An End-to-End Hybrid Architecture for Bangla OCR

Shahrear Bin Amin
Exam Roll No.: Curzon Hall-514
Registration No.: 2015-016-816

Asif Zaman
Exam Roll No.: Curzon Hall-519
Registration No.: 2015-516-776

A thesis submitted for the degree of Bachelors of Science at University of Dhaka

Department of Computer Science and Engineering
Faculty of Engineering

December 30, 2019

Declaration

We, hereby, testify that the work presented here in this report is the result of our study under the supervision of Md. Mahmudur Rahman, Lecturer, Department of Computer Science and Engineering, University of Dhaka. This project is not subsided by any external organization and neither is there an active collaboration on our part with other researchers or groups.

Candidate 1
Shahrear Bin Amin
Registration No: 2015-016-816
Class Roll: FH-055

Candidate 2
Asif Zaman
Registration No: 2015-516-776
Class Roll: SH-015

Supervisor
Md Mahmudur Rahman
Lecturer
Computer Science and Engineering
University of Dhaka

Abstract

This work illustrates a novel approach towards Optical Character Recognition (OCR) of Bangla printed documents. Bangla and other related Indic scripts provide ample hardship in building a full-fledged OCR engine owing to their overlap between characters, cursiveness, variable height of characters and so on. A Clustering algorithm has been used instead of traditional approaches for segmenting lines and then a hybrid architecture is applied for training on the lines with character encoding. Our proposed model has Convolutional Neural Network (CNN) layers to extract features from the input line image. On top of the convolutional network, a Recurrent Neural Network (RNN) is built for making prediction for each frame of the feature sequence, outputted by the convolutional layers. The Bidirectional Long Short Term Memory (BLSTM) outputs a matrix which contains a probability distribution over the characters at each image position. Decoding this matrix yields the final text which is done by the connectionist temporal classification (CTC) operation. Though our model consists of several networks, we trained jointly on one loss function. Compared with previous systems for Bangla text recognition, the proposed architecture possesses two distinctive properties: (1) it is end-to-end trainable, in contrast to most of the existing algorithms whose OCR components includes error prone segmentation process. (2) Our model can be used for handwritten Bangla text recognition given sufficient dataset.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

OCR or Optical Character Recognition is a technology that makes it possible for one to convert various types of document images and scanned papers into searchable and editable documents.

1.1 Motivation

The usage of OCR has a vast landscape. Development of various natural language processing (NLP) tools such as lemmatizer, stemmer, spell checker, dependency parser and other tools require large amount of data which can be obtained with less effort using a reliable OCR running through a well-made dataset. OCR has several usages such as converting books, newspaper and documents into text files, automation of record-keeping in office or publishing any text on websites. Thus OCR has a great contribution to the advancement of automation processes and capable of improving interface between man and machine in several practical purposes. In case of Bangla language, the literature is quite ancient and resourceful. Also lots of books, manuscripts are already available as hard copies. Hence, converting them to searchable and editable portable documents will provide to us data for conducting further research related to Bangla language.

As mentioned earlier, Bangla language has a really vast and rich literature collection and they are required to be conserved in digitized format. But doing this work by hand is tedious and time consuming. OCR can be a feasible solution for resolving the issue. There is a large collection of necessary documents piling up in the government and non-government offices in Bangladesh which is both handwritten and printed. So digitizing existing important government and other official documents will allow for developing a more efficient and automated system with reduced wasted, tiresome and monotonous working efforts.

All past OCR had four steps, preprocessing, segmentation, recognition and post-processing. Cursiveness, conjuncts character, overlap between characters makes segmentation process hard. Furthermore, there is no work in Deep learning based. We want to conduct some research on several datasets as reliable and consistent data for Bangla is necessary for our research as well as future research. Moreover, we will try to accomplish this task for fonts used in different time frame and scripts of different styles and manners.

1.2 Problem Formulation

Many researchers have worked on different components of Bangla OCR, namely preprocessing of the text image, segmentation of lines, words and characters, and recognition of segmented characters. Challenging part of OCR is character segmentation form word. For viable character segmentation, touching character makes major issue. Segmentation of compound character is very difficult. Many researchers have worked on segmentation, but did not get expected result because of cursive style of Bangla script. We mainly focus on improving segmentation of only lines from the page in printed Bangla character image. Also, we will analyze several datasets for developing a robust dataset for our present research and future work as well.

Although many researches had conducted research involving segmentation process, we found it necessary for introducing a new research in segmentation process.But segmenting whole text page into text lines, words and finally characters is a time and memory consuming process. Rather than segmenting into atomic components we focus on making our model learn to recognize by text lines.

- The line segmentation process can be formulated as a clustering problem where every line should be considered as cluster. Thus,the entire line will be taken as the unit of recognition rather than more fragmentary elements.
- The next phase can be defined as a mapping problem where the label is encoded according to horizontal character position in line image.

- . The Segmented lines and corresponding encoded labels will be used for building a classification model.

1.3 Research Objectives

We propose developing an OCR segmenter which will mainly focus on resolving some common cases occurring frequently or causing major performance issues through the papers.

- Functionality for handling heterogeneous font size as well as superscript and subscripts. Although the previous works done so far are mostly font-invariant, our approach should be versatile for hybrid page layout with texts, images, tables, etc. as well as various font, size and document layout.
- Various combination of overlapping characters, vowel modifiers, upper modifiers and lower modifiers which causes segmentation error will be handled by the learning process. This additionally includes adjoint characters as well as reassembling modifiers with wrapping parts.
- Rather than relying on deterministic features like header line, baseline which occasionally falters in unusual scenarios, we will attempt to develop an unsupervised learning process for segmentation based on several characteristics of Bangla script.

1.4 Contributions

The contributions of our research work are listed at the end.

- Our model is end-to-end trainable, in contrast to most of the existing algorithms whose OCR components includes error prone segmentation process.
- It can be used for handwritten Bangla text recognition with the sufficient dataset.
- An unsupervised learning procedure is applied exclusively on line segmentation process. OPTICS algorithm is used to cluster each line of a text image. This modified algorithm is inspired from pastor [35].
- Every single characters along with singular characters with vertical modifier and compound characters with various lengths are encoded as a unique label.

- As visual ordering of string and actual ordering of string is different in Bangla we have encoded groundtruth text and got better result than unencoded groundtruth.
- We studied and analysed the ISIDDI [3] (Indian Statistical Institute Degraded Document Image) dataset. As the documented pages are degraded and damaged, we applied variable image manipulation techniques for noise free and consistent data. Different lingual and graphical characteristics of Bangla is studied.

1.5 Organization

We have organized this report into the following chapters:

- Literature Review and Background Study includes an overview of the associated terminology and components of image processing, text encoding and machine learning this discussion is followed by a description of related literature including both prominent and recent works in the field of OCR for Bangla. Also, there are definition of the principle problems and drawbacks of other approaches we intend to work out. An Overview of the Bangla script describes the primary features and characteristics of the Bangla script. Also, a brief overview of evolution and history of the Bangla script are described.
- Proposed Approach contains an elaborate description of our work flow in the research and details regarding our proposed methodology.
- Experimental Results provide a set of test images and recognized output on them. Also, visualization about various factors are provided.

Chapter 2

Literature Review and Background Study

2.1 Introduction

In this section, we are going to look into some of the terms used throughout this book and the basic components of an OCR engine. Then we will look into some of the recent and prominent works done in the field of OCR for both Bangla and other languages.

2.2 Terminology

- **Optical Character Recognition (OCR):** Optical Character Recognition is the system for identifying characters of a certain language in unicode or some other editable format from an image of text.

- **Binarization:** Binarization of an image is the process of converting the image into binary format so that each pixel only holds one of two color values, either black or white. This makes processing an image easier since the image no longer consists of multiple levels of colors. Binarization requires finding a threshold value based on the whole image. If a pixel value is greater than the threshold value, then it is replaced by the high value, and if it is lower, it is replaced by the low value.

- **CUDA:** CUDA is invented by NVIDIA as a parallel computing and programming model. It utilizes the tremendous processing power of the Graphical Processing Unit (GPU) thus increasing the total available computational power [2].

- **Otsu Thresholding:** In image processing, Otsu's method, named after Nobuyuki Otsu, is used to perform automated clustering-based image thresholding as well as the reduction of a gray level image to a binary image. [6] We use Otsu's threshold in our project for the purpose of binarizing text images.

- **Shirorekha :** It is the top header line visually recognizable in a number of Indie languages including Devanagari, Bangla and Tibetan. According to one author's description of an Indie script, "The letters looked like clothes hung out to dry on a line and they looked more like musical notation than writing"[32]- This line is the Shirorekha or the Matra.

- **OpenCV :** OpenCV is a programming library built principally for real-time computer vision. It is known for its cross-platform performance, portability and apprehensibility. [5]

- **TensorFlow :** TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications. [10]

- **Numpy:** NumPy is the fundamental package for scientific computing with Python. Numpy is a N-dimensional array object which is a container for a grayscale image as a 2D array or matrix or rgb image as a 3D array. It has also useful linear algebra, Fourier transform, and random number capabilities.[4]

2.3 Related Works

We have studied several papers related to segmentation of both printed and hand written scripts of Bangla and other Indo-European languages. The Comparative analysis is performed among these papers by answering some vital research questions. The research question along with approach of different papers are described below

2.3.1 Literature Review

Preprocessing techniques adopted by researchers
After getting an image from documents the image contains various types of noise due to machine error. Sometimes scanned doesn't have an absolute white background, lack of contrast. Therefore, noise removal techniques are applied to the images. There are many noise removal techniques such as filtering, smoothing, etc.

Chaudhuri *et al.* [23] used Low pass filter for removing noise. Images can be tilted to one side at the time of image acquisition. Using Hough transform on the pixels the skew angle is detected. Then Otsu's binarization method was used by researchers.

Sarkar *et al.* [37] proposed method for complete segmentation. Initially isolated bangla word images were collected from various types of scripts printed in various fonts and sizes. Then image is scanned in 300dpi resolution. For binarization adaptive thresholding technique was chosen. The average of the highest and lowest gray level values in each document image was chosen as threshold. A sequence of erosion and dilation was used to abolish a noisy pixel and to smooth the contours of data, on the input printed word images.

Manab *et al.* [30] applied Otsu's thresholding. Then removed noise by applying adaptive Gaussian filter. Then used horizontal projection profile for detecting skew angle and rotated the image accordingly.

Chaudhury *et al.* [24] proposed a method for segmenting lower zone modifier. Indian governments "Million Book Project" stores thousands of old book written in Devnagari & Bengali. Authors objective was to make those books digitize for Digital Library of India (DLI). Traditional segmentation approach is not applicable here because most of the books are old and modifiers are not exactly the same as today we use. Books in DLI had non uniform skew because they had different font. First image was binarized, noise removed, skew corrected, text area identified.

Zahan *et al.* [42] scanned image of text at 100 to 300 dpi resolution, then noises are removed with bilateral filters. Otsu's state of the art technique was used for binarization and skew was corrected.

Gupta *et al.* [27] performed two phase preprocessing. First Phase binarizes the word image using Otsu's method. In later phase, the word is thinned to one pixel width.

Bhowmik *et al.* [21] performed preprocessing in several steps. The first step is to find the width of writing which is determined by row- and column-wise scanning. The frequency distribution of the run-lengths is considered for where run-length occurring most frequently is determined as thickness of writing. Next, writing zones are detected from the binary image by analyzing horizontal projection profile. This process divides the word into three horizontal zone which are upper, middle and lower zone. Then matra region is picked from the upper zone. For the final step of preprocessing significant contour zones are selected based on significant value of a region. The value is compared with a certain threshold to decide if that contour region is significant or not.

Khan *et al.* [28] binarized word image using Otsu's method.

Challenges of segmentation and segmentation techniques adopted for those challenges

A Bangla word can be spliced horizontally into three zones. The portion on and above the header line is identified as the 'upper zone'. The main body of the characters in a word below the header line lies in 'middle zone' and the zone containing lower modifiers are the 'lower zone.

Chaudhuri *et al.* [23] used a well-known technique for detection of matra using the row histogram. The row with the highest histogram value is considered as matra and it is removed. Authors used bell shaped fuzzy function for detecting upper zone of matra. It is a challenging task to separate the overlapped characters. Here researchers used piecewise linear scanning to separate connected components.

Sarkar *et al.* [37][38] segmented word into four regions, unlike other authors. The top row of the upper area, middle area, the middle row of the middle area, the bottom row of the middle area and the lower area was identified in each word image as R_1, R_2, R_3, R_4 and R_5. First image was scanned row-wise form top and marked first black pixel as R_1, scanning from bottom first black pixel gives R5. Image was scanned from R_1 to $\mathrm{R}_{\mathrm{HALF}}$ where $\mathrm{R}_{\mathrm{HALF}}$ is $\frac{(R_1+R_5)}{2}$, then black pixels longest run was calculated. And row having maximum sum of pixel is added in R2. Sum of all transition point between foreground and background pixels for all rows from $\frac{(R_1+R_5)}{2}$ to R_5 is calculated which is η. η is the average number of transition points in the the image.From this R_4 and R_3 is calculated.

To detect matra region researchers used bell shaped fuzzy function . A pixel is Matra pixel if its value exceed the mean of the bell shaped curve. Authors also vertically segmented on Matra region using fuzzy function.

Chaudhury *et al.* [24] used hpp to detect ”Matra” line. The peak value of hpp is the matra line. As books in DLI had variation in their width and alignment, headline removal from the texts often gets affected. This problem was handled by the Canny Edge Detector and then applying Probabilistic Hough Transform (PHT). Finally headline was removed. Once the headline is deleted, the headline will be disconnected and a smaller modifier will remain with the main character part. Then thinning technique was applied on each individual component. Then Parker’s image skeletonization method was applied. The text line below the headline was divided horizontally into two equal halves, called top and bottom. Authors proposed an algorithm for detecting other modifier.

Manab *et al.* [30] segmented each word with traditional horizontal projection profile. Each line corresponds to a valley followed by a peak in histogram. They used 80% threshold to select the Shirorekha region and marked it’s upper and lower boundary. Thy improved Baseline Detection algorithm proposed by Khan *et. al*[28], unlike Khan *et. al* they have computed the mode value over the entire page. Unlike Khan *et. al* they didn't locate baseline for lower modifier segmentation, as lower modifier can fall above the projected baseline for a number of anomalous reasons. They proposed a method for character segmentation by investigating the column histogram. Then assigned upper ligature with corresponding bounding box.

Zahan *et al.* [42] first segmented each line with horizontal projection profile(hpp). But there was error in some cases. To handle this statistical approach was used. Each word was segmented with a vertical projection profile(vpp). Here they divided each character into two zone, above Matra and below Matra. By removing matra each character gets topologically disconnected. Chain approximation algorithm with 8-Connectivity was used by authors. To resolve the problem associated with overlapped characters connected component analysis method was used. Accuracy mostly depends on detection of Matra line. For detecting start and end of Matra line they used first order difference of each words hpp.

Gupta *et al.* [27] started segmentation process with character segmentation where the words are binarized and thinned. Then polygonal approximation algorithm is applied on the thinned word which determines digital straightness of the word having width of one-pixel. Junction line and header line is detected in succession using the approximated points. Header junction points are traversed thoroughly which determines segmentations points and applies segmentation process. Having the character segmentation process accomplished upper modifier segmentation process is performed where header line is bounded in box which determines segmentation points. Finally baseline is detected which is followed by lower modifier segmentation process.

Bhowmik *et al.* [21] implemented a segmentation technique based on learning unlike other papers which depend on a deterministic algorithm. After Preprocessing step several steps are followed for segmentation. Candidate segmentation points are detected by contour tracing which traverses first from the left-most object pixel of the word to the right-most object pixel of same word in clockwise direction and then from the right-most object pixel to the left-most object pixel but in counterclockwise direction. An 8-directional chain code is generated by the contour tracing process. Then the header line is detected by approximation of the the least square line fitted by the points. Also for significant skewed matra line de-skewing method is applied. The next step is feature extraction where several script and characteristic based feature is extracted for applying learning algorithm. For the last step SVM classifier with linear and radial basis function(rbf) is used. This step consists of two steps where the feature vector is labeled via clustering and classifying test data to test effectiveness of the semi-automatic annotation technique.

Khan *et al.* [28] segmented by detecting base features and splitting from the key points. The matra line is detected along with its width by extracting maximum valued index from y-axis histogram which is wired with a certain threshold. Shirorekha splitter algorithm of Tesseract is used in order to split consonant character conjuncts, simple vowels and modifiers by putting spaces on the header. A statistical approach is used next for finding out the baseline. Finally by using connected components top and bottom modifiers are splitted.

2.3.2 Works on Foreign Scripts

In 2001, Cheung, Bennamoun and Bergmann proposed a recognition based segmentation technique for Arabic OCR [25]. At first documents are scanned, and then the scanned images are binarized and smoothed. Then words are segmented based on the morphing property of Arabic script characters. As a character can retain three shapes in Arabic, the words are segmented by finding a path between a tentative ending character and a tentative beginning character using a 3x3 mask.

In the next step, characters are segmented, either using Amin's character segmentation algorithm [17], or the convex dominant points (CDP) detection by Bennamoun[20]. External features known as chain code are then extracted from these hypothesized segments. The codes of the fragments are fed into a state machine with a feedback loop. The state machine is designated to output individual characters. This process is done twice (right to left and then left to right) and results of these two loops are compared and merged if necessary.

Their word segmentation accuracy is reportedly 99%, whereas their recognition capability hovers around 85%. Among the reasons for failure are : some Arabic characters being visually similar to fragments of other characters, characters sometimes morphing to a yet unintelligible degree, and the encact placement of the dots and the diacritical marks not being static. This last problem is shared by many Bangla OCR systems, and some of the ideas in this paper can be appropriated for our current purpose.

In 2002, Bansal and Sinha outlined a segmentation technique for fused or compound Devanagari characters[19]. The proposed two-pass algorithm extensively uses structural properties of the Devanagari script. In pass-1, words are first segmented into basic or composite characters with the help of row-wise and column wise histogram and the existence of a hypothesized Shirorekha. In pass-2, composite characters are predicted and examined for further segmentation using collapsed horizontal production among other properties.

The segmentation rate him been measured at 83.07 percent with this algorithm, an input of 20386 characters. Their solution can be adapted for Bangla with some reservations. As Bangla's compound characters are joined top to bottom in addition to left to fight, this paradigm must be introduced if we are to split the compounds into constituents.

In 2010, Muaz outlined an Urdu OCR system with separate segmentation and recognition modules[16]. In this OCR system, the strokes of the text are first thinned as a part of the preprocessing phase. Then a traversal algorithm starts from the lower-leftmost point and traverses all points connected to it until a junction point or a free point is found. This algorithm breaks the text image into individual segments, and with an 8 x 8 mask, original shapes are restored. In the recognition phase, the system generates constant sized frames. Discrete cosine transform is applied on all the frames of a segment to create a single feature set for the particular segment.

The accuracy for the segmentation and the recognition phase were reported to be 97.59% and 92.19% respectively. Some of the errors in recognition occurred because of segmentation failure, while others were due to uneven diacritic association. This work, however, was too font—specific to be of any real value to us; the system only work with Noorie N'astaliq font of size 12.

In 2017, Manisha and Sharmila have proposed a recognition process for Tamil OCR [31] using the Sobel Mask [40]. After segmenting the words, bounding boxes are used to separate characters, and then character-level features are selected to identify each individual character. Features such as height, width, line slopes, segment concavity with respect to centroids, and curvature information are extracted. Here, the Sahel Mask is used to extract information on whether a line is horizontal, vertical or curved.

The idea worked well for Tamil because no two adjacent characters share same region. In effect, Tamil characters are already segmented when written. The same camiot be put to guarantee for Bangla, because, in addition to having a shared Shirorekha, characters may overlap each other at several places, especially in handwriting. Moreover, their proposed method can not resolve well between similar characters.

Their proposal yields a recall rate of 95.3% and 92.8% for Word level recognition (where lines and then words have been extracted from an image) and image level recognition (where lines or words have not been extracted) respectively. The precision rate for them happen to be 97.4% and 93.2%. As both rates fall for image level identification, we can bear witness to another scenario where further segmentation proves to be indispensable for correctness in algorithm.

Meanwhile, Soora and Deshpande in 2017 reviewed various feature extraction techniques in the context of India[41]. With an abundance of scripts throughout the subcontinent, the researchers had enough motive to outgo this examination. The detailed result of this probing is too tumultuous to present here, but a quick check reveals that topological features and moment—based features performed better than Simple geometric or statistical features for most dataset. As characters can change their minute details across fonts for print and people for handwriting, this result is not totally uncalled-for.

Their discussion reveals the lack of large public database for Indie scripts which, according to them, significantly hinders the progress of OCR technologies, and a genuine poverty of systems that would work well for all Indie fonts irrespective of whether they are printed or handwritten. This gap in the existence of a more generalized OCR also brings into light the lack of a good segmenter for handwritten texts.

In 2017, Obaidullah, Goswami and others proposed a criterion to analyze documents into different sections based on which script they are written in. This separation process works for handwritten as well as printed text documents[34]. The text is first split into two broader categories, those without Shirorekha and those with it. Fractal Geometry Analysis (1-D FGA), Canny Edge Detector and Morphological Line Transform were executed on document images to extract features for the entire image (with only a prior attempt at line segmentation). These features are deemed to be different for different scripts, especially for the two larger categories. The edge detector aids in computing the pixel count of the upper zone of each line. Next, they implemented three classifiers, namely Multilayer Perceptron, BayesNet and Random Forests, to label these lines as the correct scripts. For dataset, they used 1204 text lines, with 325 in Bangla, 200 in Devanagari, 370 in Roman (English), and the remaining in Arabic/Urdu. Using different classifiers, the highest accuracies in separation achieved by Fractal Analysis, Edge Detector and Line Transform were 95.68%, 85.30%, and 76.49%. The identification accuracy was higher, in fact for the three feature selection methods coupled with the classifier most suited to them, the average output accuracies were 96.04%, 90.68% and 84.02%. Specifically, the accuracy of script identification increased from 90.84% to 95.97% on average when preceded by the script separation process, and the computational time decreased by a 15%.

This work shows how OCR technology has unexplored problems lying around precociously that need our further attention.

2.3.3 Research Gap

Here we have described what we pointed out the drawbacks where their implementation lags behind. These information will be used for further studies as well

Table 2.1: Limitation of the papers

Paper	**Limitations**
Chaudhuri *et al.* [23]	Performance is measured on clear paper. They don't address the problem of effect of noise in character image.
Chaudhury *et al.* [24][42]	Here authors haven't described about the segmentation of Compound characters, character with lower modifier. Overlapping characters and character modifier is not considered here. Variable font sized character also not considered.
Sarkar *et al.* [37]	The system segments character only vertically but considers only partial horizontal dissection. However, it didn't consider about connected components with more than two characters or how to segment them which may lead to failure. It also over segments the characters.
Gupta *et al.* [27]	The scripts of four languages are are almost identical in nature. Despite Having the fact their algorithm lacks accuracy in segmenting Bangla script.
Bhowmik *et al.* [21]	Inconsistency maybe be faced while learning as the algorithm can misclassify complex handwriting. Also the algorithm creates a ample amount of redundant data.
Khan *et al.* [28]	Baseline is detected above expected point. Some wrapping parts needed to be reassembled as they are segmented as different components. Overlapping upper and lower modifier cannot resolve conflicting elements. For characters of shorter height than the baseline any modifier attached to it remains remains above baseline, as a result this modifier can't be segmented and thus be recognized properly.
Manab *et al.* [30]	Variable font size causes failure as well as subscripts and superscripts. Heavily dependent on header line. Lower vowel is not segmented. Adjoined characters were not segmented properly.

as determining the most general problems to be solved. A summary of the related works based on the limitations is shown in TABLE 2.1. All researchers followed four traditional approaches for solving OCR problem for Bangla. They followed pre-processing, segmentation, recognition and post-processing. Each step has more or less errors which are accumulated in final stage. But our end-to-end deep learning model free from error of each steps. The process of building an OCR can be divided in two major areas of focus - character segmentation and character recognition. A complete working OCR solution depends on the performance of both of these areas. Falling behind in segmentation will cause falsely segmented characters. Thus even the best recognizer won't be able to recognize them. On the other hand, an underdeveloped recognizer will never be able to recognize even the most perfectly segmented characters. Hence it is an important design decision for an OCR solution to balance the emphasis between two core components. For example, a segmenter that can separate simple components from a compound characters can make recognition easier, reducing the complexities in identifying complex character conjuncts. On the other hand, building a recognizer that can efficiently identify compound characters require less effort from the segmentation part. Hence we are using end-to-end deep learning model.

2.3.4 Contributions

Over the past few decades many researchers have worked on different components of Bangla OCR, namely preprocessing of text image, segmentation of lines, words and characters, and recognition of segmented characters. In TABLE 2.2 we have described contribution of those researchers.

2.4 Background Study

In this section, we are going to look into some of the terms used throughout this book and the basic components of an OCR engine. narang *et al.* [33] presented some examples of similar ancient Devnagari scripts.

2.4.1 Properties of Bangla alphabet

Properties of Bangla alphabet are listed below:

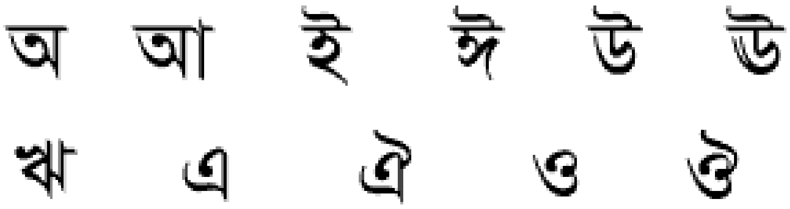
1. There are two categories of character set: basic characters and compound characters . Basic characters are the collection of consonants and vowels. Bangla has 39 consonants and 11 vowels. Small form of vowel is known as 'Kar'. In the alphabet there are 11 vowel characters, 10 of which have vowel modifier. FIGURE 2.1 and FIGURE 2.2.

2. Among the consonants in the Bangla alphabet, there are 35 characters which have their independent pronunciations. There are also 4 modifying consonants who don't have a pronunciation of their own, along with a non-vowel consonant modifier called Hashanta, which mutes inherent vowel. They are

TABLE 2.2: Contribution by other researchers

Paper Reference	**Contribution**
Chaudhuri *et al.* [23]	Solves the problem of segmenting individual character and focuses on the recognition of basic Bangla characters.
Chaudhury *et al.* [24]	Character segmentation of old books having different typesetting and developed an algorithm for segmentation for thinned image.
Sarkar *et al.* [37]	Segmentation for variable sized font using fuzzy bell shaped function, Intends to segment basic Bangla characters and successfully segmented connected component.
Zahan *et al.* [42]	Two zone approaches reduce complexity and increase accuracy.
Gupta *et al.* [27]	This paper introduces a single approach which is capable of segmenting words of different scripts. Works without removal of header line unlike other papers. Works on words with broken header lines or skewed words up to 10 degrees. Also works for broken word.
Bhowmik *et al.* [21]	Segmentation process is learning based rather deterministic coded. Under segmentation rate is very low. Works well on different and complex hand writing with various pen width. Add-in Bootstrapping for labeling more training data for better accuracy.
Khan *et al.* [28]	Statistical analysis is applied for detecting the baseline. For handling dot symbols in the form of consonant components a depth first search based solution was improvised . Also five important vowel modifiers are separated.
Manab *et al.* [30]	Baseline detection process produces more accurate baseline. Feature extraction process were kept translation invariant.

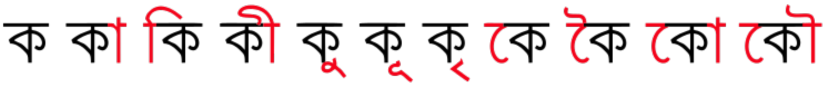
FIGURE 2.1: List of vowels in Bangla alphabet



usually found in the middle of Bangla words serving almost a similar role to a vowel modifier. FIGURE 2.4 and 2.4.

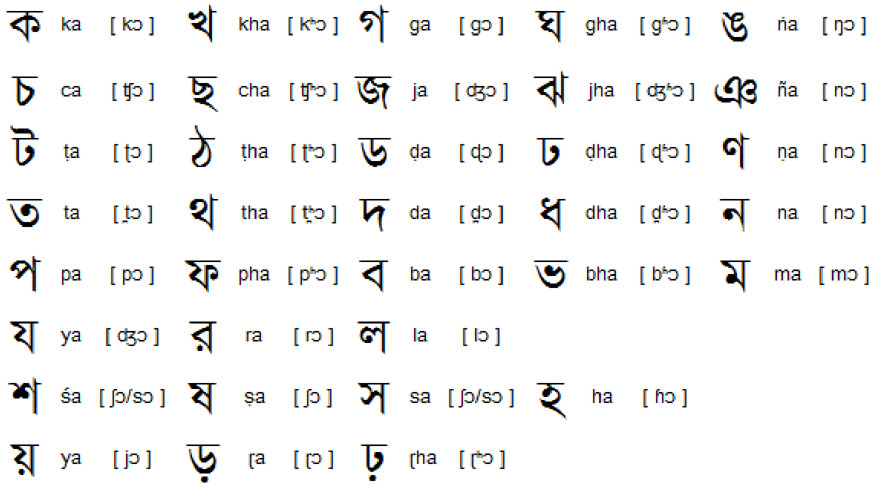
3. Another group of characters known as conjunct characters exists in Bangla language. A conjunct character is formed by adjoining two or more basic

FIGURE 2.2: List of vowel modifiers in Bangla alphabet



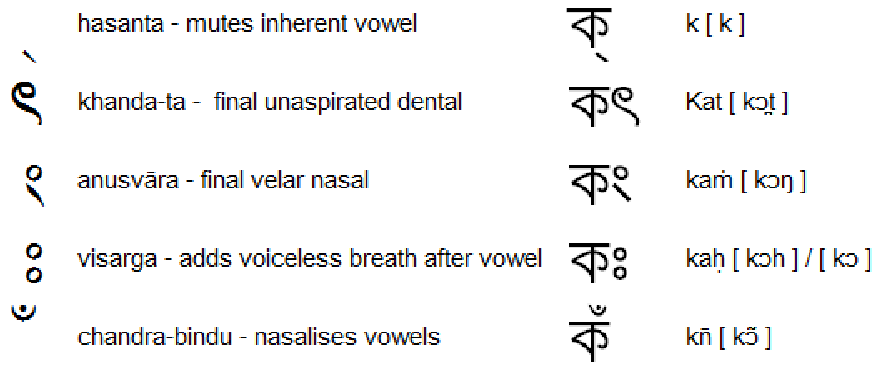
ক কা কি কী কু কূ কৃ কে কৈ কো কৌ

FIGURE 2.3: List of Bangla consonants with independent pronunciations



ক	ka	[kɔ]	খ	kha	[kʰɔ]	গ	ga	[gɔ]	ঘ	gha	[gʰɔ]	ঙ	ṅa	[ŋɔ]
চ	ca	[ʧɔ]	ছ	cha	[ʧʰɔ]	জ	ja	[ʤɔ]	ঝ	jha	[ʤʰɔ]	ঞ	ña	[nɔ]
ট	ṭa	[ʈɔ]	ঠ	ṭha	[ʈʰɔ]	ড	ḍa	[ɖɔ]	ঢ	ḍha	[ɖʰɔ]	ণ	ṇa	[nɔ]
ত	ta	[t̪ɔ]	থ	tha	[t̪ʰɔ]	দ	da	[d̪ɔ]	ধ	dha	[d̪ʰɔ]	ন	na	[nɔ]
প	pa	[pɔ]	ফ	pha	[pʰɔ]	ব	ba	[bɔ]	ভ	bha	[bʰɔ]	ম	ma	[mɔ]
য	ya	[ʤɔ]	র	ra	[rɔ]	ল	la	[lɔ]						
শ	śa	[ʃɔ/sɔ]	ষ	ṣa	[ʃɔ]	স	sa	[ʃɔ/sɔ]	হ	ha	[ɦɔ]			
য়	ya	[jɔ]	ড়	ɽa	[ɽɔ]	ঢ়	ɽha	[ɽʰɔ]						

FIGURE 2.4: List of Bangla dependent consonants and other modifiers



্	hasanta - mutes inherent vowel	ক্	k [k]
ৎ	khanda-ta - final unaspirated dental	কৎ	Kat [kɔt̪]
ং	anusvāra - final velar nasal	কং	kaṁ [kɔŋ]
ঃ	visarga - adds voiceless breath after vowel	কঃ	kaḥ [kɔh] / [kɔ]
ঁ	chandra-bindu - nasalises vowels	কঁ	kn̄ [kɔ̃]

consonant shapes of Bangla. There are 160 frequent conjunct characters in Bangla script. FIGURE 2.5.

4. Each text image line is divided into three zones: upper, middle, and lower zone. Upper zone contains a modifier or part of a character above the headline, the middle zone contains the shape in between the headline and the baseline which the primary part of the character, and lower zone contains the lower modifiers or part of compound characters below the baseline.

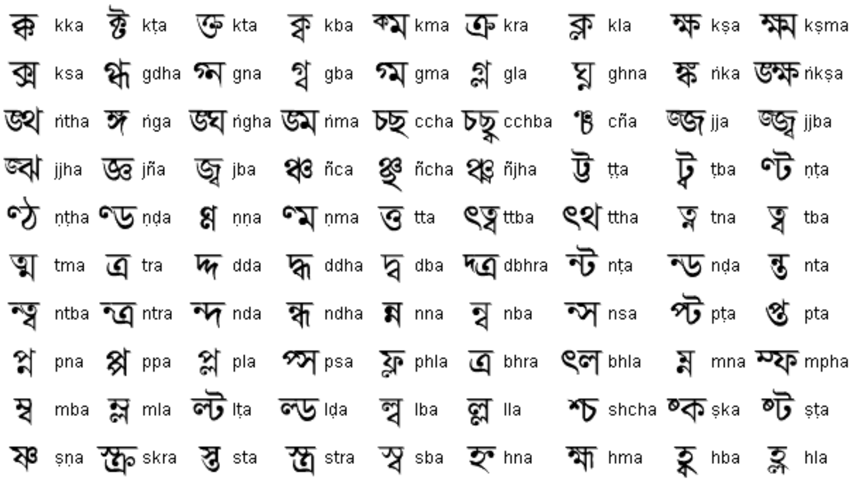
5. Bangla script has some critical characteristic specially seen in handwritten scripts.

- Skewed words

- Broken words

- Words without header line

FIGURE 2.5: List of some important consonant conjuncts in Bangla



ক্ক	kka	ক্ট	kṭa	ক্ত	kta	ক্ব	kba	ক্ম	kma	ক্র	kra	ক্ল	kla	ক্ষ	kṣa	ক্ষ্ম	kṣma
ক্স	ksa	গ্ধ	gdha	গ্ন	gna	গ্ব	gba	গ্ম	gma	গ্ল	gla	ঘ্ন	ghna	ঙ্ক	ṅka	ঙ্ক্ষ	ṅkṣa
ঙ্খ	ṅtha	ঙ্গ	ṅga	ঙ্ঘ	ṅgha	ঙ্ম	ṅma	চ্ছ	ccha	চ্ছ্ব	cchba	চ্ঞ	cña	জ্জ	jja	জ্জ্ব	jjba
জ্ঝ	jjha	জ্ঞ	jña	জ্ব	jba	ঞ্চ	ñca	ঞ্ছ	ñcha	ঞ্ঝ	ñjha	ট্ট	ṭṭa	ট্ব	ṭba	ণ্ট	ṇṭa
ণ্ঠ	ṇṭha	ণ্ড	ṇḍa	ণ্ণ	ṇṇa	ণ্ম	ṇma	ত্ত	tta	ত্ত্ব	ttba	ত্থ	ttha	ত্ন	tna	ত্ব	tba
ত্ম	tma	ত্র	tra	দ্দ	dda	দ্ধ	ddha	দ্ব	dba	দ্ভ্র	dbhra	ন্ট	nṭa	ন্ড	nḍa	ন্ত	nta
ন্ত্ব	ntba	ন্ত্র	ntra	ন্দ	nda	ন্ধ	ndha	ন্ন	nna	ন্ব	nba	ন্স	nsa	প্ট	pṭa	প্ত	pta
প্ন	pna	প্প	ppa	প্ল	pla	প্স	psa	ফ্ল	phla	ভ্র	bhra	ভ্ল	bhla	ম্ন	mna	ম্ফ	mpha
ম্ব	mba	ম্ল	mla	ল্ট	lṭa	ল্ড	lḍa	ল্ব	lba	ল্ল	lla	শ্চ	shcha	ষ্ক	ṣka	ষ্ট	ṣṭa
ষ্ণ	ṣṇa	স্ক্র	skra	স্ত	sta	স্ত্র	stra	স্ব	sba	হ্ন	hna	হ্ম	hma	হ্ব	hba	হ্ল	hla

2.4.2 Working Procedures of OCR

A character recognition system follows a few basic steps which are

1. Preprocessing

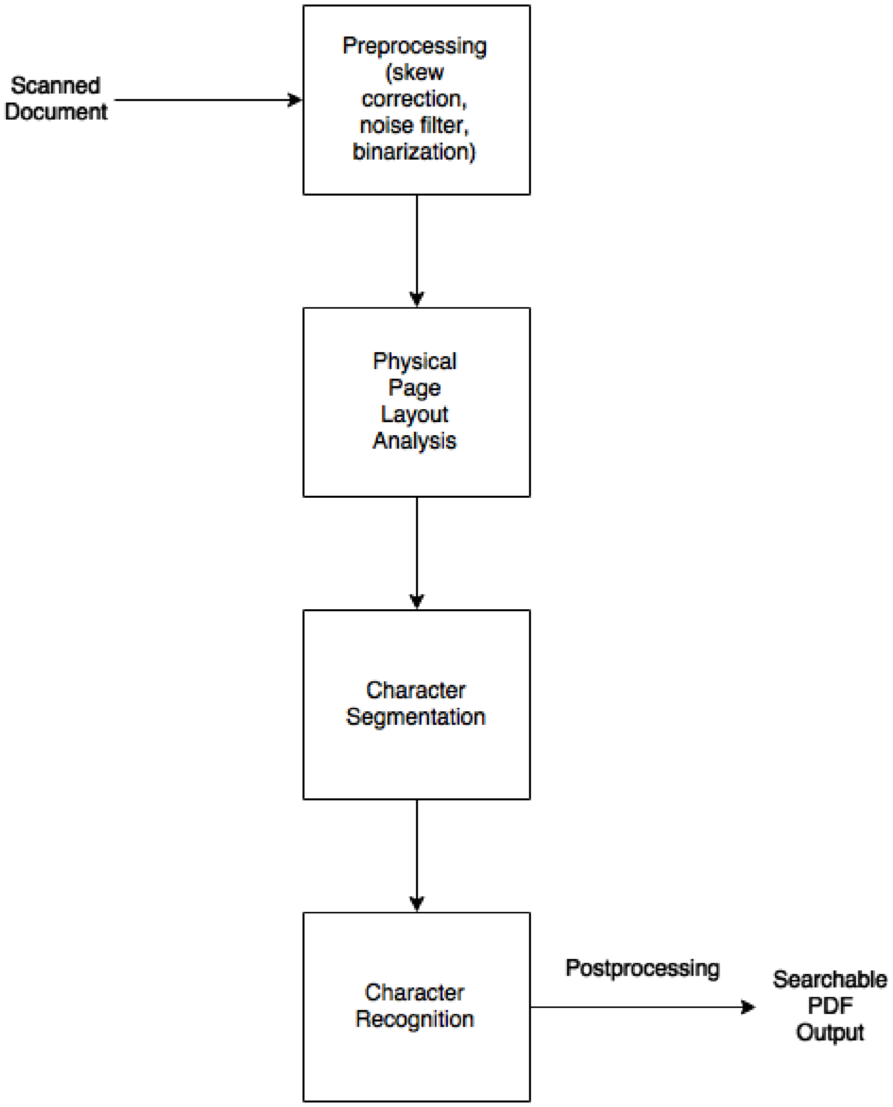
2. Segmentation

3. Recognition

4. Post-processing

Preprocessing: Input images are prepared for segmentation and recognition in the preprocessing step FIGURE 2.6. Several approaches were applied for preprocessing technique such as image acquisition, noise removal of noisy and garbage information, adjusting skewed line, image binarization and skeletonization. Also the if the document has a dual-page layout it is splitted. The images that are rescaled are either shrunk or enlarged. Image blurring is usually achieved by the image can be convolved with a low-pass filter kernel. Filters have general usage of blurring images or reducing noise. There are additional filters are available such as averaging filter, Gaussian filter, Bilateral filter, median filter. Individual filters perform differently on varying images. For instance, while some filters can successfully binarize certain images, they may fail to binarize others. Likewise, different filters may work well with those images that other filters cannot do well. There's not a single image thresholding method that fits all types of documents. Most commonly used thresholding methods are Adaptive thresholding, simple thresholding, and Otsu's thresholding.

FIGURE 2.6: An overview of the OCR process



Segmentation: After the preprocessing step being completed the image to be recognized is passed through a segmenter. A segmenter clips an image into consecutive small parts. The segmenter segments the lines and from each line segments the words. The main purpose of the segmenter is to create image fragments of meaningful separate patterns or symbol called pseudo-characters or glyphs, from those word images. A pseudo-character can be generated as

1. A single individual character

2. A single characters part which is known as over-segmentation
3. Fragment of two individual characters
4. More than one characters as a single character, known as under-segmentation.

Recognition: The segmentation process produces a list of possible characters to go through the recognition stage. A set of features are extracted from the pre-processed and segmented based on various properties of a segmented character image as well as the properties of input script (language, font, writing method etc.). Some well known feature extraction techniques are chain code based features, gabor features, structural decomposition, gradient based features and many more. A set of well defined features can recognize a character with minimum error. After extracting features, the feature sets are trained with state of-the art learning algorithms like SVM, neural network, decision tree, etc. These trained modules are used to recognize characters from test set.

Post-processing: Recognizing a character correctly is not the last work to get output. We have to merge this recognized characters in a sequence to get the actual word. Post processing is the technique for reducing the error after recognition part. Post processing of Optical Character Recognition requires several steps

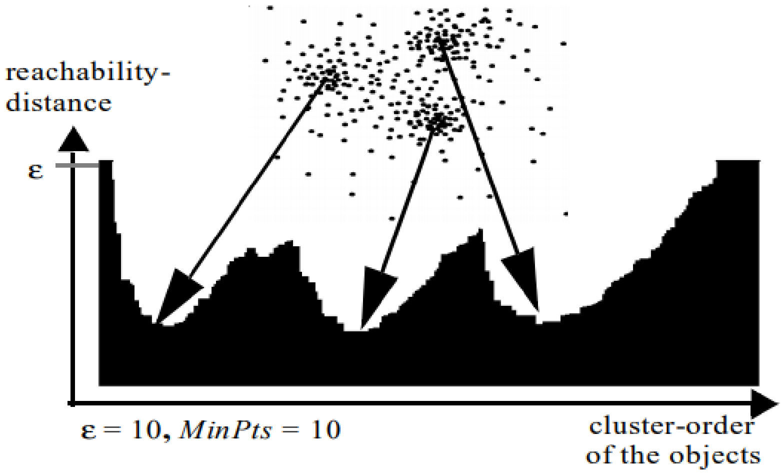
- Writing the recognized characters in the right sequence
- Spell check for word correction
- Sentence correction

Case 1 is special for Bangla characters. For English language after recognizing the characters if we write the characters in the same sequence we get them at the time of extracting there will be no change in output.

2.4.3 OPTICS

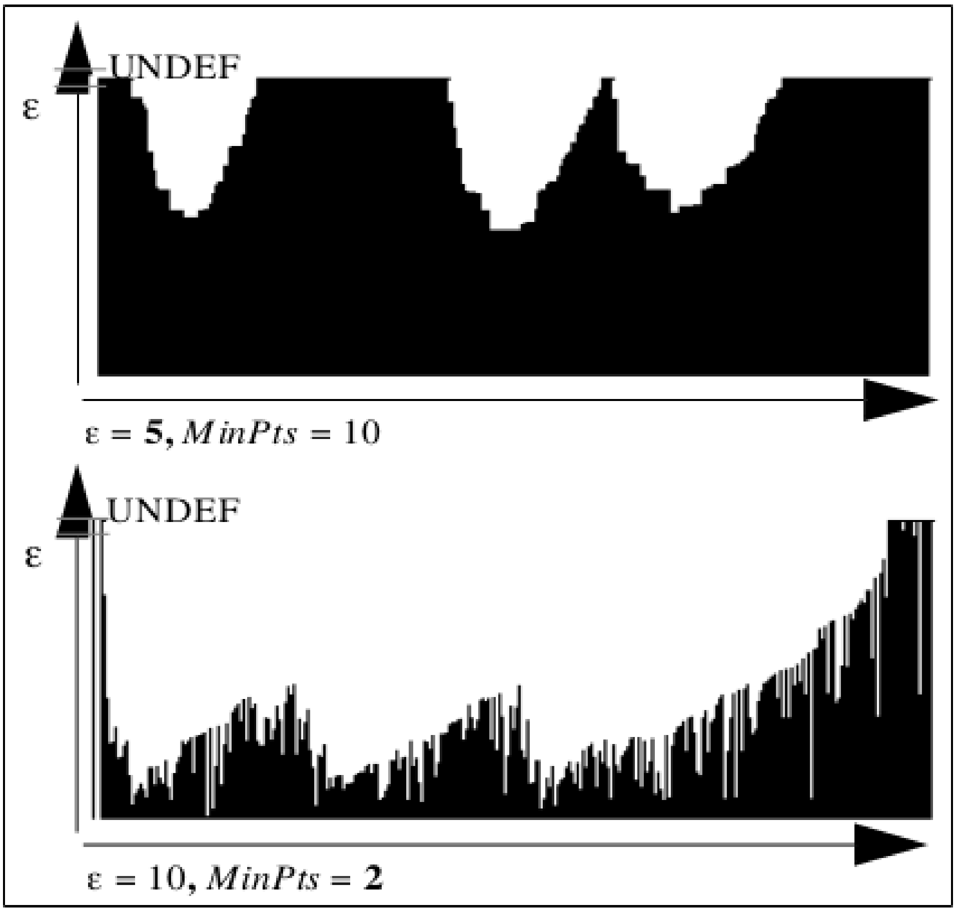
OPTICS (Ordering Points To Identify the Clustering Structure) is a density based clustering algorithm. It is a modified DBSCAN algorithm. The origin algorithm, DBSCAN can cluster objects given input parameters such as ϵ, maximum neighborhood radius and MinPts, minimum number of points required in the neighborhood of a core object, it requires the task of selecting parameter values that will lead to the discovery of acceptable clusters. This is a problem associated with many other clustering algorithms. OPTICS creates an ordering of a database, additionally storing the core-distance and a suitable reachability-distance for each object.Figure 2.7 depicts the reachability-plot for a very simple 2-dimensional data set.

Figure 2.7: Illustration of the cluster-ordering



The Reachability-plot is rather insensitive to the input parameters of the method which can be considered as an advantage of cluster-ordering a data set compared to other clustering methods. They are the generating distance εand the value for MinPts. In order to yield a good result these values have just to be large enough. The concrete values are not crucial because there is a wide range of possible values for which the clustering structure of a data set can be observed when looking at the corresponding reachability-plot. Figure 2.8 shows the effects of different parameter settings on the reachability-plot. Properties of OPTICS are explained in Ankerst *et al.* [18]

FIGURE 2.8: Effects of parameter settings on the cluster-ordering



2.4.4 RMSProp

RMSprop (root mean square propagation) is an optimization algorithm which is designed to minimize loss function in a neural network. RMSprop belongs to the category of adaptive learning rate methods. Like gradient descent with momentum it has the same concept of the exponentially weighted average of the gradients but the difference is the update of parameters.In RMSprop, largely varying gradients of successive mini-batches is mitigated by using a moving average of the squared gradient for each weight. The gradient of each mini-batch is divided by the square root of the MeanSquare. This process effectively averages the gradients over successive mini-batches so that weights can be finely calibrated. The Weight update rule equation is given below.

$$E[g^2]_t = \beta E[g^2]_{t-1} + (1-\beta)(\frac{\delta C}{\delta w})^2 \quad (2.1)$$

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{E[g^2]_t}} \frac{\delta C}{\delta w} \quad (2.2)$$

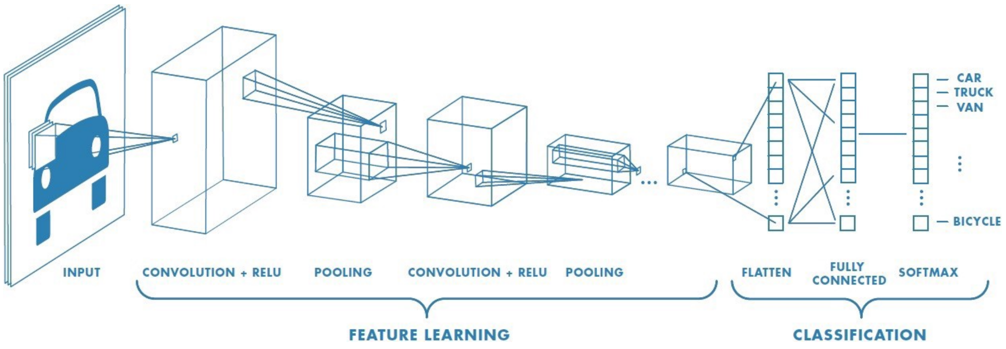
Here E[g] is the moving average of squared gradients. dC/dw is the gradient of the cost function with respect to the weight. η is the learning rate and Beta is the moving average parameter (default value is 0.9) [13] [8] [7]

2.4.5 Convolutional neural network

Convolutional Neural Network (CNN) is an widely used in computer vision. Some examples of where CNN is used are image recognition, image classifications, object detection, recognition faces etc. Image classification process takes an input image, process it and classify it under certain categories (characters label for our project). An image is an array of pixels values and it depends on the image resolution. Based on the image resolution, the array has a dimension of h $\times w \times d(h = Height, w = Width, d = Dimension)$.

In deep learning CNN model each input image will be passed through a series of convolution layers with filters (Kernals), Pooling, fully connected layers (FC) and applied Softmax function to classify an object with probabilistic values between 0 and 1. CNN has three main layers FIGURE 2.9.

Figure 2.9: Convolutional Neural Networks

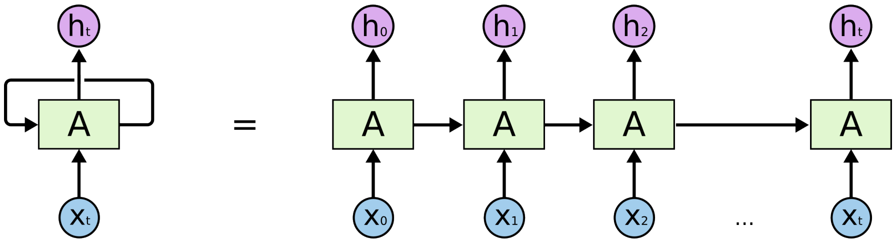


Convolution Layer: Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs which are the image matrix and another matrix named filter or kernal. Convolution of an image with different filters can perform operations such as edge detection, blur and sharpen by applying filters. Pooling Layer: The Pooling layer is responsible for reducing the spatial size of the convoluted Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model. There are two types of Pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel. Fully Connected Layer: The convolutional layer and the pooling layer, together form several layers of a CNN. After going through the above process, the model is successfully enabled to understand the features. Now the final image output will be flattened and fed to a regular Neural Network for classification purposes. [1] [12]

2.4.6 Recurrent Neural Network

Recurrent Neural Network is a generalization of feedforward neural network that has an internal memory. RNN is recurrent in nature as it performs the same function for every input of data while the output of the current input depends on the past one computation FIGURE 2.10. After producing the output, it is copied and sent back into the recurrent network. For making a decision, it considers the current input and the output that it has learned from the previous input. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. In other neural networks, all the inputs are independent of each other. But in RNN, all the inputs are related to each other. Recurrent Neural Network have some major disadvantages of which main drawback is lack of long term memory because of

FIGURE 2.10: An unrolled recurrent neural network.

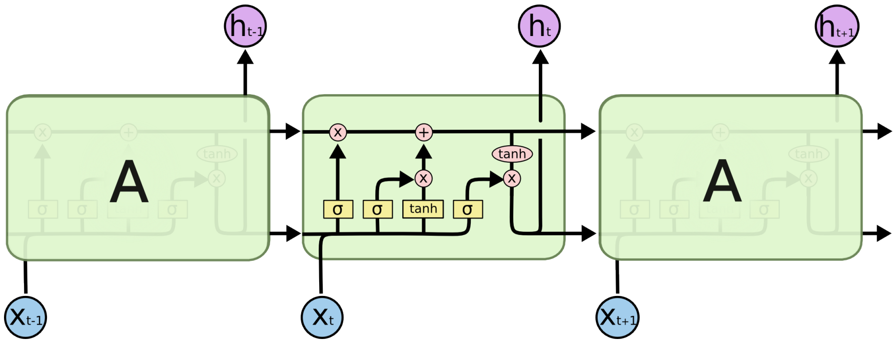


gradient vanishing. Also it cannot process very long sequences if using tanh or relu as an activation function.

2.4.7 Long Short Term Memory

Long Short Term Memory networks (LSTM) are a special kind of RNN, capable of learning long-term dependencies such as a contextt of a complex sentence. Unlike RNN they are explicitly designed to avoid the long-term dependency problem. All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer. LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way. They are input gate, forget gate, output gate and combining all three is cell state FIGURE 2.11. [14] [11]

FIGURE 2.11: The repeating module in an LSTM contains four interacting layers.



2.4.8 Connectionist Temporal Classification

Graves *et al.* [26] introduce the CTC loss and decoding operation. With a CTC output layer it is possible to train an NN from pairs of images and labels, there is no need to specify at which location a character appears in the image. CTC also handles decoding by finding the most probable labeling from the framewise label probabilities. The NN-training will be guided by the CTC loss function. We only feed the output matrix of the NN and the corresponding ground-truth (GT) text to the CTC loss function. It tries all possible alignments of the GT text in the image and takes the sum of all scores. This way, the score of a GT text is high if the sum over the alignment-scores has a high value.

2.5 Conclusion

This study provides an overview on the works of Bangla OCR. Research Question 1 identifies the common preprocessing techniques addressed by researchers. Research Question 2 identifies the common problem in segmentation and their solution techniques adapted by researchers.

Chapter 3

Proposed Approach

In this chapter we present our methodology for building a full system for Bangla OCR. This chapter starts with the details of the dataset we used in this project which is the most vital element of a machine learning project. Then the preprocessing of the image data and classification process are also explained here.

3.1 Introduction

The methods used for the proposed OCR system are presented in this chapter. Preprocessing includes extracting image lines from the entire page. We used clustering algorithm for separating those lines. Each line images is binarized and different thresholding technique was used to remove noise form image. Removing noise is important part of otherwise our NN will learn noise rather than feature. Then we encoded Bangla ground truth text. Compound character, upper and lower modifier creates problem in recognition. They take position on top of one another and our recognizer tries to predict characters at time stamp. Although Bangla has 91 unique characters in total we have taken care of most frequent 396 combination of those unique characters other can be ignored as they don't occur frequently. A hybrid neural network architecture consisting of a convolutional neural network (CNN), two layers of BLSTM cells and a connectionist temporal classification (CTC) layer have been used for recognition of the segmented text lines. CTC decoding algorithms are discussed which extract the final labeling from the RNN output. Our proposed methodology works on several steps.

- Preprocessing
- Classifier

3.2 Dataset Description

The ISIDDI [3] dataset contains 534 page images of various books. File format of the image is jpeg. The ground truth of each file is contained in a text file with the corresponding name. Total file size is 322 megabytes. The pages are taken 15 unique books. Each page contains an average of 25 lines of text. Name and page frequency of each unique books are shown on TABLE 3.2. Degraded state seen in the pages includes partially torn page, deep brown contrast of pages, noisy and miscellaneous components in the page, slanting and skewing. Likforman *et al.* [29] can be referred for historical document analysis.

TABLE 3.1: List of Unique book and their frequency in ISIDDI dataset

Book Name	**Page Frequency**
SalomanerHitopodesh	30
Bardidi	68
Gospelofmatthewi	30
ParamarthaPrasanga	30
PurbobangoOHinduSamajh	15
MohonerAgatobas	31
NewTestament	30
SankhaDarsanam	29
OnlyWayTobeSaved	29
ManahShiksha	29
MatirPath	35
DarwanMaleeKathapokathan	20
GopiChandrerGan	30
BiswasGhatakMahon	27
DhyanOSakti	101

Aside from the ISIDDI dataset we prepared some dataset from online sources [15]. The page images were extracted from the pdf version of those books also found in online sources .The page images acquired from these books contains lesser noise, slanting and skewness.

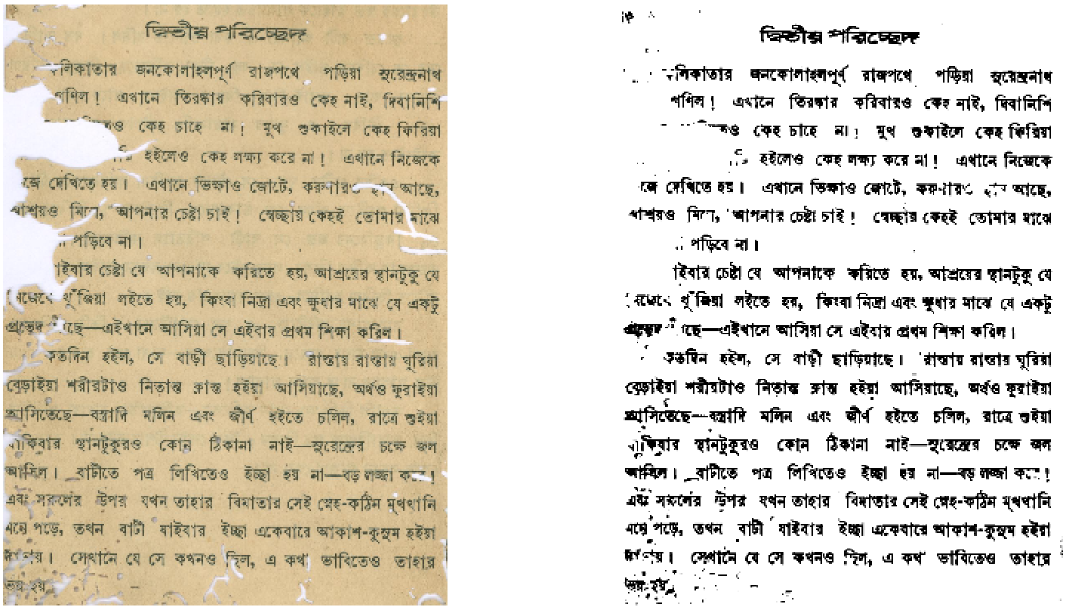
TABLE 3.2: List of Unique book and their frequency in prepared dataset

Book Name	**Page Frequency**
Baul Kabi Radharaman Geeti Sangraha	92
Kakababu O Aschorjo Dip	79

3.3 Preprocessing

The preprocessing step involves smoothing the image by Gaussian Filtering (SigmaX = SigmaY = 3) and binarizing the image by Ostu's Shareholding method.

FIGURE 3.1: Sample of original image and preprocessed image



দ্বিতীয় পরিচ্ছেদ

কলিকাতার জনকোলাহলপূর্ণ রাজপথে পড়িয়া সুরেন্দ্রনাথ ভাবিল! এখানে তিরস্কার করিবারও কেহ নাই, দিবানিশি ...ও কেহ চাহে না; মুখ শুকাইলে কেহ ফিরিয়া ... হইলেও কেহ লক্ষ্য করে না! এখানে নিজেকে ...কে দেখিতে হয়। এখানে ভিক্ষাও জোটে, করুণার... আছে, আশ্রয়ও মিলে, আপনার চেষ্টা চাই! স্বেচ্ছায় কেহই তোমার মাঝে ... পড়িবে না।

...হিবার চেষ্টা যে আপনাকে করিতে হয়, আশ্রয়ের স্থানটুকু যে ...খুঁজিয়া লইতে হয়, কিংবা নিদ্রা এবং ক্ষুধার মাঝে যে একটু প্রভেদ আছে—এইখানে আসিয়া সে এইবার প্রথম শিক্ষা করিল।

কতদিন হইল, সে বাড়ী ছাড়িয়াছে। রাস্তায় রাস্তায় ঘুরিয়া বেড়াইয়া শরীরটাও নিতান্ত ক্লান্ত হইয়া আসিয়াছে, অর্থও ফুরাইয়া আসিতেছে—বস্ত্রাদি মলিন এবং জীর্ণ হইতে চলিল, রাত্রে শুইয়া থাকিবার স্থানটুকুরও কোন ঠিকানা নাই—সুরেন্দ্রের চক্ষে জল আসিল। বাটীতে পত্র লিখিতেও ইচ্ছা হয় না—বড় লজ্জা করে! এবং সকলের উপর যখন তাহার বিমাতার সেই স্নেহ-কঠিন মুখখানি মনে পড়ে, তখন বাটী যাইবার ইচ্ছা একেবারে আকাশ-কুসুম হইয়া দাঁড়ায়। সেখানে যে সে কখনও ছিল, এ কথা ভাবিতেও তাহার ভয় হয়।

3.3.1 Image Skew Correction

The grayscaled image is first deskewed. The deskewing process takes the significantly blurred version (SigmaX = SigmaY = 5) of the image. First the canny edges are detected. By calucluating hough lines a list of points are acquired in the form representing line segments. The slopes of the lines from their paired point values are calculated. The skewness of lines are calculated by taking the inverse of tan of the slopes and converting them into radian to degree. The angle to be rotated for skew correction is the most common of the angles we get in previous step. A sample page and and preprocessed version of the corresponding page is shown on FIGURE 3.1

Algorithm 1: Image Skew Correction

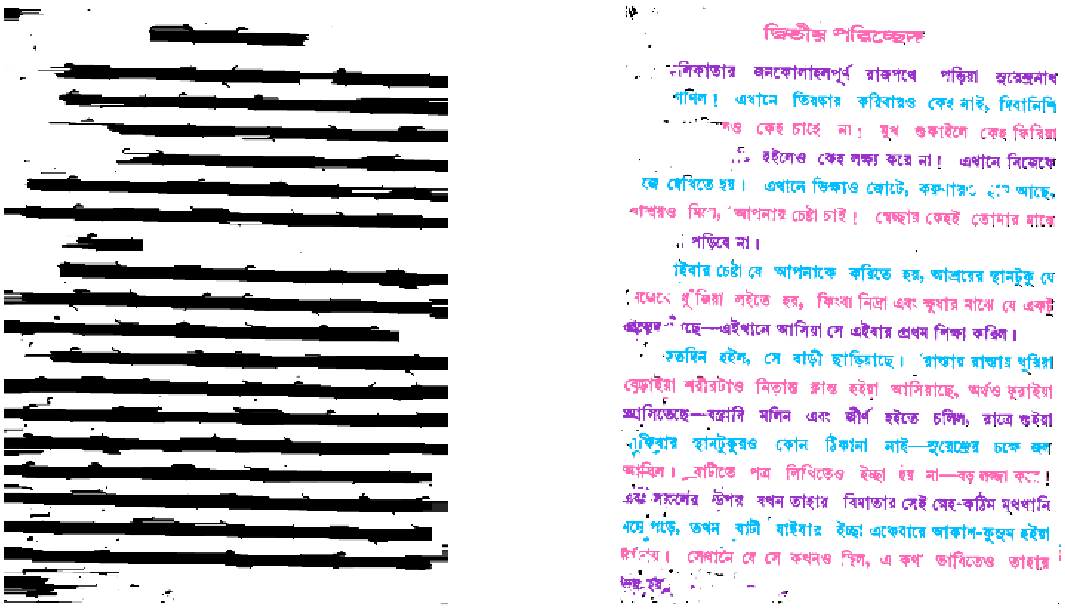
```
Input: Original Image
Output: Skew Corrected and Preprocessed Image
image = Binarization on gausian filtered(SigmaX = 3, SigmaY = 3) image
edges = Canny edges of image
hough_lines = probabilistic hough line of edges
slopes = slope of each hough_lines in hough_lines
rad_angles = Inverse_of_Tangent(slope) for each slope in slopes
deg_angles = degree angle of all rad_angles
rotation_number = most frequent angle of deg_angles
if rotation_number > 45 then
    rotation_number = -(90 - rotation_number)
else if rotation_number < -45 then
    rotation_number = 90 - abs(rotation_number)
rotatedImage = rotated image in rotation_number
return rotatedImage
```

3.3.2 Line Segmentation

The lines in the page are segmented by OPTICS clustering algorithm [9]. Before clustering is performed another modified version of the image is used. First the image is morphologically transformed by Opening. Opening is a process which first erodes and then dilates the image. It is performed 5 iterations on 1*5 kernel which opens the image in horizontally as well as rendering no change vertically[36]. This images is then reduced to lower resolution (200 pixel width) for enhancing clustering speed as OPTICS is has the most time complexity among other clustering algorithms. The coordinate of all black pixels are taken and y axis points are scaled up. After Clustering The points are rescaled to original format. The extreme points in two coordinates of each cluster are taken as a line boundary. Opened version and segmented version of image used above is shown on FIGURE 3.2

```
Algorithm 2: Line Segmentation
Input: Preprocessed and Deskewed Page Image,Clusterin_Parameter,
       scalingFactor
Output: Segmented Lines
image = morphological opening on image (kernel = [1,5] , iterations =5)
image,ratio = reduce image resolution to 200 pixel width with aspect ratio
image = Binarization on gausian filtered(SigmaX = 3, SigmaY = 3) image
coordinates = 2D positions of black pixel in image
coordinates[y_axis] = coordinates[y_axis] $\times scalingFactor$
lineClusters = Optics on coordinates with Clustering_Parameter
for each lineCluster in lineClusters do
    upperMost = min(coordinates[y_axis])
    lowerMost = max(coordinates[y_axis])
    leftMost = min(coordinates[x_axis])
    rightMost = max(coordinates[x_axis])
    lineImage = image[leftMost:rightMost,upperMost:lowerMost]
    lineImageList.add(lineImage)
return lineImageList
```

FIGURE 3.2: Morphologically opened image and line segmented image



3.3.3 Text Encoding

Bangla language contains single and compound characters as well as vowel and consonant modifiers. Compound characters and composite modifiers are comprised of multiple single characters.in order to distinguish every unique character rather than their atomic unit text encoding is performed. Every encoded characters has a encoding code along with a serial number. Every unit characters such as vowel, consonant, numericals as sysmbols are represented as v,c,n and s. The learning process performs on vertical level characters without or without lower modifiers or charcters with 'Chandrabindu' ◌ঁ need to be distinguished. For this purpose characters bounded with another character or modifier in equal or partial vertical zones are encoded as compound character represented as cmp. Compound characters composed of two or three characters are encoded as cmp. TABLE 3.3 contains one member of each granular characters of the dataset. TABLE 3.4 and TABLE 3.5 contains characters with bounded upper and lower modifiers and compound character with varous length respectively.

Algorithm 3: Ground Truth Encoding

```
Input: Ground Truth in Bangla
Output: Encoded Ground Truth
for each image in dataset do
    bangla_line = dataset[ground truth of the image]
    unicode_line = convert bangla_line to unicode
    unicode_compound_map = replace by compound code in unicode_line
    encoded_line = convert unicode_compound_map line to custom tags
    reordered_line = reorder modifier in encoded_line
    dataset[ground truth of the image] = set reordered_line
```

Table 3.3: Sample of atomic elements

Original Character	Unicode	Encoded Key
অ	\u0985	v1
ড়	\u09dc	c33
◌ৌ	\u09cb	m11
৩	\u09e9	n3
।	\u0964	s10

Table 3.4: Sample of characters with lower modifier

Original Character	Composing Characters	Unicode	Encoded Key
আঁ	আ+◌ঁ	\u0986 \u0981	cmp213
টু	ট+◌ু	\u099f \u09c1	cmp240
ষূ	ষ+◌ূ	\u09b7 \u09c2	cmp298

Table 3.5: Sample of compound characters of various length

Original Character	Composing Characters	Unicode	Encoded Key
র্শ	র +◌্ + শ	\u09b0 \u09cd \u09b6	cmp177
স্তূ	স +◌্ + ত + ◌ূ	\u09b8 \u09cd \u09a4 \u09c2	cmp56
দ্ধৃ	দ+◌্ + ধ + ◌ৃ	\u09a6 \u09cd \u09a7 \u09c3	cmp32
স্ক্র	স +◌্ + ক +◌্ + র	\u09b8 \u09cd \u0995 \u09cd \u09b0	cmp17
ন্দ্র	ন +◌্ + দ +◌্ + র	\u09a8 \u09cd \u09a6 \u09cd \u09b0	cmp7

3.4 Classifier

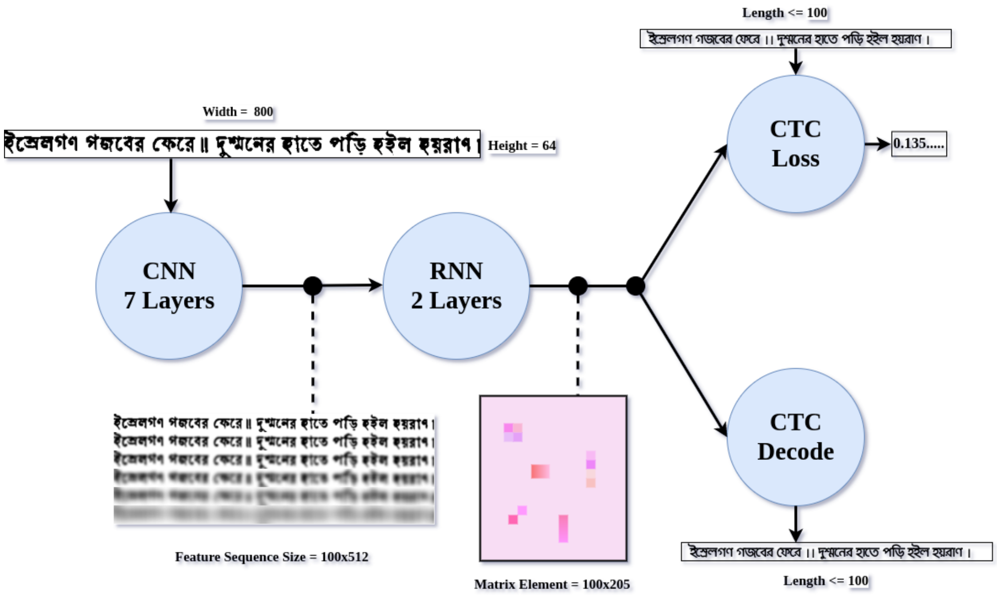
Our recognition system will take line of a image and outputs transcription of the image at unicode. In this context, the following issues are important

- Transcription of the image of a text line needs either recognition of its individual words or the characters (including space). Thus, it requires a word segmentation and/or a character segmentation module. Segmentation of the words from a given line is relatively easier than that of the characters from a word. On the other hand, if a word classifier is employed to solve the problem, the number of classes would be very large and consequently obtaining good accuracy becomes difficult. Therefore, we need to consider a trade-off.

- Implementation of a character classifier needs to include a large number of character classes because Bangla has many modified and conjunct characters and the total number character classes in the present database is 320. But segmentation of characters is error prone and buggy. Although having a good recognizer would give poor result because of incorrect character segmentation. We would recognize at character level without segmentation process. But we don't have any aligned dataset, we only have line images and their corresponding ground truth text.
- Connectionist Temporal Classification (CTC) is a valuable operation to tackle sequence problems where timing is variable. Without CTC, we would need an aligned dataset, which in the case of character recognition, would mean that every character of a transcription, would need to be aligned to its exact location in the image file. Therefore, CTC makes training such a system a lot easier.

A hybrid NN architecture consisting of 7 CNN blocks followed by two RNN blocks (LSTM) is used here. Then CTC best path decoding was integrated. Model overview is shown in FIGURE 3.3.

FIGURE 3.3: CNN layers


Length <= 100
ইস্রেলগণ গজবের ফেরে ।। দুশ্মনের হাতে পড়ি হইল হয়রাণ ।
Width = 800
ইস্রেলগণ গজবের ফেরে ॥ দুশ্মনের হাতে পড়ি হইল হয়রাণ ।
Height = 64
CTC Loss
0.135.....
CNN 7 Layers
RNN 2 Layers
Feature Sequence Size = 100x512
Matrix Element = 100x205
CTC Decode
ইস্রেলগণ গজবের ফেরে ।। দুশ্মনের হাতে পড়ি হইল হয়রাণ ।
Length <= 100

3.4.1 Architecture

Our proposed architecture was mainly inspired by Baoguang *et al.* [39]. It is shown in TABLE 3.6.

TABLE 3.6: Architecture of our model Abbreviations: average (avg), bidirectional (bidir), vertical (vert), dimension (dim), batch normalization (BN), convolutional layer (Conv).

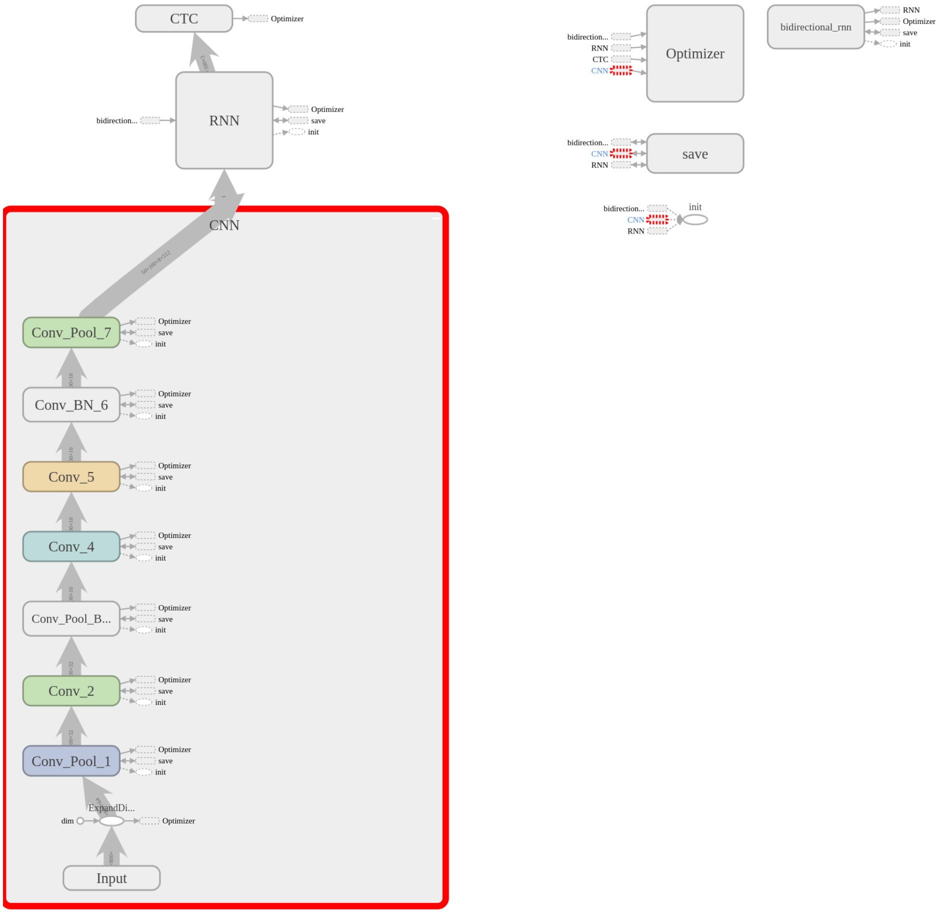
Type	**Description**	**Output Size**
Input	gray-value line-image	800 × 64 × 1
Conv+Pool	kernel 5 × 5, pool 2 × 2	400 × 32 × 64
Conv	kernel 5 × 5	400 × 32 × 128
Conv+Pool+BN	kernel 5 × 5, pool 2 × 2	200 × 16 × 64
Conv	kernel 3 × 3	200 × 16 × 256
Conv	kernel 3 × 3	200 × 16 × 256
Conv+BN	kernel 3 × 3	200 × 16 × 512
Conv+Pool	kernel 5 × 5, pool 2 × 2	100 × 8 × 512
LSTM	bidir, 512 hidden cell	100 × 8 × 512
LSTM	bidir, 512 hidden cell	100 × 8 × 512
Mean	avg. along vert. dim.	100 × 1 × 512
Collapse	remove dimension	100 × 512
Project	project onto 396 classes	100 × 397
CTC	decode or loss	$\leq$ 100

CNN : The input lines are fed into the CNN layers. This block is used to extract relevant feature. Our input image is a gray scaled image of size 800 × 61 × 1. As we turned our image to gray scale we have only one channel instead of three RGB channel. In the first convolution layer, applies kernel of size 5 × 5 having same padding and stride 1. Then a non-linear RELU functions was applied. Finally pooling layer summarizes the region. We used vallid padding for pooling layer which downsized the image input, results in output of size 400 × 32 × 64. Then the second convolution layer applies kernel of size 5 × 5. We extracted feature using 128 different filter thus our output size is 400 × 32 × 128. Third layer takes 400 × 32 × 128 size inputs and applies kernel of size 5 × 5 having same padding and stride 1. Then after applying RELU, pooling layer of size 2 × 2 is applied, which gives output of size 200 × 16 × 64. In fourth and fifth layer kernel size is 3 × 3 with same padding and stride 1. Output of the fifth layer is of size 200 × 16 × 256. Then output is fed into sixth layer. It's same as previous convolution layers but kernel size is 3 × 3. We also used batch normalization to prevent internal covariate shift problem. Output of the sixth layer is of size 200 × 16 × 512. Seventh layer is as same as our first layer which gives output of size 100 × 8 × 512. This output is then fed into LSTM cells. FIGURE 3.4 shows stacked CNN layers.

LSTM : CNN was used as feature extractor. Output of the last convolution block was fed into a BLSTM. This allow us to process temporal order of the CNN output. Feature sequence contains 512 features per time stamp, BLSTM propagates relevant information through this sequence. We used 512 hidden cell in our BLSTM. Output of the BLSTM is 100 × 8 × 512 then we have taken mean along vertical axis and resized it to 100 × 1 × 512. Single dimension is removed and we get feature of shape 100 × 512. Then the output sequence is mapped into a matrix of size 100 × 397. The ISIDDI [3] dataset consists of 90 different characters, but all combination in vertical position must be considered. So we have encoded our ground truth text and we got most frequent 396 single and compound character further one additional character is needed for the CTC operation (CTC blank label), therefore there are 397 entries for each of the 100 time-steps. As our line length can be 100, we have calculate probability of each timestamp. We have used BLSTM here, because it is able to propagate information through longer distances and provides more robust training-characteristics than vanilla RNN. FIGURE 3.5 depicts computation inside LSTM

CTC : During our training session, the CTC is given output of our previous

FIGURE 3.4: CNN layers



block(LSTM) and ground truth text. And it tries to predict characters at each timestamp, if two consecutive characters are same then they are merged. But two consecutive character can occur like মম .

This case is taken care of by inserting a CTC-blank character between those consecutive character. We transformed above Bangla word to ম-ম in our preprocessing step. Till now we have only used best path decoding FIGURE 3.7. We will improve performance by word beam search in our next phase. In case of inferring CTC only give the output matrix of LSTM then it predicts the final text. Here our ground truth and predicted text can be at most 100 characters long. FIGURE 3.6 shows CTC decoder in our model.

FIGURE 3.5: LSTM layers

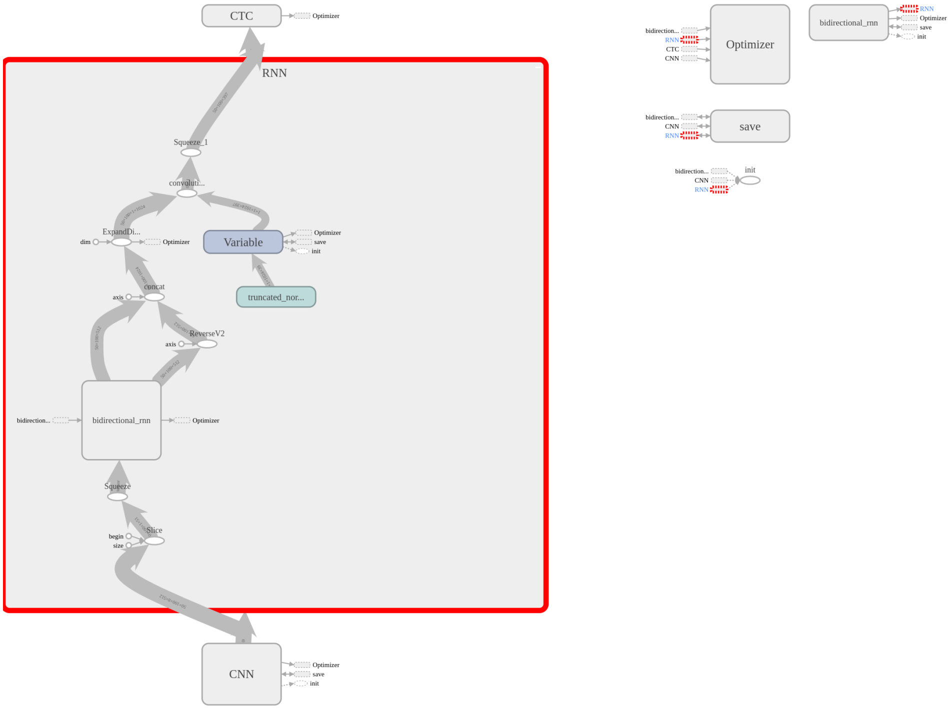
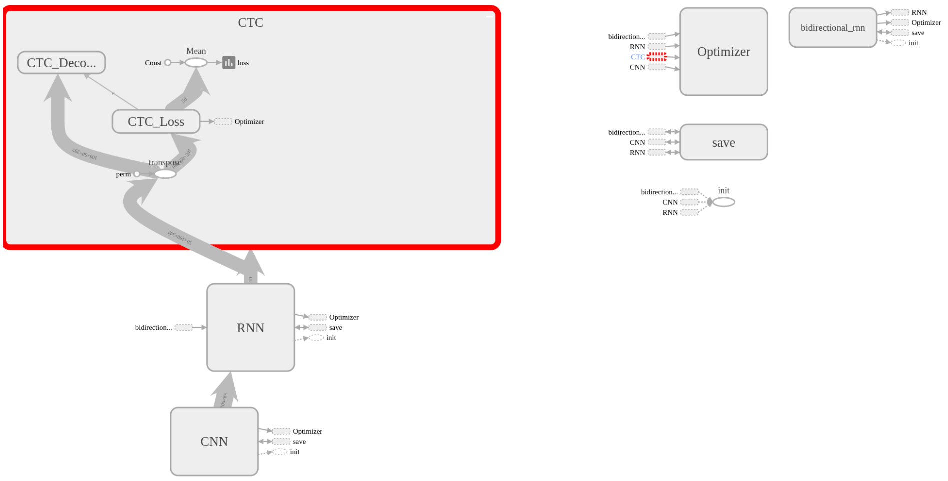


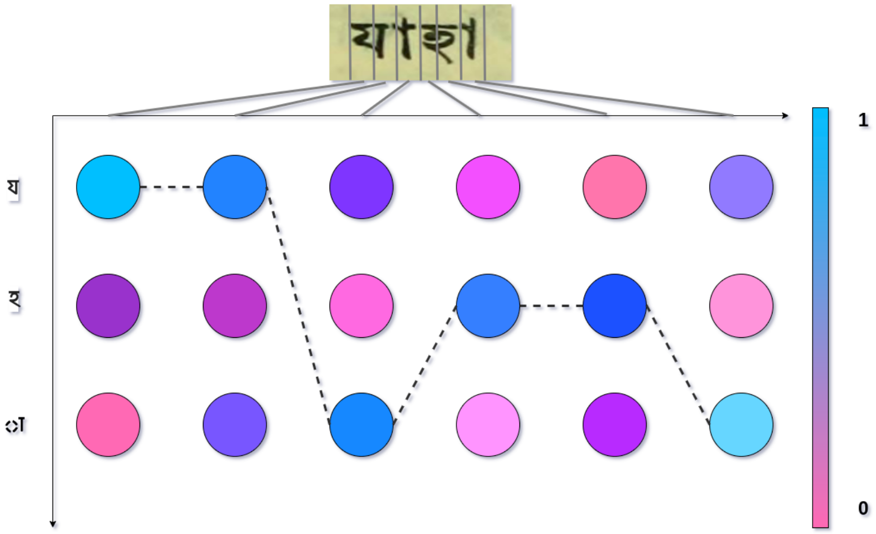
FIGURE 3.6: CTC decoder



3.5 Conclusion

This chapter presented the relevant parts of the proposed OCR system. An algorithm is shown which preprocess our input image page which segments any types of line by clustering algorithm. Then each line in fed into a hybrid architecture. A neural network architecture consisting of a convolutional neural network (CNN),

Figure 3.7: Best path decoding picks the most probable label for each time-step.



two layers of BLSTM cells and a connectionist temporal classification (CTC) layer have been used for recognition of the segmented text lines. We used best path decoding technique which selects character with the highest probability.

Chapter 4

Performance Evaluation

The performance analysis of our project on different aspect is reported in this chapter. Accuracy is measured on either encoded or without encoded version or on character and word level. Some sample line images along with their prediction and error count are provided.

4.1 Introduction

Our result analysis can be described in two different parts.

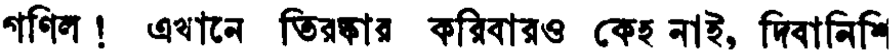
1. Results with encoding
2. Results without encoding

Our proposed scheme was executed upon document images from sources as described in the previous section. These texts were from real publications such as novels, newspaper reports to accurately reflect the distribution of characters in natural Bangla text.

4.2 With Encoding

As we already discussed in Data description section, we don't have annotated the images at each horizontal position (which we call time-step from now on). We have line images and their ground truth text. Our NN training will be guided by CTC loss functions. We only fed the output matrix of RNN and ground truth text to the CTC loss functions. It tries all possible alignment of the groundtruth text and takes sum of all scores. In case of English visual ordering of a text in image is as same as actual ordering of string in groundtruth text. But for Bangla visual order of image and actual ordering is different for some modifier. In FIGURE 4.1 actual ordering of characters is গ ণ ি ল ! এ খ া ন ে ত ি র স ্ ক া র ক র ি ব া র ও ক ে হ ন া ই , দ ি ব া ন ি শ ি but visual ordering is গ ি ণ ল ! এ খ া ে ন ি ত র স ্ ক া র ক ি র ব া র ও ে ক হ ন া ই , ি দ ব া ি ন ি শ. As a result our encoded version performs better then unencoded version. As evaluation metric we have used character error rate (CER). CER is

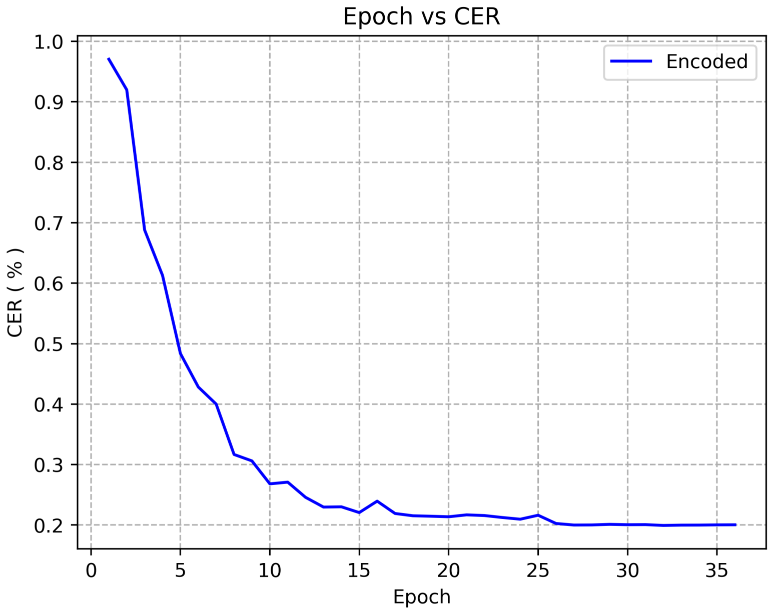
FIGURE 4.1: A single line from dataset



গণিল ! এখানে তিরস্কার করিবারও কেহ নাই, দিবানিশি

calculated by counting the number of editing operations to transfer the recognized text into the ground truth text, divided by the length of the ground truth text [22]. The numerator essentially is the Levenshtein edit-distance and the nominator normalizes this edit-distance. [22]. We have achieved 20% character error rate, that means 80% character accuracy. In FIGURE 4.2 we see observe that after

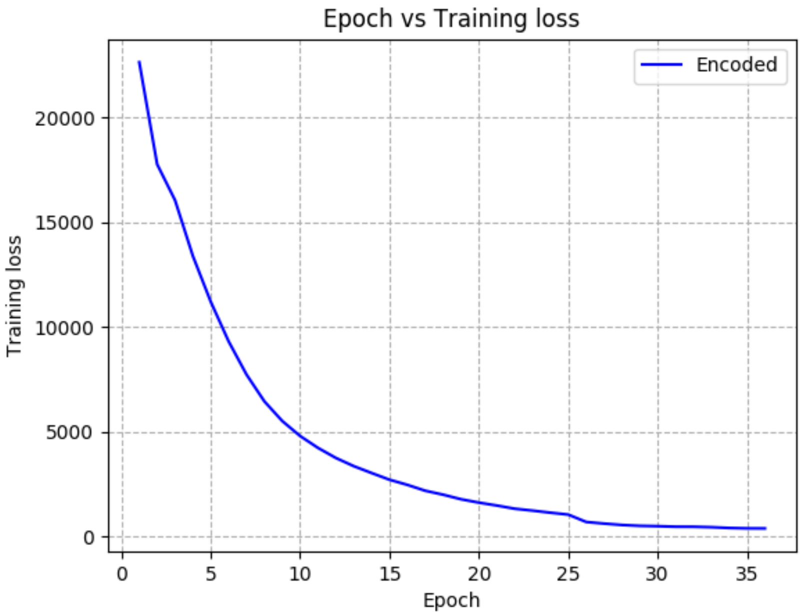
FIGURE 4.2: Epoch vs character error rate in validation set



30 epochs our CER is not decreasing and saturated. Our document image was degraded and too old. After traditional pre-processing we 40% CER which is quite high. Then we pre-processed our dataset properly and achieved CER 20%.

In FIGURE 4.3 we observe first epoch having training loss 22625 and after 36 epochs training loss is decreased to 377.

FIGURE 4.3: Epoch vs training loss in training set



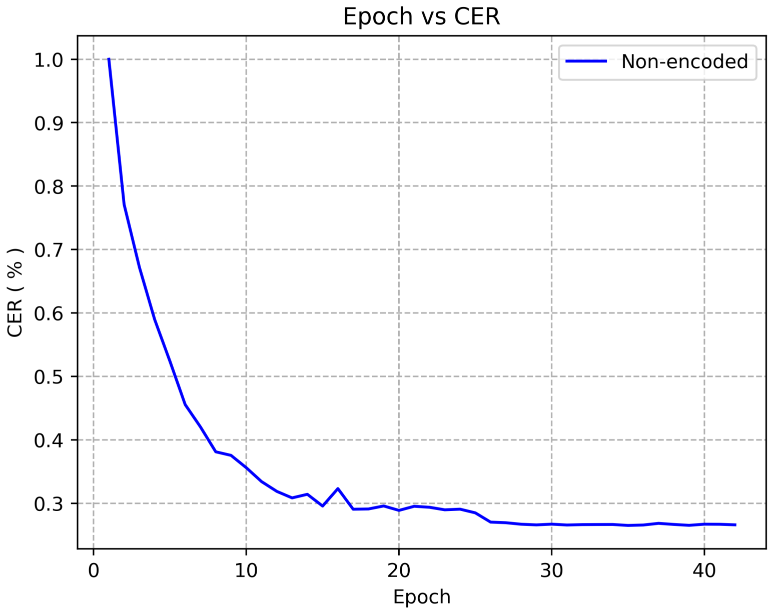
4.3 Without Encoding

We tested our model without character encoding. It gives poor performance relative to encoded part and it's expected. Without encoding, we can't recognize any compound character. In FIGURE 4.4 we see that after 40 epochs we achieved 26.50%. In Table 4.1 comparsion between encoded and non-encoded versions CER and character accuracy is depicted.

TABLE 4.1: Comparison between encoded and non-encoded

Type	CER	Character Accuracy
Encoded	19.95%	80.05%
Non-encoded	26.50%	73.5%

Figure 4.4: Epoch vs character error rate loss in validation set



4.4 OCR's Result on Computer-composed Texts

We ran our OCR on old documents. Output in shown in TABLE 4.2. For each table row first line is image from test set, second line is ground truth text for that image, third line contains output for encoded version and minimum edit distance with groundtruth. The fourth line contains same information as third line for the unencoded version. TABLE 4.2, TABLE 4.3 and TABLE 4.4 shows some example.

4.5 Conclusion

This chapter presented the results for the proposed OCR system. Multiple experiments have been conducted to analyze the influence of different parts of the system.

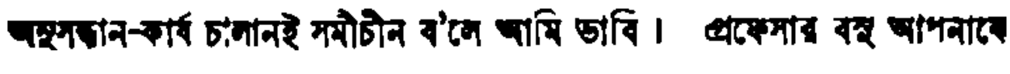
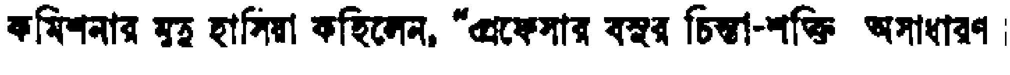
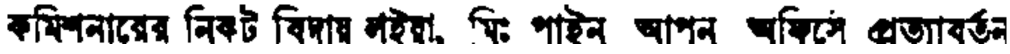
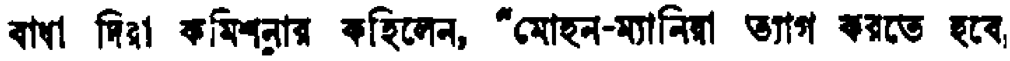
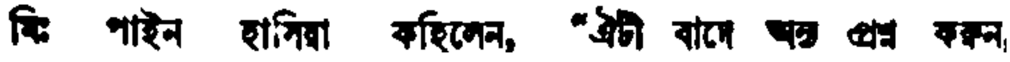
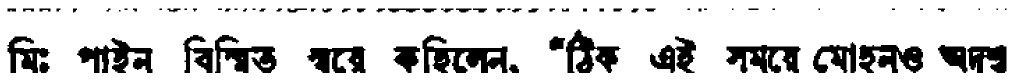
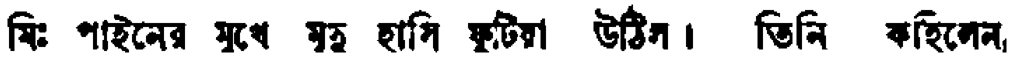
No.	Line Image	
	Ground Truth	
	Prediction with encoding	**Edit Dis.**
	Prediction without encoding	**Edit Dis.**
1		
	অনুসন্ধান-কার্য চালানই সমীচীন ব’লে আমি ভাবি । প্রফেসার বসু আপনাকে	
	অনুসন্ধানকায চলানই সমীটীন ব’লে আমি ভাবি । প্রফেসার বসু আপনাকে	4
	অনসঙধান-কার্য ালানই সমীযীন ব’লে আমি ভাবি । প্রফসার বসু আপনাক	7
2		
	কমিশনার মৃদু হাসিয়া কহিলেন “প্রফেসার বসুর চিন্তাশক্তি অসাধারণ ।	
	কমিশনার মত হাসিয়া কহিলেন “প্রফেসার বসুর চিন্তাশক্তি আধাণ ।	4
	কমিশর মত হাসিয়া কহিলেন, “প্রফেসার বস্তর চিনতা-শকি অসাধারণ ।	14
3		
	কমিশনারের নিকট বিদায় লইয়া, মিঃ পাইন আপন অফিসে প্রত্যাবর্তন	
	কমিশনারের নিকট বিদায় ইয়া মিঃ পাইন আপন অফিসে প্রত্যবখন আধাণ ।	3
	কমিশনারের নিকট বিদায় হাইয়া, মি পাইন আপন অফিসে প্রভ্যাবর্তন অসাধারণ ।	8
4		
	বাধা দিয়া কমিশনার কহিলেন “মোহনম্যানিয়া ত্যাগ করতে হবে	
	বাধা দিয়া কমিনার কহিলেন “মোহনম্যানি়া ত্যাগ করতে হবে	2
	বাধা দিয়া কমিশনার কহিলেন, “মোহন-দ্যানিয়া ত্যাগ করতে হযে,	6
5		
	মি পাইন হাসিয়া কহিলেন “ঐটী বাদে অন্য প্রশ্ন করুন	
	মিঃ পাইন হাসিয়া কহিলেন “ঐটী বাদে অন্য প্রশ্ন কন	2
	ি পাইন হাসিয়া কহিলেন, “ইটী বো্ অন্ু প্রা্ করুন,	12
6		
	মিঃ পাইন বিস্মিত স্বরে কহিলেন “ঠিক এই সময়ে মোহনও অদৃশ্য	
	কি পাইর দিরি মনে মরিশে তিীর হৈ অনে বাধন ও ব়ং ন কন	31
	কি পাইন দিনিক সেন করিযেখে ধরীইক এই বেক পোধেও হরয	36
7		
	মিঃ পাইনের মুখে মৃদু হাসি ফুটিয়া উঠিল । তিনি কহিলেন	
	ছিঃ পাইনের মুখে ময় হাসি ফুট়া উঠিল । তিনি কহিলেন	4
	মিঃানের মুখে মতু চাসি ঘুটিয়া উঠিল । তিনি কহিলেন,	9

TABLE 4.2: Line image with ground truth text and minimum edit distance in encoded and non-encoded version

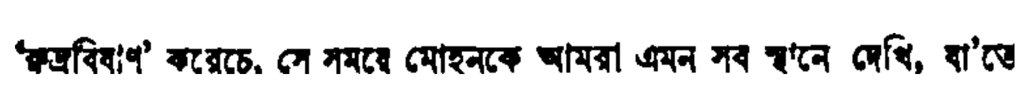
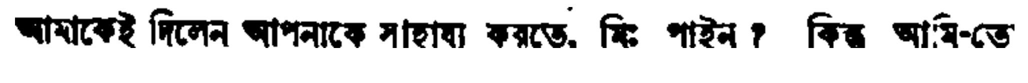
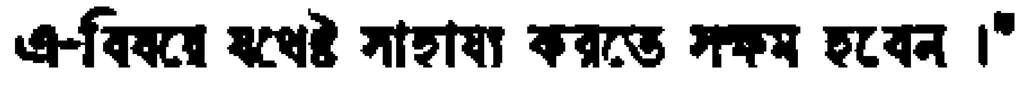
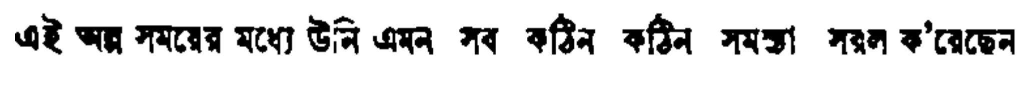
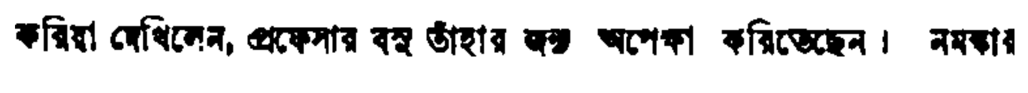
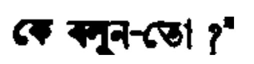
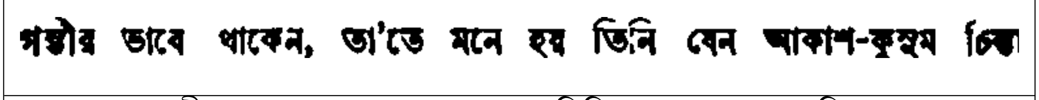
8	 'রুদ্রবিষাণ' করেচে. সে সময়ে মোহনকে আমরা এমন সব স্থানে দেখি, যা'তে	
	রুদ্রবিষাণ’ করেচে সে সময়ে মোহনকে আমরা এমন সব স্থানে দেখি যা’তে	
	তদ্রবিষাণ’ কয়রেচে সে সময়ে মোহনকে আমরা এমন সব স্থানে দেখি যাতে	3
	'নদরিষাণ’ করেছে, সে সময়ে মোহনকে আমরা এমন সব সথানে দেখি, যা’তে	6
9	 আমাকেই দিলেন আপনাকে সাহায্য করতে. মিঃ পাইন ? কিন্তু আমি-তো	
	আমাকেই দিলেন আপনাকে সাহায্য করতে মিঃ পাইন কিন্তু আমিতো	
	আমাকেই দিলেন আপনকে সাহায্য করতে মিঃ পাইন কিন্তু আমিত	2
	আমাকেই দিলেন আপনাকে সাহায্য করতে, মি পাইন ? ক্িতু আমি-তো	3
10	 এ-বিষয়ে যথেষ্ট সাহায্য করতে সক্ষম হবেন ।"	
	এবিষয়ে যথেষ্ট সাহায্য করতে সক্ষম হবেন ।	
	একিসে মুখই সাহাস করতে বলয় হবেন	16
	একিযে মেই মাবাস কুরতে শন্য কবেন ”	24
11	 এই অল্প সময়ের মধ্যে উনি এমন সব কঠিন কঠিন সমস্যা সরল ক'রেছেন	
	এই অল্প সময়ের মধ্যে উনি এমন সব কঠিন কঠিন সমস্যা সরল ক’রেছেন	
	এই অ সময়ের মধ্যে উনি এমন সব কঠিন কঠিন সমস্্া সল করেছেন	4
	এই অ সময়ের মধ্যে উনি এমন সব কঠিন কঠি সমস্যা সরল ক’রেছেন	4
12	 করিয়া দেখিলেন, প্রফেসার বসু তাঁহার জন্য অপেক্ষা করিতেছেন । নমস্কার	
	করিয়া দেখিলেন প্রফেসার বসু তাঁহার জন্য অপেক্ষা করিতেছেন । নমস্কার	
	করিয়া দেখিলেন প্রফেসার যসু তাঁহার জন্য অপেক্ষা করিতেছেন । নমার	2
	করিরা দেখিলেন, প্রফেসার বস্ তাঁহার জন্য অপেক্ষা করিতেছেন । নমসকার	3
13	 কে বলুন-তো ?"	
	কে বলুনতো	
	কে বুলনতো ”	2
	কে বলচকসো ?”	5
14	 গম্ভীর ভাবে থাকেন, তা'তে মনে হয় তিনি যেন আকাশ-কুসুম চিন্তা	
	গম্ভীর ভাবে থাকেন তা’তে মনে হয় তিনি যেন আকাশকুসুম চিন্তা	
	গহীর ভাবে থাকেন তা’তে মনে হয় তিনি যেন আকাশকুরুম বিযন্তা	4
	গরীর ভাবে থাকেন, তা’তে মনে হয় তিনি যেন আকাশ-কুরুম চন্য	7

Table 4.3: Line image with ground truth text and minimum edit distance in encoded and non-encoded version

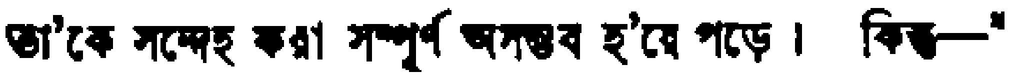
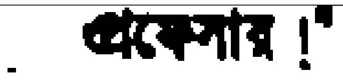
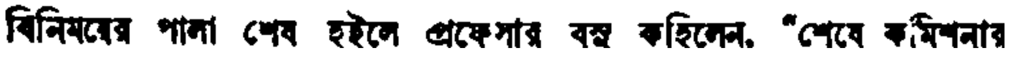
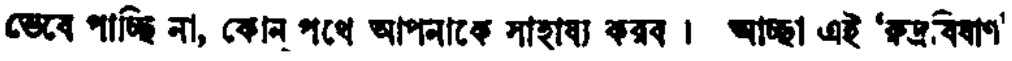
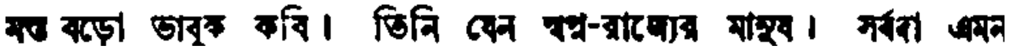
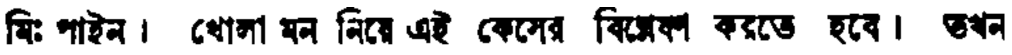
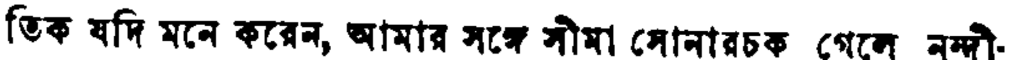
15	 তা'কে সন্দেহ করা সম্পূর্ণ অসম্ভব হ'য়ে পড়ে । কিন্তু—"	
	তা’কে সন্দেহ করা সম্পূর্ণ অসম্ভব হয়ে পড়ে । কিন্তু—”	
	ত’কে অঙ্গের করা স্পস্পর্ণ অসুম হ’য়ে গে় । কিন্তু	14
	ত ’ কতে সল্◌েত কনা স্পর্ণ অক্ষ় হ’যে পড়ে । কিু—”	19
16	 প্রফেসার !"	
	প্রফেসার ”	
	পূকেসার ।”	3
	পফেসার ।”	4
17	 বিনিময়ের পালা শেষ হইলে প্রফেসার বসু কহিলেন. "শেষে কমিশনার	
	বিনিময়ের পালা শেষ হইলে প্রফেসার বসু কহিলেন “শেষে কমিশনার ।	
	বিনিময়ের পালা শেষ হইলে প্রফেসর বসু কহিলেন “শেষে কমিশনার	1
	বিনিষয়ের পালা শেষ হইলে প্রকেসার বস় কহিলেন, “শেযে কমিশনার	4
18	 ভেবে পাচ্ছি না, কোন্ পথে আপনাকে সাহায্য করব । আচ্ছা এই 'রুদ্র.বিষাণ'	
	ভেবে পাচ্ছি না কোন্ পথে আপনাকে সাহায্য করব । আচ্ছা এই রুদ্রবিষাণ	
	ভেবে পাচ্ছি না কোন পথে আপনাকে সাহায্য করুব । আক্রা এই রুদ্রদবযাণ	6
	◌েবে পান্ছি না, কোন পথে আপনাকে সাহায্য করব । আন্ছা এই 'রুদ্রবিধুণ	8
19	 মস্ত বড়ো ভাবুক কবি । তিনি যেন স্বপ্ন-রাজ্যের মানুষ । সর্বদা এমন	
	মস্ত বড়ো ভাবুক কবি । তিনি যেন স্বপ্নরাজ্যের মানুষ । সর্বদা এমন	
	মস্ত বুড়ো ভাবুক কবি । তিনি যেন স্বপ্নরাজ্যর মানুষ । সর্ববা এন	4
	মসন্ত বতো ভাবক কবি । তিনি যেন স্বপ্-রাজযের মানুষ । সর্বা এনন	8
20	 মিঃ পাইন । খোলা মন নিয়ে এই কেসের বিশ্লেষণ করতে হবে । তখন	
	মিঃ পাইন । খোলা মন নিয়ে এই কেসের বিশ্লেষণ করতে হবে । তখন	
	নিঃ পাইন । খোলা মন নিয়ে এই কেসের বিশ্লোষণ করতে হবে । তন	3
	মি পিইন । খোলা মন নিয়ে এই কেসের ব্িলেষণ করতে হবে । তখন	5
21	 তিক যদি মনে করেন, আমার সঙ্গে সীমা সোনারচক গেলে নন্দী.	
	তিক যদি মনে করেন আমার সঙ্গে সীমা সোনারচক গেলে নন্দী	
	তিক যদি মনে করেন আমার সঙ্গে সীমা মোনারচক গেলে নমী	2
	তিক যদি মনে করেন, আমার সঙ্গে সীমা যোনারহক গেলে নৃ◌র্ী	7

TABLE 4.4: Line image with ground truth text and minimum edit distance in encoded and non-encoded version

Chapter 5

Conclusion

The Final chapter describes the findings and discovery we made while developing an OCR for Bangla language. Several problems and drawbacks of our work as well as the recommendations for our project are described here. Also numerous scopes for improvement are enlisted for further development.

5.1 Findings and Recommendations

The fact that Bangla is our mother tongue was not the sole impetus behind our work, rather the colossal need of digitizing the literature and the official documents attested the necessity to us. Our aim is to build an OCR engine for Bangla printed documents. We have not been able to assemble such a complete OCR engine yet, but we will build a complete OCR for Bangla at the end of our work. Our proposed model of segmentation of text images by lines via clustering method and optimization by recognizing text in line level is Unique contribution in development for OCR for Bangla language. On the other hand our recognizing technique is especially relevant for Bangla and other related scripts where cursiveness and displacement are principal traits. More variability is added to the data by using further data augmentation methods. This includes various brightness and contrast setting, kernel and techniques for image manipulation. This resulted in slightly better clustering purity. This model can be trained and tested on handwritten text as well. A dataset containing Bangla handwritten script of various factors are required for further developing the system. A language model has been developed for having a prior knowledge of word semantics. An auto correction system have been designed to correct a word which is few edit distances away from ground truth.

5.2 Future Work

Despite our OCR system performs with accuracy and speed, it lacks in several aspects in different phases.

- Our algorithm work on neither hybrid page layout nor paragraph with significantly different font sizes. The model will be developed so that it can detect images, tables and text with different language and fonts.
- Preprocessing methods should be made more rigid specially for degraded images for making the performance of the classifier fairly more robust. Skew correcting and slanting removal of the text results in text lying approximately in horizontal direction enabling tighter cropping of the text-lines.
- Density based clustering method should be implemented with modification for rather than using traditional python library. In both cases the several parameters for clustering method should be able to tune dynamically via paragraph analysis.

Bibliography

[1] A comprehensive guide to convolutional neural networks — the eli5 way. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way. (Accessed on 11/04/2019).

[2] Cuda - wikipedia. https://en.wikipedia.org/wiki/CUDA. (Accessed on 11/04/2019).

[3] Indian script character databases. https://www.isical.ac.in/~ujjwal/download/database.html. (Accessed on 11/05/2019).

[4] Numpy — numpy. https://numpy.org/. (Accessed on 11/04/2019).

[5] Opencv. https://opencv.org/. (Accessed on 10/24/2019).

[6] Otsu's method - wikipedia. https://en.wikipedia.org/wiki/Otsu%27s_method. (Accessed on 10/24/2019).

[7] Rmsprop - engmrk. https://engmrk.com/rmsprop/. (Accessed on 12/29/2019).

[8] Rmsprop - wiki | golden. https://golden.com/wiki/RMSprop. (Accessed on 12/29/2019).

[9] sklearn.cluster.optics — scikit-learn 0.21.3 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.OPTICS.html#sklearn.cluster.OPTICS. (Accessed on 11/04/2019).

[10] Tensorflow. https://www.tensorflow.org/. (Accessed on 10/24/2019).

[11] Understanding lstm networks – colah's blog. https://colah.github.io/posts/2015-08-Understanding-LSTMs/. (Accessed on 11/05/2019).

[12] Understanding of convolutional neural network (cnn) — deep learning. https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning. (Accessed on 11/04/2019).

[13] Understanding rmsprop — faster neural network learning. https://towardsdatascience.com/

understanding-rmsprop-faster-neural-network-learning-62e116fcf29a. (Accessed on 12/29/2019).

[14] Understanding rnn and lstm - towards data science. https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e. (Accessed on 11/05/2019).

[15] – bangla book. bangla boi. bengali books. https://www.ebanglalibrary.com/. (Accessed on 12/30/2019).

[16] Muaz Ahmed. Urdu optical recognition system. *Unpublished MS Thesis, Department of Computer Science, National University of Computer Engineering Science*, 2010.

[17] Adnan Amin. Recognition of arabic handprinted mathematical formulas. *Arabian Journal for Science and Engineering*, 16(4):531–542, 1991.

[18] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: ordering points to identify the clustering structure. In *ACM Sigmod record*, volume 28, pages 49–60. ACM, 1999.

[19] Veena Bansal and RMK Sinha. Segmentation of touching and fused devanagari characters. *Pattern recognition*, 35(4):875–893, 2002.

[20] Mohammed Bennamoun and Boualem Boashash. A structural-description-based vision system for automatic object recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 27(6):893–906, 1997.

[21] Tapan Kumar Bhowmik, Swapan Kumar Parui, Utpal Roy, and Lambert Schomaker. Bangla handwritten character segmentation using structural features: A supervised and bootstrapping approach. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 15(4):29, 2016.

[22] Théodore Bluche. *Deep neural networks for large vocabulary handwritten text recognition*. PhD thesis, 2015.

[23] BB Chaudhuri and U Pal. A complete printed bangla ocr system. *Pattern recognition*, 31(5):531–549, 1998.

[24] Ayan Chaudhury and Ujjwal Bhattacharya. Efficient segmentation of characters in printed bengali texts. In *International Conference on Eco-friendly Computing and Communication Systems*, pages 389–397. Springer, 2012.

[25] Anthony Cheung, Mohammed Bennamoun, and Neil W Bergmann. An arabic optical character recognition system using recognition-based segmentation. *Pattern recognition*, 34(2):215–233, 2001.

[26] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning*, pages 1764–1772, 2014.

[27] Deepika Gupta and Soumen Bag. Handwritten multilingual word segmentation using polygonal approximation of digital curves for indian languages. *Multimedia Tools and Applications*, pages 1–26, 2019.

[28] Muhammad Asif Hossain Khan, Anindya Sundar Paul, Muhammad Jawad Iqbal, and Mohammad Shoyaib. Printed bangla character image segmentation: A font invariant approach. 2019.

[29] Laurence Likforman-Sulem, Abderrazak Zahour, and Bruno Taconet. Text line segmentation of historical documents: a survey. *International Journal of Document Analysis and Recognition (IJDAR)*, 9(2-4):123–138, 2007.

[30] Meem Arafat Manab, Muhammad Asif Hossain Khan, and Melody Soptaka Kunder. Partial projection profile based character segmenter and transformation-invariant recognizer for printed bangla documents. 2016.

[31] S Manisha and T Sree Sharmila. Effective printed tamil text segmentation and recognition using bayesian classifier. In *Computational Intelligence in Data Mining*, pages 729–738. Springer, 2017.

[32] CG. Marquez and G. Rabassa. *One Hundred Years of Solitude*, page 179. New York: Avon Press, 1971.

[33] Sonika Rani Narang, MK Jindal, and Munish Kumar. Line segmentation of devanagari ancient manuscripts. *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, pages 1–8, 2019.

[34] Sk Md Obaidullah, Chitrita Goswami, KC Santosh, Nibaran Das, Chayan Halder, and Kaushik Roy. Separating indic scripts with matra for effective handwritten script identification in multi-script documents. *International Journal of Pattern Recognition and Artificial Intelligence*, 31(05):1753003, 2017.

[35] Moisés Pastor. Text baseline detection, a single page trained system. *Pattern Recognition*, 94:149–161, 2019.

[36] Satadal Saha, Subhadip Basu, Mita Nasipuri, and Dipak Kr Basu. A hough transform based technique for text segmentation. *arXiv preprint arXiv:1002.4048*, 2010.

[37] Ram Sarkar, Samir Malakar, Nibaran Das, Subhadip Basu, Mahantapas Kundu, and Mita Nasipuri. A font invariant character segmentation technique for printed bangla word images. In *Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012 (INDIA 2012) held in Visakhapatnam, India, January 2012*, pages 739–746. Springer, 2012.

[38] Ram Sarkar, Samir Malakar, Nibaran Das, Subhadip Basu, and Mita Nasipuri. A script independent technique for extraction of characters from handwritten word images. *International Journal of Computer Applications*, 1(23):85–90, 2010.

[39] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304, 2016.

[40] Irwin Sobel. An isotropic 3x3 image gradient operator. *Presentation at Stanford A.I. Project 1968*, 02 2014.

[41] Narasimha Reddy Soora and Parag S Deshpande. Review of feature extraction techniques for character recognition. *IETE Journal of Research*, 64(2):280–295, 2018.

[42] Tasnim Zahan, Muhammed Zafar Iqbal, Mohammad Reza Selim, and Mohammad Shahidur Rahman. Connected component analysis based two zone approach for bangla character segmentation. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–4. IEEE, 2018.