7: Noise removal

4.2.2 Segmentation process:

4.2.2.1 Hough line detection: The Hough Transform is a technique for isolating features or characteristics of a particular shape in an image. The Hough transform method's key advantage is that it understands gaps in feature boundary representations and is relatively simple by image noise. The simplest case of Hough Transform is detecting straight lines called Hough Line Transform. To identify straight lines, the Hough Line Transform is used. We used the Hough Line Transform to identify the horizontal continuous lines ('matra') over the words and dilated them to thicken them so that each word works as a connected component and individual words within a line may be identified. This will help us later on to draw the bounding box more accurately over the words.

4.2.2.2 Connected Component Analysis: Connected Component (CC) analysis is an algorithmic application of graph theory, where subsets of connected components are uniquely labeled. Connected component labeling is used to detect connected regions in binary digital images in computer vision. Once region boundaries have been discovered, it is often helpful to separate regions that are not separated by a boundary. Connected Component Analysis (CCA) is an undirected graph in which any two vertices are connected by paths, and which is connected to no additional vertices in the rest of the graph. A graph that is itself connected has exactly one component, consisting of the whole graph.

Bounding Box: We have used the Connected Component (CC) analysis to draw bounding boxes over each connected region.

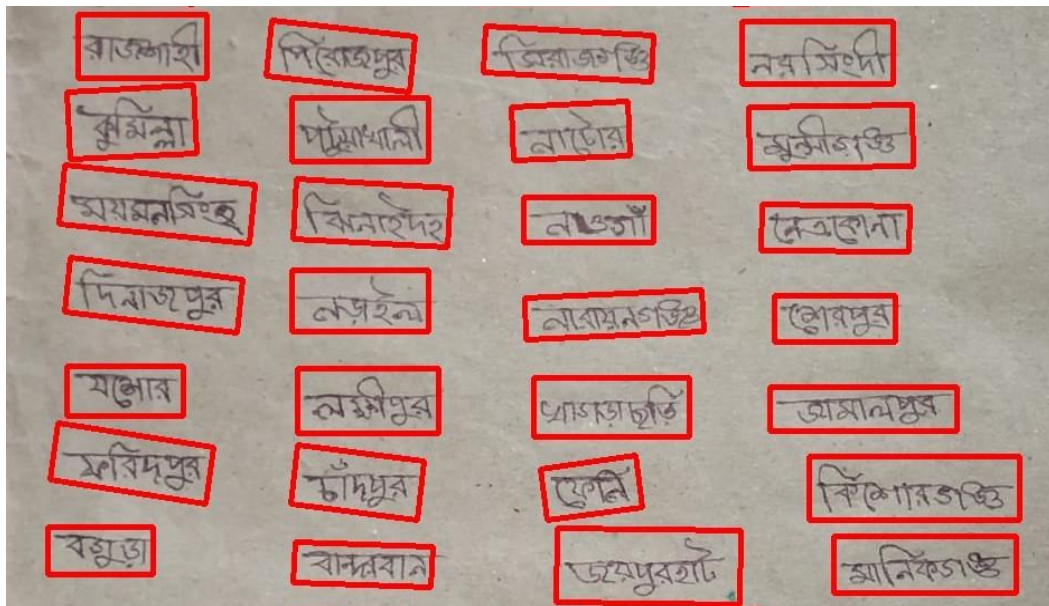


Figure 4.8: Bounding box over the words.

4.2.3 Word Segmentation and Cropping: In order to visualize the words, we have used a rectangle over each of them. We crop individual words taking the full length of the bounding box. We considered the ‘Bounding Boxes’ top and bottom points from the connected components to determine the height of the cropped words.

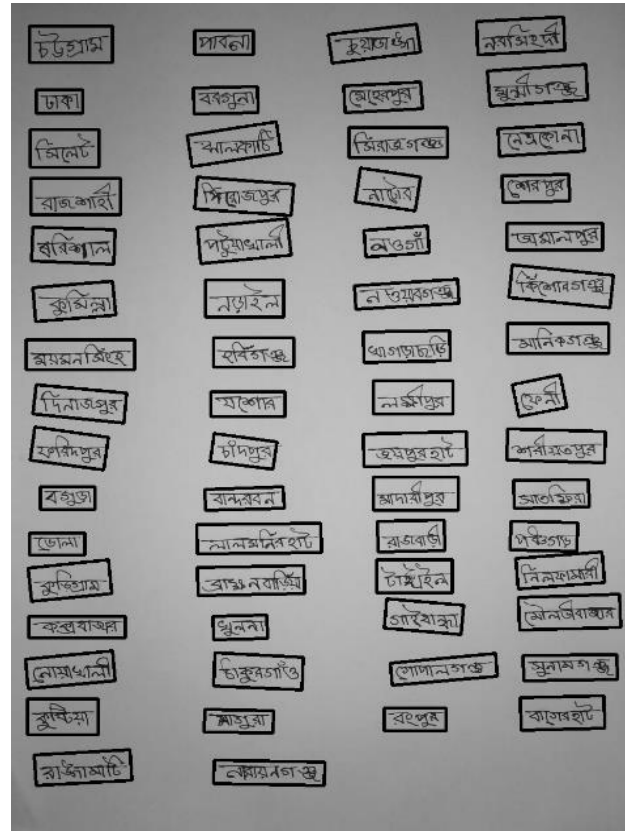


Figure 4.9: Gray Image



Figure: 4.10: Segmented image

4.3 Observation of Datasets:

Case 1: The good intensity images segmented almost all handwritten word properly and those were segmented automatically.

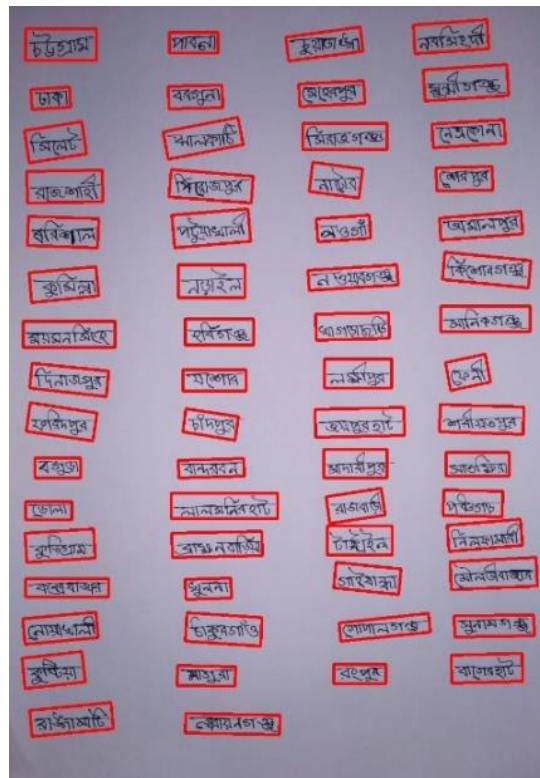


Figure 4.11: Best Case

Case 2: The images which was not segmented properly due to less resolution or intensity, we segmented them manually.

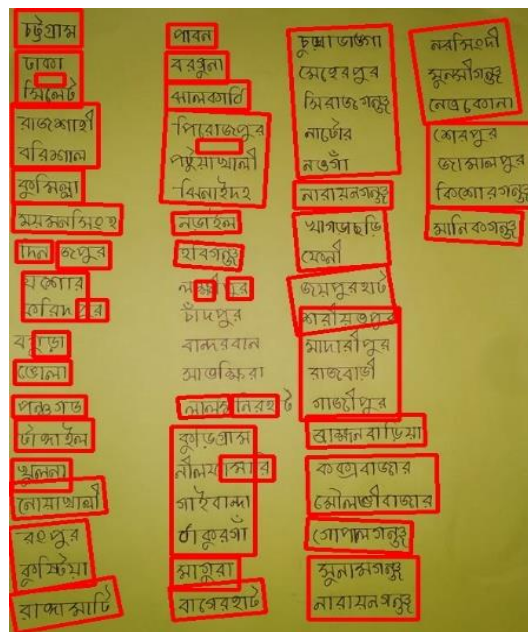


Figure 4.12: Worst Case

4.4 The manual segmentation process is given below:

4.4.1 Manual Segmentation: To begin annotating, we used a snipping tool to manually crop the words. It's a function which is also available on Windows. It allows us to save the whole or segments of an image, as well as quickly cut words or photos from the entire screen or specific areas of the PC screen. This tool can be used to make modifications or annotations, then save and share the results. Text annotations and free-hand artwork are made easier with the Snipping Tool. We followed the instructions below to use this tool:

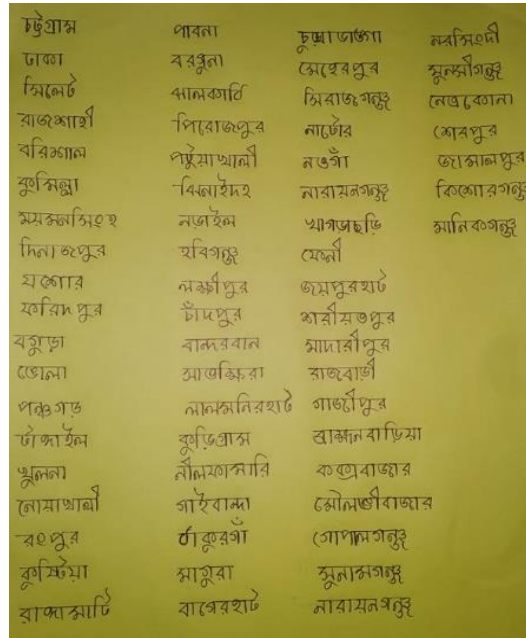


Figure 4.13: An example of a dataset image.

1. To use snipping tool, go to Start and type snipping tool, then click Enter.

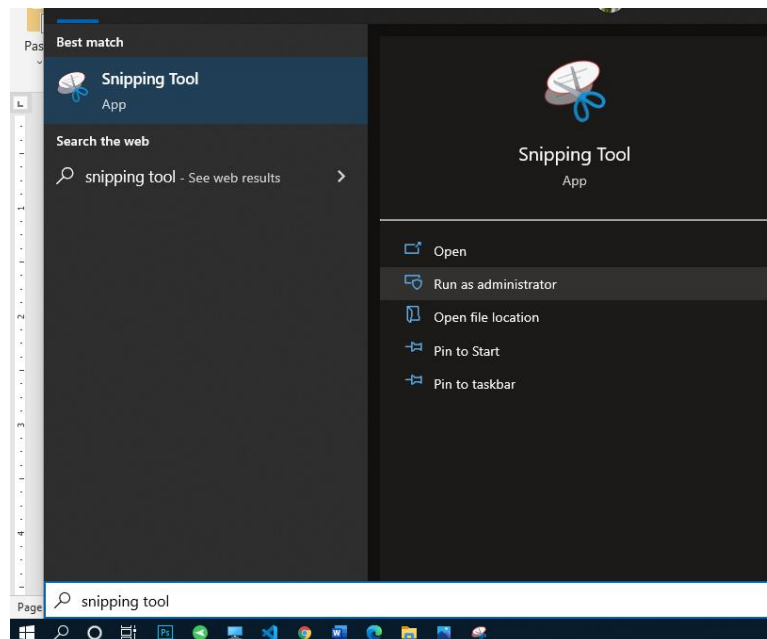


Figure 4.14: Opening of Snipping Tool from Start

2. Click on Mode and then select Free-form snip.

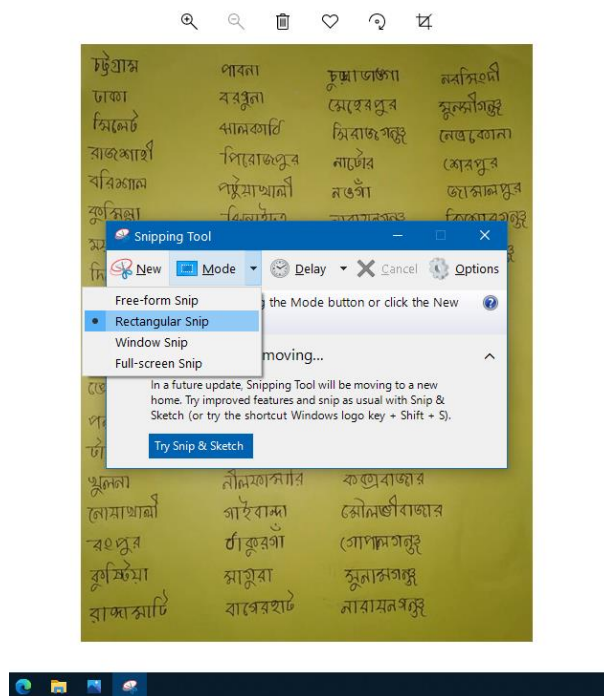


Figure 4.15: Select Rectangular-form snip from Mode

1. To begin a new snipping action, select New.

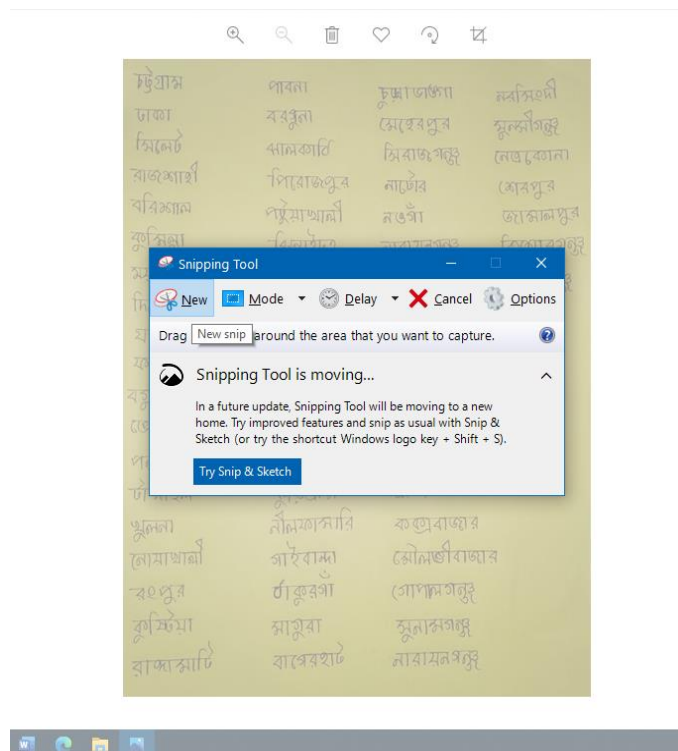


Figure 4.16: Selecting New

2. After selecting New we can crop any portion of image simply by dragging.

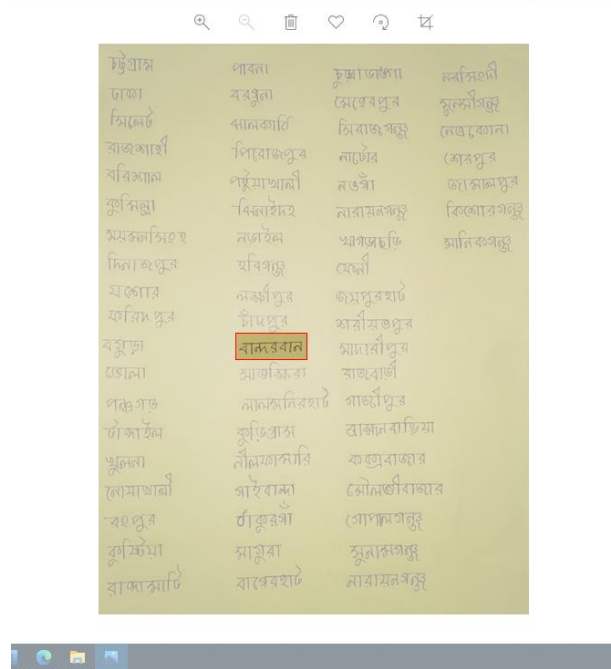


Figure 4.17: Dragging an specific portion of image.

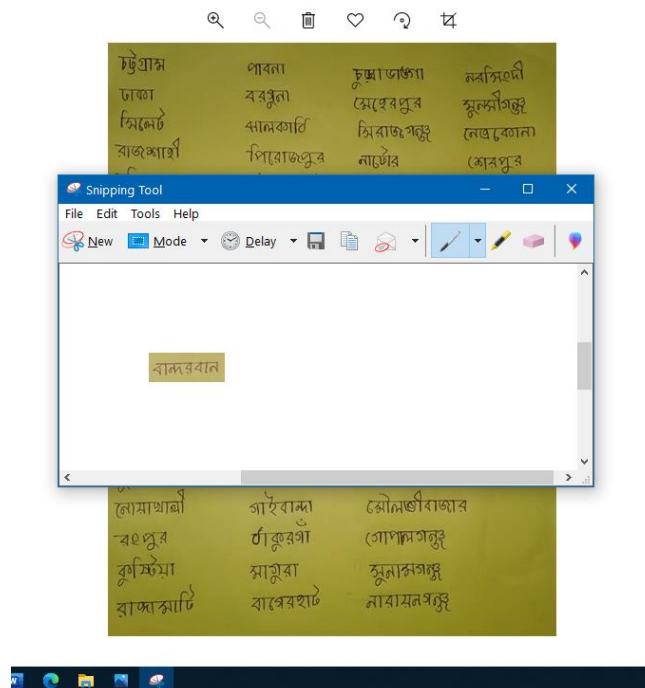


Figure 4.18: Word Cropped

1. Then save the segmented word .

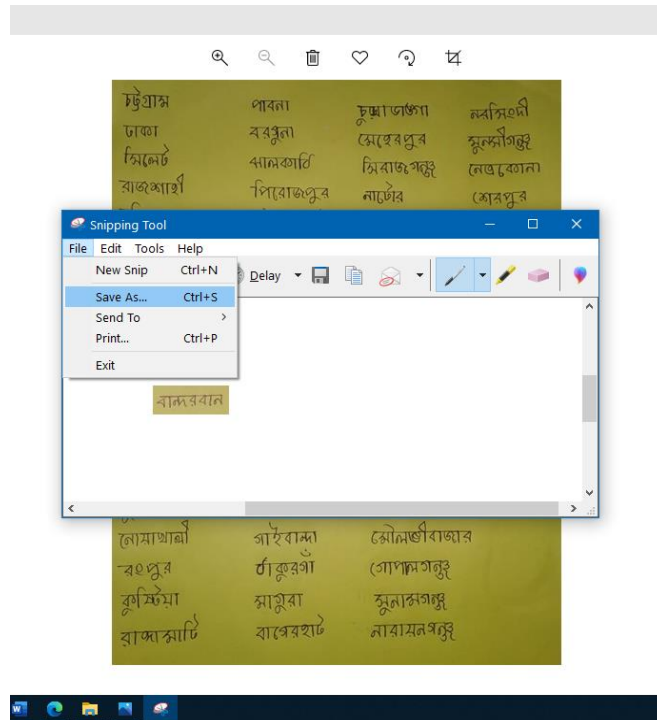


Figure 4.19 Saving the segmented word

4.5 Structure of our dataset:

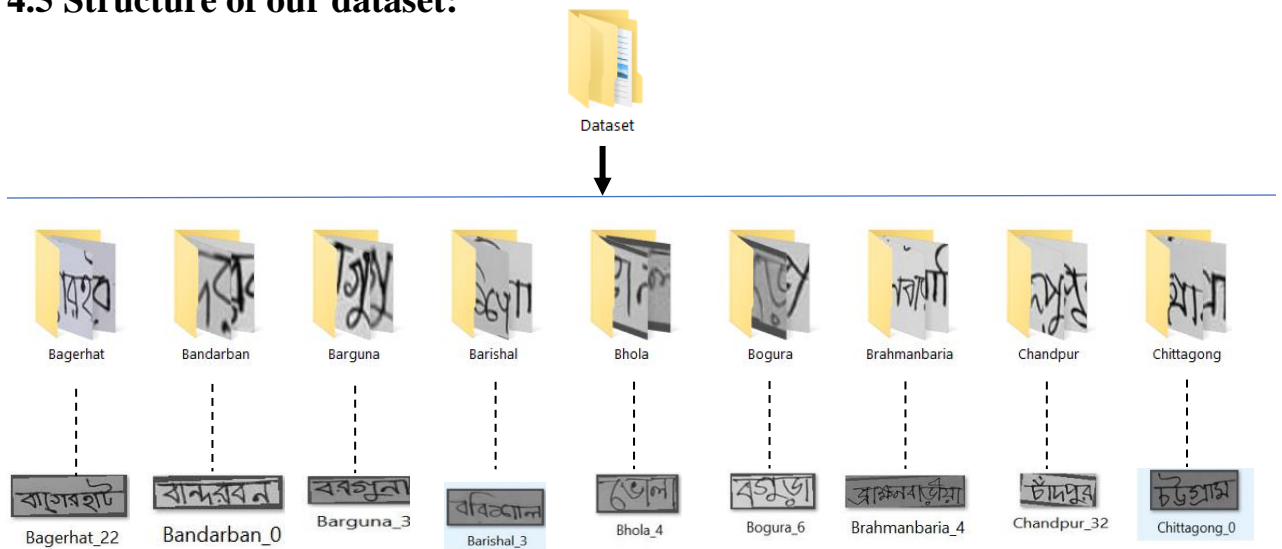


Figure 4.20: Structure of the Dataset

Chapter 5

Proposed CNN Model

5.1 Working Procedure of proposed model:

We have used Convolutional Neural Network (CNN), which is a Deep Learning algorithm able to take in an input image, assigning importance (learnable weights and biases) to various aspects/objects in the image, and distinguishing one from the other. Here we have built a multi-layer perceptron. This is also known as a feed-forward neural network. We have an input layer into which we feed our features. These perceptron functions then compute an initial set of weights while passing control to any number of hidden layers. The number of times this happens is determined by the parameters we have passed to the algorithms, the method we have taken for the loss and activation function. This is determined by the parameters that has been passed to the algorithms we choose for the loss and activation function. The network is controlled by the number of nodes we have allowed to use. The full solution is revealed later in the output. There is only one input and one output layer.

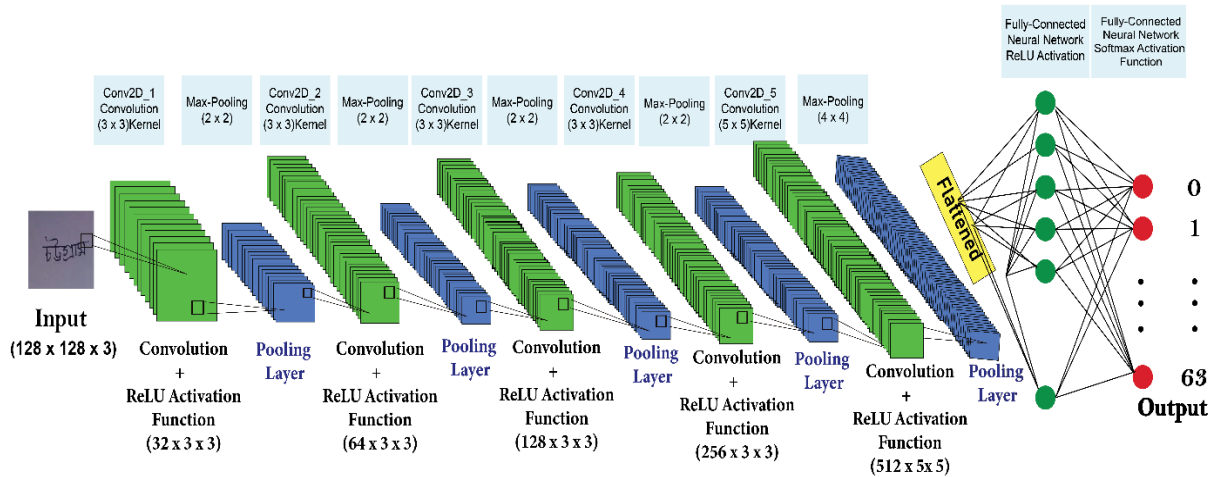


Figure 5.1: Architecture of proposed CNN model

5.2 Image Preprocessing: Our dataset contains images that are in various formats along with different resolutions and quality. In order to gain consistency and to get better feature extraction, final images desired to be used as dataset for deep neural network classifier are preprocessed. It is assured that images contain all the needed information for feature learning. All the images were resized to 128 by 128 pixels before feeding the images into the model. By resizing the images to smaller pixels, we reduced the training time and complexity. After this step the images and datasets were ready to be used in the next step.

5.3 Dataset Augmentation: Image augmentation is a technique for enhancing original images by employing various transformations including zooming, flipping, translating and simple image rotations to them, resulting in a large number of altered copies of the same image. The main goal of using augmentation is to expand the dataset and introduce minor distortion to the images, which helps to reduce overfitting during the training stage. Data augmentation techniques are often used along with traditional machine learning algorithms or deep learning algorithms to enhance the accuracy of classification. In this system, the image

augmentation method was employed by using the Keras deep learning library in Python. Keras ImageDataGenerator class provides a fast and simple augmenting image.

Argument	Parameters
Rescale	1./255
Shear_Range	0.3 degree
Zoom_Range	0.2
Horizontal_Flip	False
Brightness_Range	[0.6,1.0]
Rotation_Range	30

Table 5.1: Applied augmentation process

For the augmentation process, simple image rotations were applied, as well as rotations on the different axis by various degrees, zooming and brightness range were applied. Transformations applied in augmentation process are illustrated in Figure 5.2, where the first image is the original image and rest of the images are obtained by applying shifting, zooming and rotation.

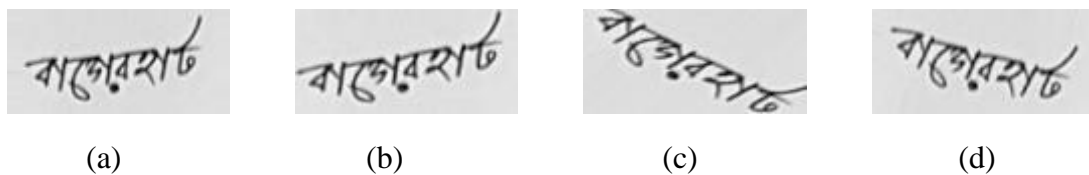


Figure 5.2: Image Augmentation: (a) original image (b) shear image (c) Zoomed (d) Rotation

5.4 Architecture of proposed CNN Model:

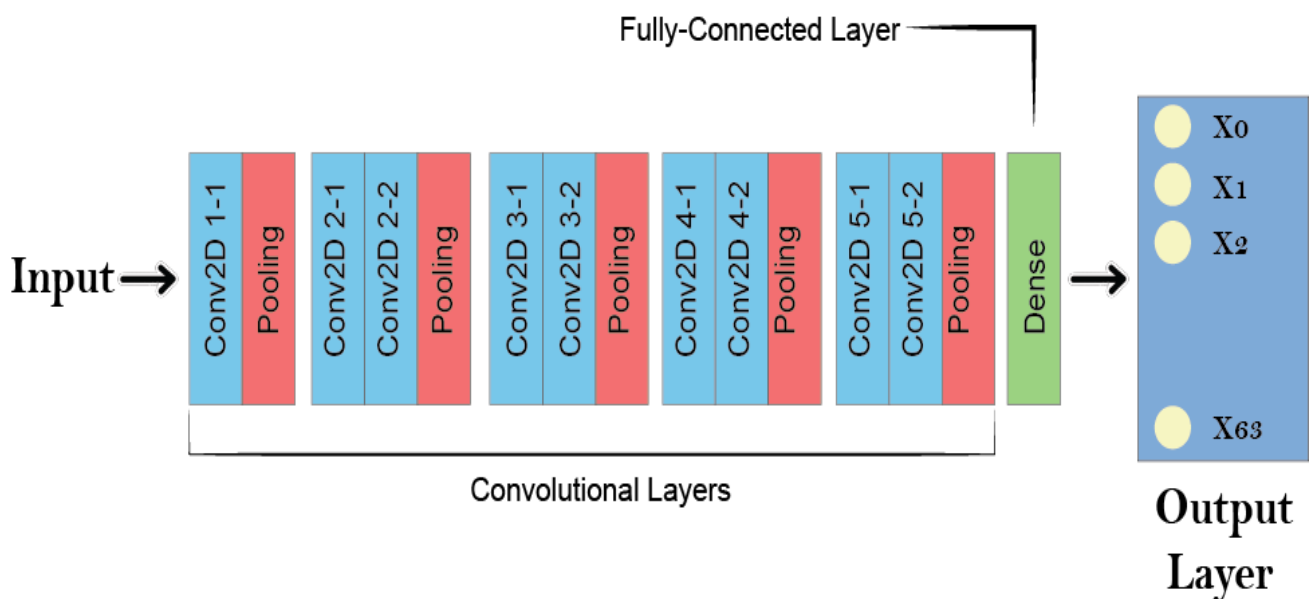


Figure 5.3: Layers used to build proposed CNN model

The process of building a Convolutional Neural Network always involves four major steps.

Step - 1: Convolution

Step - 2: Pooling

Step - 3: Flattening

Step - 4: Full connection

In addition to these four layers, there are two more important parameters which are noted below:

- the dropout layer and
- the activation function

5.5 Importing all of the keras packages:

Let's start by importing all of the keras packages the model have needed to create our CNN. Ensure that each package is correctly installed in the system.

Let's have a look at what each of the following packages is used for:

- To make our neural network model a sequential network, we have imported Sequential from keras.models. A neural network can be started in one of two ways: as a sequence of layers or as a graph.
- We've imported Conv2D from keras.layers to conduct the convolution operation on the training pictures, which is the first step of a CNN. We're using Convolution 2-D since we're working with images, which are essentially two-dimensional arrays.
- The next step in the process of establishing a CNN is to import MaxPooling2D from keras.layers, which is utilized for pooling operations. We have implemented a Maxpooling function to construct this neural network; however, many types of pooling procedures exist, such as Min Pooling, Mean Pooling, and so on. We have required the maximum value pixel from the corresponding region of interest in MaxPooling.
- Flatten, which is used for Flattening, has been imported from keras.layers. Flattening is the process of transforming all of the 2D arrays into a single long continuous linear vector.
- Finally, we've imported Dense from keras.layers, which is used to complete the whole connection of the neural network, which is the fourth stage in the CNN construction process. That is called fully connected layers.

5.6 Model creation: At first, we have created an object of sequential model. Then we followed the next process as below.

Layer (type)	Output Shape	Parameters
conv2d (Conv2D)	(None, 128, 128, 32)	896
conv2d_1 (Conv2D)	(None, 128, 128, 32)	9248
conv2d_2 (Conv2D)	(None, 64, 64, 64)	18496
conv2d_3 (Conv2D)	(None, 64, 64, 64)	36928
conv2d_4 (Conv2D)	(None, 32, 32, 128)	73856
conv2d_5 (Conv2D)	(None, 32, 32, 128)	147584
conv2d_6 (Conv2D)	(None, 16, 16, 256)	295168
conv2d_7 (Conv2D)	(None, 16, 16, 256)	590080
conv2d_8 (Conv2D)	(None, 8, 8, 512)	3277312
conv2d_9 (Conv2D)	(None, 8, 8, 512)	6554112
dense (Dense)	(None, 1024)	2098176
dense_1 (Dense)	(None, 64)	65600
Total params:		13,167,456
Trainable params:		13,167,456
Non-trainable params:		0

Table 5.2: CNN Model architecture

5.6.1 Input Layer: First of all, the following code generated a Keras sequential model. This entails gradually adding layers to the neural network, one at a time. Then, using the "Conv2D" function, we constructed a convolution layer. The Conv2D function takes four arguments:

- The first parameter is the number of filters, which in this case is 32,
- The second is the form each filter will take, which in this case is 3x3,
- The third is the input shape. INPUT [128 x 128 x 3] will save the image's raw pixel values, in this model an image with a width of 128 pixels, a height of 128 pixels, and three-color channels (R, G, B).

- The fourth input specifies the activation function to be used; in this case, 'ReLU' denotes a rectifier function.

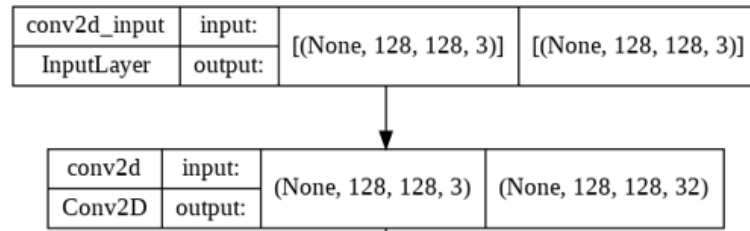


Figure 5.4: Input layer

5.6.2 Convolution layers: In our model, we used five convolution layers and numerous filters to generate the output feature maps. The following expression is used to compute the output feature maps:

$$Z_n^{[m]} = \sum_{k=1}^{n_{m-1}} X_{n,k}^{[m]} p_k^{[m-1]} + b_n^{[m]} \dots\dots\dots (1)$$

1st Convolution: Initially input shape is 128x128. In this layer we have used 32 filters where the kernel size is 3x3. As an activation function, we have used 'ReLU' because relu is used to achieve a non-linear transformation of the data whereas sigmoid functions "squash" their inputs resulting in the vanishing gradient problem. To reduce the spatial dimensions of the output volume in the Max Pooling layer we have used 2x2 matrix so that will have minimum pixel loss and get a precise region where the feature is located. After maxpooling the new input shape is 64x64. We have used Dropout rate 0.2 which randomly sets input units to 0 with a frequency of rate at each step during training time in order to reduce Overfitting.

2nd Convolution: From 1st convolution we got input shape of 64x64 as input of this layer. In this layer we have used two convolution layers having 64 filters where the kernel size is 3x3. We have used relu activation function in this layer. Then after using the maxpooling now the input shape is 32x32. This output will be the input of 3rd convolution layer. To reduce overfitting, we have used dropout rate as 0.2.

3rd Convolution: From 2nd convolution we got input shape of 32x32 as input of this layer. In this layer we have used two convolution layers having 128 filters where the kernel size is 3x3. We have used relu activation function in this layer. Then after using the maxpooling now the input shape is 16x16 where is size of pool is (2,2). This output will be the input of 4th convolution layer. To reduce overfitting, we have used dropout rate as 0.2.

4th Convolution: From 3rd convolution we got input shape of 16x16 as input of this layer. In this layer we have used two convolution layers having 256 filters where the kernel size is 3x3. We have used relu activation function in this layer. Then after using the maxpooling now the input shape is 8x8 where is size of pool is (2,2). This output will be the input of 5th convolution layer. To reduce overfitting, we have used dropout rate as 0.2.

5th Convolution: From 4th convolution we got input shape of 8x8 as input of this layer. In this layer we have used two convolution layers having 512 filters where the kernel size is 5x5. We have used relu activation function in this layer. Then after using the maxpooling

now the input shape is 8x8 where is size of pool is (4,4). This output will be the input of the fully connected layer. To reduce overfitting, we have used dropout rate as 0.2.

5.6.3 Fully Connected layer: From 5th convolution we got input shape of 2x2 as input of this layer. Then flatten is used to convert all the pooled images into a one-dimensional single vector. Then we have used 1024 nodes that should be present in this hidden layer. And the activation function will be a rectifier function. To reduce overfitting, we have used dropout rate as 0.2. This output will be the input of the output layer. In output layer the size of the neuron is used as 64 as our system has 64 classes. And in this model, we have used Softmax as an activation function. The Softmax function is based on the equation below where X is the total number of classes:

$$\sigma(\vec{z})_m = \frac{e^{z_m}}{\sum_{n=1}^{X=64} e^{z_n}} \dots\dots\dots (2)$$

We have preferred Softmax activation over sigmoid because Softmax activation evenly distributes the probability across each output node (class). Our system is classifying a multiclass classification and for multi-class classification, Softmax performs well compared to sigmoid activation function.

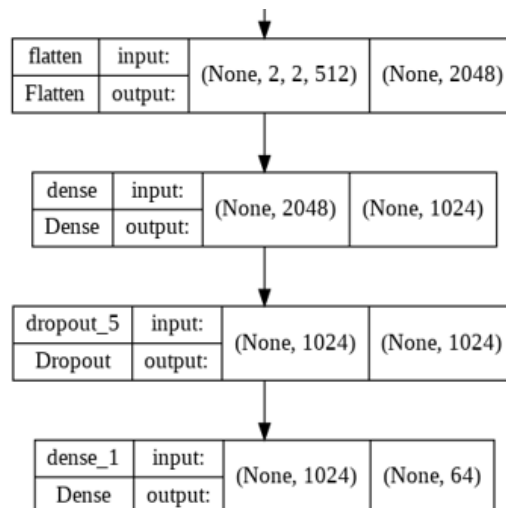


Figure 5.5: Output Layer

5.7 Model Compile: After we've finished building our CNN model, the next step is to compile the model. Model compile takes the followings as input:

Optimizer: This parameter is to choose the stochastic gradient descent algorithm. We have used RMSprop optimizer for this system. RMSprop uses an adaptive learning rate instead of treating the learning rate as a hyperparameter. This means that the learning rate changes over time.

Loss: This parameter is to choose the loss function. We have used categorical_crossentropy as a loss function for multi-class classification model where there are two or more output labels. The output label is assigned one-hot category encoding value in form of 0s and 1.

Metrics: Finally, the metrics parameter is to choose the performance metric. Metrics calculates how often prediction equal labels. This metric creates two local variables, total and count that are used to compute the frequency with which y_pred matches y_true.

5.8 Fitting the model: After compiling the model the next step is to fit the data to the model. In our model, steps per epoch holds the number of training images divided by the no of batch size that is used here. A single epoch is a single step in training a neural network. In our system we have used 50 epochs to train the model. Validation data is the data on which to evaluate the loss and any model metrics at the end of each epoch. In this system, we have used test data as validation data. Validation steps is the total number of steps (batches of samples) to draw before stopping when performing validation at the end of every epoch. In our system, validation steps hold the number of test images divided by the no of batch size. Callback has three parameters such as earlystop, ModelCheckpoint, PlotLossesKeras. Earlystop stops training when the loss is at its min, i.e., the loss stops decreasing. ModelCheckpoint callback is used in conjunction with training using model. Fit () to save a model or weights (in a checkpoint file) at some interval, so the model or weights can be loaded later to continue the training from the state saved. PlotLossesKeras is used for tracking the live training loss. Verbose is set to True to get real time visualization of the model while training.

Chapter 6

Experiments & Result

This section contains the results of the experiments carried out in this study. For image classification, CNN is widely used. In this study, the performance of different CNN architectures was evaluated for the classification. In this experiment we have used several neural network architectures along with a pretrained architecture. In this system, we have used CNN model, Classifier, AlexNet architecture. As a pretrained model we have used InceptionV3 architecture. As a result of testing the model, we have achieved 98% accuracy in CNN model, 91% accuracy in classifier, 93% accuracy in AlexNet and 80% accuracy in InceptionV3.

6.1 Training accuracy vs Validation accuracy:

The dataset we have created contains 7040 images. For the system training and evaluation, we have split the dataset into 80% train set and 20% test set. But overfitting is one of the most important things to avoid when training a machine learning model. This occurs when the model fits the training data well but is unable to generalize and make accurate predictions for new data. A technique known as cross-validation is used to determine whether a model is overfitting. In cross validation the data is divided into two parts: the training set and testing set. The training set is used to train the model, whereas the validation set is only used to evaluate its performance. The goal is to classify the images into 64 categories.

Our CNN model has an accuracy of 97% on the training set and 97% on the validation set. This means that you can expect your model to perform with 97% accuracy on new data.

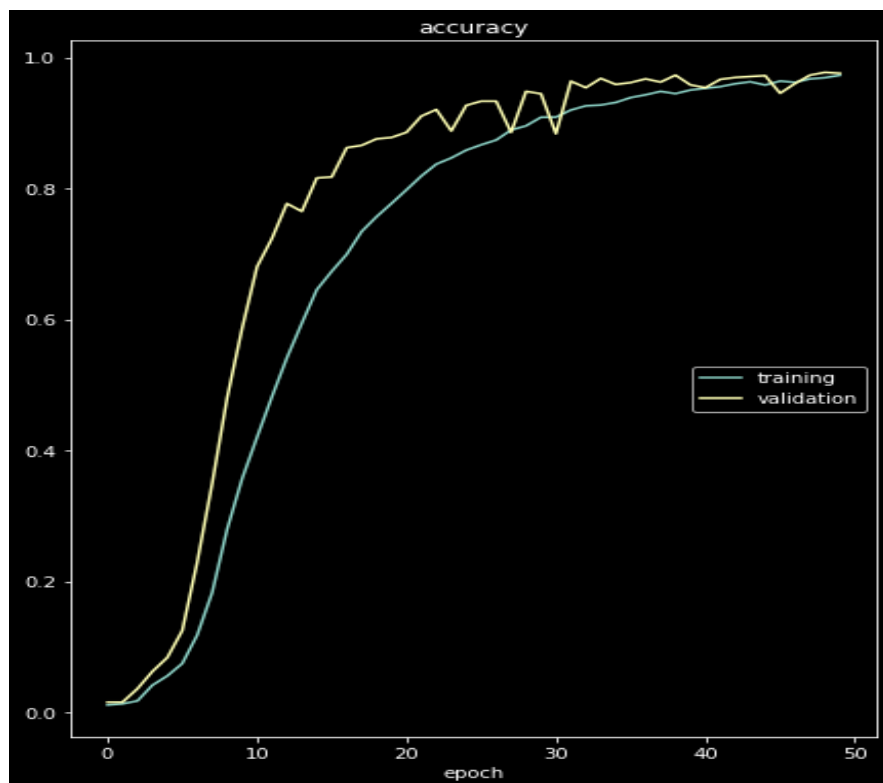
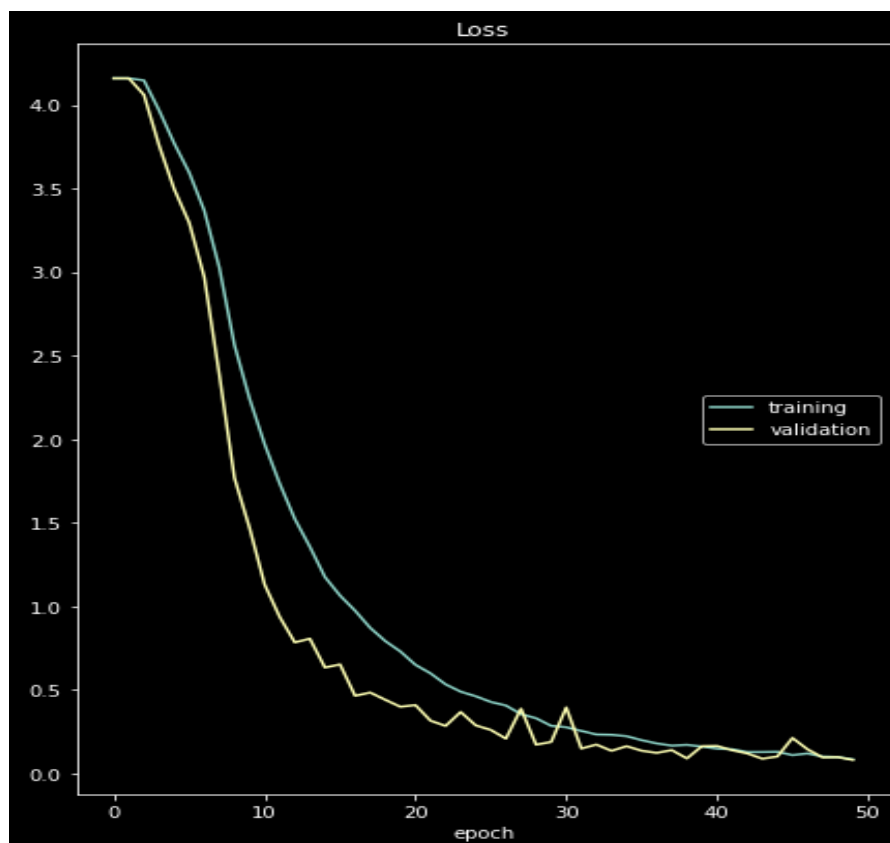


Figure 6.1: Evaluation of accuracy with the number of epochs in our model

We notice that in between 2 to 20 epochs validation accuracy increases compared to training accuracy. This could be case of diverse probability values in cases where softmax is being used in output layer. This has happened because we have used random selection of images in our validation set. So, there's a possibility that in our training set there are some images that are noisy but in validation set there's no such low intensity image. Besides the network was struggling to fit to the training data. From 26 to 30 epochs, validation accuracy is decreasing and training accuracy is increasing. It means our model is fitting the training set better, but losing its ability to predict on new data, indicating that our model is starting to fit on noise and is beginning to some overfit. We have added dropout as regularization to prevent the overfitting problem. Finally, in 50th epochs validation accuracy became equal to training accuracy and thus we achieved 97% accuracy on training accuracy and validation accuracy both.

6.2 Training Loss vs Validation Loss:



Figure

Evaluation of loss with the number of epochs in our model

6.2:

We notice that our model has 8% of training loss and 8% of validation loss. In between 1 to 20 epochs training loss and validation loss both are decreasing but the training loss is greater than validation loss. Because the training loss is calculated over the entire training dataset and the validation loss is calculated over the entire validation dataset. The training set is typically 4 times larger than the validation set. As the training continues, the training loss and validation loss are approaching one another. In 26th and 45th epochs validation loss increases and intersects training loss and the validation loss becomes greater than training loss which we can call it some overfitting. To reduce this overfitting, we have used data augmentation

and regularization techniques. Thus, we achieved 8% loss in both training and validation. As both values ended up to be same so it means the model built is learning and working fine.

6.3 Comparison among classifier, AlexNet, InceptionV3 and our CNN model:

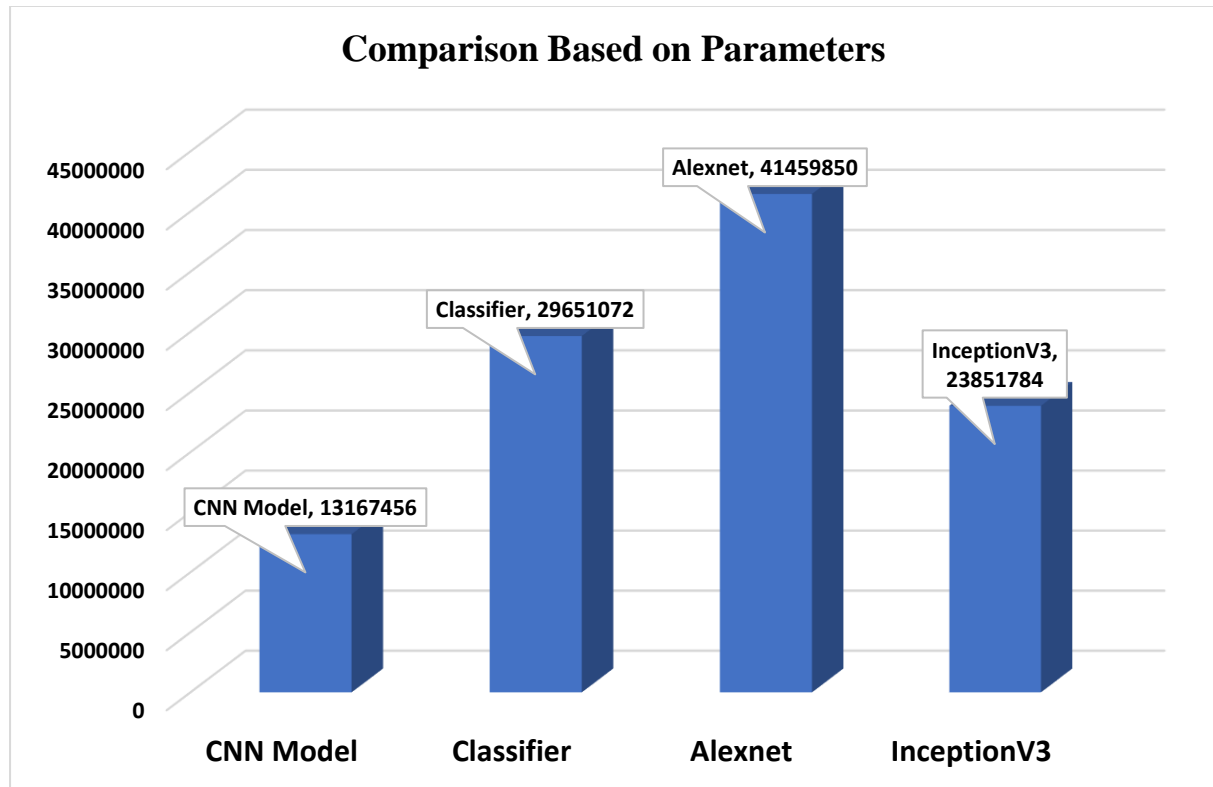


Figure 6.3: Comparison of models based on parameters

We have compared our customized CNN model with the classifier, AlexNet architecture and InceptionV3 model. Our customized model gets an accuracy of 98% while classifier, AlexNet, InceptionV3 gets an accuracy of 91%, 93%, 80% respectively using the dataset we have created.

6.3.1 Classifier: We have achieved 91% training accuracy and 90% validation accuracy using classifier. In classifier 29651072 parameters are trained while in our customized model 13167456 parameters are trained. We have used 70 epochs to evaluate the model. As epochs goes from 5 to 12, training accuracy decreases, while validation accuracy increases. After that the training accuracy improves over time but the validation accuracy remains constant, which means model is fitting the training set better but losing its ability to predict new data and this is called overfitting. But after completing 70 epochs the overfitting problem cannot be solved using regularization too.

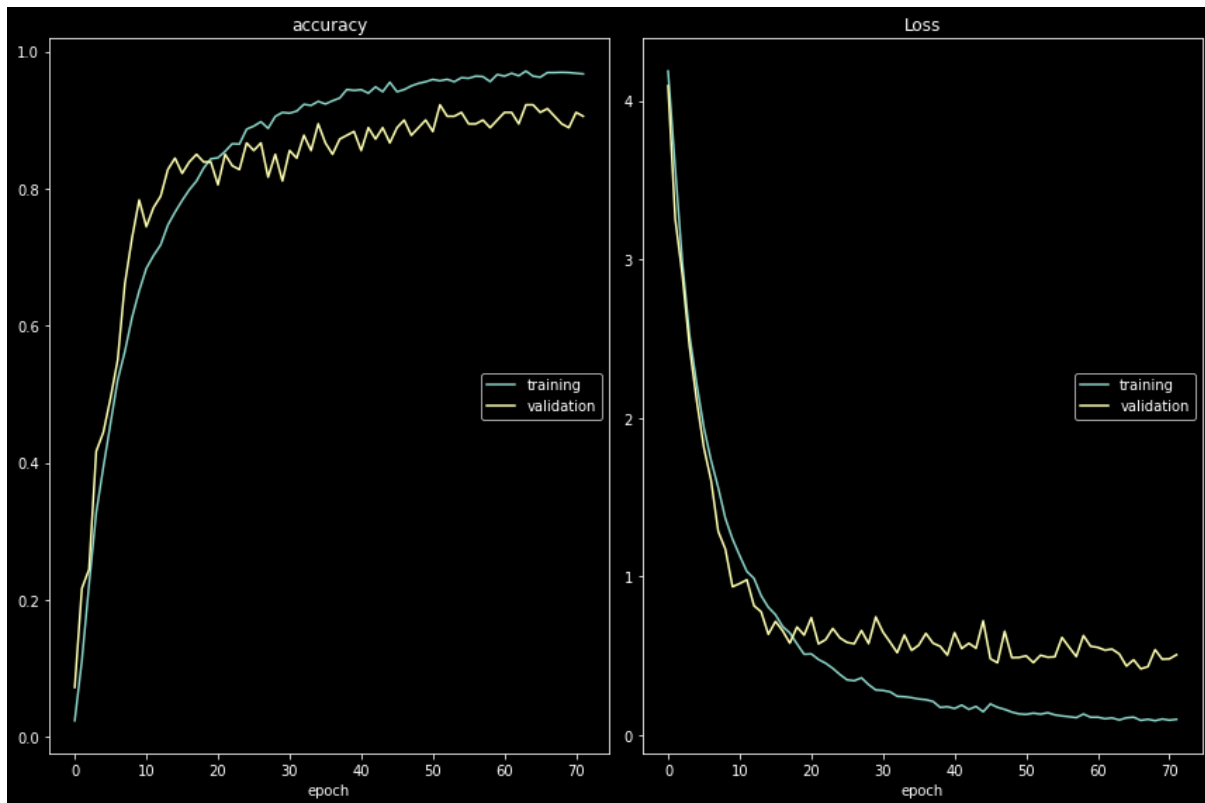


Figure 6.4: Evaluation of accuracy and loss with the number of epochs in classifier

From 0 to 15 epochs validation loss improved but after 15 epochs training loss decreases but validation loss remains same. Finally, we got minimum training loss of 10% and validation loss of 50%.

6.3.2 AlexNet: We have achieved 93% training accuracy and 93% validation accuracy using AlexNet. In AlexNet 41,459,850 parameters are trained while in our customized model 13,167,456 parameters are trained. We have used 35 epochs to evaluate the model. After 5th epochs training accuracy and validation accuracy both increased but there is fluctuation in validation accuracy which occurred for overfitting problem. It means that some portion of our examples is classified randomly, which produces fluctuations, as the number of correct random guesses always fluctuate. Because the validation loss is rapidly increasing so the validation accuracy is fluctuating. It means the model is fitting very nicely on the training data but failed to generate correctly to unseen data. To prevent this regularization technique is used and with the increase of epochs, validation accuracy and validation loss both improved, yet the fluctuations can't be reduced. After completing 35 epochs we got 14% training loss and 26% validation loss.

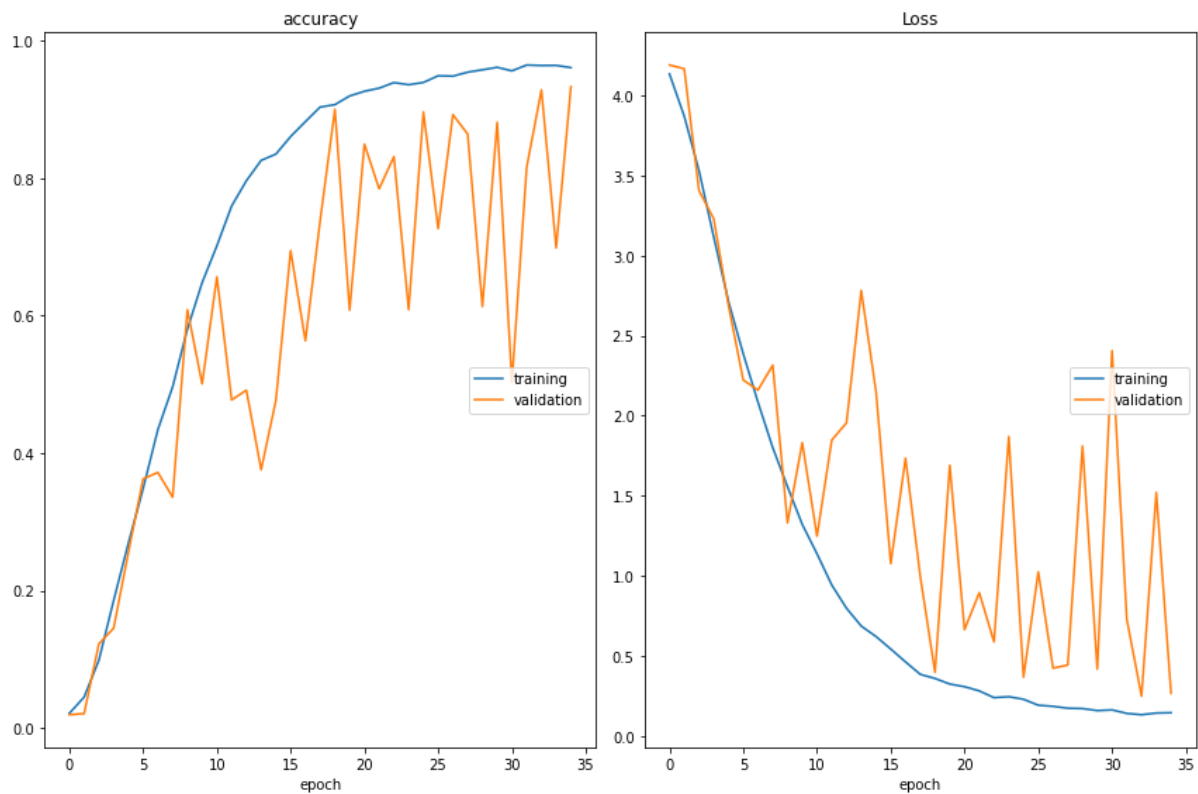


Figure 6.5: Evaluation of accuracy and loss with the number of epochs in AlexNet

6.3.3 InceptionV3: We have achieved 80% training accuracy and 92% validation accuracy using a pretrained model called InceptionV3. In InceptionV3 23,851,784 parameters are trained while in our customized model 13,167,456 parameters are trained. We have used 86 epochs to evaluate the model. After 16th epochs validation accuracy gets higher than training accuracy. This indicates the presence of high bias in dataset. It is underfitting. A higher use of dropout can cause this problem. At the end of 86th epochs validation loss is 32% and training loss is 65%.

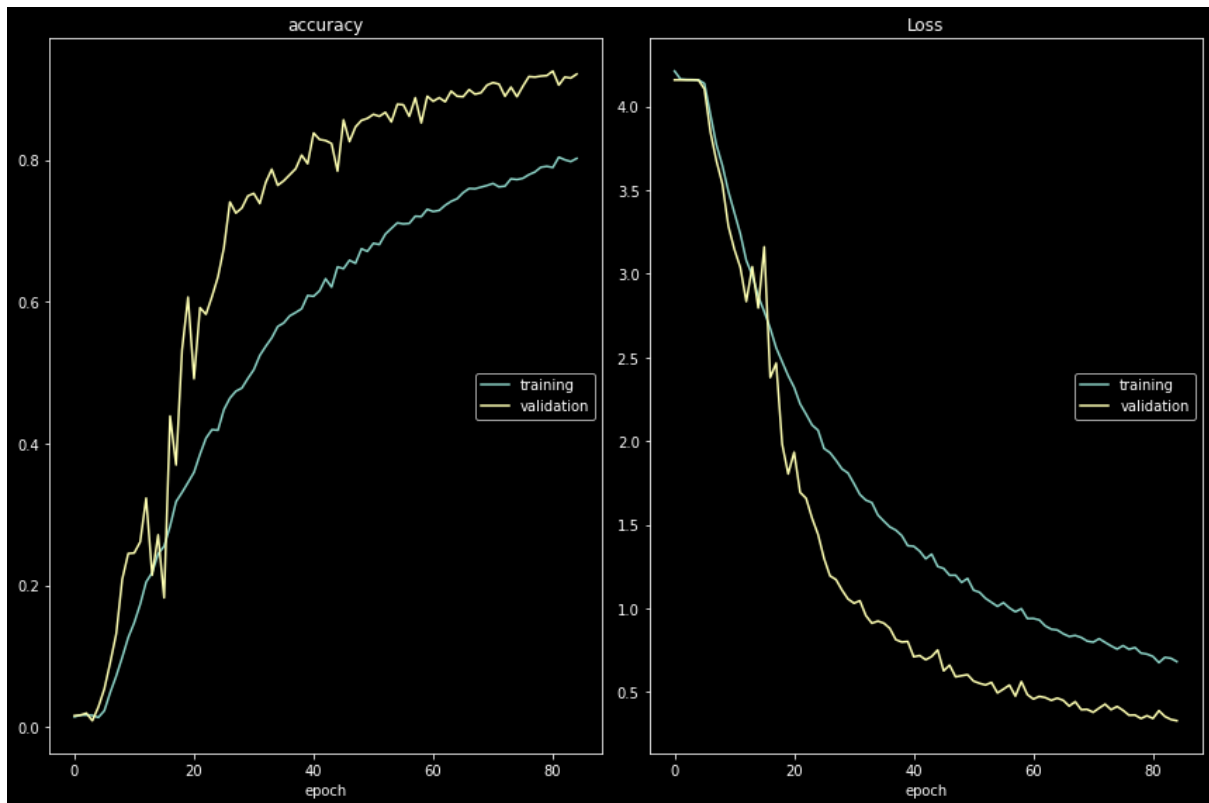


Figure 6.6: Evaluation of accuracy and loss with the number of epochs in InceptionV3

Our CNN model reduced the overfitting problem while classifier, AlexNet and InceptionV3 failed to reduce. Our proposed CNN model outperformed other models with high accuracy in less epochs.

6.4 Model comparison in terms of test accuracy:

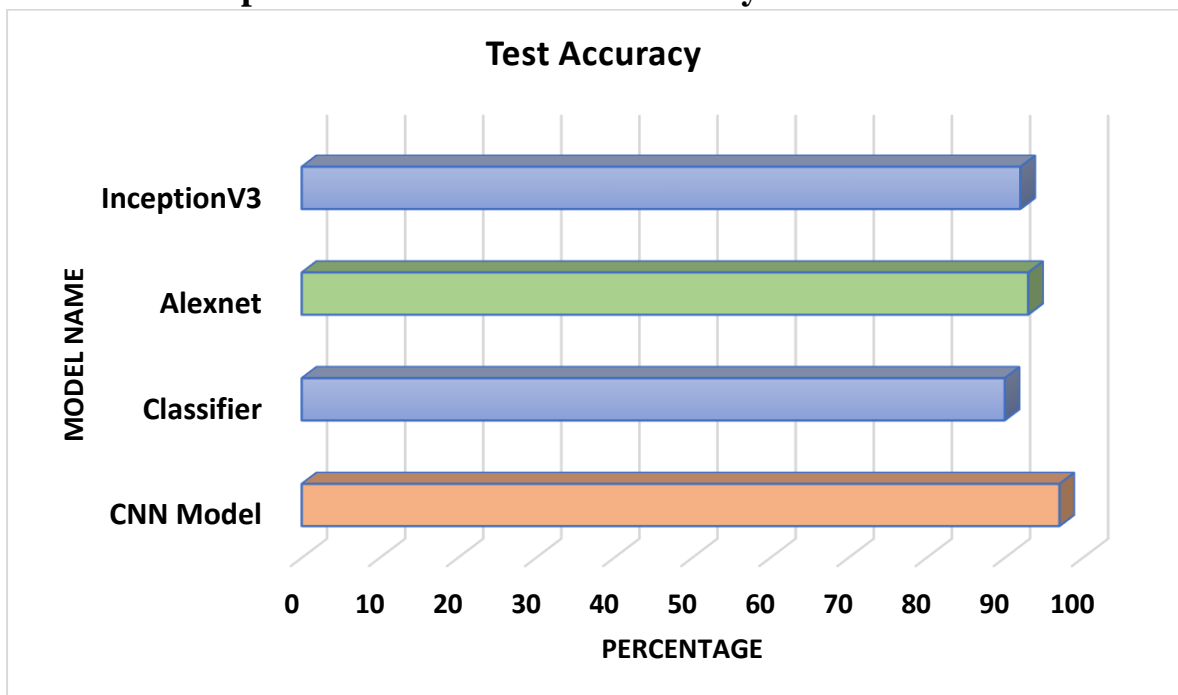


Figure 6.7: Comparison based on Test Accuracy

6.5 List of Hyperparameters:

Models	Batch size	Optimizer	Best Weight Epochs
CNN Model	32	RMSprop	50
Classifier	32	RMSprop	78
Alexnet	32	RMSprop	35
InceptionV3	32	RMSprop	86

Table 6.1: Hyper-parameters used in each model for district name recognition

6.6 Effects of the activation function on CNN Model:

	Proposed Dataset	
Activation Function	Relu	Tanh
Training Accuracy	97%	94%
Test Accuracy	97%	95%

Table 6.2: Effects of activation function

6.7 Comparison of models on proposed and benchmark datasets:

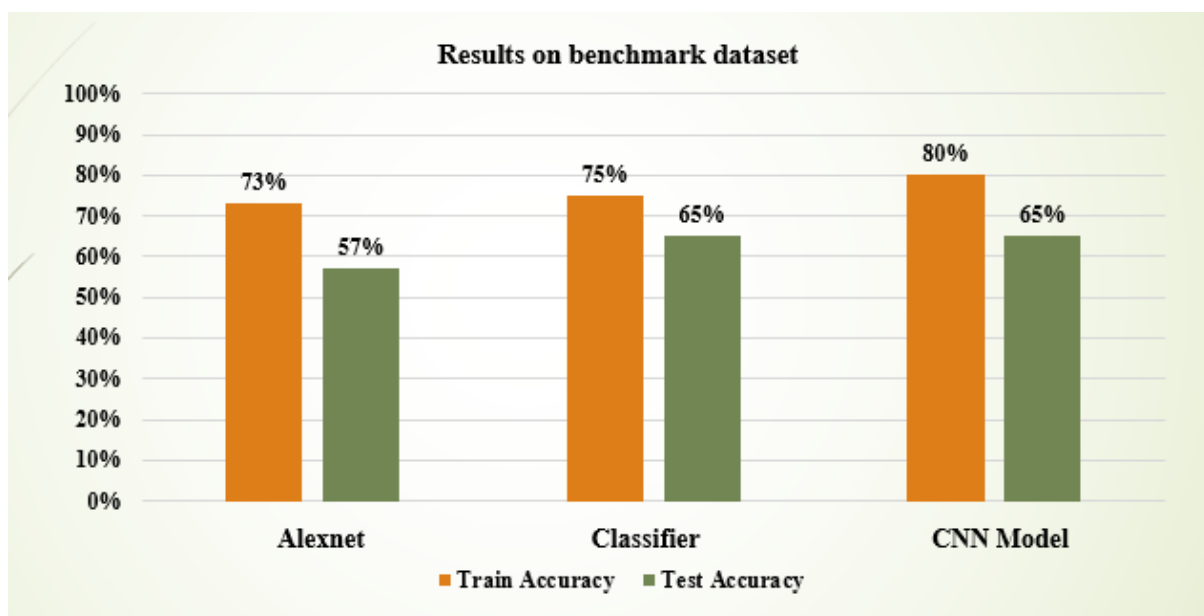


Figure 6.8: Results on BN-HTRd dataset

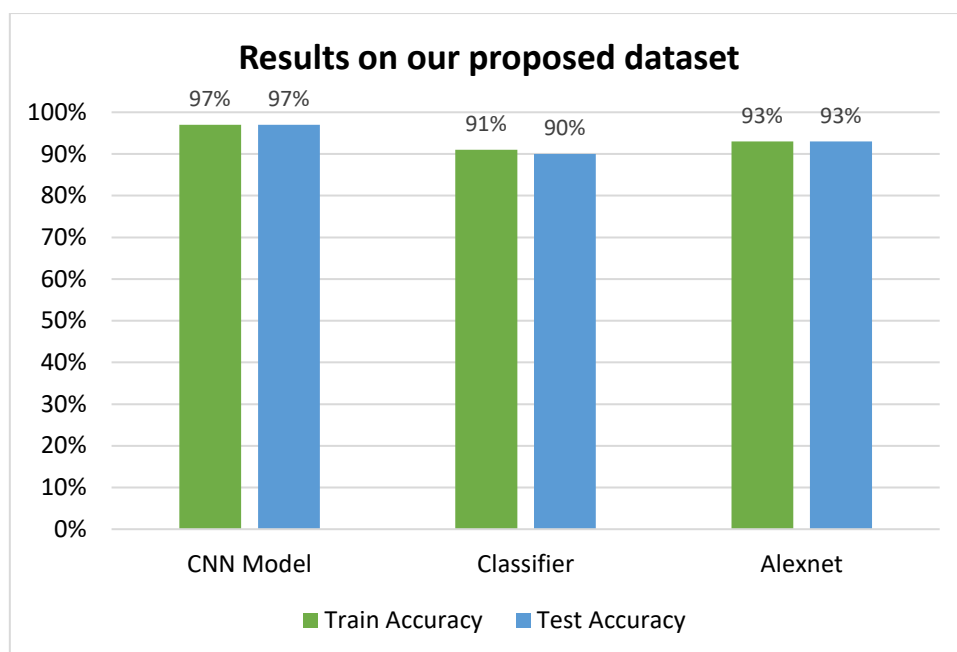


Figure 6.9: Results on our proposed dataset

6.8 Classification of District name recognition:

We have tested a number of images for detecting their exact classes. Through the process, we have noticed that some of the images have classified correctly and some of the images failed to classify correctly. Examples of classification are given below:



Figure 6.10: Correct class prediction

Above figure shows that our system predicted most of the images accurately. Accurately predicted images are denoted by green text. Top of each class, the text indicates predicted label and next to this indicates actual label which is shown using parenthesis.

6.9 Miss Classification of District name recognition:

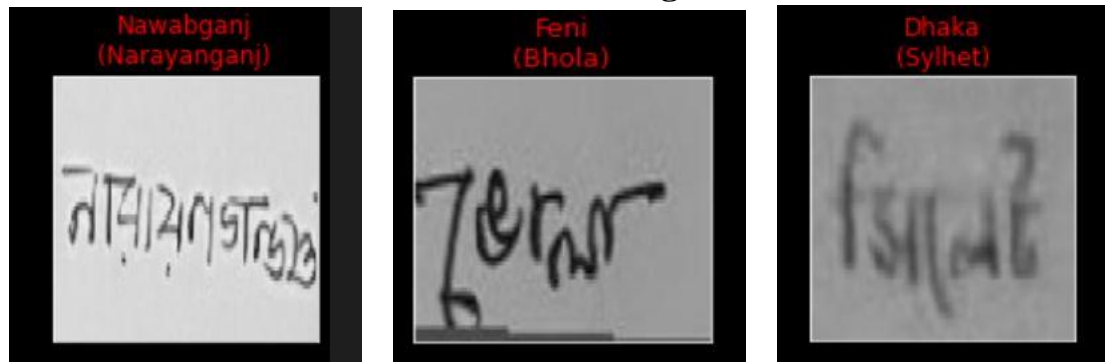


Figure 6.11: Incorrect class prediction

Above figure shows that our system failed to predict few images. Here wrong predicted images are denoted with red text. Our model failed to predict images due to low intensity gradient and curly handwritings.

6.10 Comparison among Classifier, Alexnet and proposed CNN models in terms of Recall and F1 score:

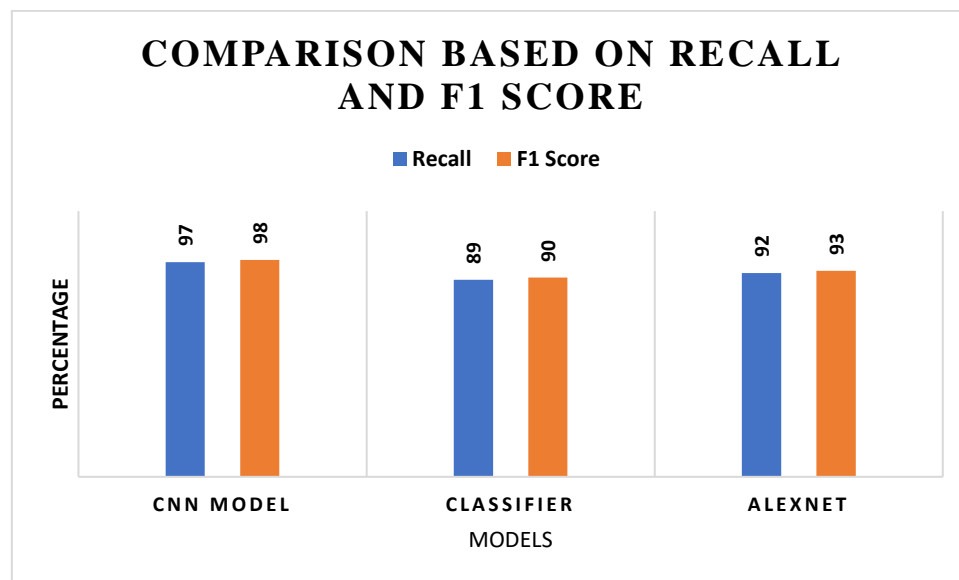


Figure 6.12: Comparison of models based on Recall and F1 Score.

6.11 Confusion Matrix:

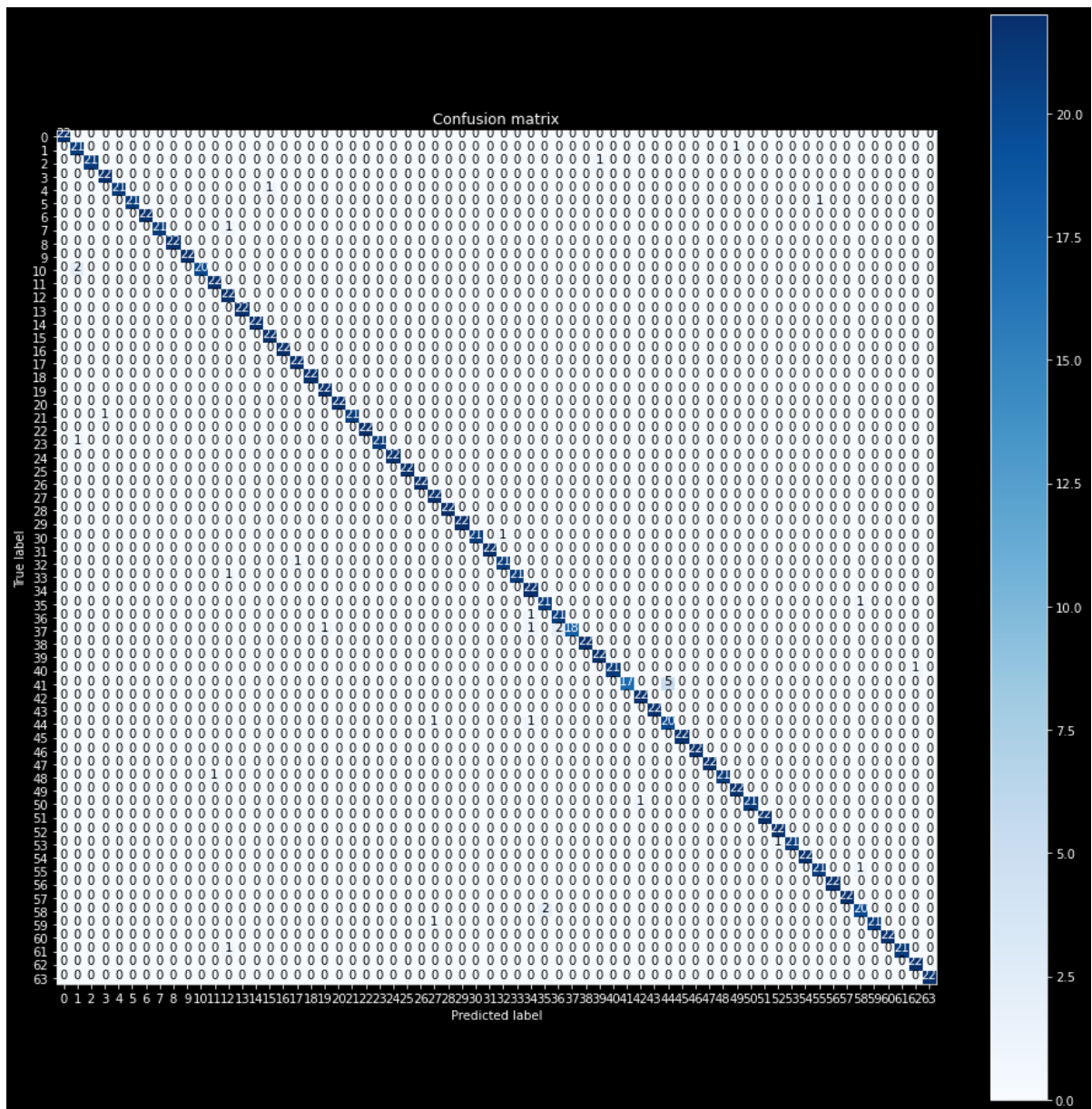


Figure 6.13: Confusion matrix

6.12 Representation of sensitivity for some selected classes:

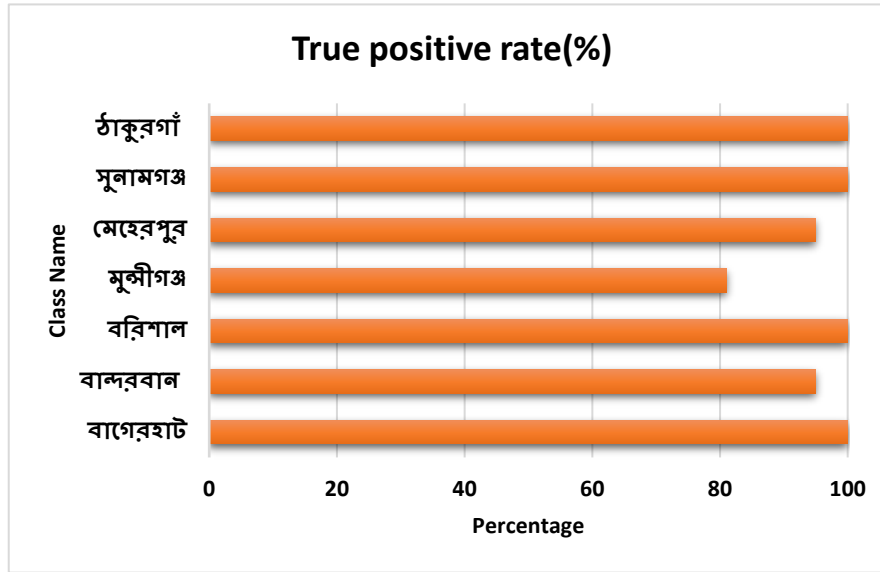


Figure 6.14: Percentage of true positive rate.

6.13 Calculating TP, FP, TN, FN for some selected class from Confusion Matrix of our proposed model:

Now, calculating the precision value for some selected class from the confusion matrix by applying the following formula we get:

Precision for Bagerhat:

$$P = \frac{TP}{TP+FP}$$

$$= \frac{22}{22+0}$$

$$= 1.00$$

Precision for Bandarban:

$$P = \frac{TP}{TP+FP}$$

$$= \frac{21}{21+0}$$

$$= 0.88$$

Now calculating the Recall value for each class from the confusion matrix by applying the following formula we get:

Recall for Barishal:

$$\begin{aligned} R &= \frac{TP}{TP+FN} \\ &= \frac{22}{22+0} \\ &= 1.00 \end{aligned}$$

Recall for Meherpur:

$$\begin{aligned} R &= \frac{TP}{TP+FN} \\ &= \frac{21}{21+1} \\ &= 0.9545 \end{aligned}$$

Now calculating the f1 score value of some selected class from the confusion matrix by applying the following formula we get:

F1 score for Sunamgonj:

$$\begin{aligned} F &= 2 \times \frac{P \times R}{P+R} \\ &= 2 \times \frac{1 \times 1}{1+1} \\ &= 1 \end{aligned}$$

F1 score for Thakurgoan:

$$\begin{aligned} F &= 2 \times \frac{P \times R}{P+R} \\ &= 2 \times \frac{1 \times 1}{1+1} \\ &= 1 \end{aligned}$$

6.14 Classification report based on True Positive Rate:

Class No	Class_Name	True positive % rate	Class No	Class_Name	True positive % rate
0	বাগেরহাট	100	32	মাদারীপুর	96
1	বান্দরবান	96	33	মাগুরা	96
2	বরগুনা	96	34	মানিকগঞ্জ	100
3	বরিশাল	100	35	মেহেরপুর	96
4	ভোলা	96	36	মৌলভীবাজার	96
5	বগুড়া	96	37	মুন্সীগঞ্জ	82
6	ব্রাহ্মণবাড়ীয়া	100	38	ময়মনসিংহ	100
7	চাঁদপুর	96	39	নওগাঁ	100
8	চট্টগ্রাম	100	40	নড়াইল	96
9	চুয়াডাঙ্গা	100	41	নারায়ণগঞ্জ	78
10	কক্সবাজার	91	42	নরসিংদী	100
11	কুমিল্লা	100	43	নাটোর	100
12	ঢাকা	100	44	নওয়াবগঞ্জ	91
13	দিনাজপুর	100	45	নেত্রকোনা	100
14	ফরিদপুর	100	46	নীলফামারী	100
15	ফেনী	100	47	নোয়াখালী	100
16	গাইবান্ধা	100	48	পাবনা	96
17	গাজীপুর	100	49	পঞ্চগড়	100
18	গোপালগঞ্জ	100	50	পটুয়াখালী	96
19	হবিগঞ্জ	100	51	পিরোজপুর	100
20	জামালপুর	100	52	রাজবাড়ী	100
21	যশোর	96	53	রাজশাহী	96
100	ঝালকাঠি	100	54	রাঙ্গামাটি	100
23	ঝিনাইদহ	96	55	রংপুর	96
24	জয়পুরহাট	100	56	সাতক্ষিরা	100
25	খাগড়াছড়ি	100	57	শরীয়তপুর	100
26	খুলনা	100	58	শেরপুর	91
27	কিশোরগঞ্জ	100	59	সিরাজগঞ্জ	96
28	কুড়িগ্রাম	100	60	সুনামগঞ্জ	100
29	কুষ্টিয়া	100	61	সিলেট	96
30	লক্ষীপুর	96	62	টাঙ্গাইল	100
31	লালমনিরহাট	100	63	ঠাকুরগাঁ	100

Table 6.3: Classification report based on True positive rate

6.15 Classification Report:

Class No	Class_Name	Precision	Recall	F1 Score	Support
0	বাগেরহাট	1	1	1	22
1	বান্দরবান	0.88	0.95	0.91	22
2	বরগুনা	1	0.95	0.98	22
3	বরিশাল	0.96	1	0.98	22
4	ভোলা	1	0.95	0.98	22
5	বগুড়া	1	0.95	0.98	22
6	ব্রাহ্মণবাড়ীয়া	1	1	1	22
7	চাঁদপুর	1	0.95	0.98	22
8	চট্টগ্রাম	1	1	1	22
9	চুয়াডাঙ্গা	1	1	1	22
10	কক্সবাজার	1	0.91	0.95	22
11	কুমিল্লা	0.96	1	0.98	22
12	ঢাকা	0.88	1	0.94	22
13	দিনাজপুর	1	1	1	22
14	ফরিদপুর	1	1	1	22
15	ফেনী	0.96	1	0.98	22
16	গাইবান্ধা	1	1	1	22
17	গাজীপুর	0.96	1	0.98	22
18	গোপালগঞ্জ	1	1	1	22
19	হবিগঞ্জ	0.96	1	0.98	22
20	জামালপুর	1	1	1	22
21	যশোর	1	0.95	0.98	22
22	ঝালকাঠি	1	1	1	22
23	ঝিনাইদহ	1	0.95	0.98	22
24	জয়পুরহাট	1	1	1	22
25	খাগড়াছড়ি	1	1	1	22
26	খুলনা	1	1	1	22
27	কিশোরগঞ্জ	0.92	1	0.96	22
28	কুড়িগ্রাম	1	1	1	22
29	কুষ্টিয়া	1	1	1	22
30	লক্ষীপুর	1	0.95	0.98	22
31	লালমনিরহাট	1	1	1	22
32	মাদারীপুর	0.95	0.95	0.95	22
33	মাগুরা	1	0.95	0.98	22
34	মানিকগঞ্জ	0.88	1	0.94	22
35	মেহেরপুর	0.91	0.95	0.93	22
36	মৌলভীবাজার	0.91	0.95	0.93	22

37	মুল্লীগঞ্জ	1	0.82	0.9	22
38	ময়মনসিংহ	1	1	1	22
39	নওগাঁ	0.96	1	0.98	22
40	নড়াইল	1	0.95	0.98	22
41	নারায়ণগঞ্জ	1	0.77	0.87	22
42	নরসিংদী	0.96	1	0.98	22
43	নাটোর	1	1	1	22
44	নওয়াবগঞ্জ	0.8	0.91	0.85	22
45	নেত্রকোনা	1	1	1	22
46	নীলফামারী	1	1	1	22
47	নোয়াখালী	1	1	1	22
48	পাবনা	1	0.95	0.98	22
49	পঞ্চগড়	0.96	1	0.98	22
50	পটুয়াখালী	1	0.95	0.98	22
51	পিরোজপুর	1	1	1	22
52	রাজবাড়ী	0.96	1	0.98	22
53	রাজশাহী	1	0.95	0.98	22
54	রাঙ্গামাটি	1	1	1	22
55	রংপুর	0.95	0.95	0.95	22
56	সাতক্ষিরা	1	1	1	22
57	শরীয়তপুর	1	1	1	22
58	শেরপুর	0.91	0.91	0.91	22
59	সিরাজগঞ্জ	1	0.95	0.98	22
60	সুনামগঞ্জ	1	1	1	22
61	সিলেট	1	0.95	0.98	22
62	টাঙ্গাইল	0.96	1	0.98	22
63	ঠাকুরগাঁ	1	1	1	22

Table 6.4: Classification report based on Precision, Recall and F1 Score

6.16 Calculation of average F1 score:

$$\text{Average F1 score} = \frac{\text{Sum of F1 score of each class}}{\text{Total no of class}}$$

$$\begin{aligned}
& 1+0.91+0.98+0.98+0.98+0.98+ \\
& 1+0.98+1+1+0.95+0.98+0.94 \\
& +1+1+0.98+1+0.98+1+0.98 \\
& +1+0.98+1+0.98+1+1+1+0.96 \\
& +1+1+0.98+1+0.95+0.98+0.94+ \\
& 0.93+0.93+0.90+1+0.98+0.98+ \\
& 0.87+0.98+1+0.85+1+1+1+0.98+0.98 \\
& +0.98+1+0.98+0.98+1+0.95+1+1+0.91 \\
& +0.98+1+0.98+0.98+1 \\
& = \frac{62.51}{64} = 0.98
\end{aligned}$$

Chapter 7

Future Work and Conclusion

7.1 Future Work: We have already recognized the handwritten district name. Our future work would be detecting the handwritten words. We will also try to generate text from given input handwritten images. We will improve dataset quality by removing noisy images and we will enlarge our dataset by adding more images.

7.2 Conclusion: In this study, we have proposed a benchmark dataset as well as robust and efficient CNN based classification system to recognize handwritten district names. We have used unsupervised word segmentation to extract words from Bangla handwritten images and a CNN model has been trained to recognize handwritten district name. Our proposed CNN model achieved 97% accuracy on our proposed dataset. We have used a benchmark dataset to analyze our model. Our proposed CNN model achieved 80% accuracy on a benchmark dataset. We have compared our proposed model with three other CNN architecture and our proposed CNN model outperformed other models with high accuracy in less epochs.

References:

- [1] Alom, Md. Zahangir & Asari, .. (2017). Handwritten Bangla Character Recognition Using the State-of-Art Deep Convolutional Neural Networks. Computational Intelligence and Neuroscience. 2018. 10.1155/2018/6747098.
- [2] M. N. Hoq, N. Anjum Nipa, M. M. Islam and S. Shahriar, "Bangla Handwritten Character Recognition: an overview of the state-of-the-art classification algorithm with new dataset," 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), 2019, pp. 1-6, doi: 10.1109/ICASERT.2019.8934641.
- [3] Wikipedia contributors, "Bengali alphabet," *Wikipedia, The Free encyclopedia*, https://en.wikipedia.org/w/index.php?title=Bengali_alphabet&oldid=1082593770 (accessed April 15, 2022).
- [4] Wikipedia contributors. (2021, July 1). List of languages by number of native speakers. In Wikipedia, The Free Encyclopedia. Retrieved 10:14, July 9, 2021, from https://en.wikipedia.org/w/index.php?title=List_of_languages_by_number_of_native_speakers&oldid=1031431808
- [5] M. N. Hoq, N. Anjum Nipa, M. M. Islam and S. Shahriar, "Bangla Handwritten Character Recognition: an overview of the state-of-the-art classification algorithm with new dataset," 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), 2019, pp. 1-6, doi: 10.1109/ICASERT.2019.8934641.
- [6] Rahman, Md Ataur; Paul, Mitu; Tabassum, Nazifa; Pal, Riya; (2021), "First Step towards End-to-End Bangla Handwritten Text Recognition (HTR)", Bachelor's Thesis, Premier University, Chittagong.
- [7] Balakrishnan, K., 2017. Offline handwritten recognition of Malayalam district name-a holistic approach. *arXiv preprint arXiv:1705.00794*.
- [8] Ashrafee, A., Khan, A.M., Irbaz, M.S., Nasim, A. and Abdullah, M.D., 2022. Real-time Bangla License Plate Recognition System for Low Resource Video-based Applications. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 479-488).
- [9] Pervej, Masud et al. "Real-Time Computer Vision-Based Bangla Vehicle License Plate Recognition using Contour Analysis and Prediction Algorithm" International Journal of Image and Graphics. 202, doi.org/10.1142/S021946782150042X
- [10] Rahman, Md Ataur; Paul, Mitu; Tabassum, Nazifa; Pal, Riya; Das, Bipon; Tasnim, Raisa; Gony, Osman; Noor, Fatin; Jubaer, Sheikh Mohammad; Chowdhury, Mehanaz; Akter, Yeasmin Ara; Islam, Mohammad Khairul (2021), "BN-HTRd: A Benchmark Dataset for Document Level Offline Bangla Handwritten Text Recognition (HTR)", Mendeley Data, V1, doi: 10.17632/743k6dm543.1
- [11] M. M. Shaifur Rahman, M. Mostakim, M. S. Nasrin and M. Z. Alom, "Bangla License Plate Recognition Using Convolutional Neural Networks (CNN)," 2019 22nd International Conference on Computer and Information Technology (ICCIT), 2019, pp. 1-6, doi: 10.1109/ICCIT48885.2019.9038597.
- [12] Ashrafee, Alif, Akib Khan, Mohammad Sabik Irbaz and Md Abdullah Al Nasim. "Real-time Bangla License Plate Recognition System for Low Resource Video-based

- Applications.” *2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)* (2022): 479-488.
- [13] Balci, Batuhan, Dan Saadati and Daniel Shiferaw. “Recognition using Deep Learning.” (2017).
- [14] Sridhar.S, Guttula Bhargav Mani Deep, Asish Bhoi, Danthuluri Swathi, Deepilli Leelarani. “Character Recognition Using Deep Learning Algorithm.” *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* Vol. 12 No. 14 (2021)
- [15] Nikitha, A., J. Geetha and D. S. Jayalakshmi. “Handwritten Text Recognition using Deep Learning.” *2020 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)* (2020): 388-392.
- [16] Nikitha, A., J. Geetha and D. S. Jayalakshmi. “Handwritten Text Recognition using Deep Learning.” *2020 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)* (2020): 388-392.
- [17] Sukhdeep Singh, Anuj Sharma, Vinod Kumar Chauhan, “Online handwritten Gurm-ukhi word recognition using fine-tuned Deep Convolutional Neural Network on offline features,” *Machine Learning with Applications*, Volume 5,2021,100037, ISSN 2666-8270, doi.org/10.1016/j.mlwa.2021.100037.
- [18] Barua, Shilpi & Malakar, Samir & Bhowmik, Showmik & Sarkar, Ram & Nasipuri, Mita. (2017). Bangla Handwritten City Name Recognition Using Gradient-Based Feature. 10.1007/978-981-10-3153-3_34.
- [19] Shamim, S. M., Miah, M. B. A., Sarker, A., Rana, M., & Al Jobair, A. (2018). Handwritten Digit Recognition Using Machine Learning Algorithms. *Global Journal of Computer Science and Technology*
- [20] Szilárd Vajda, Kaushik Roy, Umapada Pal, Bidyut B Chaudhuri, Abdel Belaïd. Automation of Indian Postal Documents written in Bangla and English. *International Journal of Pattern Recognition and Artificial Intelligence*, World Scientific Publishing, 2009, 23 (8), pp.1599-1632. ff10.1142/S0218001409007776ff. ffinria-00435501f
- [21] J, Jino P and Balakrishnan, Kannan. "Offline Handwritten Recognition of Malayalam District Name - A Holistic Approach" arXiv:1705.00794, 2017.
- [22] N. Bhattacharya and U. Pal, "Stroke Segmentation and Recognition from Bangla Online Handwritten Text," 2012 International Conference on Frontiers in Handwriting Recognition, 2012, pp. 740-745, doi: 10.1109/ICFHR.2012.275.
- [23] <https://www.kaggle.com/datasets/mdibrahimsiddiqueee/a-bangla-handwritten-district-name-dataset>
- [24] <https://www.mathsisfun.com/data/data.html>
- [25] https://byjus.com/maths/data-sets/?fbclid=IwAR1UBgoKfONR4_ekhRGBMF_BUJ7WkQh0HpycoJpywlFAZyoA6eiXIKJNcMM
- [26] https://www.frontiersin.org/files/Articles/598916/fdgth-03-598916-HTML/image_m/fdgth-03-598916-g002.jpg
- [27] <https://www.clickworker.com/wp-content/themes/clickworkerV8/assets-dist/img/ai-page/subpages/image-annotation.svg>
- [28] <https://research.aimultiple.com/audio-annotation/>
- [29] <https://www.cogitotech.com/audio-annotation-services>
- [30] <https://imerit.net/video-annotation/>

- [31] <https://dataloop.ai/wp-content/uploads/2021/10/Blog-feature-image-2240x1260-7.png>
- [32] https://www.tutorialspoint.com/dip/optical_character_recognition.htm
- [33] Rahman Ridoy, Md Abdur; (2018), “Automated Number Plate Recognition”, Bachelor’s Thesis, Islamic University of Technology (IUT), Dhaka.
- [34] Wikipedia contributors, "Optical character recognition," *Wikipedia, The Free Encyclopedia*, https://en.wikipedia.org/w/index.php?title=Optical_character_recognition&oldid=1079990571 (accessed April 16, 2022).
- [35] Quader, Abdullah Bin; Sakib, Wahid; Khalil, Md. Ibrahim; (2021), “Bangla Handwritten Word Recognition by Localizing & Classifying Characters”, Bachelor’s thesis, Premier University, Chittagong.
- [36] <https://www.flatworldsolutions.com/data-management/articles/key-advantages-ocr-based-data-entry.php>
- [37] <https://heartbeat.comet.ml/a-2019-guide-to-object-detection-9509987954c3>
- [38] Chowdhury, Pratim; Banik, Priota; (2022), “Bangla Numerical Sign Language Recognition on novel dataset”, Bachelor’s thesis, Premier University, Chittagong.