B.Sc. in Computer Science and Engineering Thesis

Bangla Word Segmentation Technique And Bangla OCR Using Convolutional Neural Network and Recurrent Neural Network

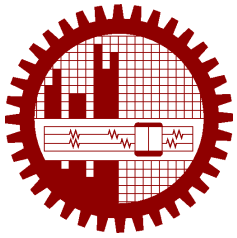
Submitted by

Sabab Aosaf
201305019

Shusmoy Chowdhury
201305108

Supervised by

Dr. M. Sohel Rahman



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

Dhaka, Bangladesh

October 2018

CANDIDATES' DECLARATION

This is to certify that the work presented in this thesis, titled, "Bangla Word Segmentation Technique And Bangla OCR Using Convolutional Neural Network and Recurrent Neural Network", is the outcome of the investigation and research carried out by us under the supervision of Dr. M. Sohel Rahman.

It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Sabab Aosaf
201305019

Shusmoy Chowdhury
201305108

CERTIFICATION

This thesis titled, **"Bangla Word Segmentation Technique And Bangla OCR Using Convolutional Neural Network and Recurrent Neural Network"**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in October 2018.

Group Members:

Sabab Aosaf

Shusmoy Chowdhury

Supervisor:

Dr. M. Sohel Rahman
Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology

ACKNOWLEDGEMENT

First of all, the authors would like to express gratitude to the Almighty God for unlimited kindness and blessings to fulfill the thesis work successfully.

The authors wish to express their heartiest gratitude and sincere thanks to supervisor Dr. M. Sohel Rahman, Professor, Department of Computer Sceince and Engineering, BUET for his proper guidance, suggestions and continuous supervision at all stages of work. The authors are indebted to him for his affectionate encouragement and helpful co-operation throughout the thesis work.

The authors would also like to thank Mr. Saddam for his support and guidance.

The authors would like to express their heartiest thanks to their families and friends for their encouragement and support this work.

Finally, the authors would like to thank all who helped them in making a successful and a productive completion of the thesis.

Dhaka
October 2018

Sabab Aosaf

Shusmoy Chowdhury

Contents

List of Figures

List of Tables

ABSTRACT

Now a Days Bengali is one of the most spoken language in the world. Around 200 million people all over the world have been speaking in this language.Being the 5th Position and sweetest language in the world declared by the UNESCO Bengali is the national language in Bangladesh and one of the major languages in India. Bangla is a rich and old language. Converting hard document, such as newspaper, printed book into editable text to modify or extend is the normal practice nowadays, OCR is the process of converting printed text images into editable text.

In our thesis work we have tried to make a complete Optical Character Recognition for Bangla Text. In our first thesis work we have introduced a new segmentation technique for bangla text. The segmentation of bangla text is not easy as other language because of many connecting word in bangla. Moreover if the image is not taken straight the output may be wrong. In our technique we first check if the image is straight or rotated. If the image is rotated then we make it straight and divide the text in many contours. We then segment the words according to the threshold value of the page and ignore the noise and punctuation of the page. Our technique has given a 95% accuracy in segmenting bangla text

For generating text from the segmented words we use Convolution Neural Network. At first we used no layer connected Convolution Neural Network. But we got a poor accuracy of 45% in this technique. Then we used fully connected Convolution Layer with stride and dropout. We have used a lot of activation function in our CNN to increase the accuracy level. The Relu and No activation function has given a good accuracy in CNN. We also used softmax_cross_entropy_with_logits as cost function in the CNN. We used Stochastic gradient descent as the optimizer in the CNN. We trained the machine with 1023 classes each having 50 different images of different fonts. Our CNN has given an accuracy of 90.1%.

We also used Recurrent Neural Network to validate the sentences if they are grammatically right or wrong. We LSTM cells in RNN to validate the sentences.

Chapter 1

Introduction

Optical character recognition is the mechanical or electronic conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo or from subtitle text superimposed on an image. It is widely used as a form of information entry from printed paper data records, whether passport documents, invoices, bank statements, computerized receipts, business cards, mail, printouts of static-data, or any suitable documentation. It is a common method of digitizing printed texts so that they can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as cognitive computing, machine translation, text-to-speech, key data and text mining. OCR is a field of research in pattern recognition, artificial intelligence and computer vision.

Early optical character recognition may be traced to technologies involving telegraphy and creating reading devices for the blind. In 1914, Emanuel Goldberg developed a machine that read characters and converted them into standard telegraph code. Concurrently, Edmund Fournier d'Albe developed the Optophone, a hand held scanner that when moved across a printed page, produced tones that corresponded to specific letters or characters.In the late 1920s and into the 1930s Emanuel Goldberg developed what he called a "Statistical Machine" for searching microfilm archives using an optical code recognition system. In 1931 he was granted USA Patent number 1,838,389 for the invention. The patent was acquired by IBM.

In 1974, Ray Kurzweil started the company Kurzweil Computer Products, Inc. and continued development of omni-font OCR, which could recognise text printed in virtually any font (Kurzweil is often credited with inventing omni-font OCR, but it was in use by companies, including CompuScan, in the late 1960s and 1970s[3][7]). Kurzweil decided that the best application of this technology would be to create a reading machine for the blind, which would allow blind people to have a computer read text to them out loud. This device required the invention of two enabling technologies the CCD flatbed scanner and the text-to-speech synthesiser. On January 13, 1976, the successful finished product was unveiled during a widely reported news

conference headed by Kurzweil and the leaders of the National Federation of the Blind. In 1978, Kurzweil Computer Products began selling a commercial version of the optical character recognition computer program. LexisNexis was one of the first customers, and bought the program to upload legal paper and news documents onto its nascent online databases. Two years later, Kurzweil sold his company to Xerox, which had an interest in further commercializing paper-to-computer text conversion. Xerox eventually spun it off as Scansoft, which merged with Nuance Communications.[citation needed] The research group headed by A. G. Ramakrishnan at the Medical intelligence and language engineering lab, Indian Institute of Science, has developed PrintToBraille tool, an open source GUI frontend[8] that can be used by any OCR to convert scanned images of printed books to Braille books.

In the 2000s, OCR was made available online as a service (WebOCR), in a cloud computing environment, and in mobile applications like real-time translation of foreign-language signs on a smart phone. Early versions of OCR were needed to be trained with images of each character, and worked on one font at a time. But the advanced systems are capable of producing a high degree of recognition accuracy for most fonts with support for a variety of digital image file format inputs.Some systems are capable of reproducing formatted output that closely approximates the original page including images, columns, and other non-textual components.

With the advent of smart-phones and smart glasses, OCR can be used in internet connected mobile device applications that extract text captured using the device's camera. These devices that do not have OCR functionality built into the operating system will typically use an OCR API to extract the text from the image file captured and provided by the device. The OCR API returns the extracted text, along with information about the location of the detected text in the original image back to the device app for further processing or display.

There are two basic types of core OCR algorithm, which may produce a ranked list of candidate character.

Matrix matching involves comparing an image to a stored glyph on a pixel-by-pixel basis; it is also known as ”pattern matching”, ”pattern recognition”, or ”image correlation”. This relies on the input glyph being correctly isolated from the rest of the image, and on the stored glyph being in a similar font and at the same scale. This technique works best with typewritten text and does not work well when new fonts are encountered. This is the technique the early physical photocell-based OCR implemented.

Feature extraction decomposes glyph into ”features” like lines, closed loops, line direction, and line intersections. The extraction features reduces the dimensionality of the representation and makes the recognition process computationally efficient. These features are compared with an abstract vector-like representation of a character, which might reduce to one or more glyph prototypes. General techniques of feature detection in computer vision are applicable to this type of OCR, which is commonly seen in ”intelligent” handwriting recognition and indeed most

modern OCR software. Nearest neighbour classifiers such as the k-nearest neighbors algorithm are used to compare image features with stored glyph features and choose the nearest match.

Software such as Cuneiform and Tesseract use a two-pass approach to character recognition. The second pass is known as ”adaptive recognition” and uses the letter shapes recognized with high confidence on the first pass to recognize better the remaining letters on the second pass. This is advantageous for unusual fonts or low-quality scans where the font is distorted.

But we need some pre-processing of data before we apply the algorithm of the OCR.Text Segmentation is the main part of the pre processing. We have to segment the text of the line into small parts. Text segmentation is the process of dividing written text into meaningful units, such as words, sentences, or topics. The term applies both to mental processes used by humans when reading text, and to artificial processes implemented in computers, which are the subject of natural language processing. The problem is non-trivial, because while some written languages have explicit word boundary markers, such as the word spaces of written English and the distinctive initial, medial and final letter shapes of Arabic, such signals are sometimes ambiguous and not present in all written languages.

Word segmentation is the problem of dividing a string of written language into its component words.In English and many other languages using some form of the Latin alphabet, the space is a good approximation of a word divider (word delimiter), although this concept has limits because of the variability with which languages emically regard collocations and compounds. Many English compound nouns are variably written with a corresponding variation in whether speakers think of them as noun phrases or single nouns; there are trends in how norms are set, such as that open compounds often tend eventually to solidify by widespread convention, but variation remains systemic. In contrast, German compound nouns show less orthographic variation, with solidification being a stronger norm.However, the equivalent to the word space character is not found in all written scripts, and without it word segmentation is a difficult problem. Languages which do not have a trivial word segmentation process include Chinese, Japanese, where sentences but not words are delimited, Thai and Lao, where phrases and sentences but not words are delimited, and Vietnamese, where syllables but not words are delimited.In some writing systems however, such as the Ge’ez script used for Amharic and Tigrinya among other languages, words are explicitly delimited (at least historically) with a non-white space character.The Unicode Consortium has published a Standard Annex on Text Segmentation, exploring the issues of segmentation in multi script texts.Word splitting is the process of parsing concatenated text to infer where word breaks exist.Word splitting may also refer to the process of hyphenation.Compare speech segmentation, the process of dividing speech into linguistically meaningful portions.There are two approach of segmentation.One is letter based approach and the other one is word based approach.

Automatic segmentation is the problem in natural language processing of implementing a com-

puter process to segment text.When punctuation and similar clues are not consistently available, the segmentation task often requires fairly non-trivial techniques, such as statistical decision-making, large dictionaries, as well as consideration of syntactic and semantic constraints. Effective natural language processing systems and text segmentation tools usually operate on text in specific domains and sources. As an example, processing text used in medical records is a very different problem than processing news articles or real estate advertisements.

The process of developing text segmentation tools starts with collecting a large corpus of text in an application domain. There are two general approaches: Manual analysis of text and writing custom software Annotate the sample corpus with boundary information and use machine learning

Some text segmentation systems take advantage of any markup like HTML and know document formats like PDF to provide additional evidence for sentence and paragraph boundaries

The OCR result can be stored in the standardized ALTO format, a dedicated XML schema maintained by the United States Library of Congress.For a list of optical character recognition software see Comparison of optical character recognition software. OCR accuracy can be increased if the output is constrained by a lexicon a list of words that are allowed to occur in a document. This might be, for example, all the words in the English language, or a more technical lexicon for a specific field. This technique can be problematic if the document contains words not in the lexicon, like proper nouns. Tesseract uses its dictionary to influence the character segmentation step, for improved accuracy.[22]

The output stream may be a plain text stream or file of characters, but more sophisticated OCR systems can preserve the original layout of the page and produce, for example, an annotated PDF that includes both the original image of the page and a search able textual representation.”Near-neighbor analysis” can make use of co-occurrence frequencies to correct errors, by noting that certain words are often seen together. For example, ”Washington, D.C.” is generally far more common in English than ”Washington DOC”.Knowledge of the grammar of the language being scanned can also help determine if a word is likely to be a verb or a noun, for example, allowing greater accuracy.The Levenshtein Distance algorithm has also been used in OCR post-processing to further optimize results from an OCR API.

In recent years, the major OCR technology providers began to tweak OCR systems to better deal with specific types of input. Beyond an application-specific lexicon, better performance can be had by taking into account business rules, standard expression, or rich information contained in color images. This strategy is called ”Application-Oriented OCR” or ”Customized OCR”, and has been applied to OCR of license plates, invoices, screen shots, ID cards, driver licenses, and automobile manufacturing.Various commercial and open source OCR systems are available for most common writing systems, including Latin, Cyrillic, Arabic, Hebrew, Indic, Bengali (Bangla), Devanagari, Tamil, Chinese, Japanese, and Korean characters.

Now a Days Bengali is one of the most spoken language in the world. Around 200 million people all over the world have been speaking in this language.Being the 5th Position and sweetest language in the world declared by the UNESCO Bengali is the national language in Bangladesh and one of the major languages in India. Bangla is a rich and old language. Converting hard document, such as newspaper, printed book into editable text to modify or extend is the normal practice nowadays, OCR is the process of converting printed text images into editable text. Optical Character Recognition using optical techniques such as mirrors & lenses and Digital Character Recognition using scanners and computer algorithms were originally considered separate fields. Since few applications survive using true optical techniques, the OCR has been broadened to digital image processing as well.

Bengali OCR involves reading text from paper and translating the images into a form (say ASCII code/Unicode) that the computer can manipulate. OCR system is still in preliminary level in case of language like Bengali, cause of its complexities in character shapes, top bars and end bars. More over it has some modified vowel and compound characters.

But it is a matter of great regrets that there has been not so much computerization done in this field. For computerization of any language, one of the vital tasks is to develop an efficient and effective Optical Character Recognition (OCR) system for the respected language so that we can store million pages of paper document in electronic form. . For Bangla, there is no good OCR solution till now.Lot of researches has been done to recognize Bengali, English and other major languages using Optical Character Recognition (OCR). To recognize Bengali character from text images and convert into editable text, Self Organizing Map (SOM) - kind of neural network has been used. To collect the character, documents are scanned, which is pre processed with the Image to Binary Conversion Algorithm. In the binary image, character area is represented by 0 (zero) and rest of the image area is represented with 1 (one). After detecting and correcting the skew and noise, the binary image is processed and grouped, which can be mapped and recognized by SOM. Considering efficiency and fastness, character grouping process has been introduced But, our government and private organizations have huge quantities of Bangla paper documents that are so important that those should be stored for a long period of time. To do so, making electronic copies of those documents are unparalleled and it can be done by using a high-quality Bangla OCR system. But, the written form of Bangla documents is more complex than that of many other languages, Bangla script segmentation is of great importance for creating a Bangla OCR system.

Our objective of thesis is to make a complete Optical Recognition for Bangla language for printed documents. It could be very useful in digitizing the deeds and official documents of the court. This document are typed by the typewriter. So we do not get soft document of this data. By using the OCR we can get the soft document of this data so that we can store this data an use it when it is needed. Many Research work are done in the field of Bangla OCR . But all this work are done for single font of Devangiri or Bangla. We will make a complete server where our user will upload the photo of the printed document. Our system will give soft document such as text file of the image.

In our thesis we have designed a system where the image of the printed document will be the input of the system. At first we will segment the image into different small parts. We have used word segmentation technique for this purpose. After this step we get the segmented image of the words of the input image. Our machine has trained using Convolutional neural network. We have used a data set of 200 words classes each class have image of 50 fonts. After getting the segmented image our machine works with the Convolutional neural network algorithm and generate the text file of the input image. We have get an accuracy of 90% in our system.

Chapter 2

Literature Review

[1] - Optical Character Recognition (OCR) systems have been effectively developed for the recognition of printed script. The accuracy of OCR system mainly depends on the text pre-processing and segmentation algorithm being used. When the document is scanned it can be placed in any arbitrary angle which would appear on the computer monitor at the same angle. This paper addresses the algorithm for correction of skew angle generated in scanning of the text document and a novel profile based method for segmentation of printed text which separates the text in document image into lines, words and characters

[2] In this paper, They propose a novel scheme towards the recognition of multi-oriented and multi-sized isolated characters of printed script. For recognition, at first, distances of the outer contour points from the centroid of the individual characters are calculated and these contour distances are then arranged in a particular order to get size and rotation invariant feature. Next, based on the arranged contour distances, the features are derived from different class of characters. Finally, They use these derived features of the characters to statistically compare the features of the input character for recognition. They have tested our scheme on printed Bangla and Devnagari multi-oriented characters and They obtained encouraging results.

[3] In this paper A complete Optical Character Recognition (OCR) system for printed Bangla is presented. They claimed that this is the first OCR system among all script forms used in the Indian sub-continent. The problem is difficult because (i) there are about 300 basic, modified and compound character shapes in the script, (ii) the characters in a word are topologically connected and (iii) Bangla is an inflectional language. In their system the document image captured by Flat-bed scanner is subject to skew correction, text graphics separation, line segmentation, zone detection, word and character segmentation using some conventional and some newly developed techniques.For single font clear documents 95.50% word level recognition accuracy has been obtained.

[4]In this paper, they present a review of the OCR work done on Indian language scripts. The review is organized into 5 sections. Sections 1 and 2 cover introduction and properties on Indian scripts. In Section 3, we discuss different methodologies in OCR development as well as research work done on Indian scripts recognition. In Section 4, they discuss the scope of future work and further steps needed for Indian script OCR development. In Section 5 they conclude the paper.

[5]In this paper a complete OCR system is described for documents of single Bangla (Bengali) font. The character shapes are recognized by a combination of template and feature matching approach. Images digitized by flatbed scanner are subjected to skew correction, line, word and character segmentation, simple and compound character separation, feature extraction and finally character recognition. A feature based tree classifier is used for simple character recognition. Preprocessing like thinning and skeletonization is not necessary in this scheme and hence the system is quite fast. At present, the system has an accuracy of about 96%. Also, some character occurrence statistics have been computed to model an error detection and correction technique in the near future.

[6]In this paper for developing a Bangla OCR they introduce a bunch of algorithm and methods. There were many effort went on for developing a Bangla OCR. But all of them failed to provide an error free Bangla OCR. Each of them has some lacking. In this paper they discussed about the problem scope of currently existing Bangla OCR's and present the basic steps required for developing a Bangla OCR and a complete workflow for development of a Bangla OCR with mentioning all the possible algorithms required.

[7]In this paper they present a system towards the recognition of off-line handwritten characters of Devnagari. The features used for recognition purpose are mainly based on directional information obtained from the arc tangent of the gradient. To get the feature, at first, a 2times2 mean filtering is applied 4 times on the gray level image and a non-linear size normalization is done on the image. The normalized image is then segmented to 49times49 blocks and a Roberts filter is applied to obtain gradient image. Next, the arc tangent of the gradient (direction of gradient) is initially quantized into 32 directions and the strength of the gradient is accumulated with each of the quantized direction. Finally, the blocks and the directions are down sampled using Gaussian filter to get 392 dimensional feature vector. A modified quadratic classifier is applied on these features for recognition. We used 36172 handwritten data for testing our system and obtained 94.24% accuracy using 5-fold cross-validation scheme.

[8]In this paper they select the training data set from the script of the specified domain. They choose Hidden Markov Model (HMM) for character classification due to its simple and straightforward way of representation. They examine the primary error types that mainly occurred at preprocessing level and carefully handled those errors by adding special error correcting module as a part of recognizer. Finally they added a dictionary and some error specific rules to correct the probable errors after the word formation is done. The entire technique significantly increases the performance of the OCR for a specific domain to a great extent.

[9] Being the 5th Position and sweetest language in the world declared by the UNESCO Bengali is the national language in Bangladesh and one of the major languages in India. Lot of researches has been done to recognize Bengali, English and other major languages using Optical Character Recognition (OCR). To recognize Bengali character from text images and convert into editable text, Self Organizing Map (SOM) - kind of neural network has been used. To collect the character, documents are scanned, which is pre processed with the Image to Binary Conversion Algorithm. In the binary image, character area is represented by 0 (zero) and rest of the image area is represented with 1 (one). After detecting and correcting the skew and noise, the binary image is processed and grouped, which can be mapped and recognized by SOM. Considering efficiency and fastness, character grouping process has been introduced.

[10] This is a complete Optical Character Recognition system for printed Bangla text with a perspective of implementation.Suggestions have been made on the basis of the problems confronted in developing the software. This paper describes the efficient ways involving line and word detection, zoning, character separations and character recognition. Employing Skewness correction, thinning and better approach in scaling have obviously enhanced the performance of the OCR system in comparison with the existing ones. Application of neural network in detection of characters has made the process even faster with optimum performance.

Chapter 3

Preliminary

3.1 Convolutional Neural Networks

In machine learning, a convolutional neural network (CNN, or ConvNet) is a class of deep, feed-forward artificial neural networks that has successfully been applied to analyzing visual imagery. CNNs use a variation of multilayer perceptrons designed to require minimal preprocessing. A convolutional neural network (CNN, or ConvNet) is a class of deep, feed-forward artificial neural networks that has successfully been applied to analyzing visual imagery. CNNs use a variation of multilayer perceptrons designed to require minimal preprocessing. It is also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics. Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field. CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage.

A CNN consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers. Description of the process as a convolution in neural networks is by convention. Mathematically it is a cross-correlation rather than a convolution. This only has significance for the indices in the matrix, and thus which weights are placed at which index.

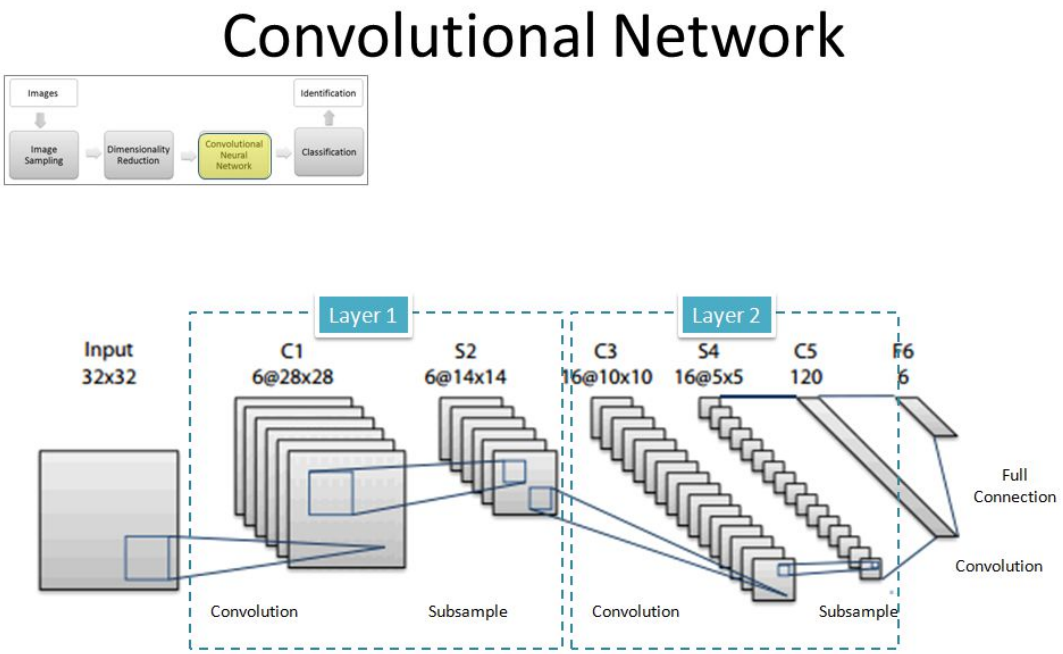


Figure 3.1: Convolutional Neural Networks.

3.1.1 Input and Output

When a computer sees an image (takes an image as input), it will see an array of pixel values. Depending on the resolution and size of the image, it will see a 32 x 32 x 3 array of numbers (The 3 refers to RGB values). Just to drive home the point, let's say we have a color image in JPG form and its size is 480 x 480. The representative array will be 480 x 480 x 3. Each of these numbers is given a value from 0 to 255 which describes the pixel intensity at that point. These numbers, while meaningless to us when we perform image classification, are the only inputs available to the computer. The idea is that we give the computer this array of numbers and it will output numbers that describe the probability of the image being a certain class.

3.1.2 What We Want the Computer to Do

Now that we know the problem as well as the inputs and outputs. What we want the computer to do is to be able to differentiate between all the images its given and figure out the unique features that make a dog a dog or that make a cat a cat. This is the process that goes on in our minds subconsciously as well. When we look at a picture of a dog, we can classify it as such if the picture has identifiable features such as paws or 4 legs. In a similar way, the computer is able perform image classification by looking for low level features such as edges and curves,

and then building up to more abstract concepts through a series of convolutional layers. This is a general overview of what a CNN does. Lets get into the specifics.

3.1.3 First Layer

The first layer in a CNN is always a Convolutional Layer. First thing to make sure that what is the input to this Convolutional and the region that it is shining over is called the receptive field. Now this filter is also an array of numbers . This numbers are called weights or parameters. A very important note is that the depth of this filter has to be the same as the depth of the input , so the dimensions of this filter is 5 x 5 x 3. Now, lets take the first position the filter is in for example. It would be the top left corner. As the filter is sliding, or convolving, around the input image, it is multiplying the values in the filter with the original pixel values of the image also known as computing element wise multiplications. These multiplications are all summed up mathematically this would be 75 multiplications in total. So now we have a single number. Remember, this number is just representative of when the filter is at the top left of the image. Now, we repeat this process for every location on the input volume. Next step we would be moving the filter to the right by 1 unit, then right again by 1, and so on. Every unique location on the input volume produces a number. After sliding the filter over all the locations, we will find out that what were left with is a 28 x 28 x 1 array of numbers, which we call an activation map or feature map. The reason we get a 28 x 28 array is that there are 784 different locations that a 5 x 5 filter can fit on a 32 x 32 input image. These 784 numbers are mapped to a 28 x 28 array.

3.1.4 Fully Connected

Now that we can detect these high level features, the icing on the cake is attaching a fully connected layer to the end of the network. This layer basically takes an input volume and outputs an N dimensional vector where N is the number of classes that the program has to choose from. For example, if we wanted a digit classification program, N would be 10 since there are 10 digits. Each number in this N dimensional vector represents the probability of a certain class. For example, if the resulting vector for a digit classification program is [0 .1 .1 .75 0 0 0 0 0 .05], then this represents a 10% probability that the image is a 1, a 10% probability that the image is a 2, a 75% probability that the image is a 3, and a 5% probability that the image is a 9. The way this fully connected layer works is that it looks at the output of the previous layer and determines which features most correlate to a particular class. For example, if the program is predicting that some image is a dog, it will have high values in the activation maps that represent high level features like a paw or 4 legs, etc. Similarly, if the program is predicting that some image is a bird, it will have high values in the activation maps that represent high

level features like wings or a beak, etc. Basically, a Fully Connected layer looks at what high level features most strongly correlate to a particular class and has particular weights so that when we compute the products between the weights and the previous layer, we get the correct probabilities for the different classes.

3.2 Recurrent neural network

A recurrent neural network (RNN) is a class of artificial neural network where connections between units form a directed graph along a sequence. This allows it to exhibit dynamic temporal behavior for a time sequence. Unlike feed forward 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.

Recurrent neural networks are used somewhat indiscriminately about two broad classes of networks with a similar general structure, where one is finite impulse and the other is infinite impulse. Both classes of networks exhibit temporal dynamic behavior.[4] A finite impulse recurrent network is a directed acyclic graph that can be unrolled and replaced with a strictly feed forward neural network, while an infinite impulse recurrent network is a directed cyclic graph that can not be unrolled.

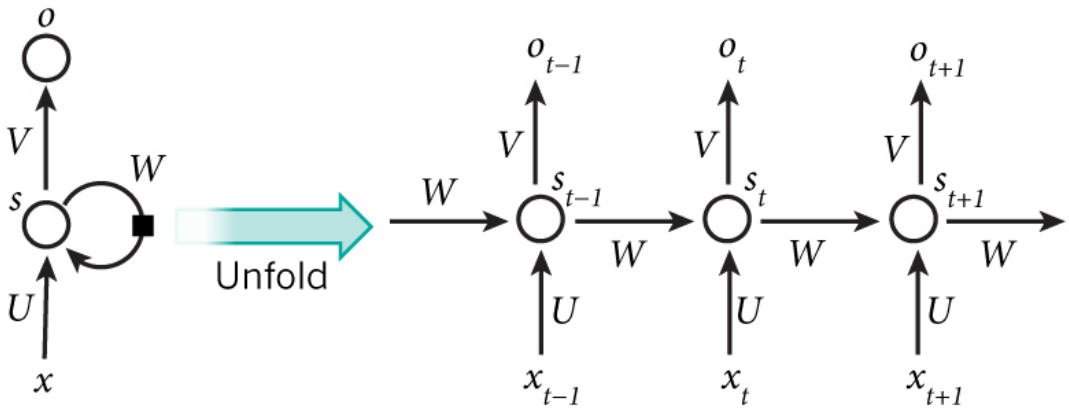


Figure 3.2: Recurrent neural network.

Both finite impulse and infinite impulse recurrent networks can have additional stored state, and the storage can be under direct control by the neural network. The storage can also be replaced by another network or graph, if that incorporates time delays or has feedback loops. Such controlled states are referred to as gated state or gated memory, and are part of Long short-term memorys (LSTM) and gated recurrent units

3.2.1 Long Short-Term Memory Units (LSTMs)

In the mid-90s, a variation of recurrent net with so-called Long Short-Term Memory units, or LSTMs, was proposed by the German researchers Sepp Hochreiter and Juergen Schmidhuber as a solution to the vanishing gradient problem.

LSTMs help preserve the error that can be back propagated through time and layers. By maintaining a more constant error, it allow recurrent nets to continue to learn over many time steps (over 1000), thereby opening a channel to link causes and effects remotely. This is one of the central challenges to machine learning and AI, since algorithms are frequently confronted by environments where reward signals are sparse and delayed, such as life itself.

LSTMs contain information outside the normal flow of the recurrent network in a gated cell. Information can be stored in, written to, or read from a cell, much like data in a computers memory. The cell makes decisions about what to store, and when to allow reads, writes and erasures, via gates that open and close. Unlike the digital storage on computers, however, these gates are analog, implemented with element-wise multiplication by sigmoids, which are all in the range of 0-1. Analog has the advantage over digital of being differentiable, and therefore suitable for back propagation.

Those gates act on the signals they receive, and similar to the neural networks nodes, it blocks or passes on information based on its strength and import, which they filter with their own sets of weights. Those weights, like the weights that modulate input and hidden states, are adjusted via the recurrent networks learning process. That The cells learn when to allow data to enter, leave or be deleted through the iterative process of making guesses, back propagating error, and adjusting weights via gradient descent.

The diagram below illustrates how data flows through a memory cell and is controlled by its gates.

Starting from the bottom, the triple arrows show where information flows into the cell at multiple points.That combination of present input and past cell state is fed not only to the cell itself, but also to each of its three gates, which will decide how the input will be handled.

The black dots are the gates themselves, which determine respectively whether to let new input in, erase the present cell state, and/or let that state impact the networks output at the present time step. S_c is the current state of the memory cell, and g_y_in is the current input to it. Remember that each gate can be open or shut, and they will recombine their open and shut states at each step. The cell can forget its state, or not; be written to, or not; and be read from, or not, at each time step, and those flows are represented here.

The large bold letters give us the result of each operation.

Heres another diagram for good measure, comparing a simple recurrent network (left) to an

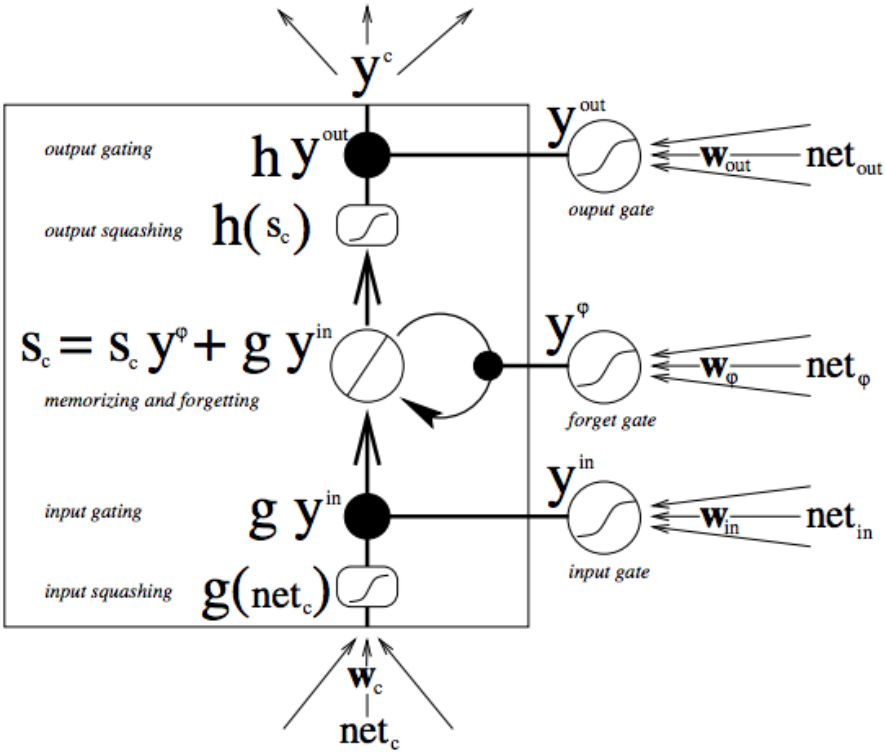


Figure 3.3: Data flow of Long Short-Term Memory Units using gates

LSTM cell (right).

Its important to note that LSTMs memory cells give different roles to addition and multiplication in the transformation of input. The central plus sign in both diagrams is essentially the secret of LSTMs. The basic change helps them preserve a constant error when it must be back propagated at depth. Instead of determining the subsequent cell state by multiplying its current state with new input, they add the two, and that quite literally makes the difference.

Different sets of weights filter the input for input, output and forgetting. The forget gate is represented as a linear identity function, because if the gate is open, the current state of the memory cell is simply multiplied by one, to propagate forward one more time step.

Furthermore, while were on the topic of simple hacks, including a bias of 1 to the forget gate of every LSTM cell is also shown to improve performance.

We may wonder why LSTMs have a forget gate when their purpose is to link distant occurrences to a final output. If we are analyzing a text corpus and come to the end of a document, For example, we may have no reason to believe that the next document has any relationship to it whatsoever, and therefore the memory cell should be set to zero before the net ingests the first element of the next document.

In the diagram below, we can see the gates at work, with straight lines representing closed gates, and blank circles representing open ones. The lines and circles running horizontal down

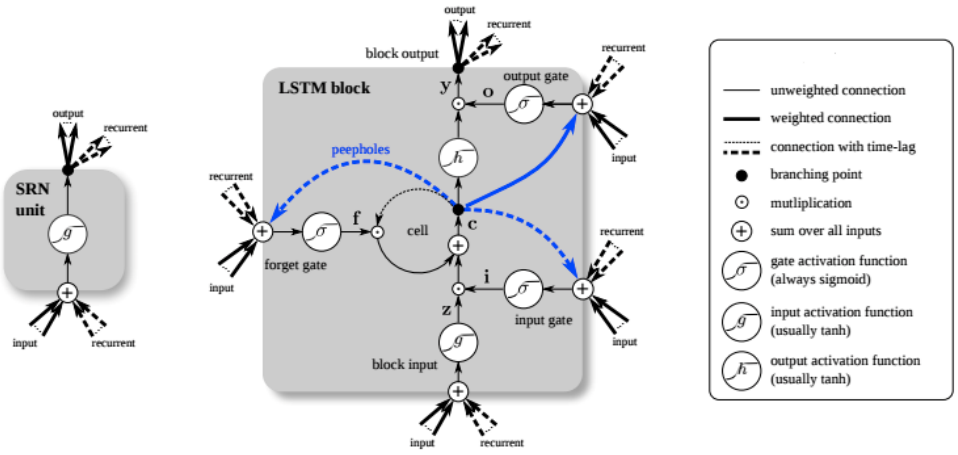


Figure 3.4: Comparing Long Short-Term Memory Units with Recurrent Neural Network

the hidden layer are the forget gates.

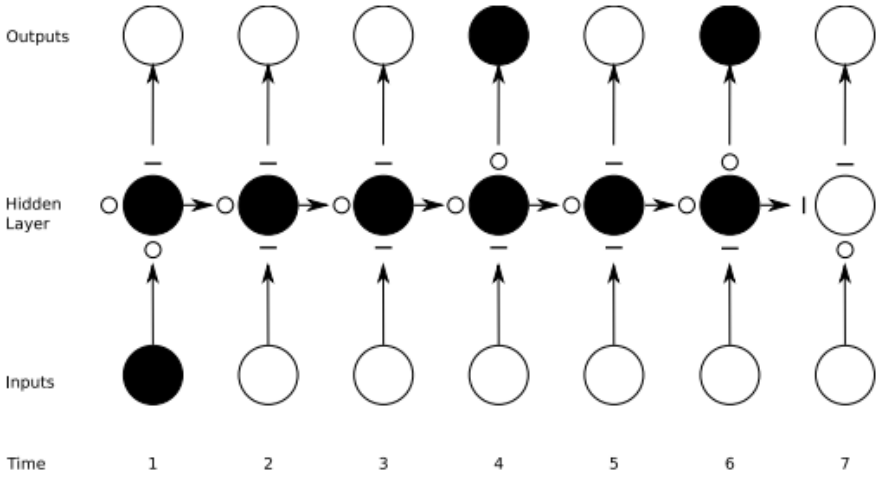


Figure 3.5: Gates work in Long Short-Term Memory

It should be noted that while feed forward networks map one input to one output, recurrent nets can map one to many, as above (one image to many words in a caption), many to many (translation), or many to one (classifying a voice).

3.2.2 LSTM Hyperparameter Tuning

Here are a few ideas we have to keep in mind when manually optimizing hyperparameters for Recurrent Neural Network:

- We should watch out for over fitting, which happens when a neural network essentially memorizes the training data. Over fitting means we get great performance on training data, but the networks model is useless for out-of-sample prediction.
- We should use regularization methods include l1, l2, and dropout among others
- We should use larger network. The larger the network, the more powerful, but its also easier to over fit. We should not want to try to learn a million parameters from 10,000 examples.
- We should use more data as it is almost always better, because it helps fight overfitting.
- We should train over multiple epochs (complete passes through the dataset)..
- We should evaluate test set performance at each epoch to know when to stop (early stopping).
- The learning rate is the single most important hyperparameter.We should tune this using deeplearning4j-ui
- In general, stacking layers can help.
- For LSTMs, we should use the softsign (not softmax) activation function over tanh (its faster and less prone to saturation (0 gradients)).
- Updaters: RMSProp, AdaGrad or momentum (Nesterovs) are usually good choices. AdaGrad also decays the learning rate, which can help sometimes.
- Finally, we should remember data normalization, MSE loss function + identity activation function for regression, Xavier weight initialization

3.3 Segmentation

3.3.1 Text Segmentation

Text segmentation is the process of dividing written text into meaningful units, such as words, sentences, or topics. The term applies both to mental processes used by humans when reading text, and to artificial processes implemented in computers, which are the subject of natural language processing. The problem is non-trivial, because while some written languages have explicit word boundary markers, such as the word spaces of written English and the distinctive initial, medial and final letter shapes of Arabic, such signals are sometimes ambiguous and not present in all written languages.Compare speech segmentation, the process of dividing speech into linguistically meaningful portions.

Word Segmentation

Word segmentation is the problem of dividing a string of written language into its component words. In English and many other languages using some form of the Latin alphabet, the space is a good approximation of a word divider (word delimiter), although this concept has limits because of the variability with which languages emically regard collocations and compounds. Many English compound nouns are variably written with a corresponding variation in whether speakers think of them as noun phrases or single nouns; there are trends in how norms are set, such as that open compounds often tend eventually to solidify by widespread convention, but variation remains systemic. In contrast, German compound nouns show less orthographic variation, with solidification being a stronger norm. However, the equivalent to the word space character is not found in all written scripts, and without it word segmentation is a difficult problem. Languages which do not have a trivial word segmentation process include Chinese, Japanese, where sentences but not words are delimited, Thai and Lao, where phrases and sentences but not words are delimited, and Vietnamese, where syllables but not words are delimited. In some writing systems however, such as the Ge'ez script used for Amharic and Tigrinya among other languages, words are explicitly delimited (at least historically) with a non-whitespace character. The Unicode Consortium has published a Standard Annex on Text Segmentation which is Unicode Standard Annex #29 exploring the issues of segmentation in multiscript texts. Word splitting is the process of parsing concatenated text to infer where word breaks exist. Word splitting may also refer to the process of hyphenation.

Intent Segmentation

Intent segmentation is the problem of dividing written words into key-phrases (2 or more group of words). In English and all other languages the core intent or desire is identified and become the corner-stone of the key-phrase Intent segmentation. Core product/service, idea, action & or thought anchor the key-phrase.

Sentence Segmentation

Sentence segmentation is the problem of dividing a string of written language into its component sentences. In English and some other languages, using punctuation, particularly the full stop/period character is a reasonable approximation. However even in English this problem is not trivial due to the use of the full stop character for abbreviations, which may or may not also terminate a sentence. For example, Mr. is not its own sentence in "Mr. Smith went to the shops in Jones Street." When processing plain text, tables of abbreviations that contain periods can help prevent incorrect assignment of sentence boundaries.As with word segmentation, not all written languages contain punctuation characters which are useful for approximating sentence boundaries

Topic Segmentation

Topic analysis consists of two main tasks: topic identification and text segmentation. While the first is a simple classification of a specific text, the latter case implies that a document may contain multiple topics, and the task of computerized text segmentation may be to discover these topics automatically and segment the text accordingly. The topic boundaries may be apparent from section titles and paragraphs. In other cases, one needs to use techniques similar to those used in document classification.

Segmenting the text into topics or discourse turns might be useful in some natural processing tasks: it can improve information retrieval or speech recognition significantly by indexing/recognizing documents more precisely or by giving the specific part of a document corresponding to the query as a result. It is also needed in topic detection and tracking systems and text summarizing problems.It is quite an ambiguous task people evaluating the text segmentation systems often differ in topic boundaries. Hence, text segment evaluation is also a challenging problem.

3.3.2 Image segmentation

In computer vision, image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super-pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image. Each of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic(s). When applied to a stack of images, typical in medical imaging, the resulting contours after image segmentation can be used to create 3D reconstructions with the help of interpolation algorithms like Marching cubes.

Chapter 4

Segmentation Technique

4.1 Column and Row Process

At first we tried to segment the word in column row process. In this process we segment the words according to the pixel of their column and row. But there is problem in noise detection. The noises are also treated as segmented words in this method. Moreover Rotated words create problem in this method. It works less efficiently for bangla.

4.2 Contour and Threshold Based Method

4.2.1 Image Input

In our algorithm we take pictures from books, pdf or handwritten notes. This pictures are taken by a software made by us. The brightness level must be good enough ta segment the words properly. There are no predefined rules to take pictures. People can take pictures manually and give that to our algorithm. Our algorithm will automatically do all changes to make it appropriate for the segmentation technique.It means if the user take picture which is not straight then our technique will calculate the angle of deviation of the image from straight image. Then it will automatically rotate the image and make it straight. This straight image is very much appropriate input for the segmentation technique.

4.2.2 Turning image into Grey Scale

We have to turn the image into grey scale. Our algorithm will do it automatically. We increase the brightness, hue and contrast value of the image. We change this value appropriately and

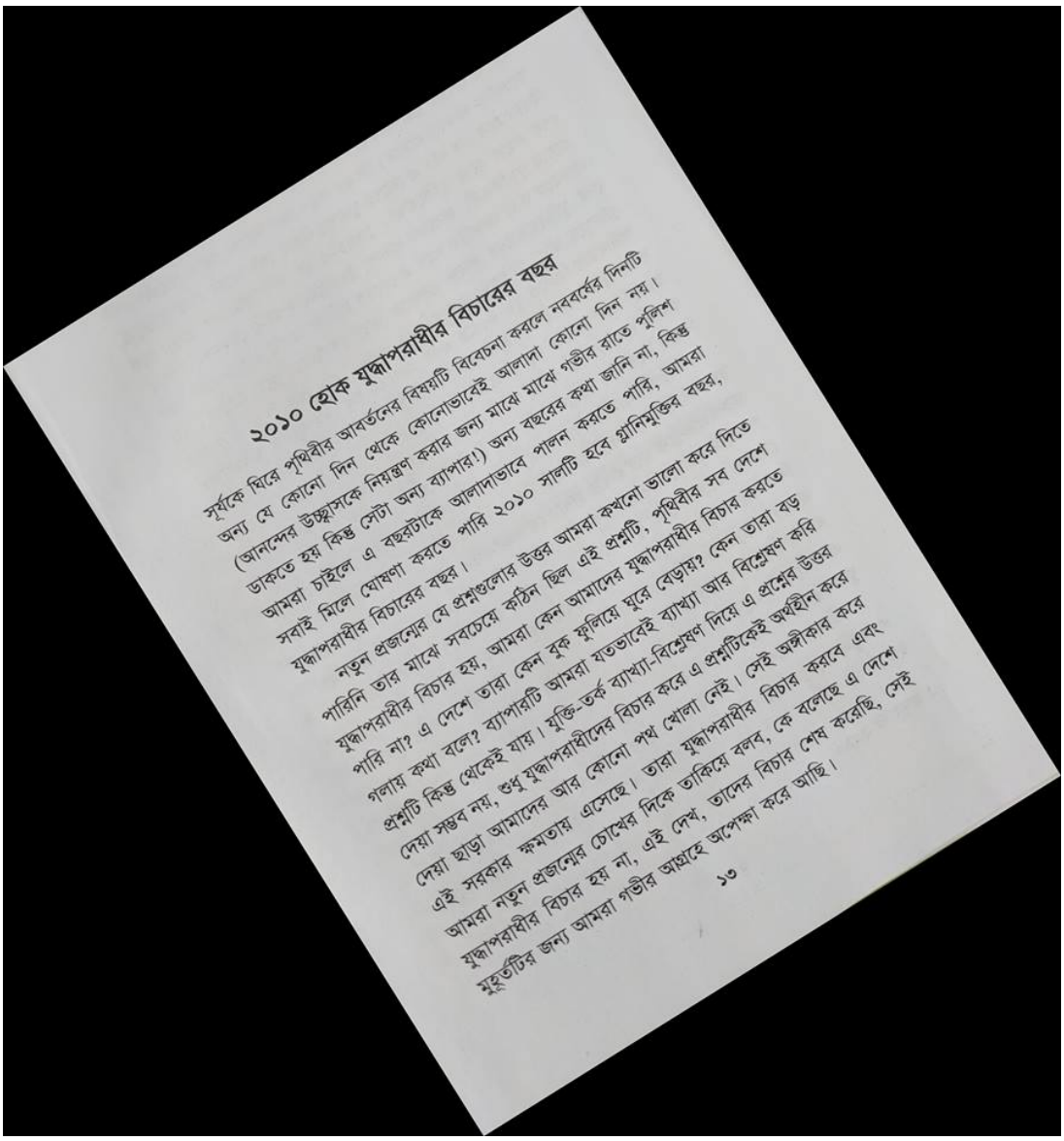
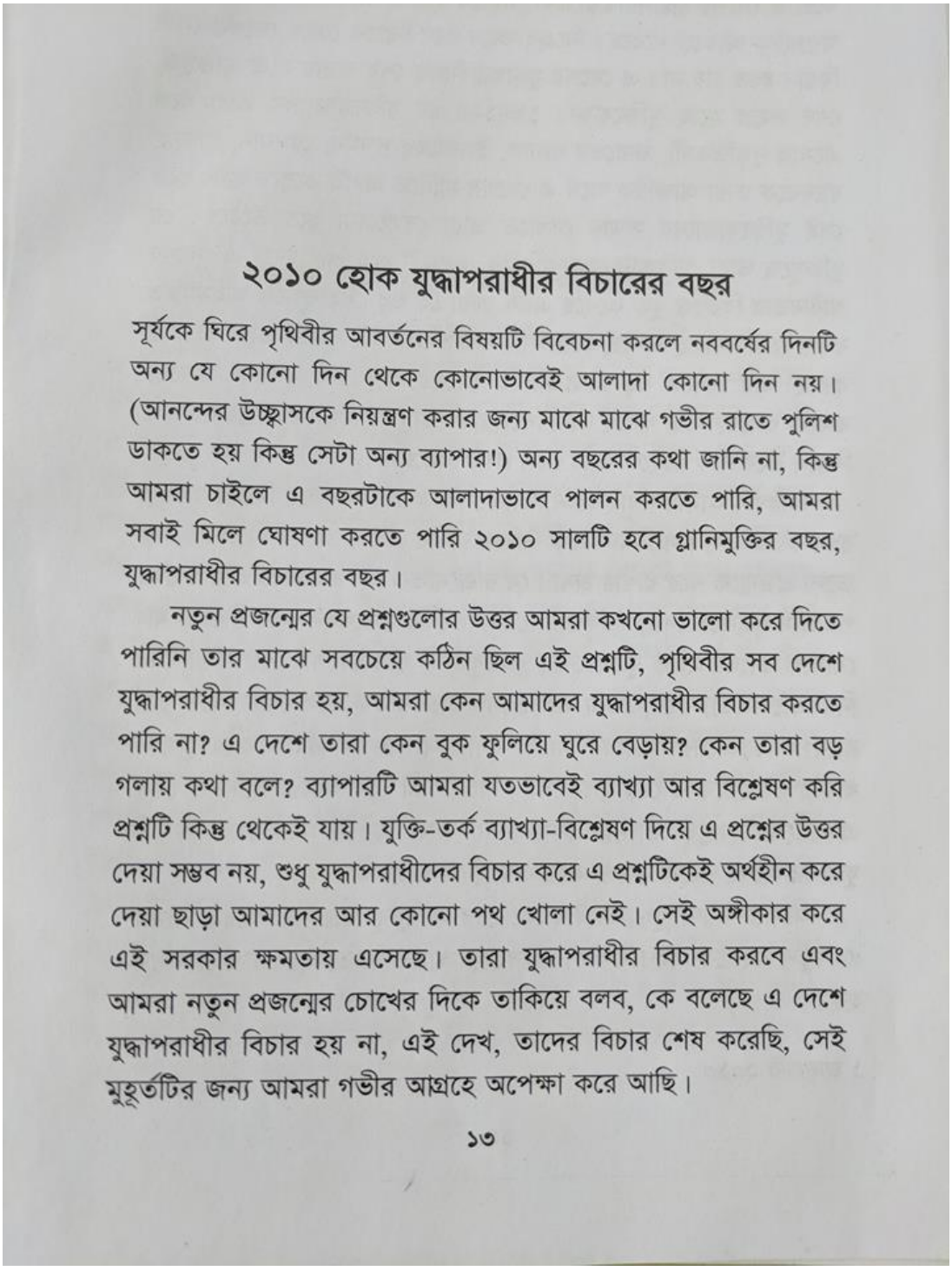


Figure 4.1: Rotated Image



২০১০ হোক যুদ্ধাপরাধীর বিচারের বছর

সূর্যকে ঘিরে পৃথিবীর আবর্তনের বিষয়টি বিবেচনা করলে নববর্ষের দিনটি অন্য যে কোনো দিন থেকে কোনোভাবেই আলাদা কোনো দিন নয়। (আনন্দের উচ্ছ্বাসকে নিয়ন্ত্রণ করার জন্য মাঝে মাঝে গভীর রাতে পুলিশ ডাকতে হয় কিন্তু সেটা অন্য ব্যাপার!) অন্য বছরের কথা জানি না, কিন্তু আমরা চাইলে এ বছরটাকে আলাদাভাবে পালন করতে পারি, আমরা সবাই মিলে ঘোষণা করতে পারি ২০১০ সালটি হবে গ্লানিমুক্তির বছর, যুদ্ধাপরাধীর বিচারের বছর।

নতুন প্রজন্মের যে প্রশ্নগুলোর উত্তর আমরা কখনো ভালো করে দিতে পারিনি তার মাঝে সবচেয়ে কঠিন ছিল এই প্রশ্নটি, পৃথিবীর সব দেশে যুদ্ধাপরাধীর বিচার হয়, আমরা কেন আমাদের যুদ্ধাপরাধীর বিচার করতে পারি না? এ দেশে তারা কেন বুক ফুলিয়ে ঘুরে বেড়ায়? কেন তারা বড় গলায় কথা বলে? ব্যাপারটি আমরা যতভাবেই ব্যাখ্যা আর বিশ্লেষণ করি প্রশ্নটি কিন্তু থেকেই যায়। যুক্তি-তর্ক ব্যাখ্যা-বিশ্লেষণ দিয়ে এ প্রশ্নের উত্তর দেয়া সম্ভব নয়, শুধু যুদ্ধাপরাধীদের বিচার করে এ প্রশ্নটিকেই অর্থহীন করে দেয়া ছাড়া আমাদের আর কোনো পথ খোলা নেই। সেই অঙ্গীকার করে এই সরকার ক্ষমতায় এসেছে। তারা যুদ্ধাপরাধীর বিচার করবে এবং আমরা নতুন প্রজন্মের চোখের দিকে তাকিয়ে বলব, কে বলেছে এ দেশে যুদ্ধাপরাধীর বিচার হয় না, এই দেখ, তাদের বিচার শেষ করেছি, সেই মুহূর্তটির জন্য আমরা গভীর আগ্রহে অপেক্ষা করে আছি।

১৩

Figure 4.2: Straight Image

check if the picture is converted into grey scale or not. After converting the image into grey scale our algorithm will start to work.

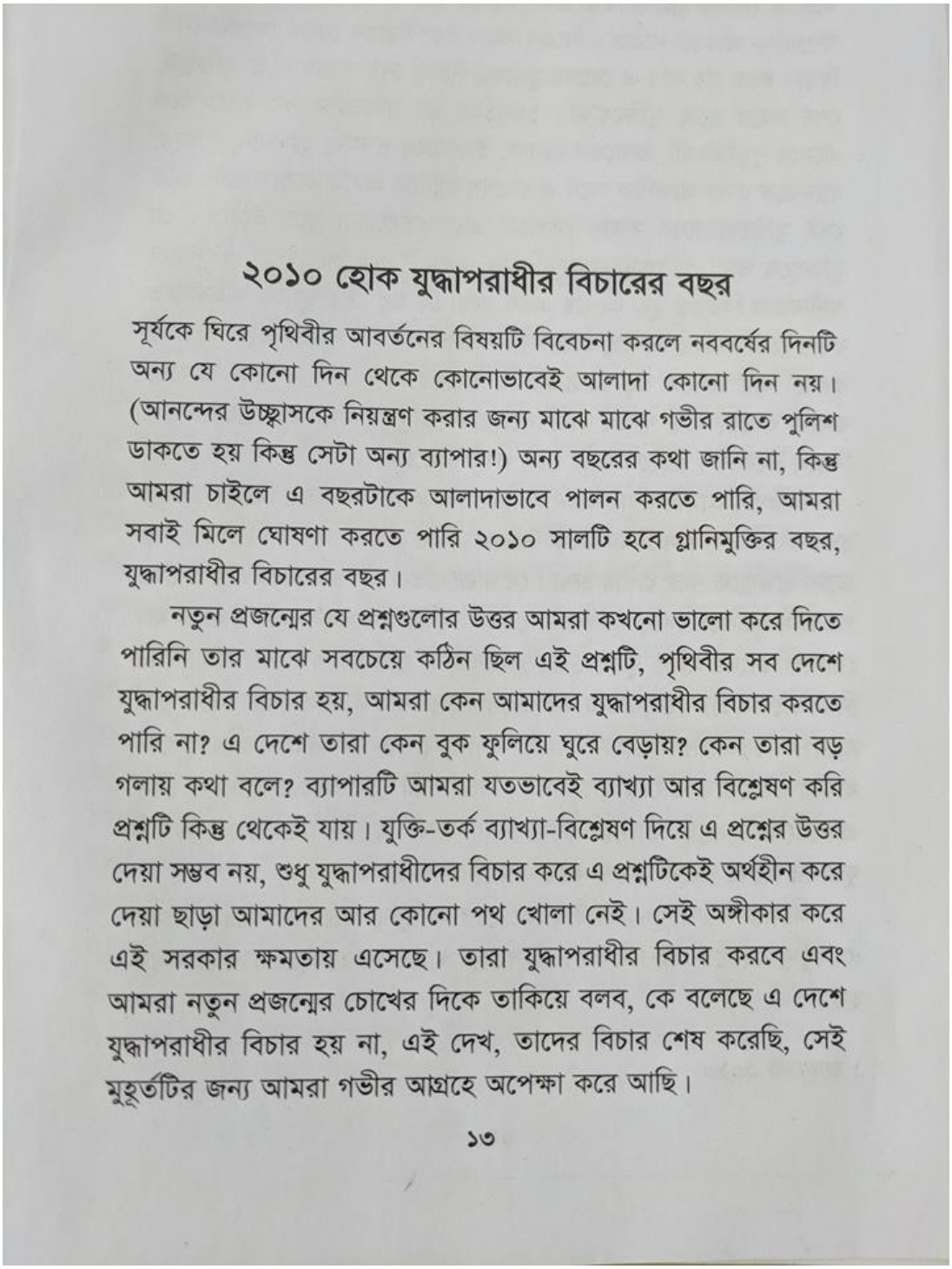
4.2.3 Finding contours and Threshold values

Contours mean the components that are connected to each other in a page. Our technique first divide the words into different contour groups. We take all kinds of possible permutations. Then we find the threshold values for this contours. At first the values of the threshold increases from zero. After reaching a maximum value the threshold value decreases. This maximum

Table 4.1: Word Count in Segmentation Technique

Threshold Value	number of Contours
0	171
10	7512
20	10188
30	7184
40	4393
50	2606
60	1714
70	1544
80	2031
90	3195
100	3903
110	4451
120	7550
130	9331
140	8636
150	8836
160	6679
170	1904
180	241
190	26
200	0
210	0
220	0
230	0
240	0
250	0

value is called the first maxima. After sometime the threshold values again start to increase and reach a maximum value. This maximum value is called second maximum. We have to do this method for all the contours of the image. Then we sort the contours according to the threshold hierarchy. Then we start find the hierarchy level between which maximum number of contours



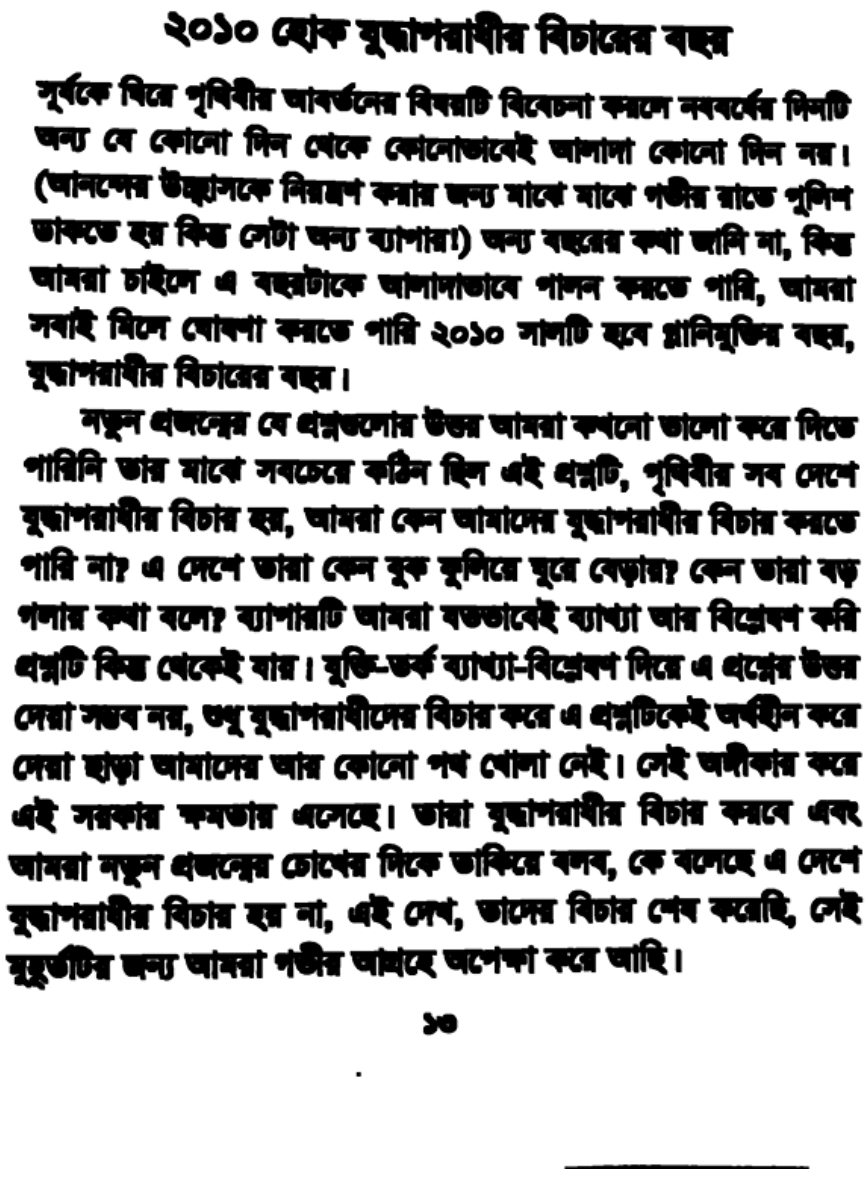
২০১০ হোক যুদ্ধাপরাধীর বিচারের বছর

সূর্যকে ঘিরে পৃথিবীর আবর্তনের বিষয়টি বিবেচনা করলে নববর্ষের দিনটি অন্য যে কোনো দিন থেকে কোনোভাবেই আলাদা কোনো দিন নয়। (আনন্দের উচ্ছ্বাসকে নিয়ন্ত্রণ করার জন্য মাঝে মাঝে গভীর রাতে পুলিশ ডাকতে হয় কিন্তু সেটা অন্য ব্যাপার!) অন্য বছরের কথা জানি না, কিন্তু আমরা চাইলে এ বছরটাকে আলাদাভাবে পালন করতে পারি, আমরা সবাই মিলে ঘোষণা করতে পারি ২০১০ সালটি হবে গ্লানিমুক্তির বছর, যুদ্ধাপরাধীর বিচারের বছর।

নতুন প্রজন্মের যে প্রশ্নগুলোর উত্তর আমরা কখনো ভালো করে দিতে পারিনি তার মাঝে সবচেয়ে কঠিন ছিল এই প্রশ্নটি, পৃথিবীর সব দেশে যুদ্ধাপরাধীর বিচার হয়, আমরা কেন আমাদের যুদ্ধাপরাধীর বিচার করতে পারি না? এ দেশে তারা কেন বুক ফুলিয়ে ঘুরে বেড়ায়? কেন তারা বড় গলায় কথা বলে? ব্যাপারটি আমরা যতভাবেই ব্যাখ্যা আর বিশ্লেষণ করি প্রশ্নটি কিন্তু থেকেই যায়। যুক্তি-তর্ক ব্যাখ্যা-বিশ্লেষণ দিয়ে এ প্রশ্নের উত্তর দেয়া সম্ভব নয়, শুধু যুদ্ধাপরাধীদের বিচার করে এ প্রশ্নটিকেই অর্থহীন করে দেয়া ছাড়া আমাদের আর কোনো পথ খোলা নেই। সেই অঙ্গীকার করে এই সরকার ক্ষমতায় এসেছে। তারা যুদ্ধাপরাধীর বিচার করবে এবং আমরা নতুন প্রজন্মের চোখের দিকে তাকিয়ে বলব, কে বলেছে এ দেশে যুদ্ধাপরাধীর বিচার হয় না, এই দেখ, তাদের বিচার শেষ করেছি, সেই মুহূর্তটির জন্য আমরা গভীর আগ্রহে অপেক্ষা করে আছি।

১৩

Figure 4.3: Normal Image



২০১০ হোক যুদ্ধাপরাধীর বিচারের বছর

সূর্যকে ঘিরে পৃথিবীর আবর্তনের বিষয়টি বিবেচনা করলে নববর্ষের দিনটি অন্য যে কোনো দিন থেকে কোনোভাবেই আলাদা কোনো দিন নয়। (আনন্দের উল্লাসকে নিয়ন্ত্রণ করার জন্য মাঝে মাঝে গভীর রাতে পুলিশ ডাকতে হয় কিন্তু সেটা অন্য ব্যাপার!) অন্য বছরের কথা জানি না, কিন্তু আমরা চাইলে এ বছরটাকে আলাদাভাবে পালন করতে পারি, আমরা সবাই মিলে ঘোষণা করতে পারি ২০১০ সালটি হবে গ্লানিমুক্তির বছর, যুদ্ধাপরাধীর বিচারের বছর।

নতুন প্রজন্মের যে প্রশ্নগুলোর উত্তর আমরা কখনো ভালো করে দিতে পারিনি তার মাঝে সবচেয়ে কঠিন ছিল এই প্রশ্নটি, পৃথিবীর সব দেশে যুদ্ধাপরাধীর বিচার হয়, আমরা কেন আমাদের যুদ্ধাপরাধীর বিচার করতে পারি না? এ দেশে তারা কেন বুক ফুলিয়ে ঘুরে বেড়ায়? কেন তারা বড় গলায় কথা বলে? ব্যাপারটি আমরা যতভাবেই ব্যাখ্যা আর বিশ্লেষণ করি প্রশ্নটি কিন্তু থেকেই যায়। যুক্তি-তর্ক ব্যাখ্যা-বিশ্লেষণ দিয়ে এ প্রশ্নের উত্তর দেয়া সম্ভব নয়, শুধু যুদ্ধাপরাধীদের বিচার করে এ প্রশ্নটিকেই অর্থহীন করে দেয়া ছাড়া আমাদের আর কোনো পথ খোলা নেই। সেই অঙ্গীকার করে এই সরকার ক্ষমতায় এসেছে। তারা যুদ্ধাপরাধীর বিচার করবে এবং আমরা নতুন প্রজন্মের চোখের দিকে তাকিয়ে বলব, কে বলেছে এ দেশে যুদ্ধাপরাধীর বিচার হয় না, এই দেখ, তাদের বিচার শেষ করেছি, সেই মুহূর্তটির জন্য আমরা গভীর আগ্রহে অপেক্ষা করে আছি।

১৩

Figure 4.4: Grey Scale Image

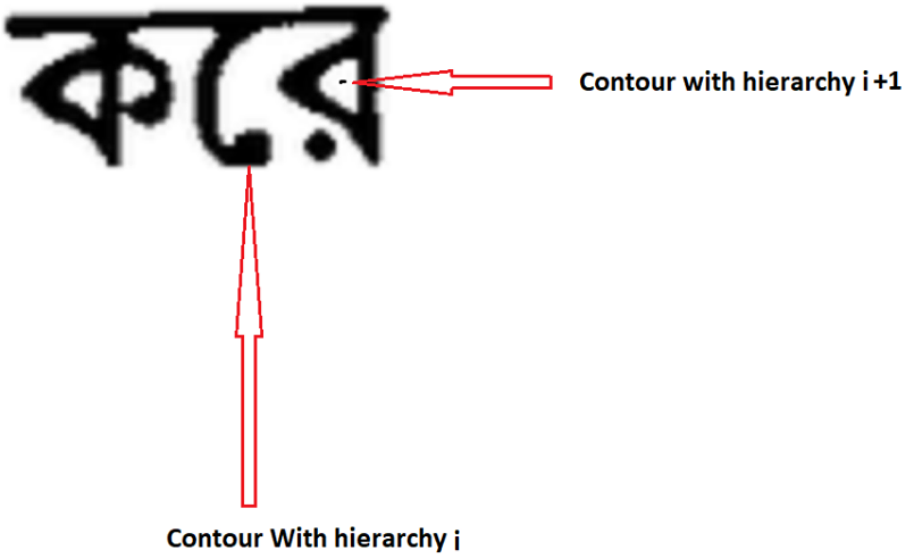


Figure 4.5: Finding Contours

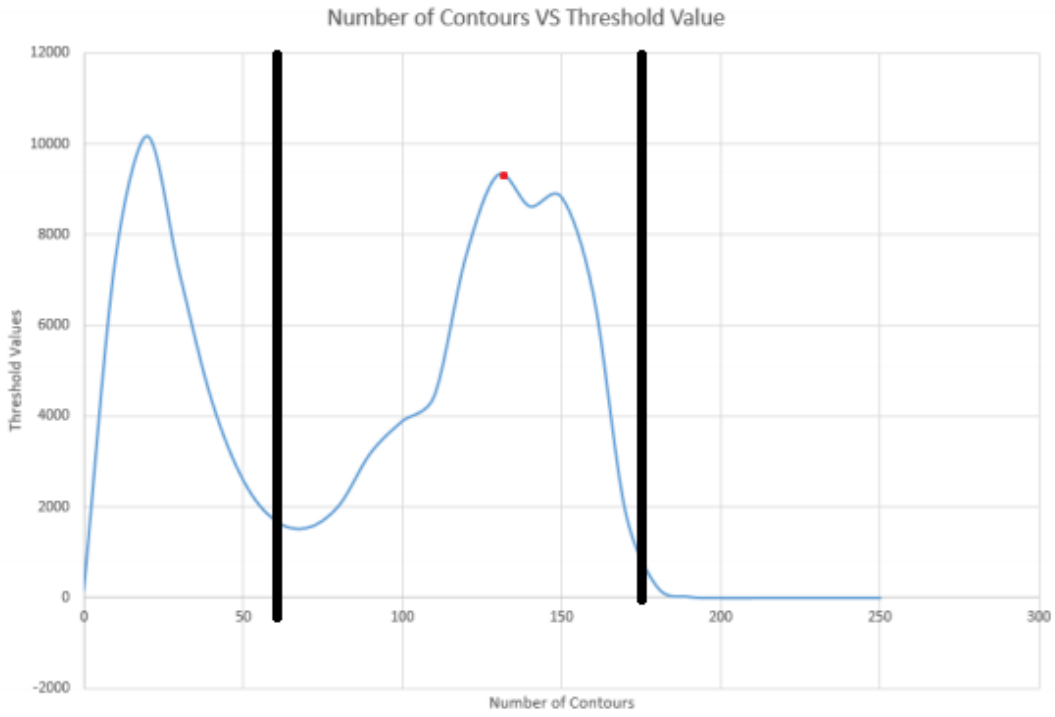


Figure 4.6: Contour VS Threshold value.

reside. It means there will be a first maxima and a second maxima between which maximum number of the contours reside. The contours which are not in this threshold hierarchy are the noise. So the noise are automatically discard through our algorithm.

4.2.4 Work flow of the segmentation Technique

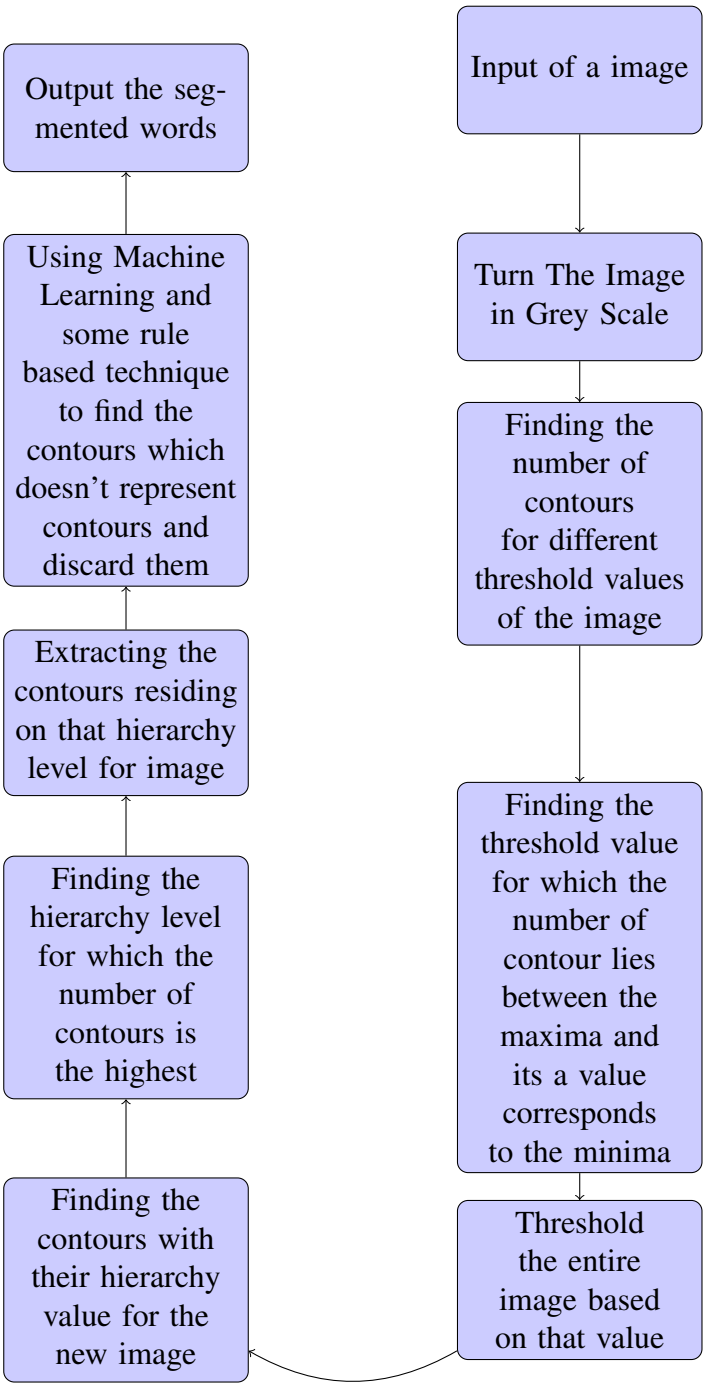


Figure 4.7: Work Flow of the Segmentation algorithm

4.3 Making dataset

4.3.1 OpenCV

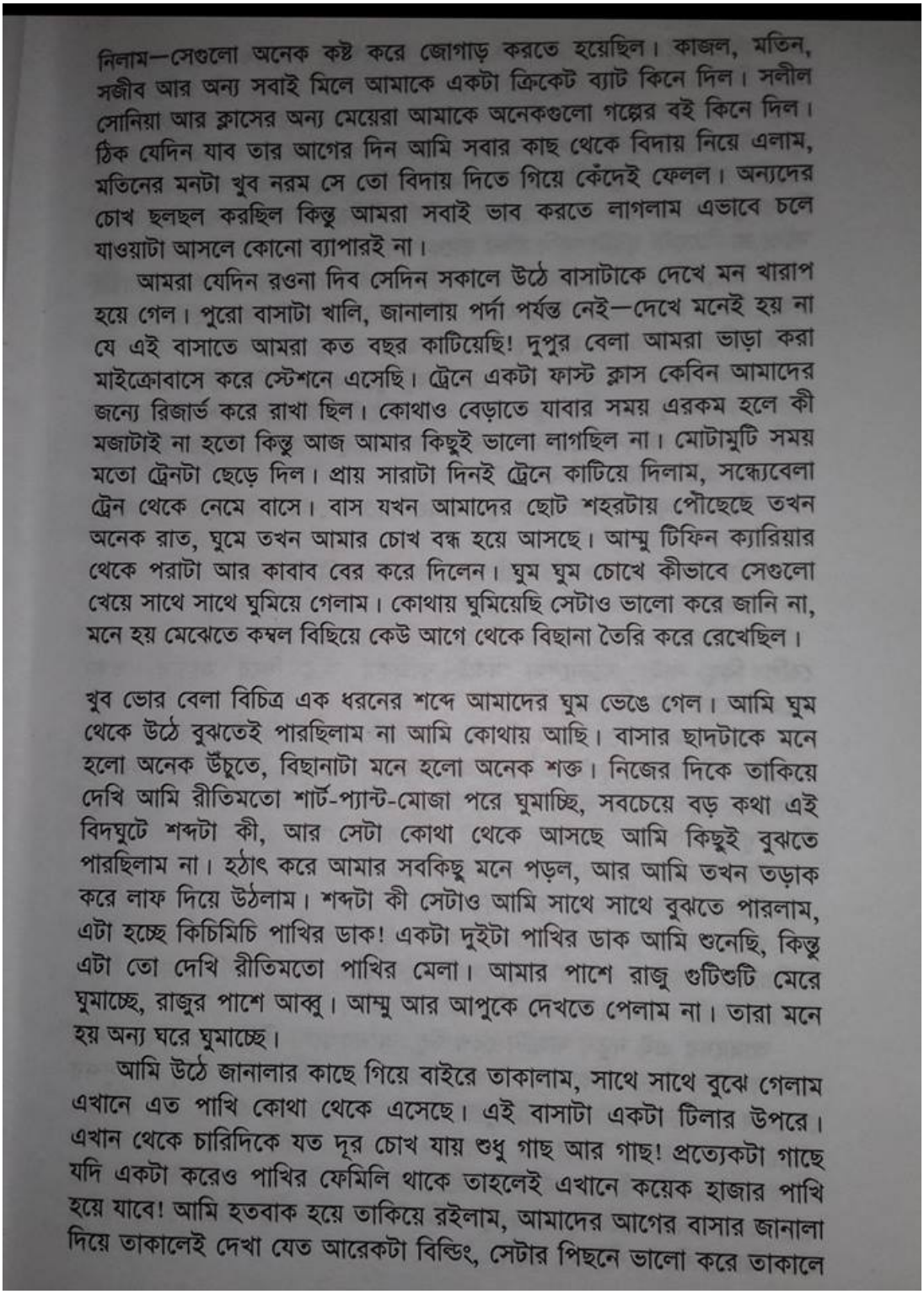
For making the dataset we use OpenCV library of python.The Usage of functions of openCV:
1. For converting colored input image to grey scale cvtColor() was used
2. For getting contours, their hierarchy level findContours() opencv as used
3. For threshold an image threshold() was used
4. For applying morphological erosion morphologyEx() was used
5. For getting area of a contour contourArea() was used
6. For getting a particular contours bounding rectangle boundingRect() was used

4.3.2 Generating Images of different Brightness

We have generated images of different brightness using python code. There are three property of a image that controls the brightness and contrast. They are h,r,v. If we increase the value of h in the images the brightness of the images are changed accordingly. In this process we get 100 images of different brightness.

4.4 Result Analysis

We see that our algorithm worked fine for 85% of the images. But for other images the result was not good.The algorithm doesn't work with Images with very high or low brightness. For images with medium brightness accuracy was almost 100%.



নিলাম—সেগুলো অনেক কষ্ট করে জোগাড় করতে হয়েছিল। কাজল, মতিন, সজীব আর অন্য সবাই মিলে আমাকে একটা ক্রিকেট ব্যাট কিনে দিল। সলীল সোনিয়া আর ক্লাসের অন্য মেয়েরা আমাকে অনেকগুলো গল্পের বই কিনে দিল। ঠিক যেদিন যাব তার আগের দিন আমি সবার কাছ থেকে বিদায় নিয়ে এলাম, মতিনের মনটা খুব নরম সে তো বিদায় দিতে গিয়ে কেঁদেই ফেলল। অন্যদের চোখ ছলছল করছিল কিন্তু আমরা সবাই ভাব করতে লাগলাম এভাবে চলে যাওয়াটা আসলে কোনো ব্যাপারই না।

আমরা যেদিন রওনা দিব সেদিন সকালে উঠে বাসাটাকে দেখে মন খারাপ হয়ে গেল। পুরো বাসাটা খালি, জানালায় পর্দা পর্যন্ত নেই—দেখে মনেই হয় না যে এই বাসাতে আমরা কত বছর কাটিয়েছি! দুপুর বেলা আমরা ভাড়া করা মাইক্রোবাসে করে স্টেশনে এসেছি। ট্রেনে একটা ফাস্ট ক্লাস কেবিন আমাদের জন্যে রিজার্ভ করে রাখা ছিল। কোথাও বেড়াতে যাবার সময় এরকম হলে কী মজাটাই না হতো কিন্তু আজ আমার কিছুই ভালো লাগছিল না। মোটামুটি সময় মতো ট্রেনটা ছেড়ে দিল। প্রায় সারাটা দিনই ট্রেনে কাটিয়ে দিলাম, সন্ধ্যেবেলা ট্রেন থেকে নেমে বাসে। বাস যখন আমাদের ছোট শহরটায় পৌঁছেছে তখন অনেক রাত, ঘুমে তখন আমার চোখ বন্ধ হয়ে আসছে। আম্মু টিফিন ক্যারিয়ার থেকে পরাটা আর কাবাব বের করে দিলেন। ঘুম ঘুম চোখে কীভাবে সেগুলো খেয়ে সাথে সাথে ঘুমিয়ে গেলাম। কোথায় ঘুমিয়েছি সেটাও ভালো করে জানি না, মনে হয় মেঝেতে কম্বল বিছিয়ে কেউ আগে থেকে বিছানা তৈরি করে রেখেছিল।

খুব ভোর বেলা বিচিত্র এক ধরনের শব্দে আমাদের ঘুম ভেঙে গেল। আমি ঘুম থেকে উঠে বুঝতেই পারছিলাম না আমি কোথায় আছি। বাসার ছাদটাকে মনে হলো অনেক উঁচুতে, বিছানাটা মনে হলো অনেক শক্ত। নিজের দিকে তাকিয়ে দেখি আমি রীতিমতো শার্ট-প্যান্ট-মোজা পরে ঘুমাচ্ছি, সবচেয়ে বড় কথা এই বিদঘুটে শব্দটা কী, আর সেটা কোথা থেকে আসছে আমি কিছুই বুঝতে পারছিলাম না। হঠাৎ করে আমার সবকিছু মনে পড়ল, আর আমি তখন তড়াক করে লাফ দিয়ে উঠলাম। শব্দটা কী সেটাও আমি সাথে সাথে বুঝতে পারলাম, এটা হচ্ছে কিচিমিচি পাখির ডাক! একটা দুইটা পাখির ডাক আমি শুনেছি, কিন্তু এটা তো দেখি রীতিমতো পাখির মেলা। আমার পাশে রাজু গুটিশুটি মেরে ঘুমাচ্ছে, রাজুর পাশে আব্বু। আম্মু আর আপুকে দেখতে পেলাম না। তারা মনে হয় অন্য ঘরে ঘুমাচ্ছে।

আমি উঠে জানালার কাছে গিয়ে বাইরে তাকালাম, সাথে সাথে বুঝে গেলাম এখানে এত পাখি কোথা থেকে এসেছে। এই বাসাটা একটা টিলার উপরে। এখান থেকে চারিদিকে যত দূর চোখ যায় শুধু গাছ আর গাছ! প্রত্যেকটা গাছে যদি একটা করেও পাখির ফেমিলি থাকে তাহলেই এখানে কয়েক হাজার পাখি হয়ে যাবে! আমি হতবাক হয়ে তাকিয়ে রইলাম, আমাদের আগের বাসার জানালা দিয়ে তাকালেই দেখা যেত আরেকটা বিল্ডিং, সেটার পিছনে ভালো করে তাকালে

Figure 4.8: Segmentation Technique Input.

4.5 Problems in Segmentation Technique

Some times the words are not segmented properly. They create mis shape of the segmented words. As a result the CNN can not detect the words properly. This is happened because of the threshold value of the contours and the noise in the page. So we had to check this kind of output after segmentation and avoid it to go to the CNN.

Figure 4.9: Faults in Segmentation Technique.

4.6 Experiment Data

Table 4.2: Word Count in Segmentation Technique

Image Number	Word Count
test.jpg	228
test1.jpg	53
test10.jpg	228
test11.jpg	767
test110.jpg	228
test111.jpg	228
test112.jpg	228
test113.jpg	227
test114.jpg	227
test115.jpg	227
test116.jpg	227
test118.jpg	227
test119.jpg	227
test120.jpg	228
test121.jpg	226
test122.jpg	226
test123.jpg	226
test124.jpg	226
test125.jpg	226
test126.jpg	226
test127.jpg	226
test128.jpg	226
test129.jpg	229
test13.jpg	229
test130.jpg	229
test131.jpg	228
test132.jpg	227
test133.jpg	227
test134.jpg	227
test135.jpg	229
test136.jpg	229
test137.jpg	229

Table 4.3: Word Count in Segmentation Technique

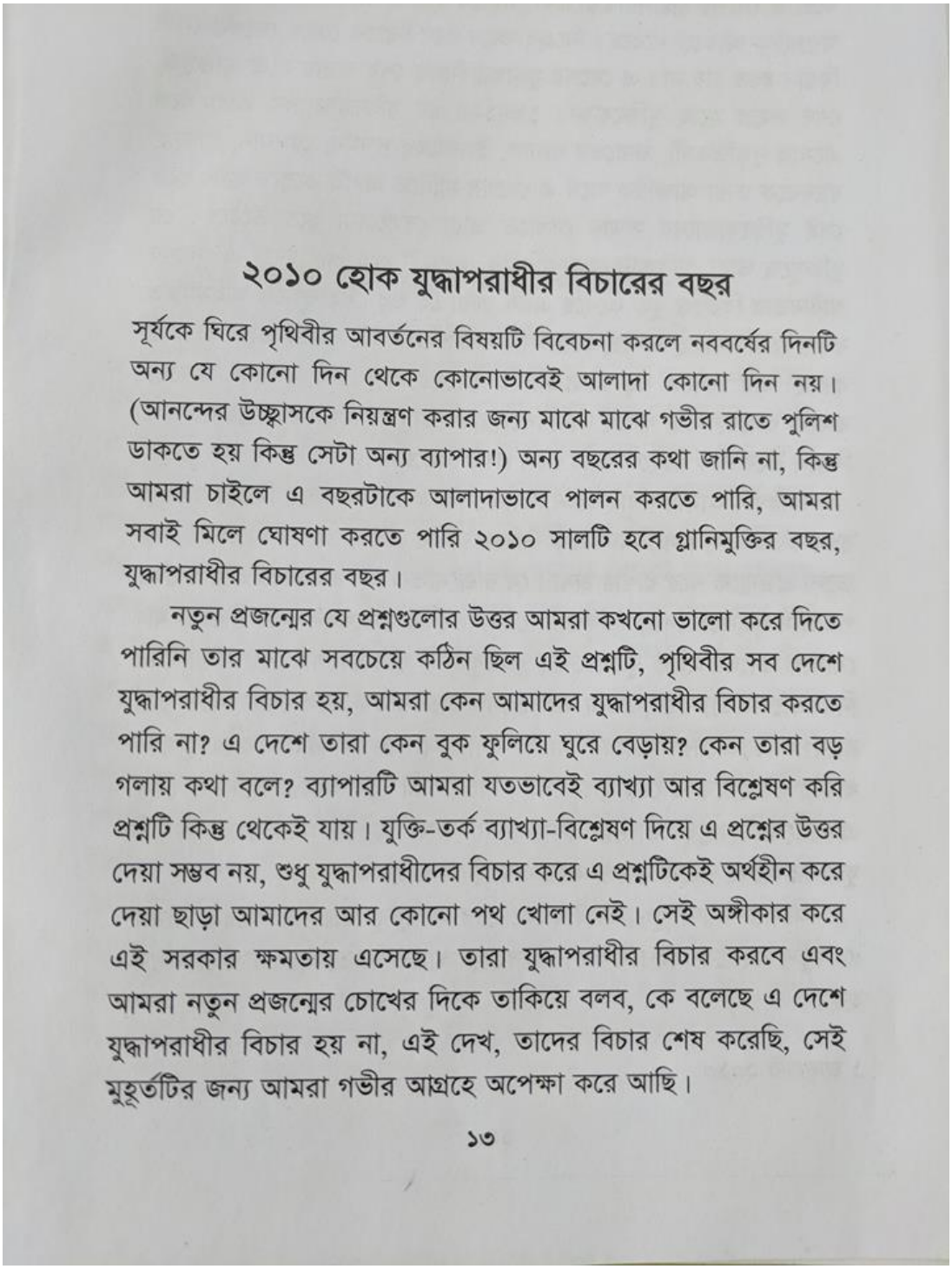
Image Number	Word Count
test138.jpg	229
test139.jpg	229
test140.jpg	228
test141.jpg	228
test142.jpg	36
test143.jpg	28
test144.jpg	37
test145.jpg	28
test146.jpg	39
test147.jpg	43
test148.jpg	46
test149.jpg	50
test15.jpg	229
test150.jpg	229
test16.jpg	229
test17.jpg	227
7 test18.jpg	227
test19.jpg	227
test2.jpg	50
test20.jpg	228
test21.jpg	229
test22.jpg	229
test23.jpg	226
test24.jpg	226
test25.jpg	226
test26.jpg	226
test27.jpg	226
test28.jpg	226
test29.jpg	226
test30.jpg	226
test31.jpg	226
test32.jpg	226
test33.jpg	227
test34.jpg	227

Table 4.4: Word Count in Segmentation Technique

Image Number	Word Count
test35.jpg	227
test36.jpg	227
test37.jpg	227
test38.jpg	227
test39.jpg	227
test4.jpg	43
test40.jpg	228
test41.jpg	228
test42.jpg	228
test43.jpg	228
test44.jpg	228
test45.jpg	228
test46.jpg	228
test47.jpg	228
test48.jpg	228
test49.jpg	228
test5.jpg	39
test50.jpg	228
test51.jpg	229
test52.jpg	228
test53.jpg	228
test54.jpg	228
test55.jpg	228
test56.jpg	228
test57.jpg	228
test58.jpg	228
test59.jpg	228
test6.jpg	28
test60.jpg	228
test61.jpg	229
test62.jpg	228
test63.jpg	228
test64.jpg	228
test65.jpg	228
test66.jpg	228
test67.jpg	228
test68.jpg	228
test69.jpg	228

Table 4.5: Word Count in Segmentation Technique

Image Number	Word Count
test7.jpg	37
test70.jpg	228
test71.jpg	228
test72.jpg	229
test73.jpg	228
test74.jpg	228
test75.jpg	228
test76.jpg	228
test77.jpg	228
test78.jpg	228
test79.jpg	228
test8.jpg	28
test80.jpg	228
test81.jpg	228
test82.jpg	228
test83.jpg	228
test84.jpg	228
test85.jpg	227
test86.jpg	227
test87.jpg	227
test88.jpg	227
test89.jpg	227
test9.jpg	36
test91.jpg	229
test92.jpg	233
test93.jpg	228
test94.jpg	207
test95.jpg	759
test96.jpg	438
test97.jpg	722
test98.jpg	747
test99.jpg	638



২০১০ হোক যুদ্ধাপরাধীর বিচারের বছর

সূর্যকে ঘিরে পৃথিবীর আবর্তনের বিষয়টি বিবেচনা করলে নববর্ষের দিনটি অন্য যে কোনো দিন থেকে কোনোভাবেই আলাদা কোনো দিন নয়। (আনন্দের উচ্ছ্বাসকে নিয়ন্ত্রণ করার জন্য মাঝে মাঝে গভীর রাতে পুলিশ ডাকতে হয় কিন্তু সেটা অন্য ব্যাপার!) অন্য বছরের কথা জানি না, কিন্তু আমরা চাইলে এ বছরটাকে আলাদাভাবে পালন করতে পারি, আমরা সবাই মিলে ঘোষণা করতে পারি ২০১০ সালটি হবে গ্লানিমুক্তির বছর, যুদ্ধাপরাধীর বিচারের বছর।

নতুন প্রজন্মের যে প্রশ্নগুলোর উত্তর আমরা কখনো ভালো করে দিতে পারিনি তার মাঝে সবচেয়ে কঠিন ছিল এই প্রশ্নটি, পৃথিবীর সব দেশে যুদ্ধাপরাধীর বিচার হয়, আমরা কেন আমাদের যুদ্ধাপরাধীর বিচার করতে পারি না? এ দেশে তারা কেন বুক ফুলিয়ে ঘুরে বেড়ায়? কেন তারা বড় গলায় কথা বলে? ব্যাপারটি আমরা যতভাবেই ব্যাখ্যা আর বিশ্লেষণ করি প্রশ্নটি কিন্তু থেকেই যায়। যুক্তি-তর্ক ব্যাখ্যা-বিশ্লেষণ দিয়ে এ প্রশ্নের উত্তর দেয়া সম্ভব নয়, শুধু যুদ্ধাপরাধীদের বিচার করে এ প্রশ্নটিকেই অর্থহীন করে দেয়া ছাড়া আমাদের আর কোনো পথ খোলা নেই। সেই অঙ্গীকার করে এই সরকার ক্ষমতায় এসেছে। তারা যুদ্ধাপরাধীর বিচার করবে এবং আমরা নতুন প্রজন্মের চোখের দিকে তাকিয়ে বলব, কে বলেছে এ দেশে যুদ্ধাপরাধীর বিচার হয় না, এই দেখ, তাদের বিচার শেষ করেছি, সেই মুহূর্তটির জন্য আমরা গভীর আগ্রহে অপেক্ষা করে আছি।

১৩

Figure 4.10: Parrent Image (test.jpg)

Table 4.6: Brightness In Different Pixels and Acuuracy Results

Image Name	Brightness In pixel (6-6)	Brightness In pixel (50-50)	Brightness In pixel (100-100)	Brightness In pixel (150-150)	Brightness In pixel (200-200)	Brightness In pixel (250-250)	Number of words found	Accuracy
test2.jpg	143	155	152	152	152	150	50	21.93%
test3.jpg	144	156	153	153	153	151	46	20.18%
test4.jpg	145	157	154	154	154	152	43	18.86%
test5.jpg	146	158	155	155	155	153	39	17.11%
test6.jpg	147	159	156	156	156	154	28	12.28%
test7.jpg	148	160	157	157	157	155	37	16.23%
test8.jpg	149	161	158	158	158	156	28	12.28%
test9.jpg	150	162	159	159	159	157	36	15.79%
test10.jpg	151	163	160	160	160	158	228	100.0%
test11.jpg	152	164	161	161	161	159	228	100.0%
test12.jpg	153	165	162	162	162	160	228	100.0%
test13.jpg	154	166	163	163	163	161	229	99.56%
test14.jpg	155	167	164	164	164	162	229	99.56%
test15.jpg	156	168	165	165	165	163	229	99.56%
test16.jpg	157	169	166	166	166	164	229	99.56%
test17.jpg	158	170	167	167	167	165	227	99.56%
test18.jpg	159	171	168	168	168	166	227	99.56%
test19.jpg	160	172	169	169	169	167	227	99.56%
test20.jpg	161	173	170	170	170	168	228	100.0%
test21.jpg	162	174	171	171	171	169	229	99.56%
test22.jpg	163	175	172	172	172	170	229	99.56%
test23.jpg	164	176	173	173	173	171	226	99.12%
test24.jpg	165	177	174	174	174	172	226	99.12%
test25.jpg	166	178	175	175	175	173	226	99.12%
test26.jpg	167	179	176	176	176	174	226	99.12%
test27.jpg	168	180	177	177	177	175	226	99.12%
test28.jpg	169	181	178	178	178	176	226	99.12%
test29.jpg	170	182	179	179	179	177	226	99.12%

Table 4.7: Brightness In Different Pixels and Acuuracy Results

Image Name	Brightness In pixel (6-6)	Brightness In pixel (50-50)	Brightness In pixel (100-100)	Brightness In pixel (150-150)	Brightness In pixel (200-200)	Brightness In pixel (250-250)	Number of words found	Accuracy
test30.jpg	171	183	180	180	180	178	226	99.12%
test31.jpg	172	184	181	181	181	179	226	99.12%
test32.jpg	173	185	182	182	182	180	226	99.12%
test33.jpg	174	186	183	183	183	181	227	99.56%
test34.jpg	175	187	184	184	184	182	227	99.56%
test35.jpg	176	188	185	185	185	183	227	99.56%
test36.jpg	177	189	186	186	186	184	227	99.56%
test37.jpg	178	190	187	187	187	185	227	99.56%
test38.jpg	179	191	188	188	188	186	227	99.56%
test39.jpg	180	192	189	189	189	187	227	99.56%
test40.jpg	181	193	190	190	190	188	228	100.0%
test41.jpg	182	194	191	191	191	189	228	100.0%
test42.jpg	183	195	192	192	192	190	228	100.0%
test43.jpg	184	196	193	193	193	191	228	100.0%
test44.jpg	185	197	194	194	194	192	228	100.0%
test45.jpg	186	198	195	195	195	193	228	100.0%
test46.jpg	187	199	196	196	196	194	228	100.0%
test47.jpg	188	200	197	197	197	195	228	100.0%
test48.jpg	189	201	198	198	198	196	228	100.0%
test49.jpg	190	202	199	199	199	197	228	100.0%
test50.jpg	192	204	201	201	201	199	228	100.0%
test51.jpg	193	205	202	202	202	200	229	99.56%
test52.jpg	194	206	203	203	203	201	228	100.0%
test53.jpg	195	207	204	204	204	202	228	100.0%
test54.jpg	196	208	205	205	205	203	228	100.0%
test55.jpg	197	209	206	206	206	204	228	100.0%
test56.jpg	198	210	207	207	207	205	228	100.0%
test57.jpg	199	211	208	208	208	206	228	100.0%
test58.jpg	200	212	209	209	209	207	228	100.0%
test59.jpg	201	213	210	210	210	208	228	100.0%
test60.jpg	202	214	211	211	211	209	228	100.0%
test61.jpg	203	215	212	212	212	210	229	99.56%
test62.jpg	204	216	213	213	213	211	228	100.0%
test63.jpg	205	217	214	214	214	212	228	100.0%
test64.jpg	206	218	215	215	215	213	228	100.0%

Table 4.8: Brightness In Different Pixels and Acuuracy Results

Image Name	Brightness In pixel (6-6)	Brightness In pixel (50-50)	Brightness In pixel (100-100)	Brightness In pixel (150-150)	Brightness In pixel (200-200)	Brightness In pixel (250-250)	Number of words found	Accuracy
test65.jpg	207	219	216	216	216	214	228	100.0%
test66.jpg	208	220	217	217	217	215	228	100.0%
test67.jpg	209	221	218	218	218	216	228	100.0%
test68.jpg	210	222	219	219	219	217	228	100.0%
test69.jpg	212	223	220	220	220	218	228	100.0%
test70.jpg	213	224	221	221	221	219	228	100.0%
test71.jpg	214	225	222	222	222	220	228	100.0%
test72.jpg	215	226	223	223	223	221	229	99.56%
test73.jpg	216	227	224	224	224	222	228	100.0%
test74.jpg	217	228	225	225	225	223	228	100.0%
test75.jpg	218	229	226	226	226	224	228	100.0%
test76.jpg	219	230	227	227	227	225	228	100.0%
test77.jpg	220	231	228	228	228	226	228	100.0%
test78.jpg	221	232	229	229	229	227	228	100.0%
test79.jpg	222	233	230	230	230	228	228	100.0%
test80.jpg	223	234	231	231	231	229	228	100.0%
test81.jpg	224	235	232	232	232	230	228	100.0%
test82.jpg	225	236	233	233	233	231	228	100.0%
test83.jpg	226	237	234	234	234	232	228	100.0%
test84.jpg	227	238	235	235	235	233	228	100.0%
test85.jpg	228	239	236	236	236	234	227	99.56%
test86.jpg	229	240	237	237	237	235	227	99.56%
test87.jpg	230	241	238	238	238	236	227	99.56%
test88.jpg	231	242	239	239	239	237	227	99.56%
test89.jpg	232	243	240	240	240	238	227	99.56%
test90.jpg	233	244	241	241	241	239	227	99.56%
test91.jpg	234	245	242	242	242	240	229	99.56%
test92.jpg	235	246	243	243	243	241	233	97.85%
test93.jpg	236	247	244	244	244	242	228	100.0%
test94.jpg	237	248	245	245	245	243	207	89.86%
test95.jpg	238	249	246	246	246	244	759	30.04%
test96.jpg	239	250	247	247	247	245	438	52.05%
test97.jpg	240	251	248	248	248	246	722	31.58%
test98.jpg	241	255	249	249	249	247	747	30.52%
test99.jpg	242	254	253	250	250	248	638	35.74%
test100.jpg	243	255	254	251	251	249	767	29.73%

Figure 4.11: Images of different brightness of test.jpg

4.6.1 Graph

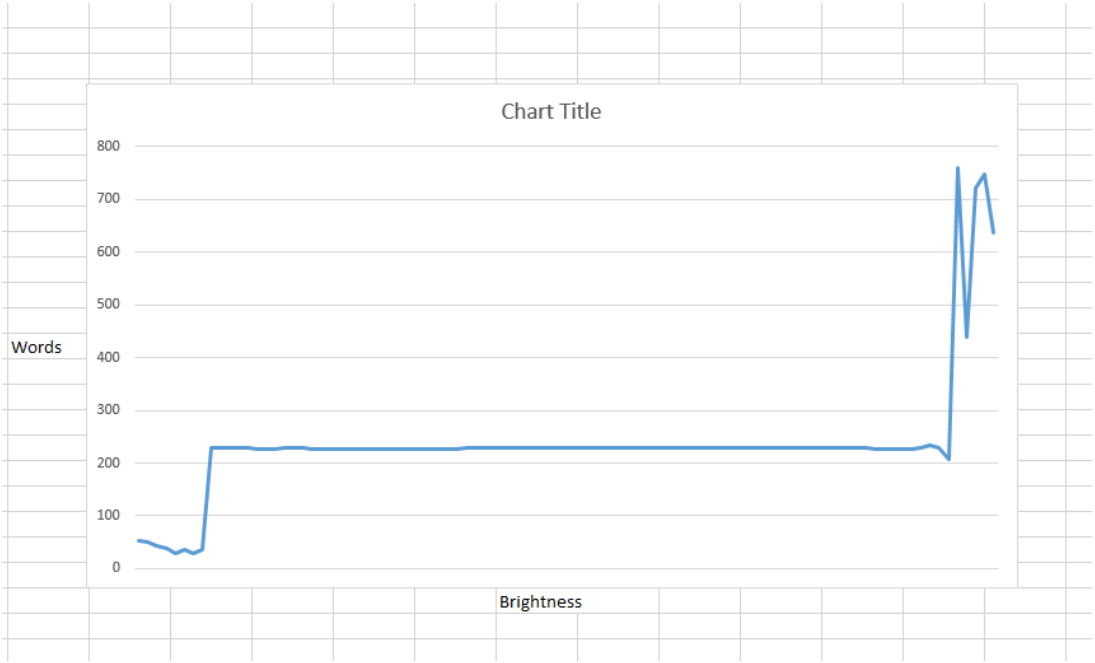


Figure 4.12: Brightness Graph of Segmentation Technique

As we increase brightness we see that the word count decreases.When we have images that has very low brightness after thresholding we have lower number of contours . The slowly as we increase brightness there are stable number of words then we we get very high brightness there are many noises after thresholding and too many noises are taken as words.

We see for very low and high brightness the also performs be badly but for a very large brightness range this performs greatly

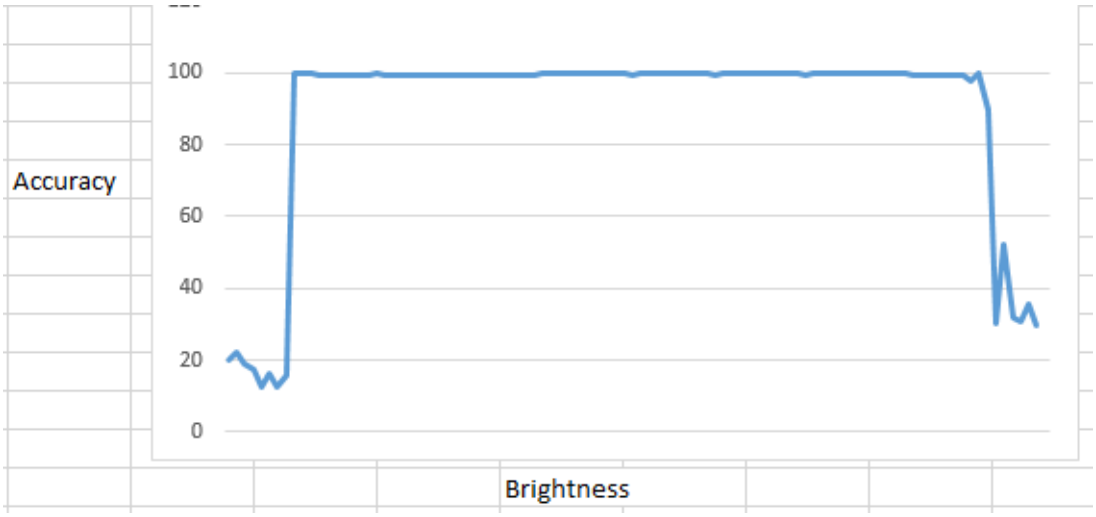


Figure 4.13: Brightness Graph of Segmentation Technique

4.7 Discussion

In our experiment we took a image of a pdf book of Jafor Iqbal sir. We first turn the image into grey scale. After that we used our algorithm to segment the words. First our algorithm find the contours of the page. After finding the contours it work with the words which are not straight in the page. First our algorithm rotate the words into straight words.

When all the images are straight then we started to find the threshold value of the pages. We get a graph with two local maxima. The contours which threshold value are in the local maxima of the page threshold value, we took these words as segmented words. The other contours are treated as noise of the page.

Our algorithm check this thing continuously for a good result. In this way the noises of the page are removed and we get the actual segmented words. We also get the segmented word counts which helps us to find the accuracy of the algorithm.

We took images of different brightness to check if our algorithm can detect the noise in the images. When the brightness of the images are increased the noises are become more bright. They look like a single word in the page. But the threshold value of the noises remained same. So when we check the threshold value maxima-minima level all the noises are removed from the set of the segmented words.

In all cases we check the word count found by our algorithm and the actual word count of the page. We get an accuracy of 90% through our algorithm. We have also successfully able to detect the punctuation through our algorithm which is very much needed in our OCR purpose. So our algorithm returns the segmented word images from the image of the page without any noises and punctuation's.

Then we use this segmented word images in CNN to get the text file of the image of the page.

In all other segmentation technique they have worked for only segmented images. They can't detect the rotated images. They treat this rotated images as noise and ignore this words. They can't also detect the punctuation's individually. They connect the punctuation with the images of the words.

Chapter 5

Methodology of OCR

5.1 WorkFlow of the Thesis

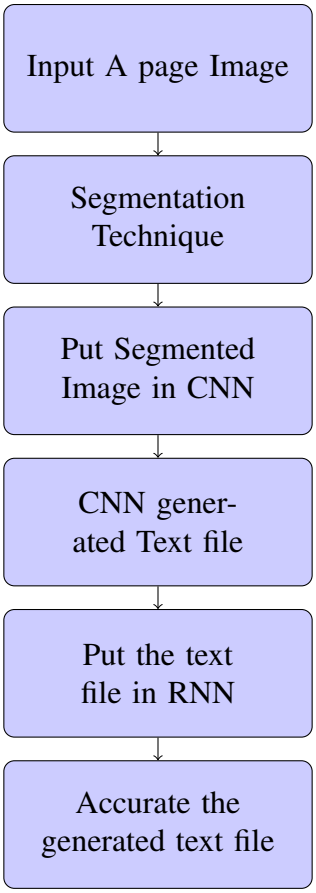


Figure 5.1: The workflow the thesis

5.2 Training CNN

We have to train our machine with Convolation neural network for our thesis purpose. For this reason we need a high configuration machine to train. For our training we used a laptop which model is $Lenovo Y50$. the configuration of the machine is defined below.

GPU : GTX 860M

RAM : 16 GB

Video Memory : 4 GB

5.3 Convolutional Layer

The convolutional layer is the core building block of a CNN. The layer's parameters consist of a set of learnable filters (or kernels), which have a small receptive field, but extend through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product between the entries of the filter and the input and producing a 2-dimensional activation map of that filter. As a result, the network learns filters that activate when it detects some specific type of feature at some spatial position in the input. Stacking the activation maps for all filters along the depth dimension forms the full output volume of the convolution layer. Every entry in the output volume can thus also be interpreted as an output of a neuron that looks at a small region in the input and shares parameters with neurons in the same activation map.

5.3.1 No Convolutional Layer

In the primary stage of our experiment we tried to train the data with no convolutional layer. For some dataset it worked perfectly. But when we increase our dataset or tried to work with big dataset thye accuracy of our experiment started to decrease. For large dataset this method results in poor accuracy.

5.3.2 Fully Connected Convolution Layer

Neurons in a fully connected layer have full connections to all activations in the previous layer, as seen in regular Neural Networks. Their activations can hence be computed with a matrix multiplication followed by a bias offset. We have used four kinds of fully connected convolution layer in our experiment.

1.[4096, 2048, 1024] : There are three layers in this fully connected convolutional layer. The First layer has 4096 nodes, the second layer has 2048 layer, the third layer has 1024 nodes. But it is not use able for our thesis work. Because The size of the image is large. So the memory of the GPU overflows. It stpos the training of the dataset.

2.[2048, 1024, 512] : There are three layers in this fully connected convolutional layer. The First layer has 2048 nodes, the second layer has 1024 layer, the third layer has 512 nodes.It has a accuracy of 90%.

3.[512, 256, 128] : There are three layers in this fully connected convolutional layer. The First layer has 512 nodes, the second layer has 256 layer, the third layer has 128 nodes.It has a accuracy of 75%.

4.[128]

There is only one layer in this fully connected convolutional layer. This layer has 128 nodes. It has an accuracy of 60-70%.

The dropout way used for our thesis is 0.9 and 0.4

5.4 Stride

Stride controls how the filter convolves around the input volume. In the example we had in part 1, the filter convolves around the input volume by shifting one unit at a time. The amount by which the filter shifts is the stride. In that case, the stride was implicitly set at 1. Stride is normally set in a way so that the output volume is an integer and not a fraction.

In our thesis we don't use stride value 1. If we use stride value 1 then the size of the images doesn't decreases significantly. As a result GPU memory overflows. So we get a bad accuracy in this case.

We have used stride valu 2 and 3. The accuracy in this case is 80-90%.

5.5 Activation function That are used

5.5.1 Activation function

Its just a thing (node) that we add to the output end of any neural network. It is also known as Transfer Function. It can also be attached in between two Neural Networks.

5.5.2 Rectified Linear Unit (ReLU

In the context of artificial neural networks, the rectifier is an activation function defined as the positive part of its argument:

$f(x) = x^+ = \max(0, x)$

where x is the input to a neuron. This is also known as a ramp function and is analogous to half-wave rectification in electrical engineering. This activation function was first introduced to a dynamical network by Hahnloser et al. in a 2000 paper in Nature[1] with strong biological motivations and mathematical justifications.[2]. It has been demonstrated for the first time in 2011 to enable better training of deeper networks [3], compared to the widely used activation functions prior to 2011, i.e., the logistic sigmoid (which is inspired by probability theory; see logistic regression) and its more practical[4] counterpart, the hyperbolic tangent. The rectifier is, as of 2018, the most popular activation function for deep neural networks

5.5.3 Tanh

The function is monotonic but function's derivative is not. The logistic sigmoid function can cause a neural network to get stuck at the training time. The softmax function is a more generalized logistic activation function which is used for multiclass classification of Tanh or hyperbolic tangent Activation Function

5.5.4 Sigmoid or Logistic Activation Function

The main reason why we used sigmoid function is because it exists between (0 to 1). Therefore, it is especially used for models where we have to predict the probability as an output.Since probability of anything exists only between the range of 0 and 1, sigmoid is the right choice.

The function is differentiable.That means, we could find the slope of the sigmoid curve at any two points. The function is monotonic but functions derivative is not. The logistic sigmoid function can cause a neural network to get stuck at the training time. The Sigmoid Function curve looks like a S-shape. The softmax function is a more generalized logistic activation function which is used for multiclass

5.5.5 Leaky ReLU

The leak helps to increase the range of the ReLU function. Usually, the value of a is 0.01 or so. When a is not 0.01 then it is called Randomized ReLU. Therefore the range of the Leaky ReLU

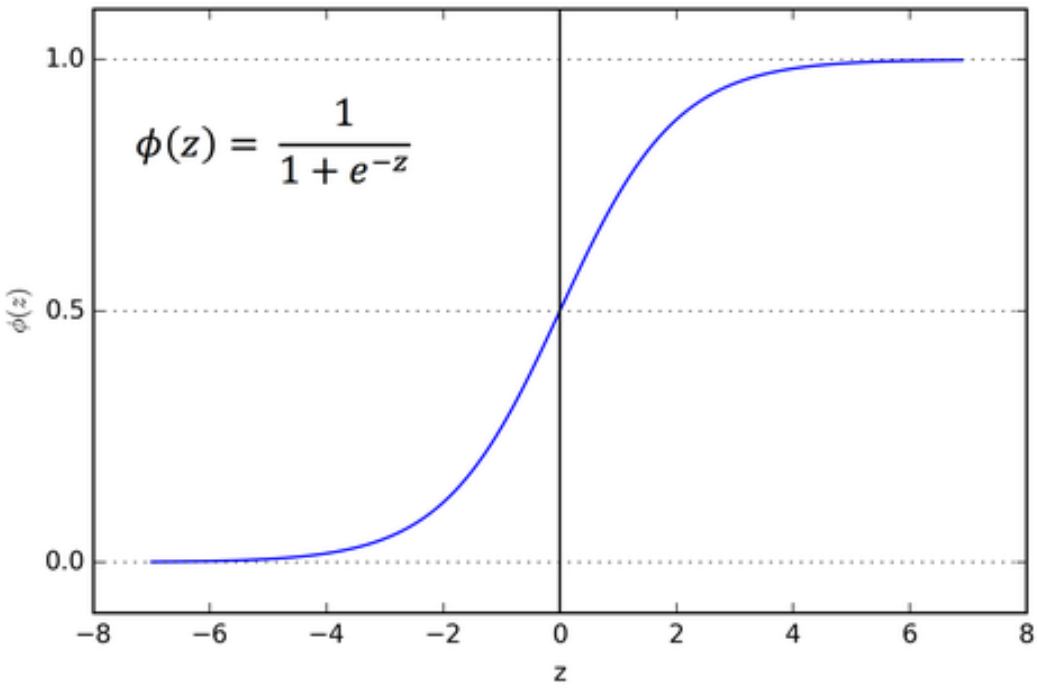


Figure 5.2: Sigmoid Function.

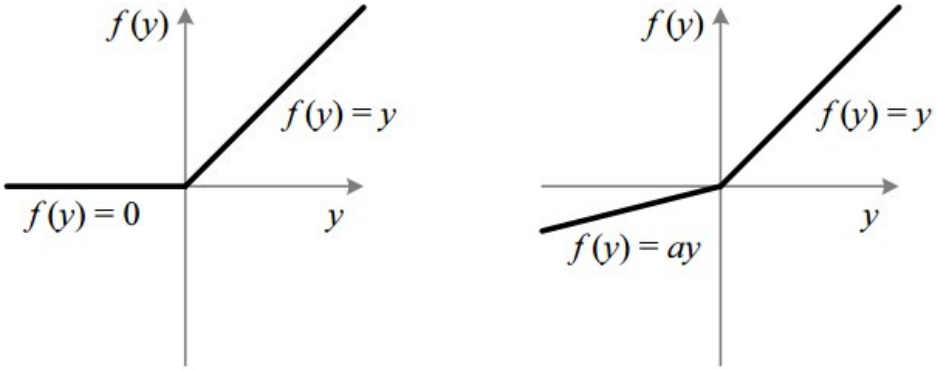


Figure 5.3: Leaky ReLU.

is (-infinity to infinity). Both Leaky and Randomized ReLU functions are monotonic in nature. Also, their derivatives also monotonic in nature.

5.5.6 Exponential Linear Unit ELU

ELU is very similiar to RELU except negative inputs. They are both in identity function form for non-negative inputs. On the other hand, ELU becomes smooth slowly until its output equal to - whereas RELU sharply smoothes. Notice that is equal to +1 in the following illustration.

5.6 Cost Function

5.6.1 softmax_ cross_ entropy_ with _ logits

It computes softmax cross entropy between logits and labels. (deprecated) It will be removed in a future version. Future major versions of TensorFlow will allow gradients to flow into the labels input on backprop by default. measures the probability error in discrete classification tasks in which the classes are mutually exclusive (each entry is in exactly one class). For example, each CIFAR-10 image is labeled with one and only one label: an image can be a dog or a truck, but not both.

5.7 Optimizer

5.7.1 Stochastic gradient descent

Stochastic gradient descent (often shortened to SGD), also known as incremental gradient descent, is a stochastic approximation of the gradient descent optimization and iterative method for minimizing an objective function that is written as a sum of differentiable functions. In other words, SGD tries to find minima or maxima by iteration. Adam is an optimization algorithm that can used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data. Adam was presented by Diederik Kingma from OpenAI and Jimmy Ba from the University of Toronto in their 2015 ICLR paper (poster) titled Adam: A Method for Stochastic Optimization

For our thesis we used Adaptive Gradient Algorithm (AdaGrad) that maintains a per-parameter learning rate that improves performance on problems with sparse gradients

5.8 Tenfold Graphs for Single Instance

5.8.1 TenFold Details

Parameters: Loss function : Softmax cross entropy.

Convolution layers : [9,64] [7,32]

Neural Network Structure : [4000,2000,1000]

Image Size : 32*300

Result of 10 fold:

84.7% , 83.2% , 70.8% , 76.5% , 77.64%, 82.14% ,78.3% , 87.3% , 73.2% , 87.22%

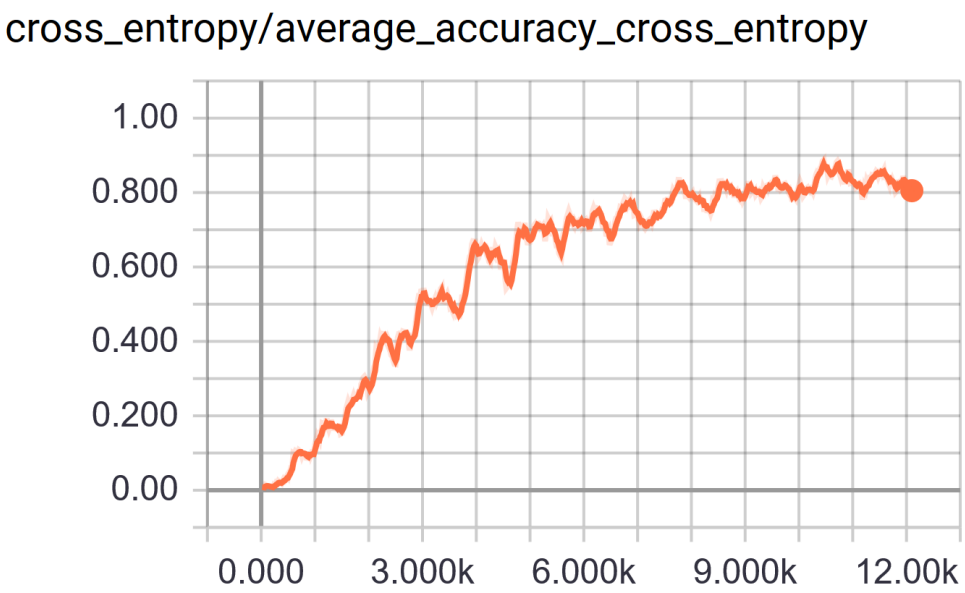


Figure 5.4: Graph Cross Entropy VS Average Accuracy Cross Entropy.

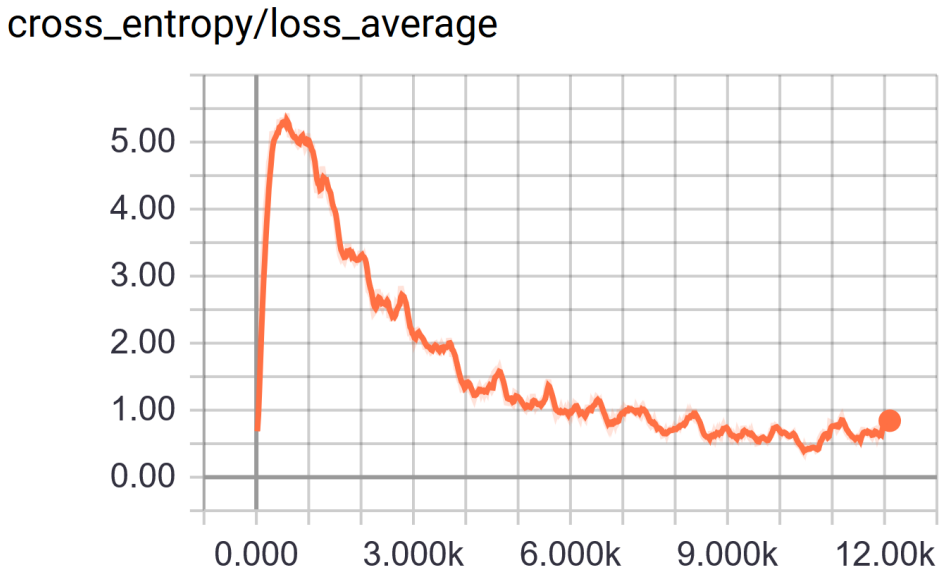


Figure 5.5: Graph Cross Entropy VS Loss Average.

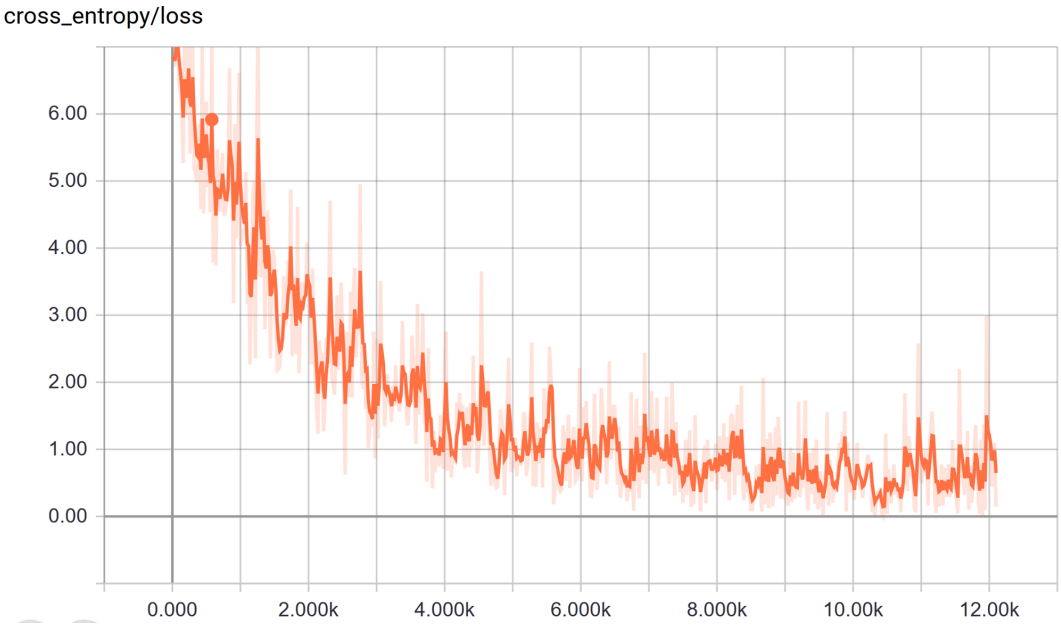


Figure 5.6: Graph Cross Entropy VS Loss.

5.8.2 Discussion

This was the result of 1023 classes but the accuracy was 90.1 . As we see the loss value was decreasing which proves to us that the neural network was constructed correctly even though it was not a very good one . The value of accuracy came to a steady state which gave us the hint the the training was enough and training even more will result in over fitting . The result was also giving a lot different values for different folds .

We got 10 accuracy after tenfold validation. They are 91.5%, 80.7%, 92.7224736048%, 93.26%,96.393288084
81.8%, 89.4570135747%, 92.4%, 91.2028657617%, 94.0573152338%

After experiments we have found out a technique by which we were able to get 90% accuracy. For this we used used images of 40*256 size which was not the main factor about the technique that was applied on the image. As we see different images of different words has different number of letters ant different boldness. So the length of the words are not same even though the size of the images were same And so for many images , the image contained a lot of space .to tackle this issue we cropped only the word from the images but this time the length of the images were very different and so we zoomed the images along along x-axis and turned every image to 40 * 256 We didn't need to zoom along y-axis because the images had words of same height

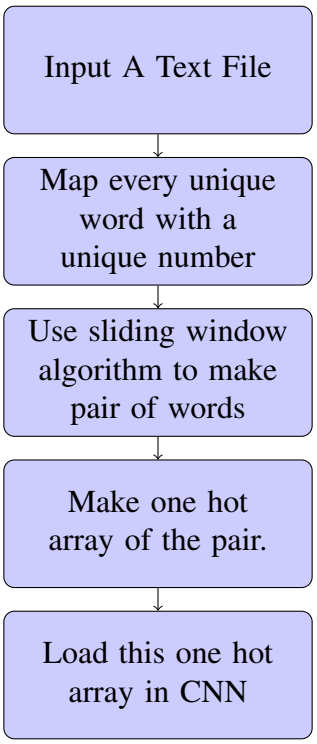
5.9 Data set of the Experiment

5.9.1 Making The Dataset

In our research, we used bangla text file as dataset. We first read the whole text file and separate every word individually. We separate the word through the split operation of the string.

5.9.2 Sliding window approach

Sliding window approach means to slide a window in every line of the dataset. We made pair of words through this sliding window approach. We took a line. Then we made pair with the first word and the next word. Through this process we made the whole dataset some pair of words. We also used n sliding window approach. In this approach we made pair with current word and next n words taking all possible combinations.



5.9.3 Making One hot array

one-hot is a group of bits among which the legal combinations of values are only those with a single high (1) bit and all the others low (0).A similar implementation in which all bits are '1' except one '0' is sometimes called one-cold. We represent the pair of words in one hot array. We create one hot array for every individual words. We gave every unique word a unique number. The maximum number of this unique number is the length of the one hot array. The one hot array only contains 0 and 1. The position of the array which is equal to the unique number of the word is 1. All the other position of the array is zero.After making this one hot array we train our convlatounal neural network with this array.

5.9.4 Using Binary Search algorithm

In our dataset there are many words which are repeated frequently. So we get the count of the every word how many times they are in the dataset through our code. ten we sort the unique words according to the count they are in the dataset. The we take 50 % words from the sorted words and check if 50% words of the dataset are covered. If 50% of the dataset is not covered then we take 50% words from the remaining sorted words and check again. Through this we get to know how many words we have to take to cover the 50% of the words.

5.9.5 Working with Training Set

In our research we are working with a huge number of image files. The size of the image files are also large. As a result if we try to train our machine with all the images at a single time it will crash. The memory will overflow and the GPU memory will be overloaded.

So we divide our dataset into small batches. For example if there are 50,000 images then the batch size can be from 128-512. If we increase the batch size the machine will train fast and the training time will decrease. but The GPU can be overloaded. Again if we decrease the batch size the training time will increase. Normally the choice of images for a batch is random. but for our research we have taken 512 images sequentially in every batch.

5.9.6 Working with Test set

For the test dataset we also use the batch method. We take a certain number of images in every batch. then we take 10 images randomly and print the accuracy for the images. If the dataset was small the accuracy was bad. but finally we get 90% accuracy for using tenfolt cross validation.

5.10 Making The CNN

5.10.1 Approach 1: Making OCR on character level

Making Dataset :

We got a dataset consisting of 500+ classes and each class consisted of 25 images . The images consisted of bangla vowels consonants and consonants with the diacritic form of the vowels .

Training :

This set gave us a very poor accuracy using both binary and grey scale version of the images . And using varied CNN structures.We also augmented them using a sine function (to add ripple in the images) and also skewed and sheered with but improvement was negligible .

5.10.2 Approach 2: Making OCR on Word level

Making Dataset :

Second approach was to use Bengali word as whole . The words consisted of all kinds of characters . vowels consonants and consonants with the diacritic form of the vowels and also compound characters .We generated the dataset by using 516 bijoy fonts . We used 1020 random words each has a class . So we had a total of almost 500000 images . The size of each image was 32*300 . Since there are so many images augmentation was not needed .

Training :

First We needed take several images into batches and turn each batch into serialized binary data. And than we saved them in SSD . Since it is not possible to keep the in RAM . Since the dataset was large we used a large number of nodes per layer . For the training we used Adam Optimizer . Out cost function was softmax cross entropy . Then used tenfold cross validation for measuring accuracy . Here are the results of the 10 different runs .We saved wrongly classified images for further analysis . Using them we found out wrongly generated images and fonts. For punctuation detection we used another CNN . But smaller in scale .

5.11 Making of the RNN

5.11.1 Letter Approach

For our classes we have correct and incorernt sentencess. But we see that even though corrent sentence has certain gramattical structure incoirrrecvt sentence can have any type of stucture . This is why we only got the output value of RNN and thesholded it to get two classes.

5.11.2 Parameter of RNN

We have used a lstm layer of size 2048, i.e., each lstm layer contains 2048 cells in our model. We need to fix a sort of a window to on which the model trains itself and predict on the next window based on its learning; we set this window to be 120 characters. Finally, we have trained using 50000 different batches. RNN took the input and it took all 120 characters of a sentence. So we have assumed a sentence has 120 character. Any sentence bigger than this was cut down to 120 characters and any sentence smaller than 120 characters was padded using a special value which did not correspond to any character value. Than the output of the last time step was put into the fully connected hidden layer.

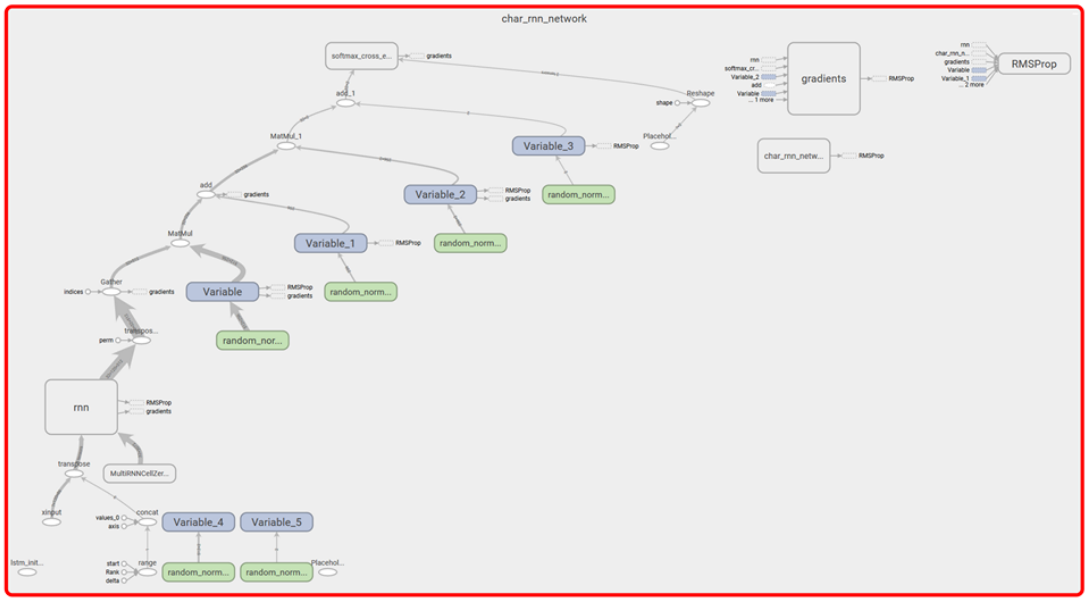


Figure 5.7: Flow Chart of Recurrent Neural Networks.

Table 5.3.3.1: RNN Hyper parameter values

Number of nodes in Input layer	120
Number of nodes in RNN layer	2048
Number of RNN layers	1
Timestep for RNN	120
Number of nodes in hidden layer after RNN	1024
Number of hidden layers	1
Output layer node numbers	2
Optimization algorithm	RMSProp
Loss Function	Softmax_cross_entropy
Learning Rate of optimization algorithm	0.9
Batch Size	64

Figure 5.8: Parameter of Recurrent Neural Networks.

5.11.3 Loss Function Graph Of RNN

While training a RNN a training dataset is used for training in combination with a validation dataset which is used for hyper parameter tuning. But there can be infinitely many kinds of wrong sentences (any permutation of letters). However, in our case, we trained the RNN without the validation dataset and focused on minimizing the loss function.

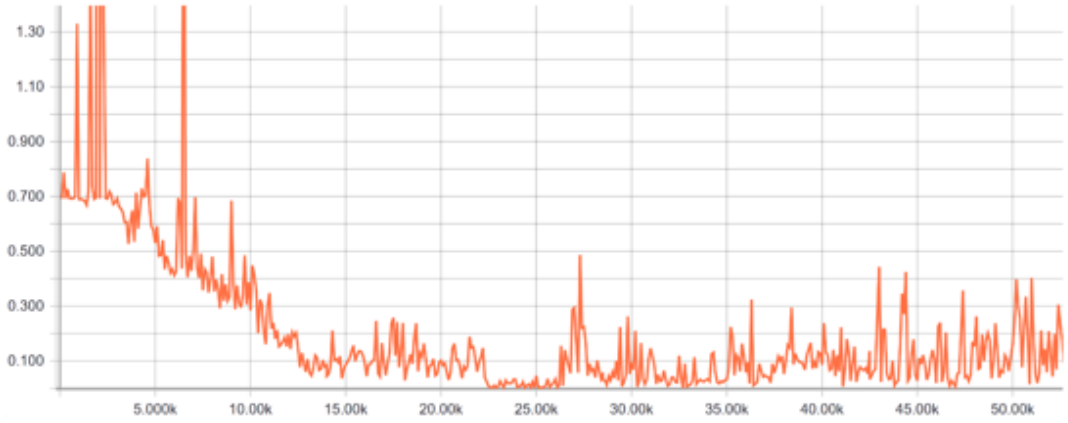


Figure 5.9: Loss Function Graph of Recurrent Neural Networks.

In Figure we present the graph representing the loss function during the training of the model and While training the RNN, the loss decreased so we can see that the network was indeed working. If we look closely, we can further see that there are some spikes in the range between 0th to 10000th iterations and also the loss value was quite high. Slowly the model becomes trained and more stable and the value of loss decreased.

5.11.4 Vector Approach

From image we see that there is a RNN cell and then we have several fully connected cells this was we constructed RNN and we had a softmax cross entropy function. For our case we took the input of sotfmax. The output of softmax was for future training.

For RNN we turned words to vector as we see this technique is widely used. We trained our own vectors using a windows of size 2 on sentences . We turned the words into one hot array mappings and then fed them to a CNN which had a word of a sentence on one side and another word on other side . for a sentence if it had 15 words there would be 5C2 pairs . This a an embedding was got for the word which was later fed into RNN .

For our classes we have correct and incorrect sentences. But we see that even though correct sentence has certain grammatical structure incorrect sentence can have any type of structure . This is why we only got the output value of RNN and thresholds it to get two classes. For

constructing the RNN, we have used TensorFlow library (written in python). TensorFlow is an open-source software library that is regularly used by developers for machine learning applications such as neural networks. In fact it is Google Brain's second generation system. While the reference implementation runs on single devices, TensorFlow can run on multiple CPUs and GPUs (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units). We have exploited its GPU capability extensively. We have used TensoFlow placeholders to take the values to be fed to RNN. The Basic LSTM Cell function of TensorFlow was used to create RNN later. After the RNN there was a fully connected hidden layer. This layer was created by TensorFlow with 2 variables, namely, weight and bias matrix. Then there was an output layer that was created just like the hidden one. The output of this layer was put into a cost function.Tensorflows sofmax cross entropy function was used. The optimization function for minimizing the output of cost function was RNNProp.

5.12 Experiment Results

Table 5.1: CNN Accuracy Result

Convolution Layer	Karnel	Stride	Fully Connected Layer	Activation Function	Dropout	Classifier	Loss function	Accuracy
”[(5,32),(5,36)]”	2	2	[1024]	Relu	1	Softmax	Softmax Cross entropy	87.1
”[(7,64),(7,36)]”	2	3	”[1024, 512]”	Relu	1	Softmax	Softmax Cross entropy	83.4
”[(7,64),(9,36)]”	2	2	[2048]	Relu	1	Softmax	Softmax Cross entropy	88.7
”[(7,64)]”	2	2	”[2048, 1024]”	Relu	1	Softmax	Softmax Cross entropy	89.1
”[(9,64)]”	2	2	[4096]	tanh	1	Softmax	Softmax Cross entropy	89.2
”[(7,64),(9,64)]”	2	2	”[4096,2048, 1024]”	Relu	1	Softmax	Softmax Cross entropy	90.1
”[(7,64),(9,64)]”	2	2	”[4096, 2048, 1024]”	sigmoid	1	Softmax	Softmax Cross entropy	88.6
”[(7,64),(9,64)]”	2	2	”[4096, 2048, 1024]”	Relu	1	Softmax	RMS	88.8
”[(7,64),(9,64)]”	2	2	”[4096, 2048, 1024]”	Relu	0.8	Softmax	Softmax Cross entropy	87.2

5.13 Discussion

Firstly we have tried with many fully connected layer structures.We see that giving a value of 4096,2048,1024 gives the highest accuracy. In one point we tried with stride value 3 which give less accuracy. We had softmax activation function.Firstly giving other activation lessens the accuracy for other parameters we see a highest tenfold accuracy of 90.1%. Changing any of the hyper meters lessens the accuracy.

For 1023 classes we had a total of 540000 images . Almost 523 images per class . for training set at each tenfold validation we chose 450000 images for training and 50000 for validation set. For testing we choose 40000 other images . Our testing accuracy was 92 percent

We also used several features of neural network like using dropout and using various activation functions. After using this function along with the new image size the accuracy improved a lot. The neural network had softmax cross entropy as is was the best classifier. After tuning values a lot we were able to get 91% accuracy which is for our 1023 classes (words).

We verifies the results by seeing automatically generated graphs. The graphs were generated by taking the accuracy values and loss function values on every iteration.

The way we trained the network was by using files as serialized object as we have 500000 images we could not read them every time we trained as it took a long time just to read them. So we first generated serialized python objects from these images.

The images were turned into their uncompressed format and then they were saved in python lists. After that several small lists were got from the large list where got from the lists containing 500000 images. 100 such lists were generated. After that they were saved in python pil files which can keep python objects. Each time we trained images from pil files were loaded. Images from 10 pil files we

Chapter 6

Conclusion

In our thesis we have tried to make a complete Optical Character Recognition for Bangla text for different Bangla font. Our motivation is to make a server where our user can submit there image of the printed document. The output of our system will be the text file of the image. It could be very useful in digitizing the deeds and official documents of the court. This document are typed by the typewriter. So we do not get soft document of this data. By using the OCR we can get the soft document of this data so that we can store this data an use it when it is needed

The first part of our thesis work is to segmentation of the image. In the past there are many method of segmentation of words. But in our case it is difficult because bangla has many connecting letters.Moreover if the image is not straight then this segmentation technique can give wrong outputs. So we introduce a new technique in segmentation process. We first calculate the angle of deviation of the image from the straight image. Then we rotate the image and make it straight. This is done in the prepossessing of the technique. Then we find the contours of the page and the threshold value of the page. If we draw a graph of this threshold values we will get local maxima and minima. We took the contours which are found between this local maxima. In this way we avoid the noise and punctuation of the page. The output of this step is the segmented image of the words in the image.

Then we pass this segmented image into our Convolution Neural Network. The CNN we used is trained with the dataset of 1020 random words each has a class of 512 bijoy fonts . So we had a total of almost 500000 images . The size of each image was 32*300. We needed take several images into batches and turn each batch into serialized binary data. And than we saved them in SSD . Since it is not possible to keep the in RAM . Since the dataset was large we used a large number of nodes per layer . For the training we used Adam Optimizer . Our cost function was softmax cross entropy . Then used tenfold cross validation for measuring accuracy. In our thesis we used stride but not the value value 1. If we use stride value 1 then the size of the images doesn't decreases significantly. As a result GPU memory overflows. So we get a bad accuracy in this case. We have used stride valu 2 and 3. The accuracy in this case is 80-90%.

We also used softmax_ cross_ entropy_ with _ logits as cost function of our thesis work. We first did every experiment in Tensorflow then we used Tenfold which gave us better result. In our thesis work we also used Recurrent Neural Network. We used RNN to validate the sentences. This is an extension of our thesis work. We used LSTM cells to build the RNN.

6.1 Future Work

We will try to extract letters from individual words in future . In bengali all the letters in words are connected which makes the task hard . In our dataset there are image whose photos were correctly generated but classification was wrong. We will use them for improving CNN even more . We will improve RNN even more to validate the sentences in the text file

References

[1] D. Archana A. Shinde, "Text pre-processing and text segmentation for ocr," *IJCSET*, vol. 2, pp. 810–812, January 2012.

[2] N. T. U PAL, "A contour distance-based approach for multi-oriented and multi-sized character recognition," *Sadhana*, vol. 34, p. 755765, October 2009.

[3] U. P. B. B. CHAUDHURI, "A complete printed banga ocr system," *Computer Vision & Pattern Recognition Unit*, vol. 31, no. 5, p. 531549, 1998.

[4] B. U.Pal, "Indian script character recognition: a survey," *Computer Vision & Pattern Recognition Unit*, vol. 3, no. 5, p. 18871899, (2004.

[5] B. U.Pal, "Ocr in bangla: an indo-bangladeshi language," *Conference C: Signal Processing*, vol. 3, (1994.

[6] M. A. N. B. Farjana Yeasmin Omee, Shiam Shabbir Himel, "A international journal of computer applications," *International Journal of Computer Applications*, vol. 21, (2011.

[7] T. W. F. K. U. Pal, N. Sharma, "Off-line handwritten character recognition of devnagari script," *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*.

[8] M. A. H. S. M. M. H. M. Khan, "A high performance domain specific ocr for bangla script," *Center for Research on Bangla Language Processing*.

[9] A.-I. Muhammad Golam Kibria, "Bengali optical character recognition using self organizing map," *IEEE/OSA/IAPR International Conference on Infonnatics*.

[10] S. A. S. H. C. M. R. Ahmed Asif Chowdhury, Ejaj Ahmed, "Optical character recognition of bangla characters using neural network: A better approach," *2nd International Conference on Electrical Engineering (ICEE)*.

Generated using Undegraduate Thesis LaTeX Template, Version 1.4. Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh.

This thesis was generated on Saturday 20th October, 2018 at 1:47pm.

tikz