

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project/internship successfully.

We really grateful and wish our profound our indebtedness to **SHEIKH ABUJAR, LECTURER**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “*Data Mining*” to carry out this project. His endless patience ,scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to **Dr. Syed Akhter Hossain**, Professor and Head, Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

Abstract

This report presents **“Bangla Handwritten Character Recognition using Convolutional Neural Network”**. Sample training data, scanned using a modest scanner. Pre-processing steps that follows are skew angle detection and correction, noise removal, line, word and character separation. The separated characters are then fed into a 10 layer Convolutional Neural Network for training. Finally, this network is used to recognize handwritten Bangla scripts. For collection of data we have developed a form that helped us getting the most out of data. The form was designed in such manner so that it helps us during the preprocessing. In preprocessing we segmented each character separately and fed that data to our own developed 10 layered neural network.

TABLE OF CONTENTS

CONTENS	PAGE
Board of examiners	ii
Declaration	iii
Acknowledgements	iv
Abstract	v
CHAPTER	
CHAPTER 1: INTRODUCTION	1-2
1.1 Introduction	1
1.2 Motivation	2
1.3 Research Questions	2
1.4 Expected Outcome	2
1.5 Report Layout	2
CHAPTER 2: LITERATURE REVIEW	4-6
2.1 Introduction	4
2.2 Related Works	4
2.3 Research Summary	5
2.4 Scope of the Problem	6
2.5 Challenges	6

CHAPTER 3: RESEARCH METHODOLOGY	7-25
3.1 Introduction	7
3.2 Form Preparation	7
3.3 Processing Handwritten Image	11
3.4 Some Necessary Corrections	11
3.4.1 Grayscale Conversion	12
3.4.2 Blurring the Image	14
3.5 Segmentation	15
3.5.1 Detecting Canny Edge	15
3.5.2 Manipulating the Contours and Crop	16
3.5.3 Skewness and Rotation	16
3.5.4 Resizing	18
3.5.5 Row Wise Cropping	18
3.5.6 Column Wise Cropping	19
3.5.7 Noise Removal	19
3.6 Algorithms	20
3.6.1 Algorithm to Separate Each Character	20
3.6.2 Remove Noise and Extra Background	21
3.7 Recognition using CNN	22
3.7.1 Convolutional Neural Network	22
3.7.2 Proposed Model	22
3.7.3 Diagram of CNN	23

3.7.4 Details of Neural Network	24
3.7.5 Gradient Descent Algorithm	25
3.7.6 Max-Pooling Layer	25
3.7.7 Dataset	25
CHAPTER 4: EXPERIMENTAL RESULTS AND DISCUSSION	26-27
4.1 Experimental Results	26
4.2 Discussion	26
CHAPTER 5: SUMMARY, CONCLUSION AND IMPLICATION FOR FUTURE RESEARCH	28-29
5.1 Summary	28
5.2 Conclusion	28
5.3 Implication for further Research	29
REFERENCES	30
APPENDICES	31-32
Appendix A: Research Reflection	31
Appendix B: Plagiarism Report	32

LIST OF TABLES

TABLE	PAGE NO
Table 1: Classification Performance of different Deep Learning Architecture	4
Table 2: Performance Comparison with the state of art method using CMATERdb Dataset	5
Table 3: Accuracy of previous works	5
Table 4: Layer wise description	24
Table 5: Accuracies of our model	26

LIST OF FIGURES

Figure		Page
Figure 3.1:	A Complete workflow	7
Figure 3.2:	All Bangla Characters with Handwritten Form	8
Figure 3.3:	Edges and Boundaries of our Developed Form	9
Figure 3.4:	The Whole Form	10
Figure 3.5:	Scanning Handwritten Forms	11
Figure 3.6(a):	Before Grayscale Conversion	12
Figure 3.6(b):	After Grayscale Conversion	13
Figure 3.7:	Blurring the Image	14
Figure 3.8:	Canny Edge Detection	15
Figure 3.9:	Contour Detection and Cropping	16
Figure 3.10(a):	Before Skew Correction	17
Figure 3.10(b):	After Skew Correction	17
Figure 3.11:	Resizing the Image	18
Figure 3.12:	Row Wise Cropping	18
Figure 3.13:	Column Wise Cropping	19
Figure 3.14:	Removal of Noises	19
Figure 3.15:	Removal of Extra Background	20
Figure 3.16:	Layered Diagram of Proposed Model	23
Figure 4.1:	Accuracy on Bornocchota	26
Figure 4.2:	Accuracy of NumtaDB	27
Figure 4.3:	Accuracy on mixed Dataset	27
Figure A1:	Plagiarism Report	32

CHAPTER 1

INTRODUCTION

1.1 Introduction

This is the time in this world when “Information” is considered as the most valuable asset. And language is the boat through which maximum of the data and information floats. But it is undeniable that there is a clear division between handwritten text and printed text that almost surrounds us in every step nowadays. By the passage of time, to reduce this gap many researchers and mathematicians produced several formula and laws and techniques for hand-engineered specifications. Especially when it comes to the recognition of handwritten characters the similarities among all the distinct characters and their shapes, neighboring characters, the overlapping issues tremendously makes the problem bigger. An OCR that detects handwritten character recognition needs to be made focusing on two major areas: extraction of features from data sets and then apply learning tools for the classification of those characters. With distinct extracting techniques, artificial neural network and convolutional neural network methodologies are considered to be one of the most result providing methods for handwritten character recognition. Morphological/Rank/Linear Neural Network(MRL-NN) where the combination of inputs in every node is made by hybrid linear and nonlinear (of the morphological/rank type) operations is well observed for recognition of handwritten digits. MLP-SVM which is a hybrid multilayer perceptron based recognition system which have been found to be providing very good results for English [1] and Chinese character recognition [2]. But Convolutional Neural Networks have provided a new dimension of recognition of patterns from pixel images with the least effort of preprocessing images. Which means CNN automatically provides some degree of translation to the whole recognition phase.

Optical Character Recognition (OCR) includes detecting text from paper and converting the images in such a format that a computer can manipulate and do various computational operations. There has been a noteworthy amount of improvement in languages like Arabic, English. But recognition of Handwritten Bengali scripts is still in its early age. This thesis proposes a neural network approach for Bangla Handwritten Character Recognition. There have been number of preprocessing steps that was used to prepare the data for the recognition. It includes preparing a forms to collect data, scanning forms, skew detection and reduction,

cropping, separation of characters, noise removal and many more. This report provides details of preprocessing steps from chapter 3 to 5. Chapter 6 explains the recognition procedure with neural network. And chapter 2 elaborates some properties of Bangla Characters and handwritten characters that are widely studied to work with OCR.

1.2 Motivation

We all are know that, Bengali is usually counted as the sixth most spoken native language in the world by population, which is approximately 250–300 million total speakers worldwide. In this modern era, technology is our most dependable option. So it is very important to enrich our computer vision through bangla language.

The field of computer vision, the rise of high performance computing(HPC) systems turn the problem of image recognition very hard. So now there exists a wide range of algorithms and architecture to manage such problems. Many models are created to solve those problems using neural networks. It has been designed on the base on some artificial function of the neural pathways of human brain. So to maximize the performance of such architecture, models have been proposed and one such implementation of a system of detector and recognizer that uses machine learning techniques is convolutional neural network or CNN.

1.3 Research Questions

Bangla OCR is already very important topic in research field. Therefore, it is difficult to find information about that where do we start from and where we can get all the necessary resources, how do we build our own database. That is the burning question of this work.

1.4 Expected outcome

Our expectation to classify and recognize Bangla handwritten characters from sample images of handwritten isolated words. Make a proper utilization for a vital model that can achieves more accurate value compared to last generated value of Bangla handwritten character recognition.

1.5 Report layout

The purpose of the project is to create a prototype of bangla handwritten character recognition system. The report related to the study, design, implementation and testing of the system. This study aims to develop a prototype system to detect and classify the handwritten character by using image processing technique as an alternative of manual detection method.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Character recognition is a large working field in every language. Implementations on Bangla handwritten characters are presented by various researcher. Most of that work are very appreciated. They set a high standard for future researcher to work and implementation on OCR.

2.2 Related Works

In past, some renowned works have been done on Bangla Handwritten recognition. Many researcher groups came with many different strategies [10], [11], [12], [13], [14]. In [10], the paper authors create 30000 samples for training purpose and test them by two different test sets, one with 6000 trained characters and another with random untrained characters with same size of samples. The authors randomly select 100 sample of each character for each test case. In this paper the authors used SVM and Bootstrap Aggregating Technique for recognition Bangla handwritten character. Where Hog feature extraction and Binary pixel features methods were combined together. This paper reported about achieving maximum 89.8% accuracy for SVM classification model and 93% for Bootstrap Aggregating model. AKM Shahariar Azad Rabby, Sadeka Haque, Sheikh Abujar and Syed Akhter Hossain proposed strategy[8] for 50 basic characters ,10 digits, 10 modifiers and 52 mostly used compound handwritten bangla character. Their proposed EkushNet used a Multilayer CNN for classifying bangla handwritten characters. This model used convolution, Max pooling layer, fully connected dense layer and some regularization method . They used 28*28 pixel datasets. For Ekush dataset they got 97.73% accuracy and 95.01% crossvalidation accuracy on CMATERdb dataset. Shyla Afroge, Boshir Ahmed and Ali Hossain proposed a strategy [11] for printed characters where Feature Extraction used for representing a character and Multilayer feed forward neural network for recognize character. In this paper author proposed a geometric technique called Discrete Frechet Distance which is used for recognition process. They did not used any standard dataset for Bangla OCR. The data they were used deliberated as trained characters or words, untrained similar fonts (solaimanlipi is the trained font and similar font is dhanshiri), scanned documents are of solaimanlipi fonts and lastly irregular shape fonts (some big or small fonts). At first, they grab the whole image, take the binary data and scale it onto 60×70 pixels which are followed by binarizations. 1 is represented for character area and 0 for rest of the area. This paper

reported that the author tested 50 trained characters which provide 100% accuracy and overall 90-95% accuracy for all basic characters of the developed method on training and the tests sets respectively. In this paper it still has some problems for similar text and dots are removed as noise. Nazmul Arefin proposed a framework [12] for segmenting the lines, words and characters from the input documents where distance-based segmentation (DBS) method is used to segment the line from the document, words from the line and characters from words. Histogram Oriented Gradients (HOG) used for features extraction and classify by Support Vector Machine (SVM). In this paper the author used the manuscripts of Tagore and several people that respectively contained 500 and 600 words that were collected to evaluate the performance of the proposed method. The paper reported that the proposed framework shows 94.32% accuracy. But the proposed framework was not shown the satisfactory result for the joint characters. Mujadded Al Rabbani Alif et al [14] proposed a modified ResNet-18 architecture which is a convolutional neural network architecture for recognizing Bangla Handwritten characters. By applied modified ResNet-18 architecture on CMATER db dataset they achieve 95.10% classification accuracy. Nibaran et al. [13] proposed a feature set representation for Bangla handwritten alphabets recognition. Their feature set is a combination of 24 shadow features, 8 distance features, 16 centroid features and 84 quad tree based longest run features. They used MLP classifier for recognition of unknown Bangla basic characters using the above mentioned 132 feature set. A training set of 8000 samples and a test set of 2000 samples were formed through random selection. The paper reported that the proposed feature shows 85.40% accuracy on 50character classes dataset

Table 1: Classification Performance of different deep learning architecture.

Deep Learning Architecture	Accuracy (%)
VGGNet-16	91.00
VGGNet-19	92.11
ResNet-18	94.52
ResNet-34	94.59
EkushNet	97.73

Table 2: Performance comparison with the state-of-art method using CMATERdb dataset.

State-of-the-art method	Accuracy (%)
GF+SVM	74.06
CH+SVM	75.04
QTLR+SVM	77.51
CH+QTLR+SVM	81.00
RS-CH+QTLR+SVM	86.43
DCNN	84.04
SL-DCNN	90.33

Table 3: Accuracies of previous works

Dataset	Accuracy(%)
Ekush	97.73
ISI	96.81
BornoNet	96.40
BanglaLekha	95.71

2.3 Research Summary

Convolutional Neural Network (CNN) has been found to be the most effective method for handwritten digit and character recognition in any language. CNN is also helping researchers to use their state of the art methodologies and procedures to solve real world problems and gain the best accuracy possible. CNN in a supervised fashion can provide better results than previously done non-CNN classification system for digits and characters. BornoNet[7] is one of the remarkable proposed model that was based on Convolutional Neural Network and it

achieved good accuracies on various large datasets that are made of handwritten Bangla digits and characters.

2.4 Scope of the problem

After we have gone through the background study and brainstorming discussion we have discovered that the compound characters in bangla are given less attention in almost every dataset. We have tried to provide more accuracy with training, testing and validating the compound characters. Also, we have covered the numerals as well as the vowels and consonants of bangla language.

2.5 Challenges

The accumulation of data from a different person has always been a big challenge. One of the challenges is create a form for collecting handwritten data. To make a training dataset without image it is not possible. There are some challenges happen that effect the system when it is running. The image produces a lot of noise which unable to process the CNN model for training and testing. The quality of the image samples was blurring and sometimes it is not able to be process. Sometimes handwritten data are not written in the fix position.

CHAPTER 03

RESEARCH METHODOLOGY

3.1 Introduction

To make a model for bangla character recognition, there must be a model or algorithm or system that can accurately recognize the character so that it can identify the character. The methodology consists of the pre-processing and segmentation of dataset. The method will be applied for the further model of bangla character recognition. Figure 3.1 showing the complete workflow.

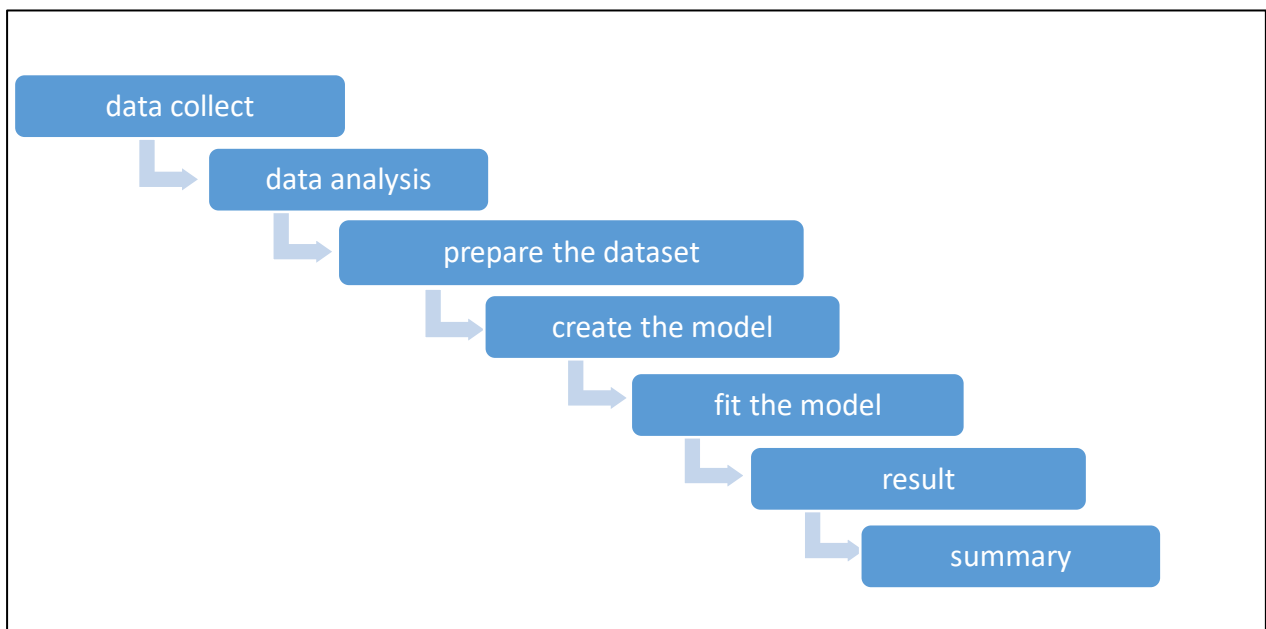


Figure 3.1: A Complete workflow

3.2 Form Preparation

Bangla characters and digits are completely different from any other languages. The reason maybe its origin is Sanskrit script which is totally different from English, Chinese and many languages available. Alignments and decorations of Bangla characters create a good challenge to researchers for its complexity. There are 50 basic characters which has 39 consonants and 11 vowels including 10 numerals and 10 modifiers. And the number of compound character is more than 300.

Due to this variety Bangla handwritten character recognition is still more challenging than other languages. If we can overcome these challenges and complexities than the application of Bangla handwritten character recognition would become more efficient and user friendly where new horizons of application may reveal. Figure 3.2 showing bangla characters with handwritten form.

Here are some examples of Bangla handwritten characters:

অ	আ	ই	ঈ	উ	ঊ	ঋ	এ	ঐ	ও	ঔ	ক	খ	গ	ঘ	ঙ	চ
ছ	জ	ঝ	ঞ	ট	ঠ	ড	ঢ	ণ	ত	থ	দ	ধ	ন	প	ফ	ব
ভ	ম	য	র	ল	শ	ষ	স	হ	ড়	ঢ়	য়	ৎ	ু	ূ	ৃ	ৄ

Figure 3.2: All Bangla Characters with Handwritten Form

Any handwriting recognition system badly needs a huge dataset for training purposes. Especially if we are talking about Neural networks and machine learning approach, the training set almost plays the lead role in recognition. For a better quality handwriting we have developed our own customized form. Which is inspired by the preprocessing procedures of Ekush dataset [8] and that will help us in processing the image and increase the recognition and boost the speed of the whole preprocessing step. Every form can collect 120 characters. The form is made up by keeping some tricky measurements in head. Such as the distance on the upper side or lower side of a cell was designed equal. Also it is necessary to print the letter under the cells as it helps a lot during processing of the image. One of the important keys was printing the forms rather than photocopying. Because the borders and the cells must be aligned perfectly. In case of contrast or copying issues the whole data of a form may become wasted.

The demo of the cell is given below:

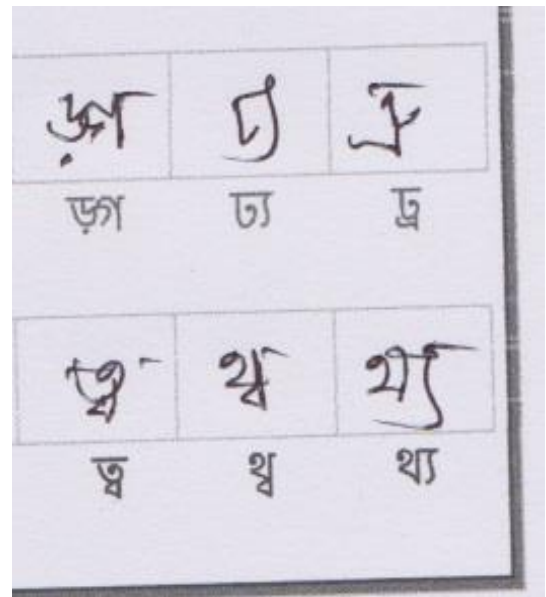
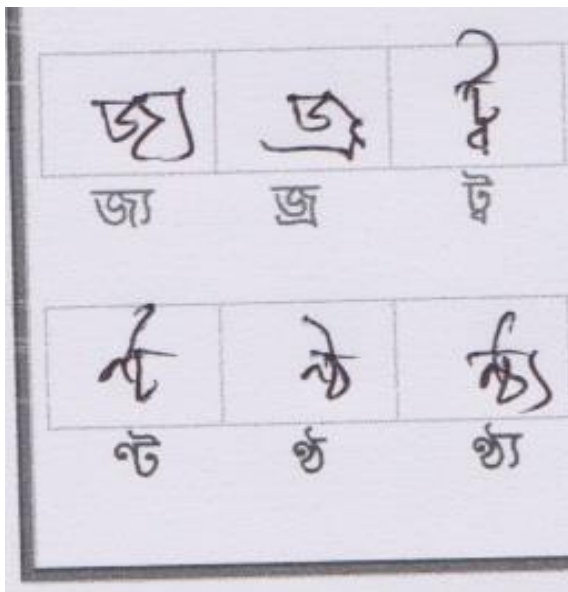
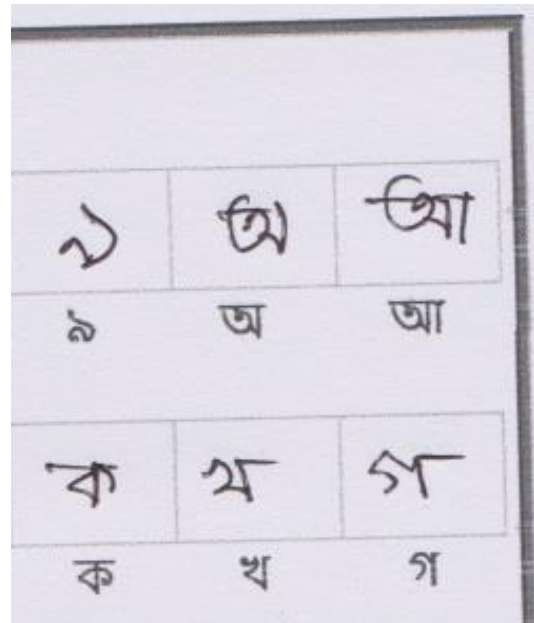
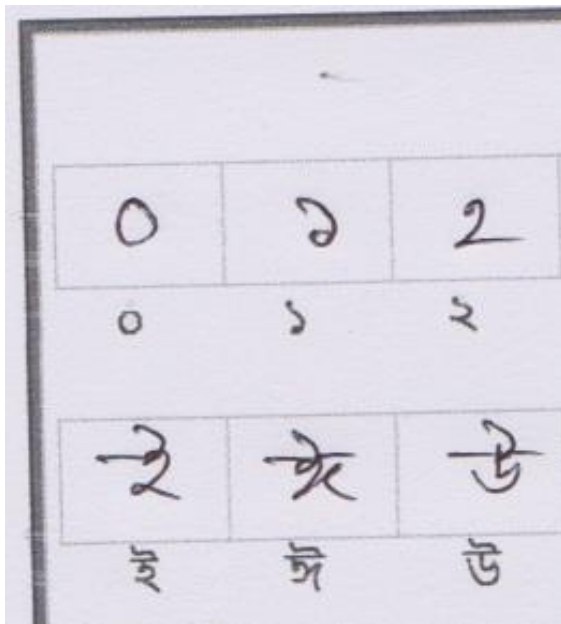


Figure 3.3: Edges and Boundaries of our Developed Form

Figure 3.3 showing the edges and boundaries of our developed form for image preprocessing.

Figure 3.4 showing the demo of the whole form:

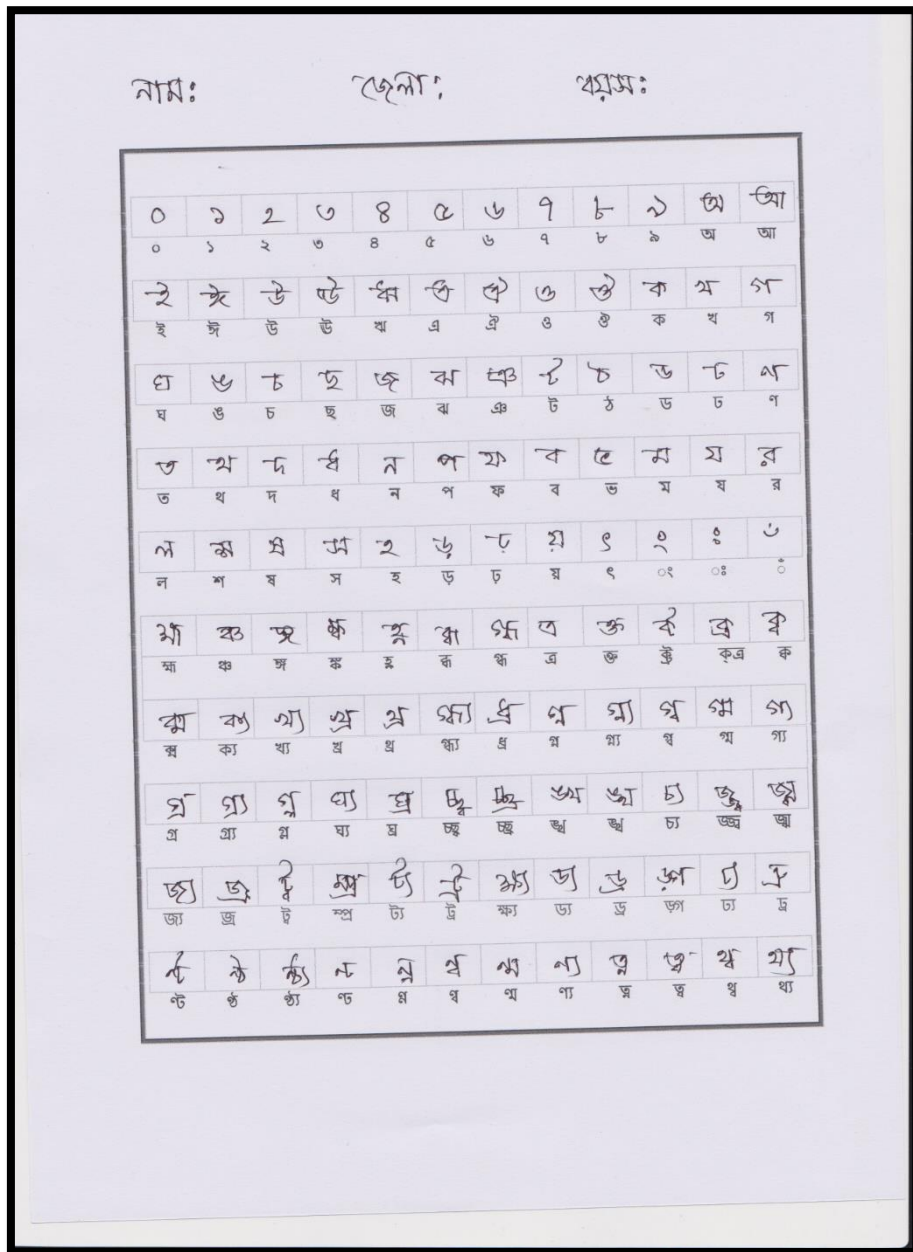


Figure 3.4: The Whole Form

3.3 Processing Handwritten Image

This step is about scanning the image to perform several operations necessary. To scan images any windows application that performs scanning job will do well but they take around 1 minute to scan a single form. There are some backlogs of windows application also. Such as most of the applications it does not provide saving multiple images at once.

For these short comings we have used the “Simple Scan” application on “Ubuntu” with 300dpi. Which takes 7-15 seconds for scanning a single form. Moreover, it also comes with some manipulating tools built in, such as cropping which helps us removing the extra unwanted black portion due to the scanner. Figure 3.5 showing the scanning process of handwritten forms.

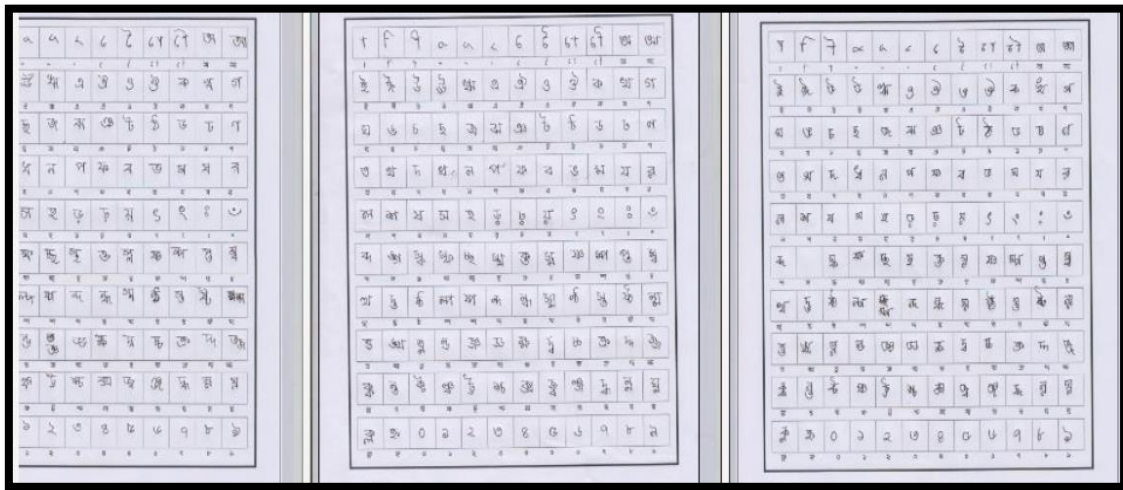


Figure 3.5: Scanning Handwritten Forms

3.4 Some Necessary Corrections

When scanning is complete then then we need to apply some primary operations on the images. Most of the time maximum of the images are not perfect for further processing. Unwanted space, Noise and other issues being detected is a common thing after scanning the images. Also it is not very wise to manually make them perfect for processing because it may take a huge amount of time. So we needed some form of algorithm that would do the job for us.

3.4.1 Grayscale Conversion

For many image process applications, color info doesn't facilitate determine vital edges or different options. There are exceptions If there's a border (a step amendment within the component value) within the tone that's tough to find in a very grayscale image, or if we'd like to spot the best-known tone objects (orange fruit against inexperienced leaves), then the colour info can be helpful. If we do not would like color, we are able to contemplate it as noise.

With the help of python OpenCV, at first we have taken the image from a directory and converted into a grayscale image. Figure 3.6 showing the effect of applying grayscale on the image.

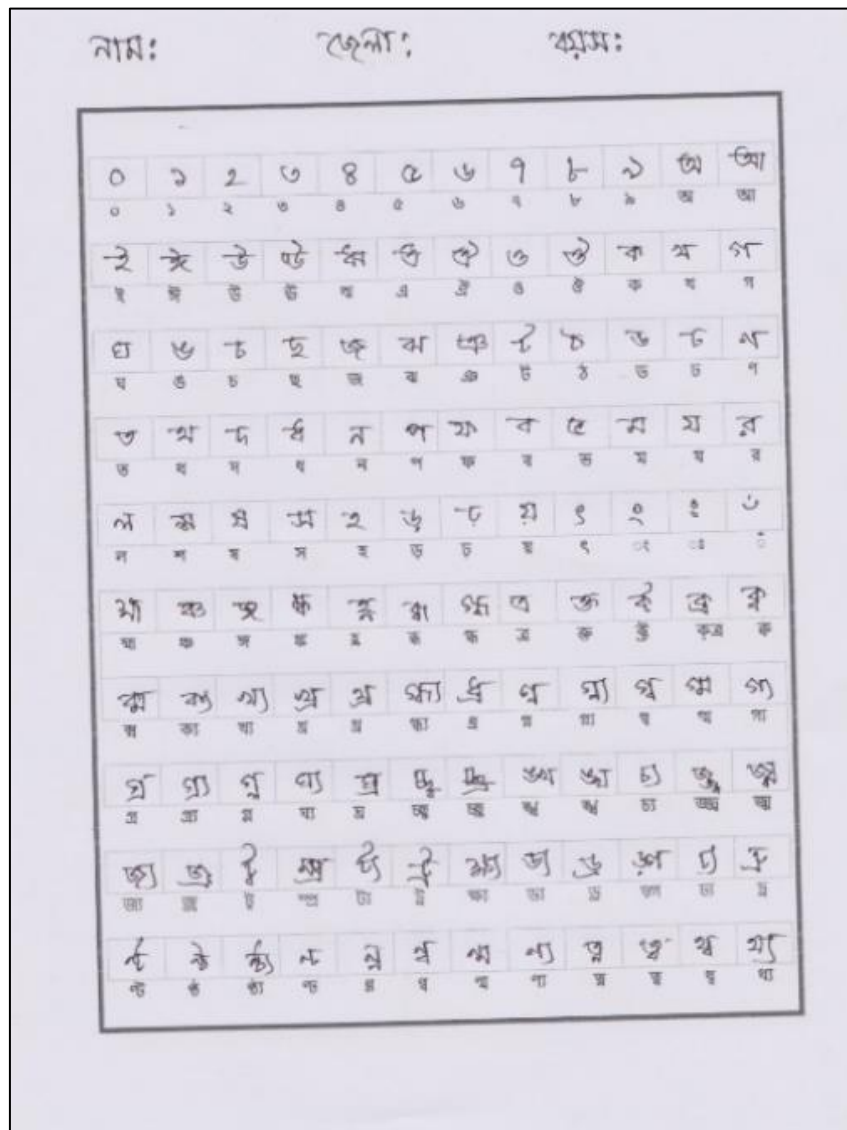


Figure 3.6 (a): Before Grayscale Conversion

3.5 Segmentation

Multiple functions, methodologies are applied in this section. Those are described below:

3.5.1 Detecting canny edge

Canny edge detection is used to remove noise from an image and it works tremendously well in the presence of Gaussian blurring. As we have used Gaussian blurring. Here is an example of our form when canny edge is used shown in figure 3.8:

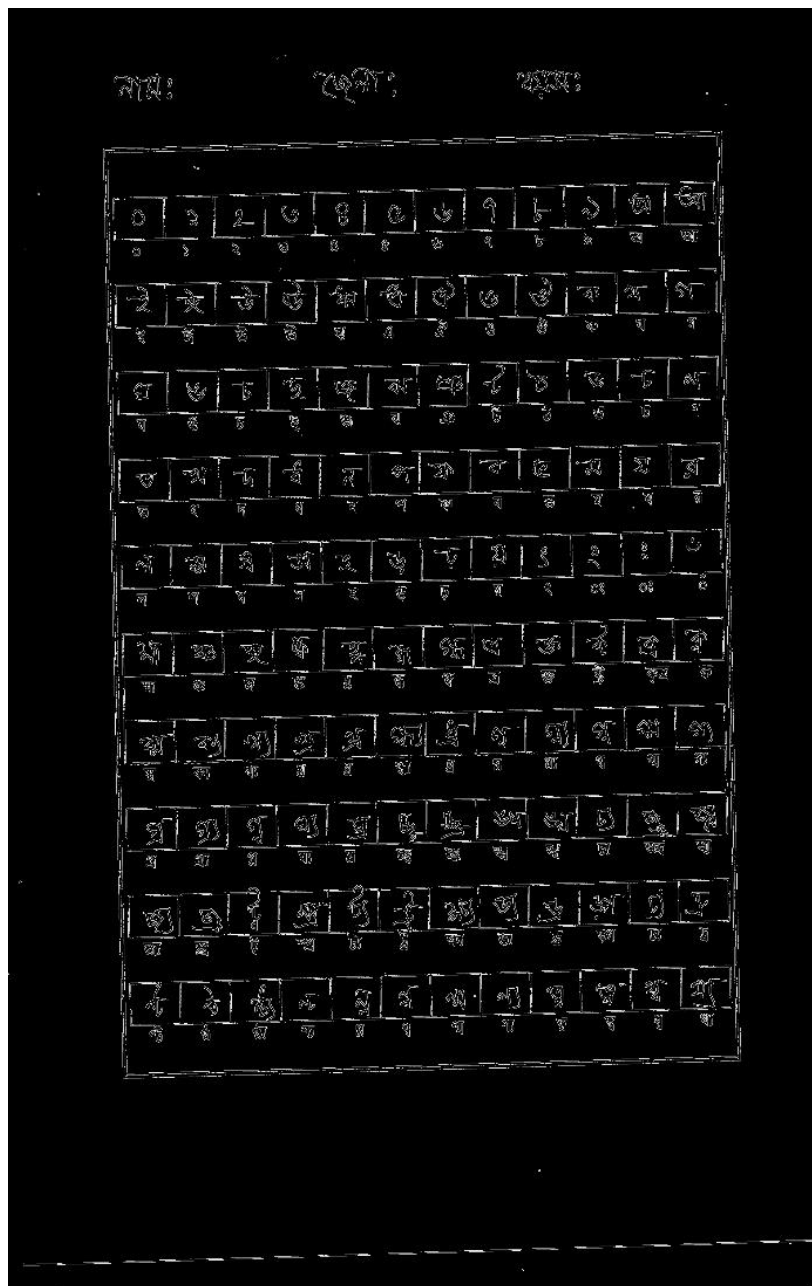


Figure 3.8: Canny Edge Detection

3.5.2 Manipulating the contours and crop

In order to getting the correct portion of the form which will contain the handwritten characters we must omit the unnecessary portion. This is the whole necessity of cropping. In other words, to get a single character out of the whole form this is the first step to be taken. The unnecessary portion may include some of the fields that is necessary for the volunteers. Such as there are fields like Age, Gender, Education etc. The white portion is also considered to be as an unnecessary portion of a form. The strategy of our cropping is finding the biggest contour and crop. Thus, we have designed the form in such manner so that it could perform well. However, the algorithm is explained at 3.6.2. Here is an example:

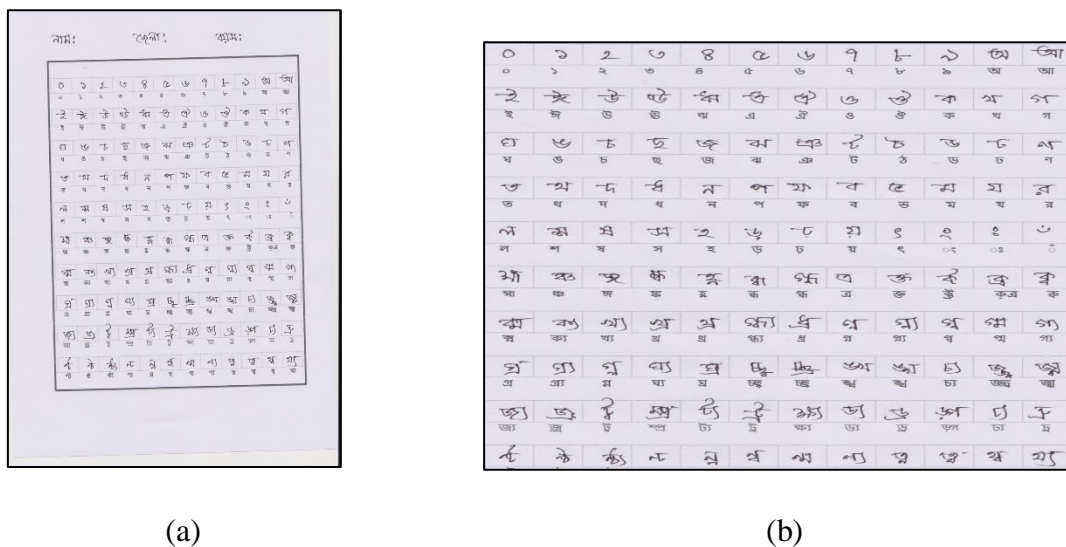


Figure 3.9: Contour Detection and Cropping

In figure 3.9 shows how to detect the biggest contour and cropped the image on that region.

3.5.3 Skewness and Rotation

Skewed image of the form is a common issue in image processing. However, to increase the quality of the processing we need images that are not skewed. Here is an example of an image that have been resolved from skewness. Rotation happens comparatively less than skewing. Basically, we rotate the image to make it correct for skewing. Figure 3.10 showing an example of skew correction.

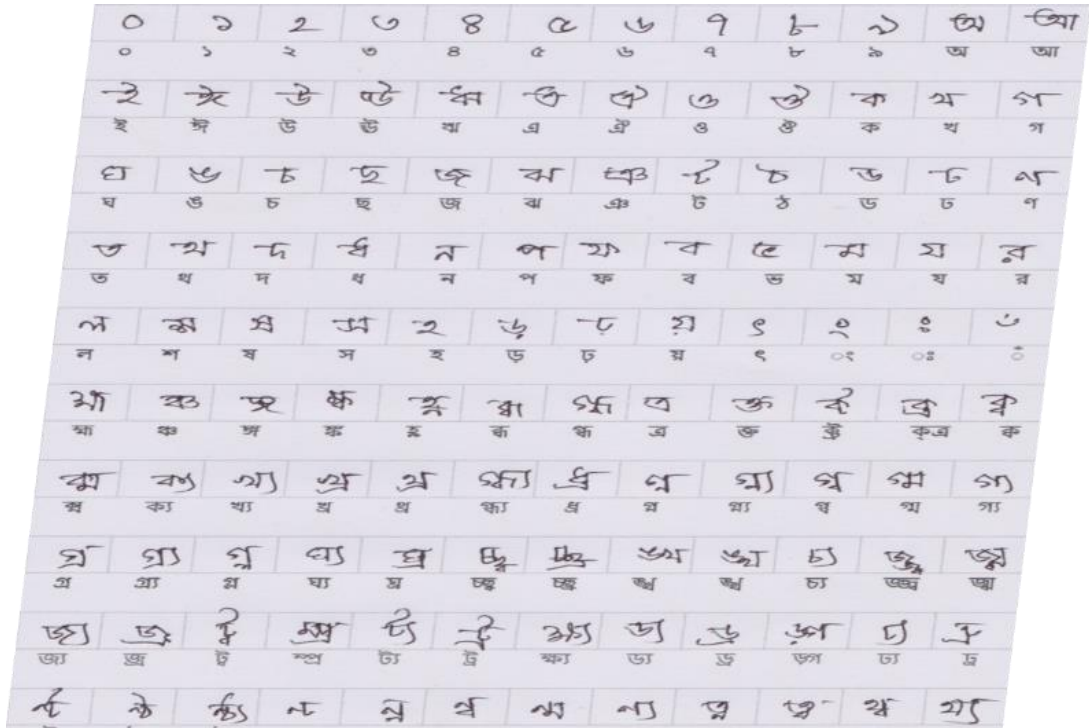


Figure 3.10(a): Before Skew Correction

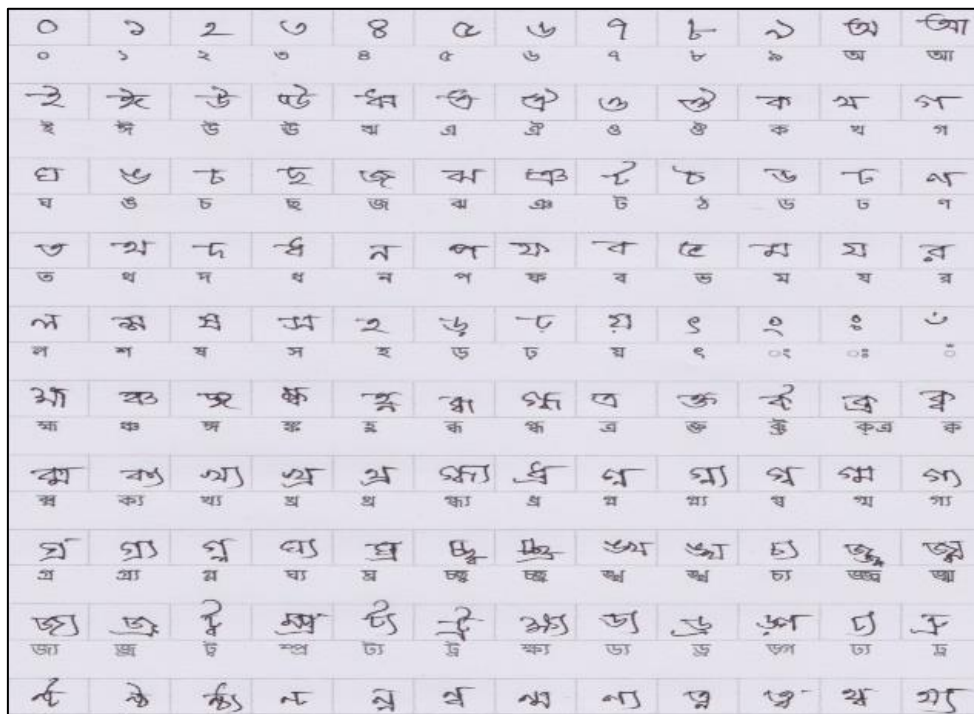


Figure 3.10(b): After Skew Correction

3.5.4 Resizing

Image is resized to get the best detail out of an image. Thus, any object or handwritten portion could be detected and manipulated easily. We resized the cropped image so it becomes more suitable for further processing which is shown in figure 3.11.

An example is given below:

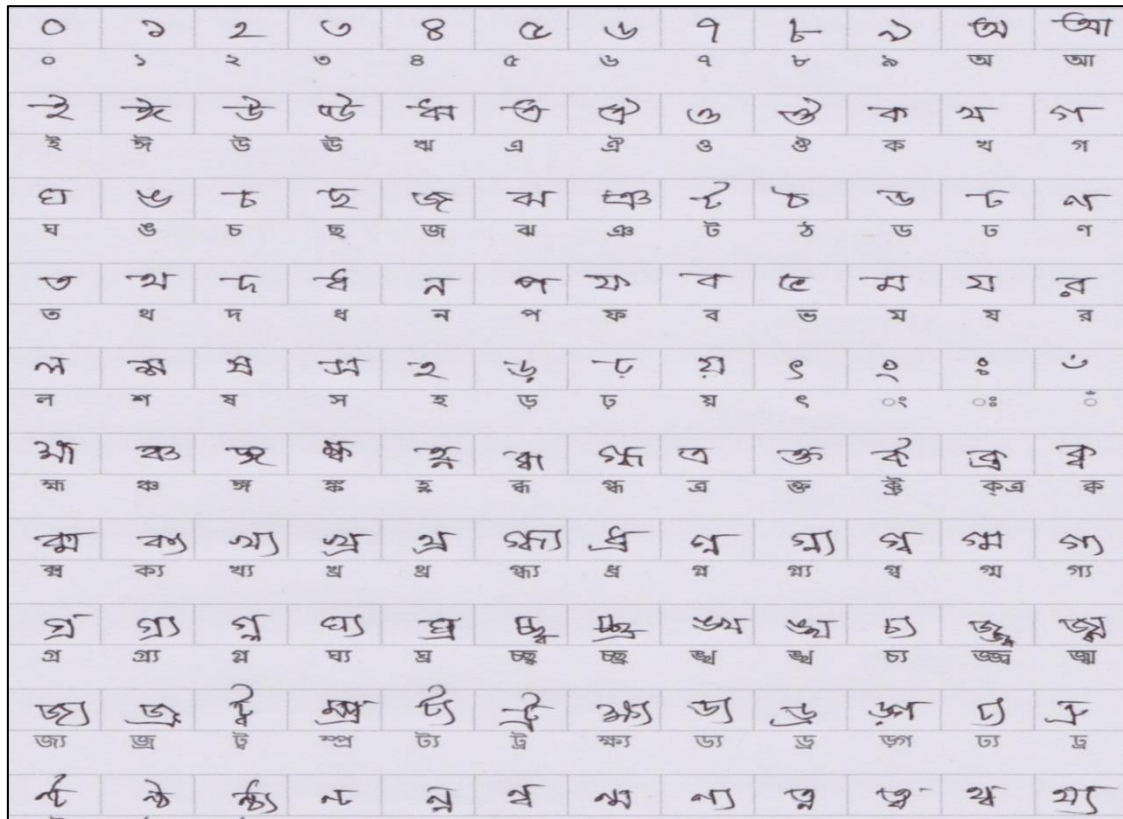


Figure 3.11: Resizing the Image

3.5.5 Row wise cropping

After we have resized the image now it is time for us to crop the image row wise. The algorithm at 6.8.1 describes it completely. The cropping starts with the start row to end row for all 10 rows in the form with OpenCv. The reason of doing so is our goal. Our goal is to segment a single character for training and testing. So in order to getting a single image it is preferable to crop and segment a single row first. Each row will be saved in a new directory then. Here is an example of cropping a row which is shown in figure 3.12.

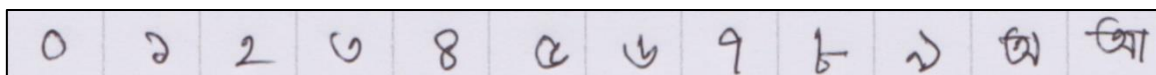


Figure 3.12: Row Wise Cropping

3.5.6 Column wise cropping

This is the step where we get a single handwritten character from an image. The image is cropped and saved in new directories for each character. Some examples are shown in figure 3.13.

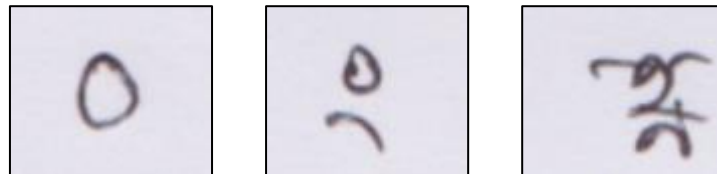


Figure 3.13: Column Wise Cropping

3.5.7 Noise Removal

In a scanned image, it is common to have some noise due to various reasons such as lack of scanner accuracy. Some black pixels of lower concentration in the position of white pixels of the images considered as noise. Sometimes it results in loss of significant information from the contents and as a result, the model fails to represent every details of an image. So highly effective filter is used to remove this noise from input image. And the image is passed through a smooth process to remove the noise within it.

Inverting makes the lighter portion dark and the darker portion white. So to remove noise and get the needed portion of the image and it is beneficiary. Here figure 3.14 showing an example:



Figure 3.14: Removal of Noises

Finally, when all possibly removable noises are omitted we converted the image into a binary image where foreground is black and background is white and added 1 pixel padding to each side of the binary image with python OpenCv. Figure 3.15 showing an example.



Figure 3.15: Removal of extra background

3.6 Algorithms

To get the perfect image and do all of the mentioned operations we have designed our own algorithms and model which are meant to provide images that are compatible for each other.

3.6.1 Algorithm to separate each character

1. *For all scanned image in the directory do:*
2. *Take an image and convert into grayscale*
3. *Blurred the image*
4. *Detect canny edge*
5. *Find all the contour*
6. *Sorted all the contour*
7. *For all sorted contour do:*
8. *Search the highest contour area*
9. *End for*
10. *Detect the highest contour in the image*
11. *Crop image on the detected contour area*
12. *Detect the 4 point of the cropped image*
13. *Compute the angle of the rotated image*
14. *Rotating the image to correct for the skew*
15. *Resize the image and save the image in new directory*
16. *Start_row = 0*
17. *End_row = Start_row + row height*

18. *RowFolder = 0*
19. *ColumnFolder = 0*
20. *For 10 times do:*
21. *Crop image in position (Start_row to End_row)*
22. *Save image in RowFolder*
23. *Start_column = 0*
24. *End_column = start column + column width*
25. *For 12 times do:*
26. *Crop rowimage in position (Start_column to End_column)*
27. *Start_column = End_column*
28. *End_column = Start_column + column width*
29. *Save image in ColumnFolder*
30. *ColumnFolder = ColumnFolder + 1*
31. *End For*
32. *Start_row = End_row + gap*
33. *End_row = Start_row + row height*
34. *RowFolder = RowFolder + 1*
35. *End For*
36. *End For*

3.6.2 Remove noise and extra background

1. *For all folder in column wise cropped directory do:*
2. *NewColumnFolder = 0*
3. *For 120 times do:*
4. *CF = 0*
5. *For all image in Folder CF do:*
6. *Convert image in binary*
7. *Remove noise from the image*
8. *Adding 1 pixel to each side of the image(padding)*
9. *Find where the black pixels are*
10. *Create a rectangle around those points*
11. *Cropped that image on that region*
12. *Save image in NewColumnFolder*

13. $NewColumnFolder = NewColumnFolder + 1$
14. *End For*
15. $CF = CF+1$
16. *End For*
17. *End For*

3.7 Recognition using CNN

3.7.1 Convolutional Neural Network:

Convolutional neural networks live associate architecturally totally different approach to the dimensioned and ordered data of the method. rather than submitting that the position of the knowledge inside the item is questionable (as do the completely connected layers), the convolutional and severe levels of body harm grouping impose the shared weight on the interpretation. This model the approach within which the human region works, and has shown to figure unlikely for perception and also the kind of different tasks. we are going to learn convolutional networks through associate recent descent of a random gradient.

Convolutional Neural Networks (CNN) as a fairly specialized neural network for method data that includes a special correlation between the knowledge points of the neighborhood that's additionally called grid topology. The samples of data, info incorporate statistic information, which might be thought-about as a 1D grid that takes samples at regular time intervals and also the data of the pictures, which might be thought-about as a second grid of pixels. Convolutional nets are implausibly safe in sensitive applications. The name "convolutional neural network" indicates that the network employs a calculation known as convolution.

3.7.2 Proposed Model

The proposed model is a 10-layer convolutional neural network. Proposed model use stochastic gradient descent algorithm optimizer. For first two-layer same padding and ReLU activation used with 32 filters with the 5x5 kernel. Then a max-pooling layer added with a 2x2 followed by 25% dropout layer. All dropout layer used to reduce overfitting. Figure 3.16 showing our proposed model.

$$ReLU(X) = \text{Max}(0, X) \dots \dots \dots (1)$$

3.7.3 Diagram of CNN:

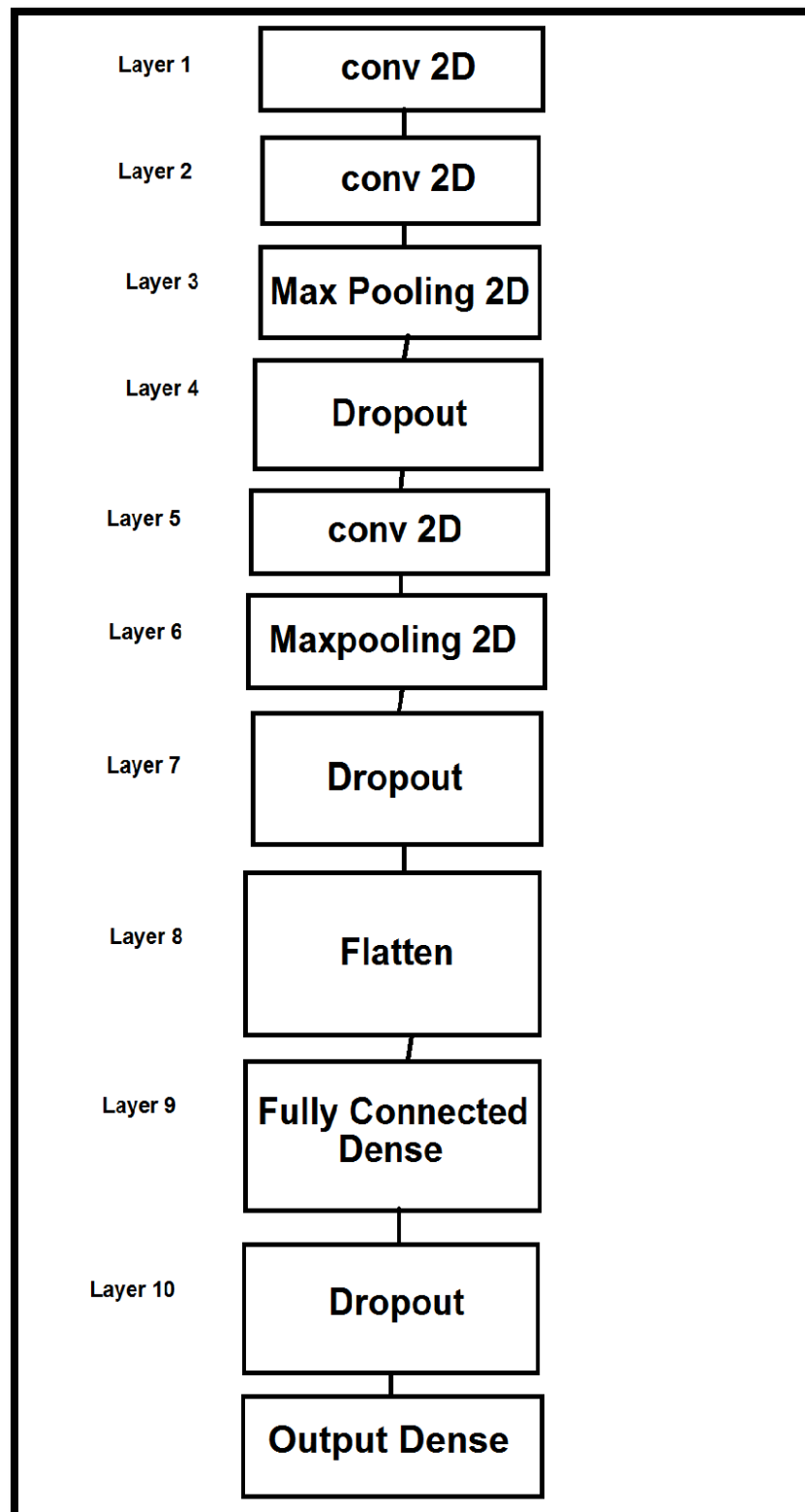


Figure 3.16: Layered Diagram of Proposed Model

3.7.4 Details of the Neural Network

This table shows the layer wise description of our model.

Table 4: Layer wise description

Layer No (type)	Output Shape	Parameter
Input Layer	28,28,1	0
Conv 2D	28,28,32	832
Conv 2D	28,28,32	25632
Maxpooling 2D	14,14,32	0
Dropout	14,14,32	0
Conv 2D	14,14,64	51264
Maxpooling 2D	7,7,64	0
Flatten	3136	0
Dense	1280	4015360
Dropout	1280	0
Output(Dense)	10	12810

Here,

Total parameters=4105898

Trainable Parameters=4105898

Non-trainable parameters=0

3.7.5 Gradient Descent Algorithm:

Gradient descent is an iterative algorithm that gradually adjusts the honorable parameters W , B to better reflect the training data by iteratively rendering the cost function. The reason of using gradient descent algorithm is as follows:

- Initiate the parameters w , b randomly
- Compute the gradient descent which points in the direction of biggest cost increase and update the parameters w and b by taking a small step in the opposite direction. The step size α is called learning rate.
- Repeat with the new parameters w and b .

3.7.6 Max-Pooling Layers

After every convolutional layer, there is also a grouping layer. The grouping layer takes little rectangular blocks of the convolutional layer and takes the subsamples to provide one output of that block. There are many ways in which to try and do this grouping, like taking the typical or most, or a linear combination of neurons within the block. Our grouping levels can forever be the utmost grouping levels; that's, they take the utmost of the block they're gathering.

As noted higher than, most grouping levels don't perform any learning severally. Instead, cut back the scale of the matter by introducing dispersion. In forward propagation, the $k \times k$ blocks are reduced to one worth. Thus, this distinctive worth acquires a calculated error of backward propagation from the previous level. This error is solely forwarded to the place wherever it came from. Since it comes solely from a degree within the $k \times k$ block, the inverse propagation errors of the utmost grouping levels are quite rare.

3.7.7 Dataset:

We have used our own collected data which is around 1320 in number. Also we have used NumtaDB[9] dataset to check how the model performs on other datasets for digit recognition. But we have also tried a mixed dataset and that helped us achieve a great percentage of accuracy.

CHAPTER 4

EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Experimental Results:

The convolutional neural network worked out well on different datasets. It has shown good accuracy on numerous datasets. The results are given below:

Table 5: Accuracies of our model

Dataset	Accuracy
Bornocchota (own dataset)	96.2%
Numta DB	91.41%
Mixed	90.1%

4.2 Discussion:

On our dataset there was 1320 data and these handwritten images were preprocessed and fed to the convolutional neural network on the input layer. After each layer weight and bias is added. Besides the gradient descent algorithm optimizes the values and reshapes it over and over so that the optimum accuracy can be achieved. Bornocchota provided the optimum accuracy of 96.2% that shown in figure 4.1 and other data sets shown remarkable results. NumtaDB has given the accuracy of 91.41% which is based on bangla numeral digits shown in figure 4.2. The visual results on all the Datasets are given below:

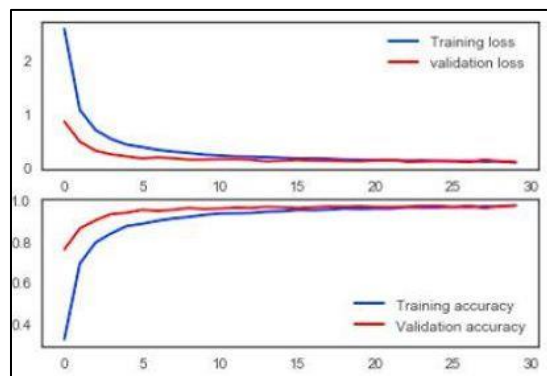


Figure 4.1: Accuracy on Bornocchota

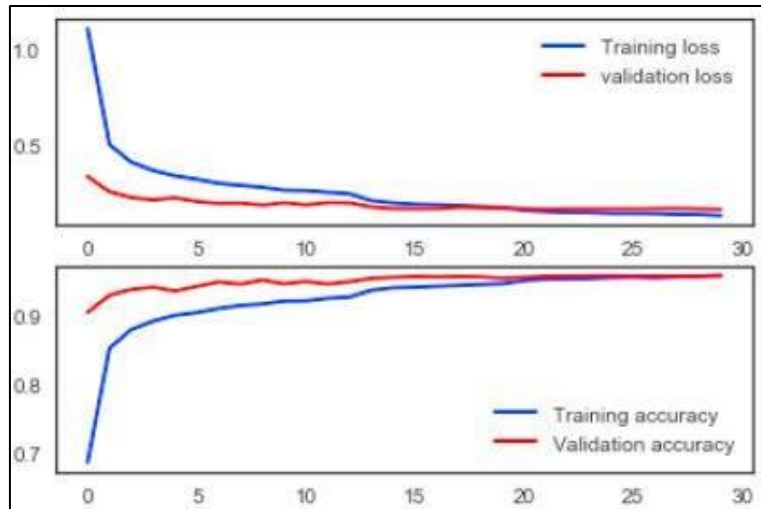


Figure 4.2: Accuracy of NumtaDB

NumtaDB has given the accuracy of 91.41% which is based on bangla numeral digits shown in figure 4.2.

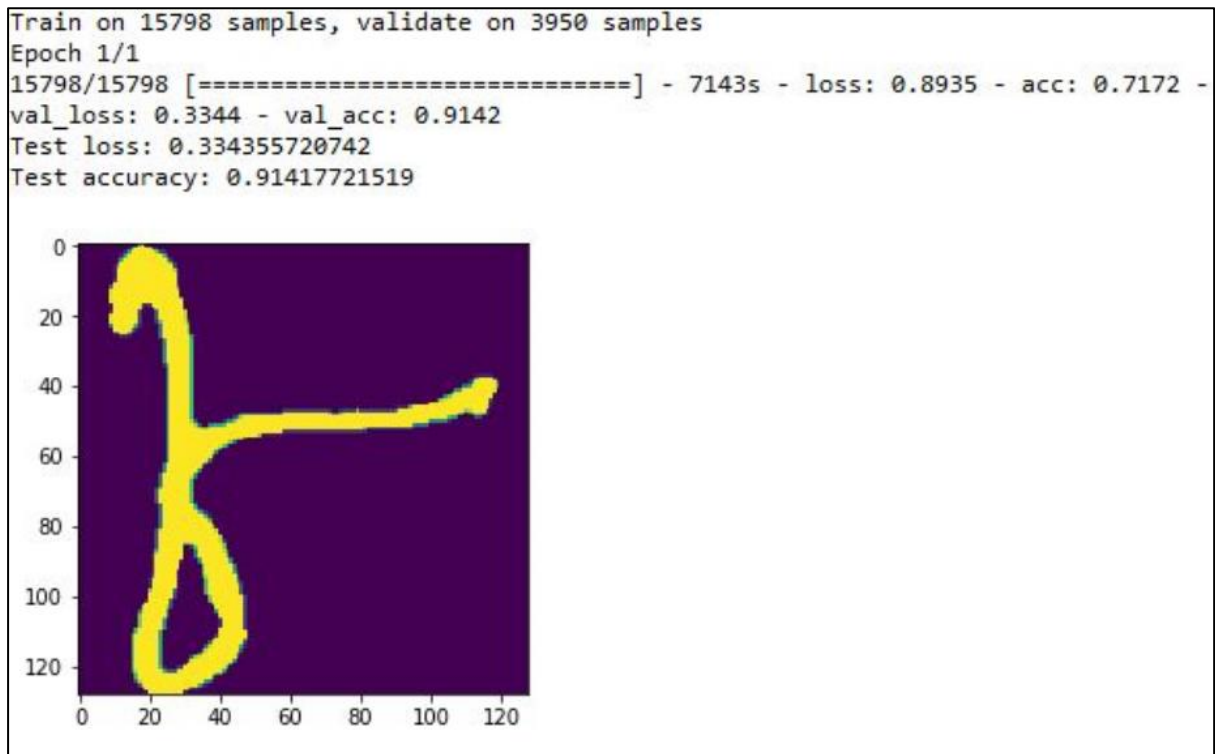


Figure 4.3: Accuracy on mixed Dataset

CHAPTER 5

SUMMARY, CONCLUSION AND IMPLICATION FOR FUTURE RESEARCH

5.1 Summary

Optical Character Recognition (OCR) is a process of converting scanned document into text document so it can be easily edited if needed and becomes searchable. OCR is the mechanical or electronic translation of images of handwritten or printed text into machine-editable text.

So in our thesis firstly we collect data and processing all the dataset. Then we segmented all the data for prepare data properly for our model.

Moreover, if the image is not straight then this segmentation technique can give us wrong outputs. So first calculate the angle of deviation of the image from the straight image and rotate the image and make it straight. This is done in the preprocessing of the technique.

In our model, at first we have tried with input and output layer which is 10- layer convolutional neural network is fully connected structures. Proposed model use stochastic gradient descent algorithm optimizer. For first two-layer same padding and ReLU activation used with 32 filters with the 5x5 kernel. Then a max-pooling layer added with a 2x2 followed by 25% dropout layer. All dropout layer used to reduce overfitting. Also we use gradient descent algorithm, max pooling.

So from our model we can see the value of recognition which gives us the best accuracy for our bangla character. We verifies the results by seeing automatically generated graphs. The graphs were generated by taking the accuracy values and loss function values on every iteration.

After using this function along with Numtadb dataset we able to get accuracy 91.41%. After that 90.1% accuracy get by the mix dataset. Finally, we were able to get 96.2% accuracy which is for our 1320 data.

5.2 Conclusion

In our thesis we have tried to make a complete Optical Character Recognition for Bangla text for different Bangla font. Our motivation is to recognition bangla character with a large database to enrich bangla character in computer vision. The output of our system will be the text file of the image. It could be very useful in digitizing all the writing page or document. By using the OCR we can get the soft document of this data so that we can store this data an use it when it is needed or for editable version. Day by day it will be make a large dataset and the accuracy will be highest accuracy for all the bangla handwritten character.

So this research introduced lead to formulating a universal way for processing bangla languages handwritten character database. That database is possible to develop handwritten character recognition systems for bangla language. Applying these propose method, researchers will be benefited in different research fields.

5.3 Implication for Further Research

We will try to extract letters from individual words as well as sentences in future. In Bengali all the letters in words are connected with “matra” which makes the task hard. Also, we would like to increase the number of dataset to include more complex Bangla characters, with word structures that includes basic and compound characters combined within single characters. We will also improve our CNN even more.

REFERENCES

- [1] L. F. C. Pessoa and P. Maragos, Neural networks with hybrid morphological/rank/linear nodes: a unifying framework with applications to handwritten character recognition, *Pattern Recognition*, vol. 33, no. 6, pp. 945-960, June 2000
- [2] Bellili, M. Gilloux and P. Gallinari, An MLP-SVM combination architecture for offline handwritten digit recognition, *Document Analysis and Recognition*, Springer-Verlag, vol. 5, no. 4, pp. 244-252, 2003.
- [3] Ekush: A Multipurpose and Multitype Comprehensive Database for Online Off-line Bangla Handwritten Characters
- [4] J. Hull, "A database for handwritten text recognition research", *IEEE Trans. on PAMI*, 16(5), 1994, pp.550–554.
- [5] Biswas, Mithun & Islam, Rafiqul & Kumar Shom, Gautam & Shopon, Md & Mohammed, Nabeel & Momen, Sifat & Abedin, Anowarul. (2017). BanglaLekha-Isolated: A multi-purpose comprehensive dataset of Handwritten Bangla Isolated characters. *Data in Brief*. 12. 103-107. 10.1016AC/j.
- [6] T.K. Bhowmik, P. Ghanty, A.Roy and S.K. Parui, SVM based hierarchical architecture for handwritten Bangla character recognition, *International Journal on Document Analysis and Recognition*, vol. 12, no. 2, pp. 97-108, 2009
- [7] BornoNet: Bangla Handwritten Characters Recognition Using Convolutional Neural Network AKM Shahariar Azad Rabby, Sadeka Haquea, Md. Sanzidul Islama, Sheikh Abujar and Syed Akhter Hossain
- [8] A Universal Way to Collect and Process Handwritten Data for Any Language AKM Shahariar Azad Rabby, Sadeka Haque, Shammi Akther Shahinoor, Sheikh Abujar and Akhter Hossain
- [9] Alam, Samiul & Reasat, Tahsin & Mohammad Doha, Rashed & Humayun, Ahmed. (2018). NumtaDB - Assembled Bengali Handwritten Digits.
- [10] Asish Kumar, Ghosh Shyla Afroge, "A Comparison between Support Vector Machine (SVM) and Bootstrap Aggregating Technique for Recognizing Bangla Handwritten Characters"
- [11] Shyla Afroge, Boshir Ahmed, Ali Hossain," Bangla Optical Character Recognition through Segmentation using Curvature Distance and Multilayer Perceptron algorithm"
- [12] Nazmul Arefin, Mahamodul Hassan, Md. Khaliluzzaman, Shayhan Ameen Chowdhury, "Bangla Handwritten Characters Recognition by Using Distance-Based Segmentation and Histogram Oriented Gradients"
- [13] N. Das, S. Basu, R. Sarkar, M. Kundu, M. Nasipuri et al., "An Improved feature descriptor for recognition of handwritten bangla alphabet".
- [14] Mujadded Al Rabbani Alif, Sabbir Ahmed, Muhammad Abul Hasan, "Isolated Bangla Handwritten Character Recognition with Convolutional Neural Network"

APPENDICES

Appendix A: Research Reflection

This topic of Bangla Hand written character recognition using convolutional neural network was an enjoyable one. We have learnt many aspects of machine learning as well as deep learning. On the other hand, we had to learn a new programming language which is “python” to implement the model and process handwritten data. It was a delightful experience. Various Machine Learning libraries and algorithms is used and we had to learn them. We have gone through tons of papers and book written by authors worldwide. Which improved our depth of knowledge in Machine learning and Deep Learning.

Appendix B: Plagiarism Report

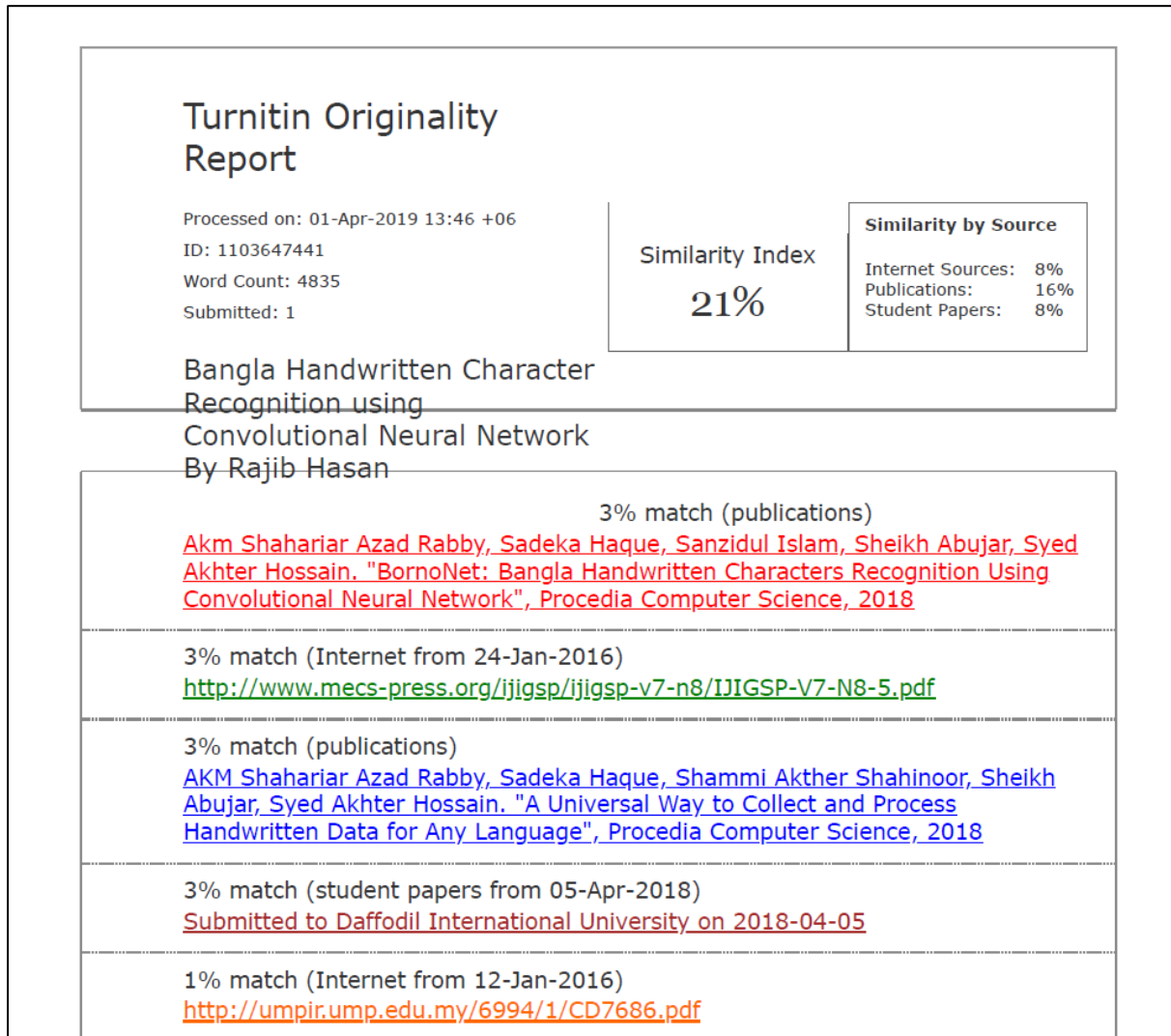


Figure A1: Plagiarism Report