Optical Character Recognition from Handwritten Bangla Texts

By

Faias Satter
Roll: 1707116

Department of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna 9203, Bangladesh
February, 2023

Optical Character Recognition from Handwritten Bangla Texts

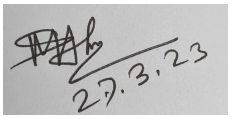
By

Faias Satter
Roll: 1707116

A thesis submitted in partial fulfillment of the requirements for the degree of
"Bachelor of Science in Computer Science and Engineering"

Supervisor:

Dr. Sk. Md. Masudul Ahsan
Professor
Department of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna, Bangladesh.



Signature

Department of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna 9203, Bangladesh
February , 2023

Acknowledgment

All glory to the Almighty Allah, whose blessing and mercy enabled us to finish our thesis work fairly. Following that, we gratefully recognize Dr. Sk. Md. Masudul Ahsan, Professor, Department of Computer Science and Engineering, Khulna University of Engineering & Technology, whose important recommendations, advice, direction, and honest cooperation enabled this work to be finished successfully. His intellectual guidance, encouragement, and direction create trust in us, and scientific study necessitates a significant amount of effort in learning and applying, as well as a broad perspective on problems from multiple perspectives.We would like to offer our heartfelt gratitude to all of the Department of Computer Science and Engineering's teachers, authorities, and staff members for their unflinching support in finishing this project. Last but not least, we'd want to thank our friends and family members for their everlasting support.

Abstract

Document processing has grown increasingly significant in a wide number of fields in Bangladesh as the use of computers has increased in the twenty-first century, including corporations, schools, hospitals, and many other industries. Reading handwritten paper and typing it into a computer takes a lengthy time. An optical character recognition system can scan a paper and extract text, making people' jobs more easier. While there are numerous OCR systems accessible in the software sector, finding a dependable equivalent solution for Bangla is tough. The goal of this system is to detect and recognize handwritten letters in an image. Adaptive thresholding, canny edge detection, Hough transform, morphological operation, and other image processing techniques are used to preprocess scanned pictures. A custom dataset is built with 3212 images each containing single/multiple characters. A total of 90 classes are used among which 50 characters, 10 numerals,17 most used compound characters and others were supportive characters. Single Shot Detector and Faster-RCNN is used to train the custom dataset. The dataset is trained using a transfer learning approach.An experimental comparison between those two models has been shown in the study. Also, The final model is built by training this learned model with images containing numerous characters. The Faster-RCNN model performs better, but the SSD model works faster. The system has performed admirably in given test data. Average macro precision, recall and F1-score are 76.76%, 78.26% and 73.61% respectively for character Detection and Recognition. While average macro precision, recall and F1-score for Faster-RCNN are 89.82%, 92.99% and 89.92%. A major limitation of this system is that it does not work with all the compound characters and punctuation marks. The system can be further improved with a rich dataset containing compound characters and by introducing a post-processing technique on word-correction.

Contents

List of Tables

List of Figures

CHAPTER I

Introduction

1.1 Background

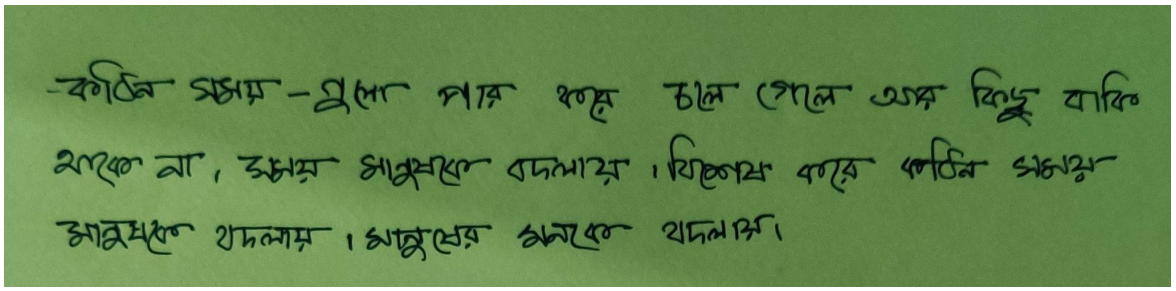
Computer vision, a field of artificial intelligence , enables computers and systems to extract useful information from digital photos, videos, and other visual inputs and to conduct actions or offer recommendations in response to that information. If AI gives computers the ability to think, computer vision gives them the ability to see, observe, and comprehend.

Computer vision has a wide range of applications, including Object Recognition and Classification in Traffic, Image Segmentation of Scans, Plant Disease Detection, Customer Behavior Tracking, and many more. The use of optical character recognition is one such application.OCR is the electronic or mechanical process of converting images of typed, handwritten, or printed text into machine-encoded text from scanned documents, photos of documents, scene photos (such as the text on signs and billboards in a landscape photo), or subtitle text superimposed on an image (for instance: from a television broadcast).

It is a common practice to digitize printed texts so they can be electronically edited, searched, stored more compactly, displayed online, and used in machine processes like cognitive computing, machine translation, (extracted) text-to-speech. This method of digitizing printed texts is used to enter data from printed paper data records such as passports, invoices, bank statements, computerized receipts, business cards, mail, printouts of static-data, or any suitable documentation. Pattern recognition, artificial intelligence, and computer vision are all areas of study in OCR. We'll work on digitizing handwritten Bangla text from pictures of papers in this study.

1.2 Motivation

Despite being the seventh most spoken language in the world, there is currently no trustworthy OCR on the internet. In Bangladesh, where there are more businesses and educational institutions than ever before, the demand for a trustworthy OCR has reached a record high. There have been several studies on character recognition, but relatively few studies on developing the entire OCR system.



(a) Input Image

কঠিন সময় গুলো পার করে চলে গেলে আর কিছু বাকি থাকে না। সময় মানুষকে বদলায়। বিশেষ করে কঠিন সময় মানুষকে বদলায়। মানুষের মনকে বদলায়।

(b) Target Output

1.3 Problem Statement

The goal of Optical Character Recognition for Handwritten Bangla Documents is to scan a picture of handwritten Bangla text, identify it, and display it.But Bangla characters are not as straightforward. When it comes to computer recognition, certain characters have very similar appearances and the letters are connected so closely that it is difficult to distinguish them as separate characters.

1.4 Bangla OCR

Characters in Bangla, unlike English, are connected by a line within a word. Characters do not have separate spaces in a handwritten document, as they do in a typed one. It's not

uncommon for pieces of one character to be absorbed by another. Because of these limits, vertical histogram projection cannot be employed with handwritten characters. This makes developing the system considerably more complicated.

1.5 Objectives

There are 50 normal characters, 10 number characters and 171 compound characters in Bangla language. The total of 231 characters is too many, however in this approach we will recognize the entire document character by character. The 50 standard characters, 10 numerals and 17 of the most common compound characters[40] will be the subject of this study.In this study, effort has been made to use some transfer learning approach.The purpose of this approach is to help students and company officials streamline their work. A bespoke dataset was used to detect characters in this procedure. A transfer learning strategy is utilized, in which a model trained with single characters serves as a checkpoint for training another model with images containing multiple characters.

1.6 Scope of the Thesis

The aim of the thesis is to process input images and give them the appropriate format so that the CNN system can perform optimally. As there is currently no high-quality Bangla OCR like that of English, Chinese, or Arabic, the system will only work with the Bangla language. The system will identify and segment handwritten characters, but it won't be able to recognize typed characters, which can be easily separated and recognized using conventional image processing and machine learning techniques. The scope of the thesis encompasses recognition of characters, sub-characters, combined characters, and digits.

1.7 Organization of the Thesis

The rest of the thesis is organized as follows:

- Chapter 2 presents brief overview of relevant research studies in image processing, skew correction, character segmentation, and character recognition.
- In chaper 3, the theoretical considerations necessary for one to understand the research is included.

- In Chapter 4, the numerous approaches that were used during the thesis study are detailed in depth.
- Explanation of the experimental results of the proposed methodologies is described in chapter 5.
- The limitations, future works and concluding remarks for the optical character recognition system are presented in Chapter 6.

CHAPTER II

Literature Review

2.1 Introduction

Many well-known studies offered many ways and produced numerous efficient mechanisms for creating an OCR. For character segmentation, several of the systems used image processing techniques including vertical histogram projection and the water reservoir approach. Some of the systems collaborated on improving the Hough transform algorithm. A couple of the systems contributed to the development of a full English OCR system. Various Convolutional Neural Networks, such as YOLO and MobileNet were used in some of the systems to detect isolated Bangla characters. Several tactics were used to successfully complete the detection and recognition tasks by those systems. Some systems used basic image processing approaches to perform the detection and recognition tasks, while others used various deep learning architectures.

2.2 Previous Works

Character Detection and recognition is done by many academics basically in two ways: deep learning-based methods (typically transfer learning approaches) or feature extraction via image processing. The following are some existing approaches that share the same domain as the proposed Bangla OCR system:

2.2.1 Image Processing Based Approaches

Bishnu and Chaudhuri [1] presented a method of recursive contour following to determine the extents within which the main portion of the character lies, in one of the zones across the height of the word. Their algorithm produces reasonable results if the subsequent characters do not contact in the contour following zone.

Pal et al. [2] proposed Water Reservoir Based Approach for Touching Numeral Segmentation. According to the water reservoir theory, if water is poured from the top and

bottom of the numeral, the cavity areas of the numerals where water will be retained are referred to as reservoirs. The segmentation approach was tested on 978 pictures of a numerical string from a French bank check with a courtesy amount. The findings were manually examined, and it was discovered that 94.34 percent of the related numbers were segmented accurately.

Kishan and Sharda [3] implemented different algorithms for skew detection correction in scanned document images. Over a variety of text document pictures, the scan line technique was found to reliably detect a range of skew angles. However, when compared to other approaches, it has the disadvantage of being extremely sluggish. In terms of skew detection, the hough transform approach was shown to be as good as the scan line algorithm while having a substantially lower time complexity.

Sekehravani et al. [4] implemented canny edge detection algorithm for noisy images. In a noisy environment, the Canny edge detection algorithm produces the best results, but its implementation is difficult and expensive. In the classic Canny technique, the Gaussian filter is utilized, which causes edge detail to seem blurry and has a poor effect in filtering salt-and-pepper noise. They employed the median filter to maintain the image's details while reducing the noise to solve this problem. Their research focused on how to apply Canny edge detection to noisy images and how to increase its accuracy. The findings revealed that the proposed method efficiently overcomes noise issues, preserves critical edge data, and improves edge detection precision.

Duan et al. [5] proposed an improved Hough transform for line detection. In terms of attributes, the suggested method is similar to the modified Hough transform (MHT) and the Windowed random Hough transform (RHT). It employs "many-to-one" mapping and the sliding window neighborhood technique to reduce computational and storage load.

Lu and Shridhar [6] gave a rundown of the most common strategies for extracting characters from handwritten words. They emphasized that the character segmentation and recognition procedures are closely intertwined in all existing handwritten word recognition methods. Handwritten numeral segmentation, handprinted word segmentation, and cursive word segmentation results were compared.

Lee et al. [7] suggested a new method for character segmentation and recognition that takes advantage of gray-scale image properties. The projection profiles and topography features retrieved from the gray-scale pictures are used to identify the character segmentation regions. To find a nonlinear character segmentation path in each character segmentation zone, they used a multi-stage graph search algorithm.

Dave [8] compared many segmentation methods for handwritten character identification, The pixel counting approach, histogram approach, smearing approach, stochastic approach, water flow approach, and mixed approach are among the methodologies discussed. The histogram strategy was discovered to be the most effective.

Zhou and Lopresti [9] looked at the issue of finding and extracting text from photos on the Internet. They described a text identification system based on color clustering and linked component analysis. Using a parameter-free clustering technique, the computer quantized the color space of the input image into a number of color classes. Based on their shapes, it then recognized text-like related components in each color class. A post-processing method is used to align text-like components into text lines. Despite the challenging nature of the input data, the experimental findings indicate that this technique is viable.

Sulem et al. [10] proposed a method for detecting text lines on handwritten sheets, which can include lines of various orientations, erasures, or annotations between main lines. The approach adopted an iterative hypothesis-validation strategy until the segmentation was complete. At each stage of the method, the best text-line hypothesis was developed in the Hough domain, taking into consideration the text-line component fluctuations. The validity of the line was then assessed in the picture domain using a proximity criterion that looked at the context in which the putative alignment was seen. Many text lines' ambiguous components are also noted.

Wang and Kangas [11] proposed a method for automatically extracting and detecting characters by locating character-like regions in natural color scene photographs. Alignment analysis was used to check the block candidates, specifically the character-like regions in each binary picture layer and the final constructed image, after linked component extraction using a multigroup decomposition approach. To get correct foreground pixels for each character in each block, priority adaptive segmentation (PAS) was used. The segmented characters were then justified using heuristic meanings such as

statistical features, recognition confidence, and alignment attributes. For a wide range of character types, shooting conditions, and color backdrops, the algorithms were found to be reliable. Their tests have given promising results in terms of real-world applications.

2.2.2 Deep Learning Based Approaches

Alom and Sidike [12] utilized a Deep Convolutional Neural Network for Bangla character recognition. In comparison to other deep learning approaches and previously proposed classical methods, their experimental results demonstrated that recent DCNN models such as DenseNet, FractalNet, and ResNet provide improved testing accuracy. In general, the DenseNet has a testing accuracy of 99.13 percent for handwritten digit recognition.

Coates et al. [13] divided the OCR problem into two parts: Text Detection and Character Recognition. They learned the features automatically from unlabeled data using unsupervised feature learning. This system used an unsupervised feature learning technique to learn a bank of image features from a set of picture patches extracted from the training data, and then assessed the features convolutionally over the training images.

Roy et al. [14] used deep learning algorithms to recognize handwritten Bangla compound characters. This system created a new Deep Convolutional Neural Network with supervised layerwise training. They used the CMATERdb 3.1.3.3 Bangla compound character dataset as a benchmark and saw a considerable reduction in error rate from 19 percent to 9.67 percent.

Fardous and Afroge [15] created a convolutional neural network (CNN) based model to distinguish handwritten solitary Bangla compound letters. The proposed model's performance was evaluated by training it on CMATERdb 3.1.3.3 and comparing the results with other current approaches for handwritten Bangla compound character recognition on a test dataset. Their network yielded a 95.5 percent accuracy rate.

Trier et al. [16] discussed feature extraction strategies for off-line recognition of segmented (isolated) characters. For different representations of the characters, such as solid binary characters, character contours, skeletons (thinned characters), or gray-level sub-images of each individual character, multiple feature extraction algorithms were

created. In terms of invariance qualities, re-constructability, and predicted distortions and variability of the characters, the feature extraction approaches were discussed.

Dutta and Chaudhury [17] used curvature features to recognize Bengali alpha-numeric character. They worked with both printed and handwritten characters. After thinning the smoothed character pictures and filtering the thinned images with a Gaussian kernel, the curvature properties were retrieved. A two-stage feed forward neural net-based recognition approach was used to classify the unknown data.

Purkaystha et al. [18] suggested a deep convolutional neural network-based technique for recognizing Bengali handwritten characters. The BanglaLekha-Isolated dataset was used to test their system. It is accurate to 98.66 percent on numerals (10 character classes), 94.99 percent on vowels (11 character classes), 91.60 percent on compound letters (20 character classes), 91.23 percent on alphabets (50 character classes), and 89.93 percent on practically all Bengali characters (80 character classes). The majority of their model's mistakes in the recognition test are due to characters' forms being too close together.

Ghosh et al. [19] proposed a model to recognize Bangla handwritten character using MobileNetV1 architecture. They employed MobileNet for handwritten character recognition in their paper. It recognized 231 classes (171 compound, 50 basic, and 10 numerals) with 96.46 percent accuracy, 171 compound character classes with 96.17 percent accuracy, 50 basic character classes with 98.37 percent accuracy, and 10 numerical character classes with 99.56 percent accuracy.

Islam and Rasel [20] proposed a Real-time Bangla license plate recognition system based on Faster R-CNN and SSD. The characters are recognized as separate things. After all of the characters and digits on the license plate have been detected, they are rearranged in the order that they appear on the plate. A total of 292 pictures were used to train the suggested model. Finally, using the given test dataset, the proposed methodology has a precision of 91.67 percent for detecting the characters on the license plate.

Chaudhuri et al. [21] suggested an English Language Optical Character Recognition System. The discrete cosine transformation is used to extract the features. Important soft computing techniques such as fuzzy multilayer perceptron (FMLP), rough fuzzy

multilayer perceptron (RFMLP), fuzzy support vector machine (FSVM), and fuzzy rough support vector machine are used to accomplish feature-based classification (FRSVM).

Bora et al. [22] proposed an OCR that combines CNN with the Error Correcting Output Code (ECOC) classifier. The CNN is used to extract features, while the ECOC is used to classify them. They investigated many common CNN classifiers in order to develop a suitable CNN for extracting features that can be utilized in conjunction with the ECOC classifier for accurate recognition of handwritten characters. The NIST handwritten character image dataset was used to train and evaluate the CNN-ECOC. Their simulation results revealed that CNN-ECOC outperforms the classic CNN classifier in terms of accuracy.

Abdullah et al. [23] proposed a YOLO-Based three-stage network for Bangla license plate recognition in Dhaka metropolitan city. They created a dataset by manually gathering 1500 different Bangladeshi vehicle license plate photos from the street to represent diverse real-world circumstances. They used the YOLOv3 algorithm to correctly locate the license plate and recognize the numbers in their research. They also created a Bangla scene character dataset with over 6400 characters, which they used to train a ResNet-20-based deep Convolutional Neural Network to recognize the character (CNN). In digit recognition, their proposed technique achieved more than 85% Intersection over Union (IoU). The ResNet-20-based CNN model correctly identified the Bangla character in the license plate 92.7 percent of the time.

Acharya et al. [24] proposed a deep learning-based system for large scale handwritten Devanagari character recognition. The Devanagari Handwritten Character Dataset (DHCD) is a new public image dataset for Devanagari script that they introduced in their research. Their dataset contains 92 thousand images of Devanagari script characters split into 46 different classes culled from handwritten papers. They also looked at how difficult it is to recognize Devanagari characters. They also offered a deep learning architecture for character recognition to go along with the dataset. In several recognition tasks, deep convolutional neural networks (CNN) outperformed standard shallow networks. To boost test accuracy, they adopted a Dropout and dataset increment strategy, which differs from the traditional Deep CNN character recognition approach. They were able to improve test accuracy by about 1% by using these strategies in Deep CNN.

Wei et al. [25] proposed a method in which with the help of a deep neural network, they were able to improve optical character recognition. Character recognition can be made more difficult in ancient or poorly printed texts because printed characters are frequently fractured and obscured. To train and perform OCR in their study, a deep neural network with Inception V3 was used. The Inception V3 network was built using 53,342 noisy character images gathered from receipts and newspapers. The suggested deep neural network demonstrated much superior recognition accuracy on low-quality text images, resulting in a 21.5 percent drop in the error rate when compared to existing OCRs, according to their experiments.

Anthony Adole et al.[41] proposed a method where the study examines the effectiveness of Faster-RCNN Inception Resnet V2 for offline handwritten Kanji character recognition and reports encouraging findings.

2.3 Discussion

The reviewed systems mostly worked with detecting a single handwritten character from an image using different machine learning and deep learning approaches. Purkaystha et al. [18] used deep convolutional neural network, Ghosh et al. [19] used MobileNetV1 and Fardous and Afroge [15] developed a convolutional neural network for detecting isolated Bangla characters and their models achieved 89.93%, 96.46% and 95.5% accuracy respectively. Bishnu and Chaudhuri [1], Pal et al. [2] and Dave [8] proposed recursive contour following, water reservoir approach and histogram strategy respectively for segmenting characters from words in a handwritten text but their results were not up to the mark to apply in an optical character recognition system.

Chaudhuri et al. [21] worked with developing a fully working OCR but the system is developed only for English language. Acharya et al. [24] proposed a deep learning-based system for large scale handwritten Devanagari character recognition which is an Indian language having a similar type of writing style to that of Bangla. Kishan and Sharda [3] proposed Hough transform for skew correction after comparing the method with other skew correction methods. Some of the systems used vertical histogram projection for character segmentation, however this method cannot be used in this Bangla OCR system

since the handwritten characters in Bangla are frequently written below or above one another.

CHAPTER III

Theoretical Considerations

3.1 Introduction

Bangla Handwritten Optical Character Recognition is a system that recognizes and converts handwritten Bangla texts into machine-encoded text. The character recognition procedure is divided into several parts.Pre-processing, Word Segmentation and Character Detection & Recognition. The word Segmentation is done by several image processing techniques such as, scanning in grayscale,binarization, noise reduction, and skew correction, dilation etc. After the words are segmented, each characters of the words will be detected and recognized.

3.2 Related Terms

Related terms include the theories of the technologies used for the system. Basic concepts of technologies such as MobileNet, Convolutional Neural Network,Single Shot Detector, Faster RCNN, ResNet, Activation function etc. are discussed in this section.

3.2.1 Convolutional Neural Network

A convolutional Neural Network are made up of three main layers:

- Convolutional layers
- Pooling layers
- Fully connected layers

Feature extraction is carried out using convolutional and pooling layers, which are referred to as feature extractor layers [26]. The classification layer is a fully linked layer that is in charge of categorizing the image.

The pooling layer receives an image and performs a series of pooling operations on it. Max pooling layers uses the maximum value on the provided kernel size. Figure 3.2 (a) shows how polling layers work.

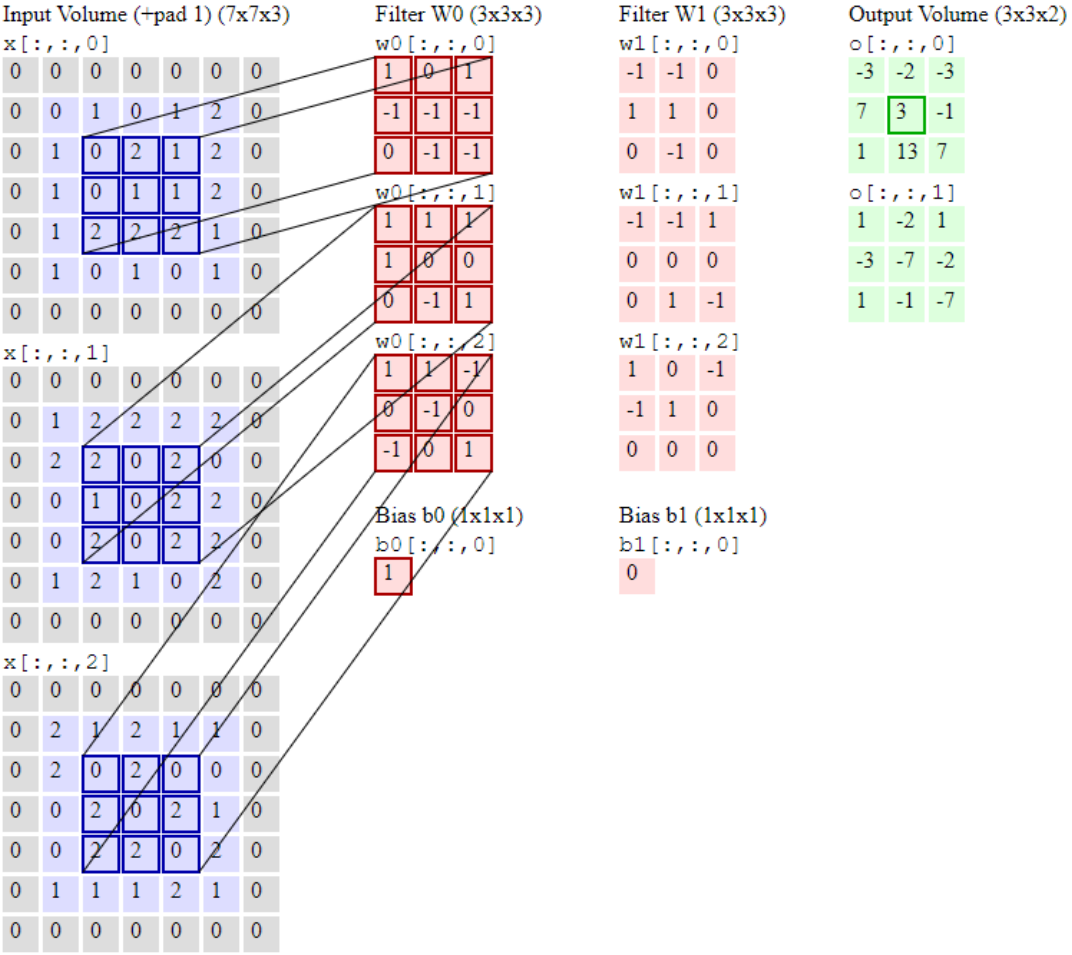


Figure 3.1: Operations of a convolutional layer [27].

Pooling layers are used to:

- Reduce the computational cost
- Make the model more generic

In a neural network, an activation function describes how a node or nodes in a layer turn the weighted sum of the input into an output. In deep neural networks, the activation function is employed to introduce nonlinearity [29].There are numerous popular activation functions, including the Binary Step Function, Exponential Linear Unit, ReLU, Linear Function, and so on. This model's activation function is ReLU. The Rectified Linear Unit activation function, or ReLU for short, is a piecewise linear function that outputs the positive input directly and zero otherwise.The fully connected layer receives input in the form of a flattened image vector (1D image) and computes a probability score for each label.

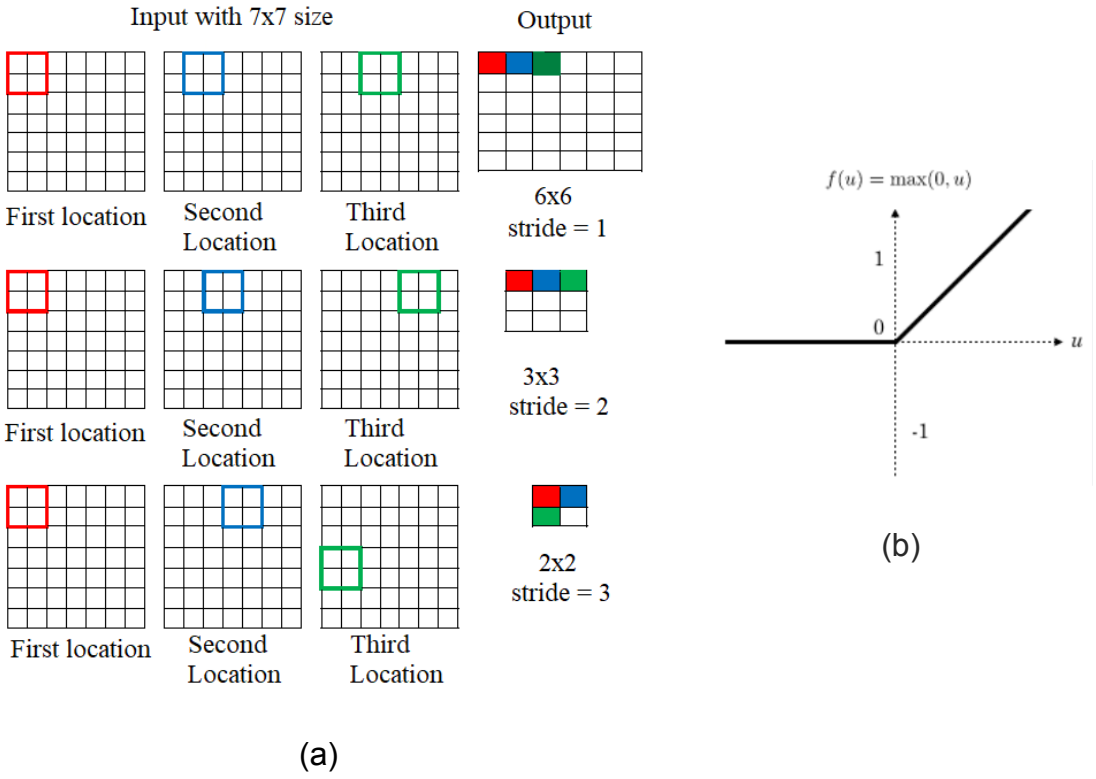


Figure 3.2: (a) Operation of pooling layers [27]. (b) Visualization of ReLU activation function [28].

In fully connected layers, the neuron applies a linear transformation to the input vector using a weight matrix. The product is then subjected to a non-linear transformation using a non-linear activation function.

3.2.2 MobilenetV2

MobileNetV2's design is a convolutional neural network. It is based on an inverted residual structure, with residual links connecting bottleneck levels. There are three layers for each bottleneck residual block.

- The first layer is pointwise convolution or 1×1 convolution with ReLU6.
- The second layer is depthwise convolution with ReLU6.
- The third layer is also pointwise convolution but without ReLU6.

Depthwise Convolution: A type of convolution in which a single convolutional filter is applied to each input channel. In this sort of convolution, a 2D convolutional filter is applied on each image channel. Depthwise convolution is seen in Figure 3.4.

The 1×1×n kernel is applied to a picture in pointwise convolution, where n is the number of channels in that image. Figure 3.5 shows an example of pointwise convolution.

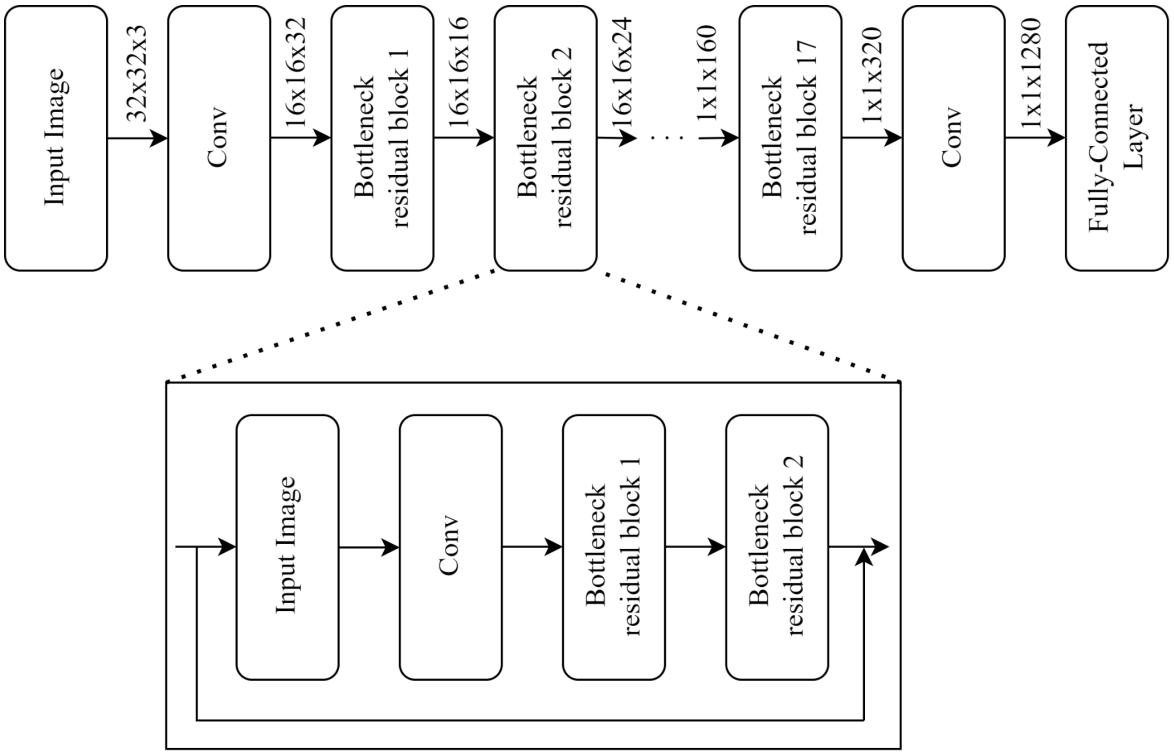


Figure 3.3: The architecture of the MobileNetv2 network [30].

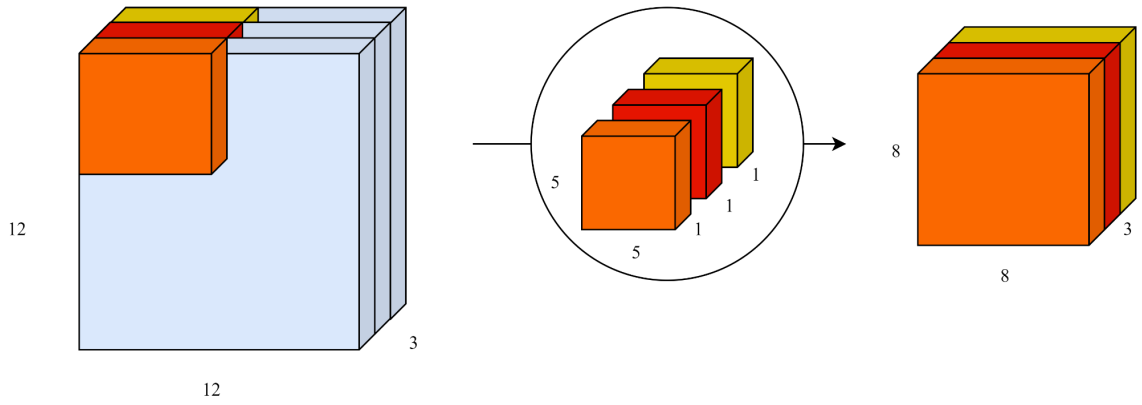


Figure 3.4: Depthwise convolution transforms a 12×12×3 image to an 8×8×3 image

using three kernels [31].

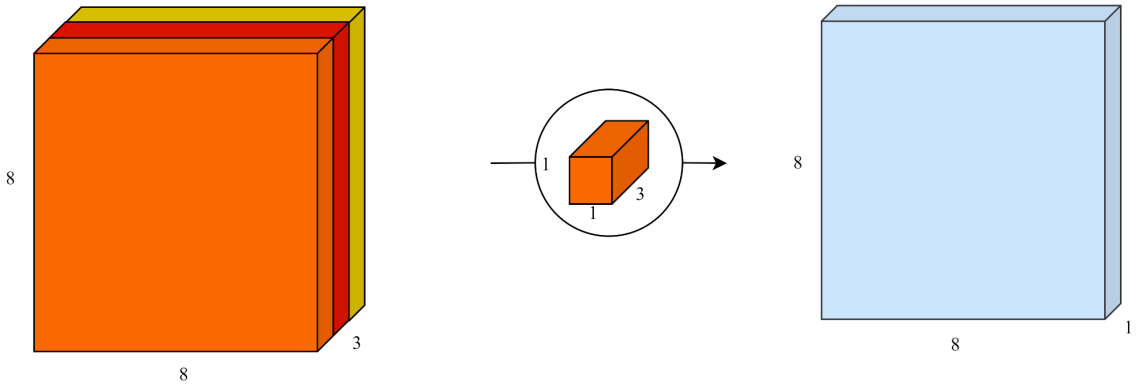


Figure 3.5: Pointwise convolution reduces a three-channel image to a single-channel image [31].

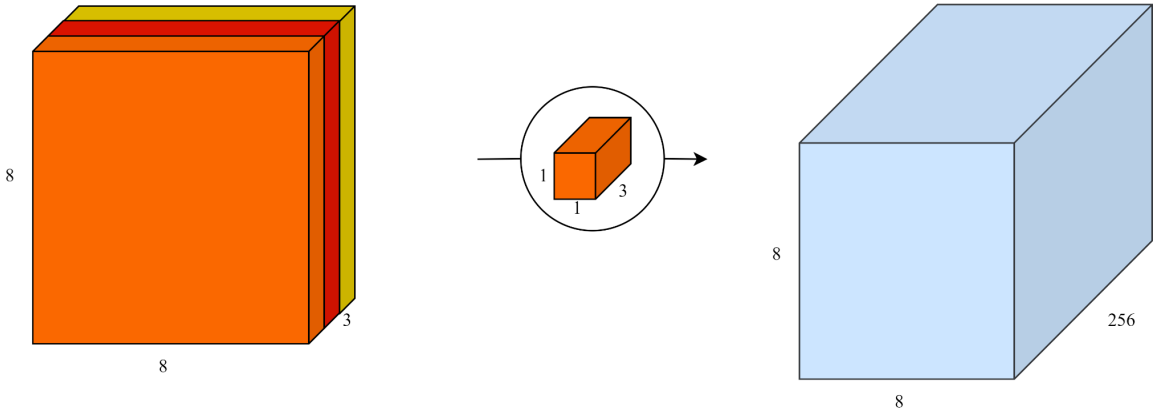


Figure 3.6: Pointwise convolution using 256 kernels that results in a 256-channel picture [31].

Pointwise Convolution: It reduces an n-channel image to a single channel image. To generate a multi-channel image, several pointwise kernels are applied to the input image. MobileNet Architecture uses depthwise and pointwise convolution instead of regular convolution. The network may be able to process more data in less time. MobileNets are low-power models with resource constraints that have been parameterized to meet the needs of diverse use cases.

3.2.3 Single Shot Detector

The Single Shot Detector (SSD) is a method for detecting image objects. Several things in an image can be detected in a single shot using a multibox [32]. To help in understanding SSD, here's an explanation of where the architecture's name comes from:

Single Shot: This refers to the network's ability to execute object localization and classification tasks in a single forward pass.

MultiBox: Szegedy invented the bounding box regression approach known as Multibox.

Detector: The network is an object detector that can also classify the objects it discovers.

Figure 3.7 demonstrates the architecture of SSD. SSD object detector has two components. One is base neural network and another one is additional convolutional layers.

Base Neural Network: As a base neural network, the MobileNetV2 CNN architecture encodes features at various scales. To generate feature maps, MobileNetV2 CNN layers without fully connected categorization layers are employed.

Additional convolutional layers: Additional convolutional layers are added to the base MobileNetV2 network for detection. This assists the model in generating box and class scores at various scales. For each image and its items, multiple bounding boxes and confidence scores are generated. Following this, non-maximum suppression is used to remove any remaining bounding boxes.

Matching Strategy: It is crucial to identify which default boxes correspond to ground truth detection during training and train the network accordingly. SSD employs a plethora of default boxes of varying sizes and aspect ratios throughout the image, as shown below. It makes use of 8732 standard boxes. This assists in determining the default bounding box

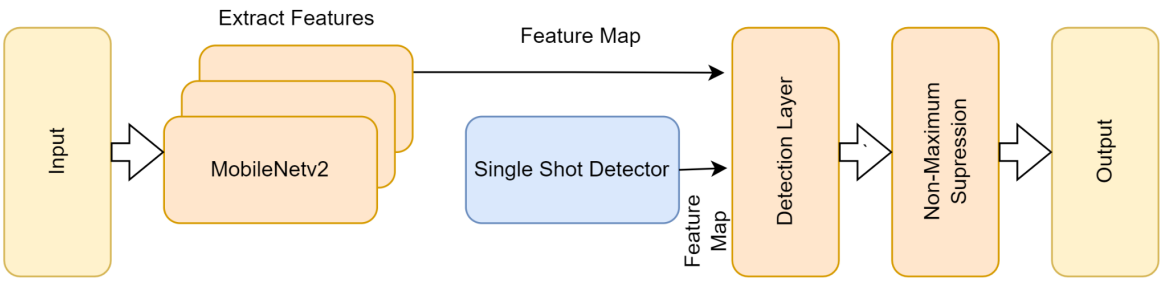


Figure 3.7: Architecture of a Single Shot MultiBox Detector (SSD).

that most nearly mimics the ground truth bounding box, including objects. Figure 3.8 depicts SSD's ground truth boxes and default bounding boxes.

The default boxes are matched to the ground truth boxes in terms of aspect ratio, location, and scale during training. Finally, the greatest overlap with reality is anticipated. Each prediction consists of numerous components:

- In the bounding box, the shape is offset. cx, cy, h, and w are offsets from the default box's center, as well as its height and width.
- Each bounding box class has its own confidence score.

The MultiBox loss function is utilized in SSD. The confidence loss and the localization loss are the two terms that make up this term.

Confidence Loss: Confidence loss measures the network's confidence in the objectivity of the computed bounding box.

Localization Loss: Localization loss measures the distance between the network's expected bounding boxes and the ground truth ones from the training set.

SSD selects the optimal default box with the greatest overlap with each ground truth box by determining the one with the highest IoU for each ground truth box (intersection over union). The boxes with the most overlap with the ground truth boxes are picked. The IoU (intersection over union) between selected boxes and ground truth is greater than 0.5. Figure 3.9 depicts a picture that matches default boxes with ground truth boxes and outputs the shape offset and confidence score for each bounding box. SSD is made up of a

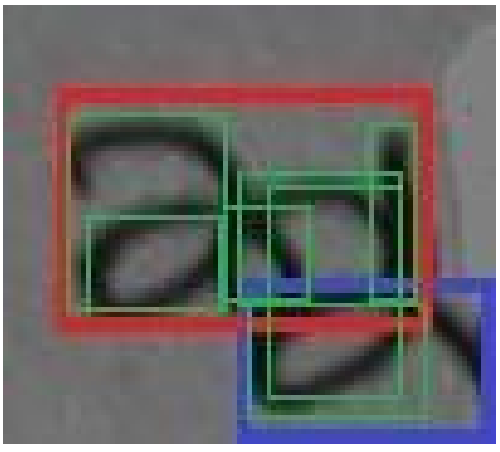


Figure 3.8: Default bounding boxes of SSD.

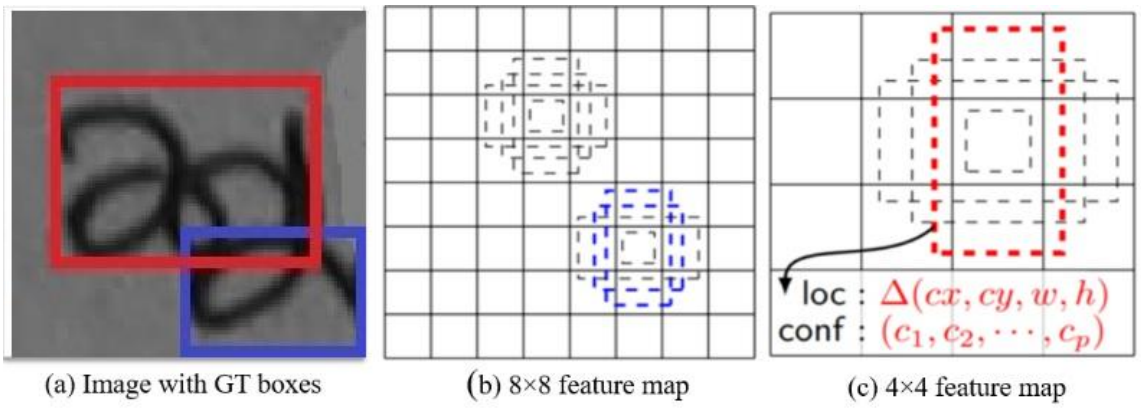


Figure 3.9: Matching technique of SSD architecture.

base network and extra convolutional layers, where the base network encodes features at various scales and the additional layers predict bounding box and class scores.

Non-Maximum Suppression: During an SSD forward pass, a large number of bounding boxes are created. Non-maximum suppression is the technique used to filter critical bounding boxes. Non-max suppression seeks to select the best bounding box for an object while rejecting or "suppressing" any others. The NMS takes two elements into account:

- The model determines the confidence score.
- The IOU (Intersection Over Union)

In addition to the bounding boxes, the model generates a confidence score. This score represents the model's belief that the requested object exists within the bounding box.

Figure 3.10 depicts the non-maximum suppression phases. Each detection box is assigned a class and a confidence rating. The greater the confidence value, the more likely it is that the box contains the class object. All objects are sorted according to their confidence rating in this strategy. If any bounding box has IoU>0.5 with a higher confidence bounding box, it will be eliminated. As a result, only the necessary bounding box will be present. Non-maximum suppression ensures that no object is identified more than once.

3.2.4 Inception-ResNetV2

The Inception-ResNet-v2 convolutional neural network was trained on over a million

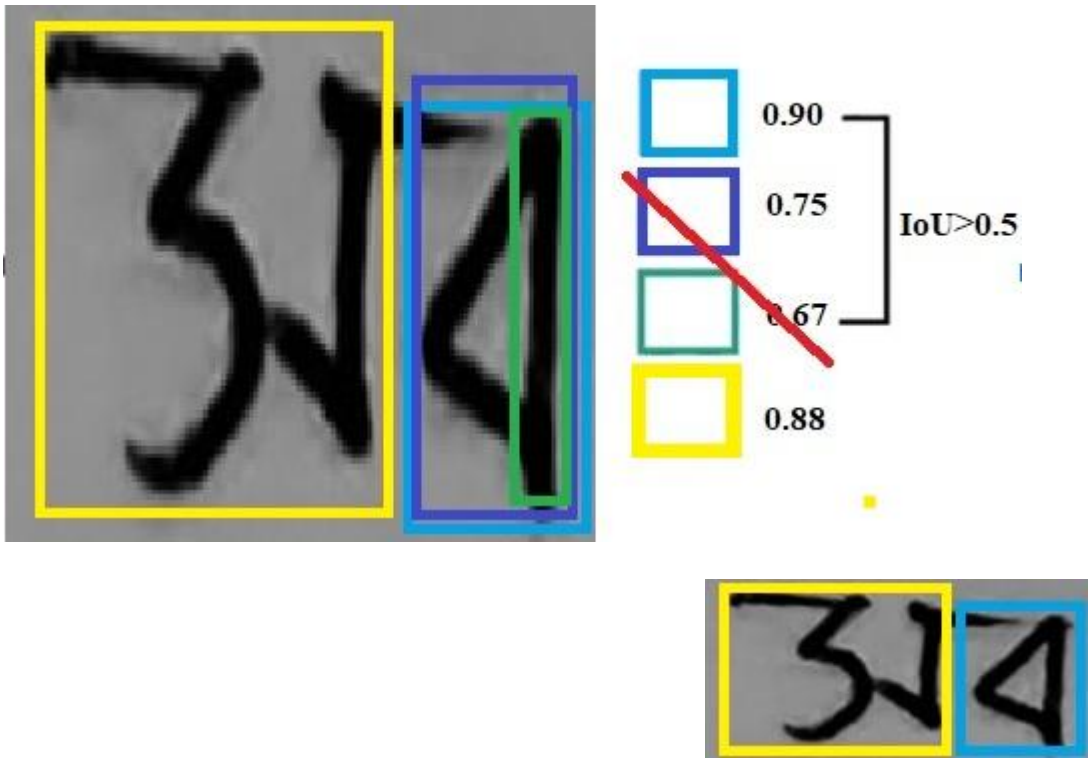


Figure 3.10: Selecting the right bounding box using non-max suppression.

photos from the ImageNet collection. The network takes a 299-by-299 picture as input and returns a list of estimated class probabilities as output. The schema-diagram of the model is stated in figure 3.11.The Inception-ResNet-v2 model makes use of residual networks, successfully increasing the original model's accuracy and convergence speed.

Figure 3.12 shows the architecture of stem block of Inception-ResNet-v2 model.

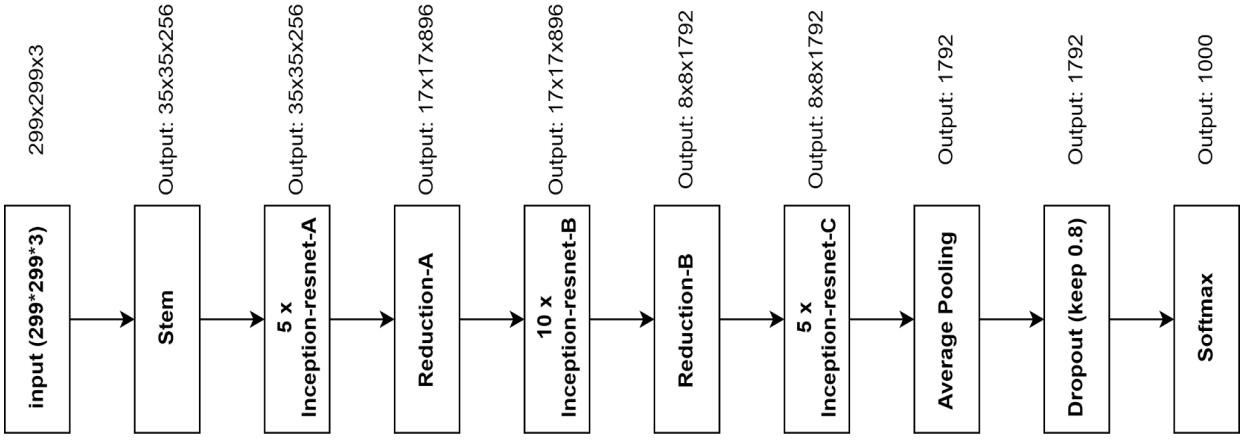


Figure 3.11: Schema Diagram for Inception-resNetv2 [42].

Each Inception block is followed by a filter expansion layer (1 x 1 convolution without activation), which is used before the addition to scale up the dimensionality of the filter bank to fit the depth of the input. Batch-normalization is employed solely on top of the standard layers in Inception-ResNet, not on top of the summations. Figure 3.13 shows the architecture of inception-resnet-A.

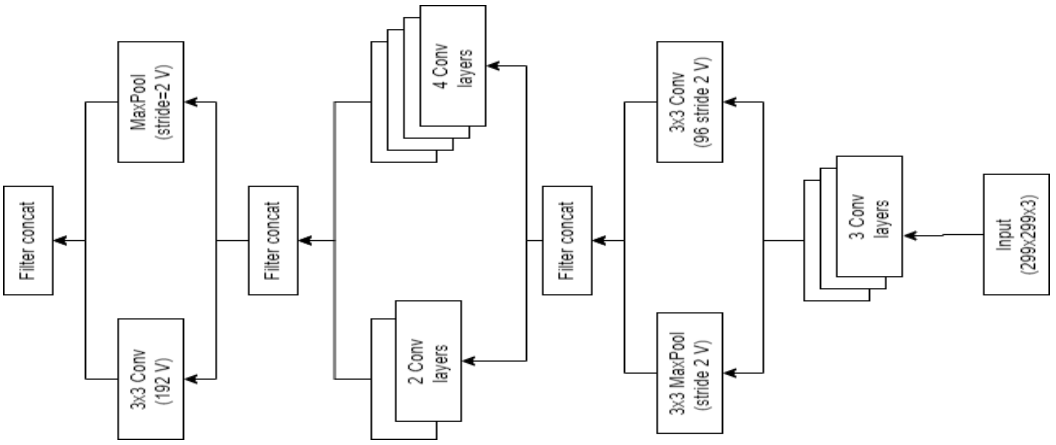


Figure 3.12:Stem architecture of Inception-ResNetV2[42]

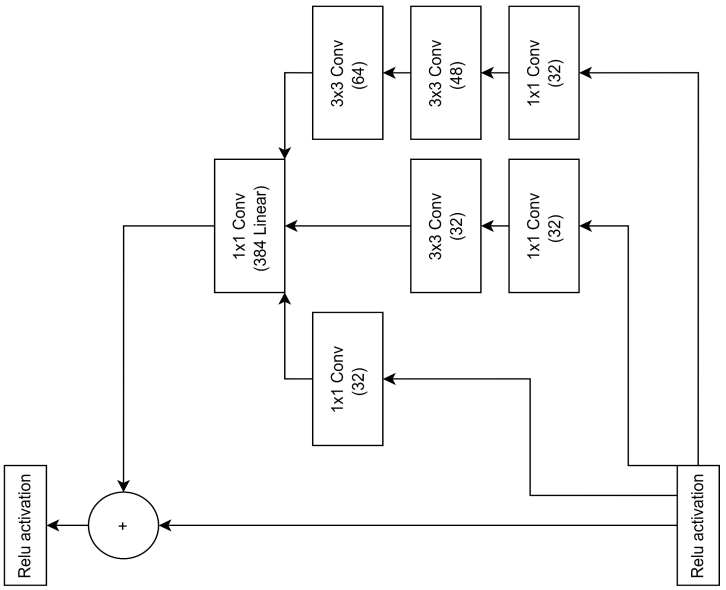


Figure 3.13: Inception-ResNet-A-Architecture[42]

3.2.5 Faster-RCNN:

In this experiment, Faster-RCNN takes Inception-ResNetV2 as it's base network for feature extraction. Faster R-CNN is a single-stage model that is trained from start to finish. It generates region proposals using a new region proposal network (RPN), which saves time over classic methods like Selective Search. It extracts a fixed-length feature vector from each region suggestion using the ROI Pooling layer.This model, basically has 3 issues to consider.

Region Proposal Network (RPN):

It is a completely convolutional network that generates ideas with different scales and aspect ratios. The RPN uses neural network terminology with attention to inform the object detection where to look.

Anchor Box:

Rather than utilizing image pyramids (many instances of the same image at different scales) or filter pyramids (multiple filters at different sizes), this work introduced the concept of anchor boxes. An anchor box is a specified scale and aspect ratio reference box. There are numerous scales and aspect ratios for a same region when there are multiple reference anchor boxes. This is similar to a pyramid of reference anchor boxes. Each zone is then transferred to a distinct reference anchor box, detecting objects at various scales and aspect ratios. There are 2 parameters for anchor box named scale and aspect-ratio.

Convolutional Layers:

The convolutional computations are shared across the RPN and the Fast R-CNN. This reduces the computational time.

Intersection Over Union(IoU):

Each anchor is assigned a positive or negative objectness score based on the Intersection-over-Union(IoU) in order to train the RPN .The area of intersection between the anchor box and the ground-truth box is divided by the area of union of the two boxes to get the IoU. The IoU scale runs from 0.0 to 1.0. The IoU is 0.0 when there is no junction. The IoU increases as the two boxes get closer together until it reaches 1.0.

Objectness Score is assigned from IoU.Here IoU is considered >0.6. Rest of the working procedure is similar to what mentioned in the SSD model architecture.

Figure 3.14 shows the architecture of Faster-RCNN model.

3.2.6 Conclusion :

Aside from these, some other theoretical considerations will be provided in brief where it will be necessary.

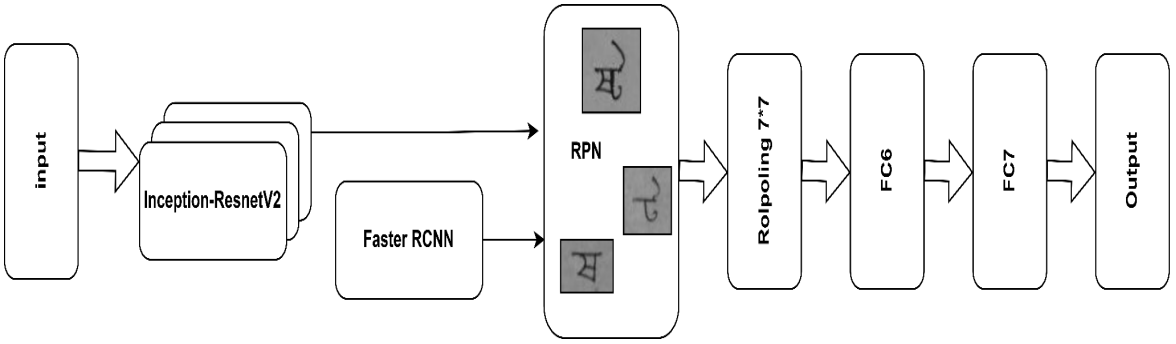


Figure 3.14: Architecture of Faster-RCNN with Inception-ResNetV2[43]

CHAPTER IV

Proposed Methodology

4.1 Introduction

Bangla Handwritten Optical Character Recognition is a system that recognizes and converts handwritten Bangla texts into machine-encoded text. The character recognition procedure is divided into several parts. After scanning in grayscale, the image is pre-processed with binarization, noise reduction, and skew correction. A bounding box is created around each word using morphological operations. Using a single shot detector and Faster-RCNN and convolutional neural network , each character is segmented and recognized from words. To obtain the final result, the identified words are rearranged.

4.2 Working Procedure

The total working procedure of the system can be divided into following parts:

- Processing Input
- Word Segmentation
- Character Segmentation and Recognition

While the Word Segmentation is based on the use of image processing, the character segmentation and recognition require the use of deep learning.

4.2.1 Processing Input

Multiple image processing techniques are used to process the input image. It is done for two purposes. word Segmentation and Character Detection and Recognition. Preprocessing is concerned with increasing image quality in order to improve system performance. Scanning, binarization, skew correction, and other processes are included.

Word detection, on the other hand, builds a bounding box around each word and delivers cropped photos for character segmentation and recognition.

A. Preprocessing

Preprocessing is done primarily to increase the identification of picture features and the accuracy of the optical character recognition system. Pre-processing is used to improve the optical character recognition system. The purpose of pre-processing is to improve the image's quality so that it can be evaluated more effectively. By preprocessing the image, unwanted distortions can be removed and certain qualities that are useful for the application can be amplified. The flow-chart is depicted in figure 4.2.

I. Grayscale Image

To achieve the best results, the image is scanned in grayscale mode. A grayscale photograph contains only information about light intensity and no color information. A grayscale image set is a collection of grayscale images with no identifiable color. Grayscale graphics are far larger than line art because each pixel in a grayscale image can store one of 256 distinct hues. While scanning in black and white allows the scanner to scan faster, some character information is lost. Figures 4.3 (a) and (b) illustrate the input and output images after grayscale transformation.

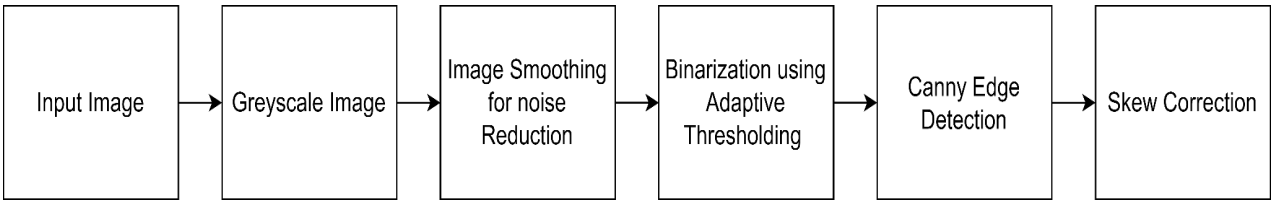


Figure 4.1: Flowchart of the Pre-processing Stage.

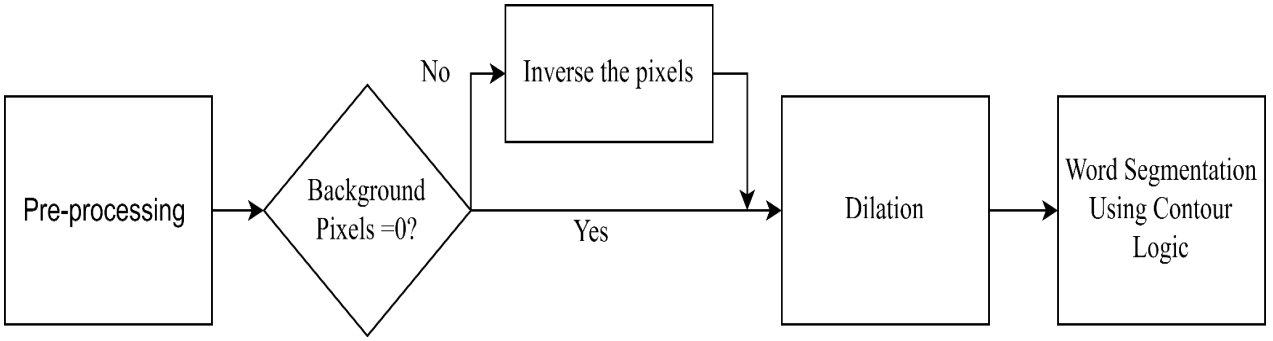
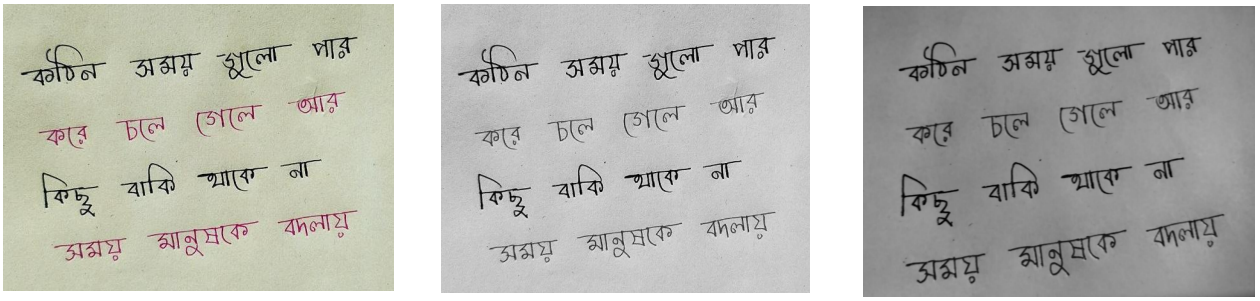
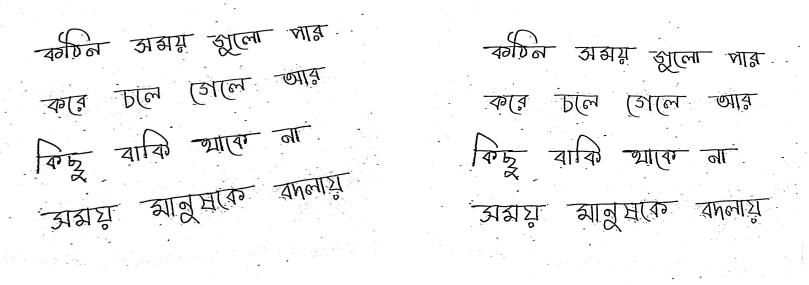


Figure 4.2: Flowchart of the Word Segmentation Stage.



(a) Input Image (b) Grayscale image (c) Smoothing



(d) Adaptive Thresholding (e) Skew-corrected Image

Figure 4.3: Input and output images of the preprocessing steps.

II. Noise Reduction

This procedure smoothens out the image by removing little dots/patches. The Non-local Means Denoising algorithm is used to de-noise images, using numerous computational enhancements. This algorithm replaces a pixel's value with an average of the values of several other pixels. Small patches centered on other pixels are compared to the patch centered on the pixel of interest, and only pixels with patches similar to the present patch are averaged.Figure 4.3(c) shows image after noise reduction.

III. Binarization

Image binarization is the process of dividing pixel values into two groups, white for the background and black for the foreground. Binarization primarily relies on thresholding. Binarization is accomplished using the Adaptive Thresholding method. This method computes a threshold for a small part of a picture based on its surroundings and location. In other words, rather of having a single defined threshold for the entire image, each little section of the image has a different threshold dependent on its location, resulting in a smooth transition [33]. There are many pixels in damaged document photos that cannot be clearly identified as foreground or background due to background noise or variations in contrast and illumination. In case of Adaptive thresholding, if there is shadow on the texts, it can be handled. But there should be enough spaces between each words. The output after Adaptive Thresholding is depicted in figure 4.3 (d)

IV. Skew Correction

Document skew identification and rectification is a difficult phase in image analysis. The image may be slightly warped at times. The skew must be corrected in order for the OCR technology to recognize lines. Skew correction techniques include the projection profile method, the Hough transformation method, the topline method, the scanline method, the base-point method, and others. In this system, the Hough Transformation method is chosen because it is the most efficient.

Canny edge detection detects edges while also increasing contrast and removing picture noise [34]. Hough lines, which employ the Hough transform, are used to identify whether or not those edges are lines. Hough Transform requires good edge detection to be efficient and produce meaningful results. The Hough transform method requires edge detection in order to generate an edge image, which will then be used as input to the algorithm.

Canny edge detection has following steps:

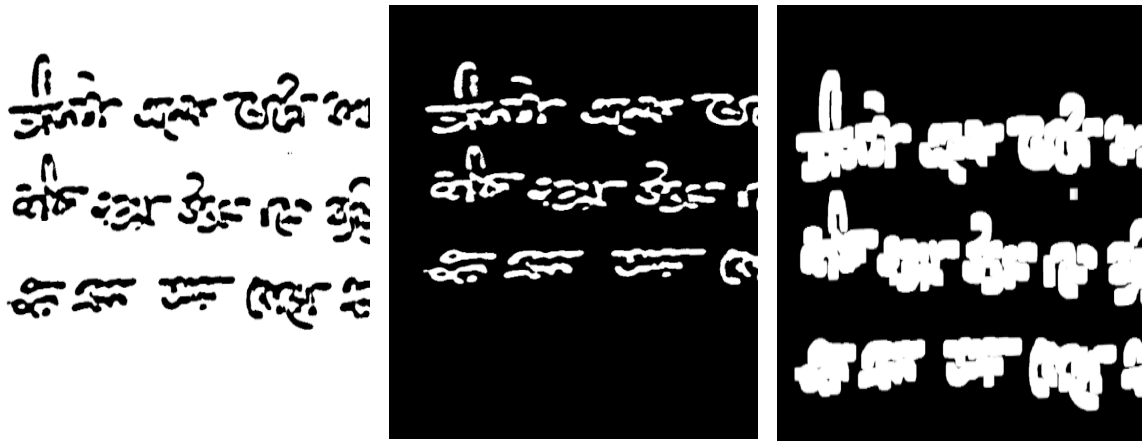
- Gaussian smoothing
- Differential operator along x and y axis
- Non-maximum suppression
- Hysteresis thresholding

Hough transform is applied after canny edge detection. From image space to parameter space, the Hough transform turns a co-linear line to intersecting points [35]. Four points are discovered using this technique. Slightly rotated scanned images are converted to skew-corrected images using these points. Figures 4.3 (e) shows skew corrected version of the input image.This version of the image will be used later by imposing the bounding boxes got in the Word Segmentation level.

4.2.2. Word Segmentation

I. Dilation

Dilation is the process of adding pixels to the edges of objects in an image. The amount of pixels added or subtracted from an image's objects is determined by the size and shape of the structuring element used to process the image. Before dilation, the pixels are inverted if the background pixels are white and foreground pixels are black. Otherwise, the pixels aren't flipped. This morphological process is done with a kernel of size 11*11. There are multiple iterations of dilations until there is a step where the current number of segments or words is less than the number of segments of previous iterations.The figure-4.4 showcases the dilation.



(a)Result after applying Thresholding

(b) Inverting the pixels

c) After 1 iteration of Dilation

Figure 4.4: Sample image showcasing iterations of dilation.

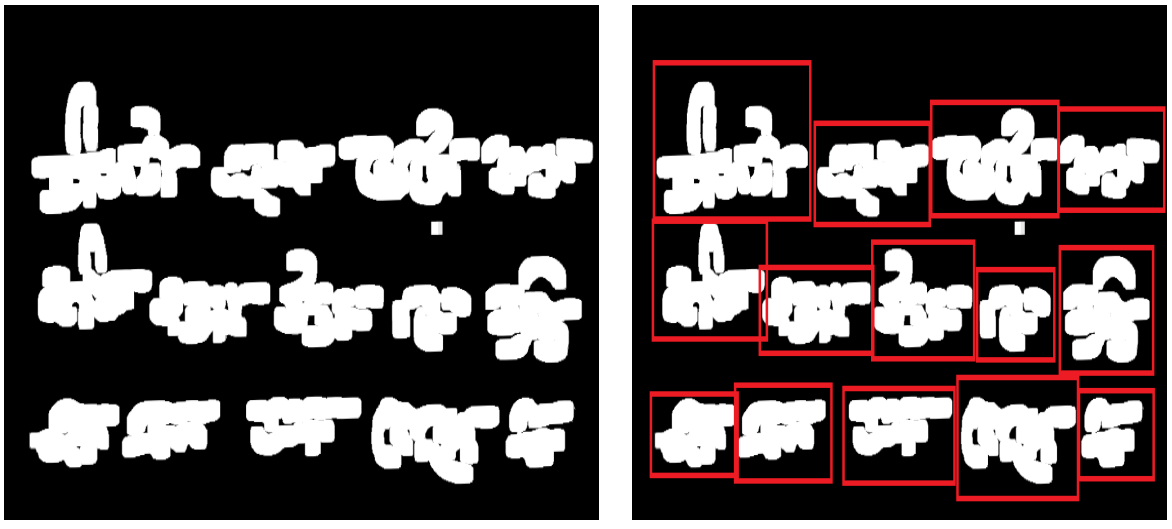
II. Word Detection using Contour Logic:

The morphological operation is used for word detection. Because characters in a word are interrelated in Bangla, a similar approach could not be employed in character segmentation.

To begin, dilation is performed, and a rectangular structural element is preferred to correspond with the size of a word. Then the connected pixels whose area is larger than a minimum threshold is considered as a contour.A bounding box wraps the contour. Thus, all the words are segmented. Figure 4.5 shows the detected words output after all the steps done. Figure-4.6 shows the initial input and the final Result after Word Segmentation.

4.2.3. Character Segmentation and Recognition

Here, the bounding boxes found in the Word Segmentation stage is collected. It is super-imposed on the images found after pre-processing.The pixels are inverted if the characters aren't black. Then the processed images are segmented. It is the input for the Character Segmentation and Recognition stage. Figure 4.7 shows the flowchart of this step. For the purpose of character segmentation and recognition from words, a custom dataset is built. Hence SSD-MobileNetV2 and Faster-RCNN-Inception-ResNetV2 are separately applied to train the dataset.



(a) Stage Before Segmentation (b) Bounding Boxes After Segmentation

Figure 4.5: Result After Word Segmentation

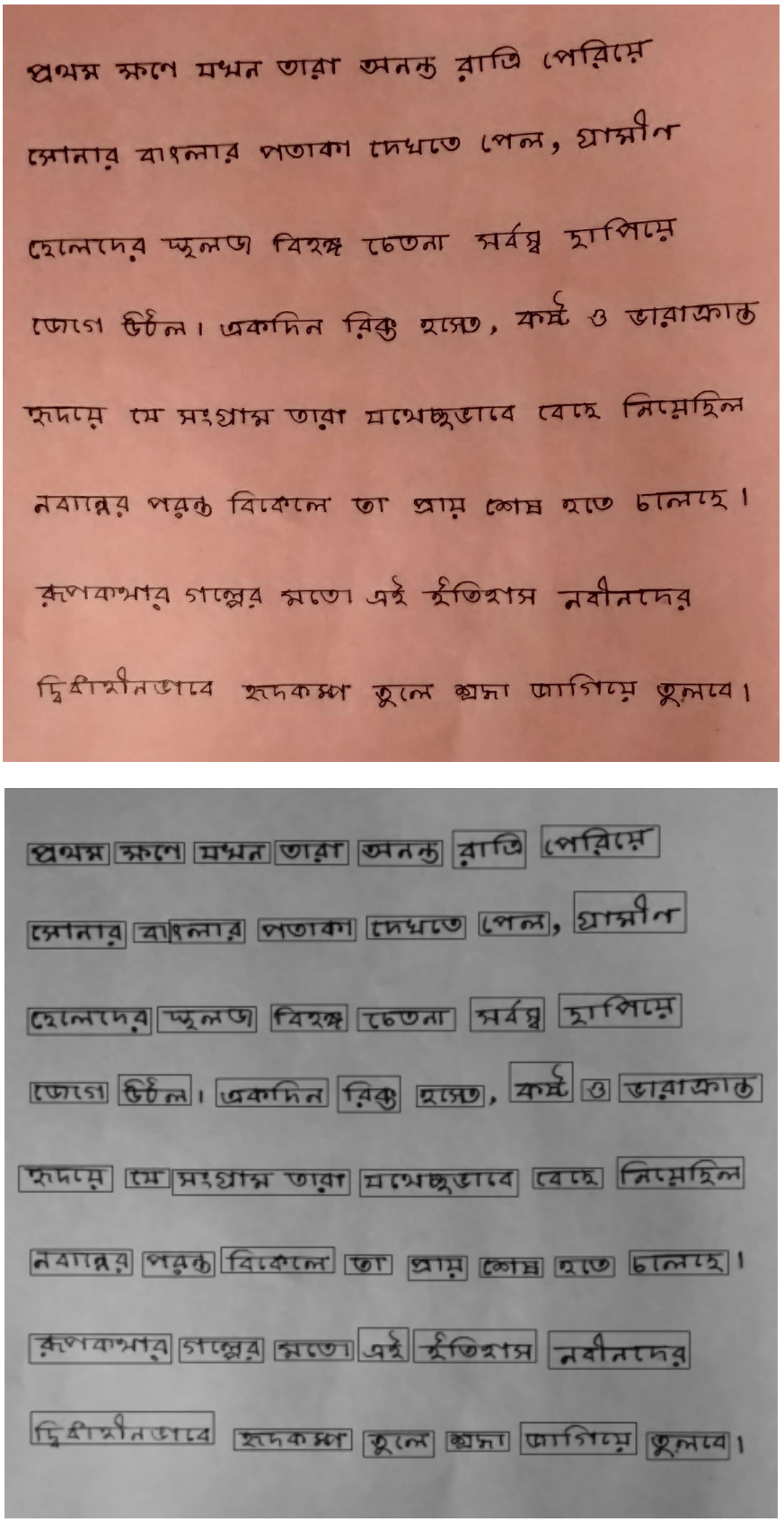


Figure 4.6: Input and Output of the Word Segmentation process.

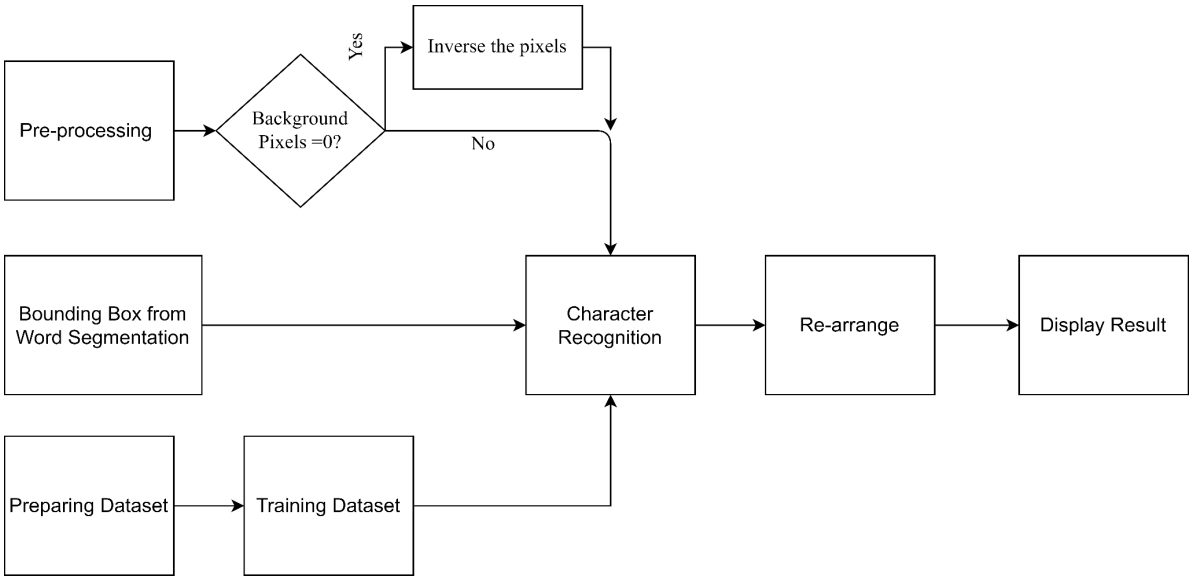
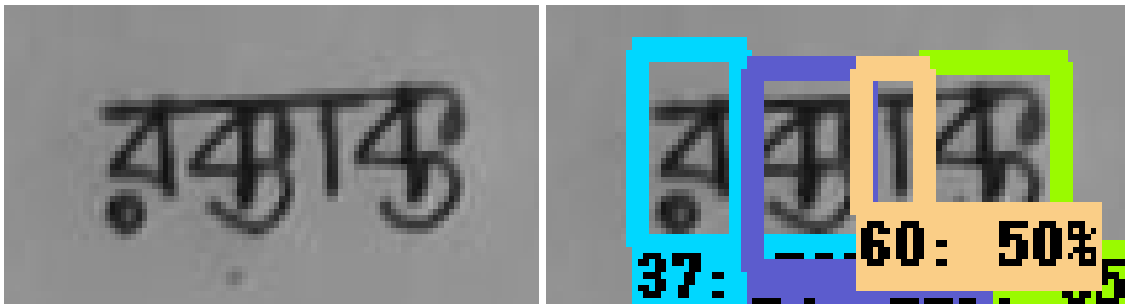


Figure 4.7: Flow-chart of the entire Character Segmentation and Recogntion process

Characters are localized and recognized after conducting models on images. Model provides bounding box and the character classes. Here in figure-4.7 , the classes of the characters and their confidence levels and their bounding boxes are shown.There could be some cases where the model could give different predictions for same character.For example, in figure 4.8, model predicted class 31,35 and 85 for the same character 'ম্প'. To eliminate the ones with the lowest confidence levels, their overlapping measurements have been measured. Each of their area and their common area have been calculated and ins1 and ins2 value have been found. ins1 = current area/common area and ins2 = previous area/common area. If either ins1 or ins2 is less than 2, that means both of the classes are strongly sharing common area. In this case, the class for which current area has been measured, was eliminated.

But, in case of Bangla characters like 'ি', 'ী', 'া', 'ে', 'ৈ', 'ো', 'ৌ' - the overlapping is inevitable as shown in figure 3.18. In that case, ins1 and ins2 value threshold will be 0.5.



(a)Input word (b) Output classes and bounding boxes

Figure 4.8: Input and Output of the Character segmentation and recognition process.

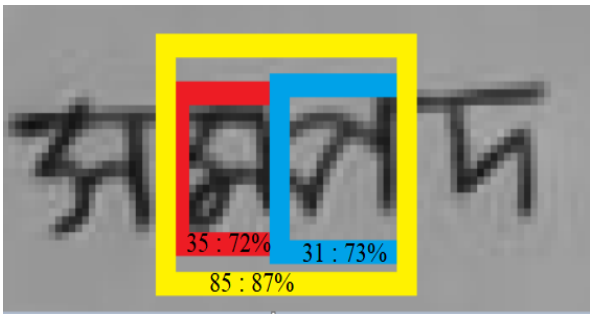


Figure 4.9 - Avoiding of several classes for same character.

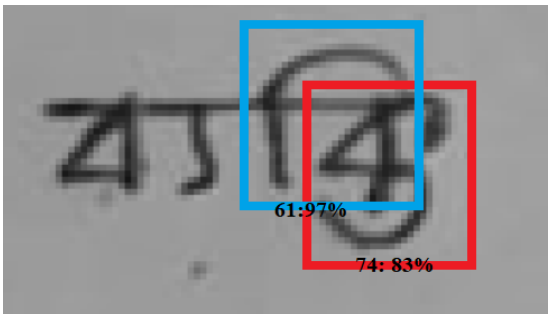


Figure4.10- Unavoidable overlapping case

4.3 Conclusion

The methodology underlying the complete proposed optical character recognition system for Bangla handwritten documents has been discussed in this chapter. Adaptive thresholding, canny edge detection, Hough transform, morphological transformation, and other image processing techniques are used in the preparation stage. Single shot detection is used for character segmentation and recognition. Convolutional neural networks are used to boost the performance even further. Some post-processing procedures could be used to improve the final outcome.

CHAPTER V

Experimental Results

5.1 Introduction

This chapter presents the findings from all of the methodologies used during the course of this thesis work. The chapter is broken down into several sections and subsections. The remainder of the chapter is structured as follows: Section 5.2 provides helpful information about the employed hardware and software for future replication. The dataset utilized is described in depth in Section 5.3. The numerous Evaluation metrics used in the segmentation and classification phase are represented in Section 5.4. The experimental findings for word detection, character recognition and segmentation are presented in Section 5.5.

5.2 Experimental Setup

The Experimental Setup section discusses useful information on the hardware used in the capture and model training process, as well as software information such as toolkit versions, OS details, and so on.

5.2.1 Hardware

Many pieces of hardware are required for the project to be completed successfully. Without the proper hardware, the software may not function properly or at all. Below is some useful information regarding the hardware utilized in the database generation and subsequent processing and detection operations.

- **Computer**
  - Processor: 2.5GHz intel Core i5 7th Gen 7200U (Turbo Boost up to 3.1GHz) with 3MB shared cache.
  - RAM: 8GB.
  - ROM: 128GB SSD.

- GPU: Integrated Intel HD Graphics 520 with 4GB GDDR5.

5.2.2 Software

Various types of software are used to make the preprocessing, training and testing steps trouble-free. Useful information regarding the used OS information, programming language information, toolkit versions, etc. are provided below.

- Operating System: Windows 10.
- Programming Language: Python 3.7.
- Application type: Python Script.
- IDE: Google Colab
- Libraries: Tensorflow 2.9.0, Cudnn 8.1, Cuda 11.2, Scikit-learn v0.20.0, OpenCV v3.4.2, Pandas v0.23.4, Numpy v1.15.3, Seaborn v0.9.0.

5.3 Dataset

The goal of this thesis is to recognize each words from the image of a text. For Word Segmentation stage, a customized dataset of 74 written documents is created. Different types of handwriting from different people have been included here to make the system more generalized.Some Sample of the dataset is shown in figure 4.1.

For the next step, both detection and classification tasks are necessary for a complete OCR. For recognizing and segmenting handwritten characters, customized dataset is used with 90 characters/labels. This dataset has 2504 images of words with multiple characters and 708 images of single characters as training images and 274 images of words with multiple words as test images.

A dataset is created for segmenting and classifying characters for proposed methodology by annotating on the images. The proposed dataset contains 90 labels of characters. Table 5.2 shows the classes with their assigned labels and number of occurrences. It can be observed that, not all classes have the same number of occurrences.

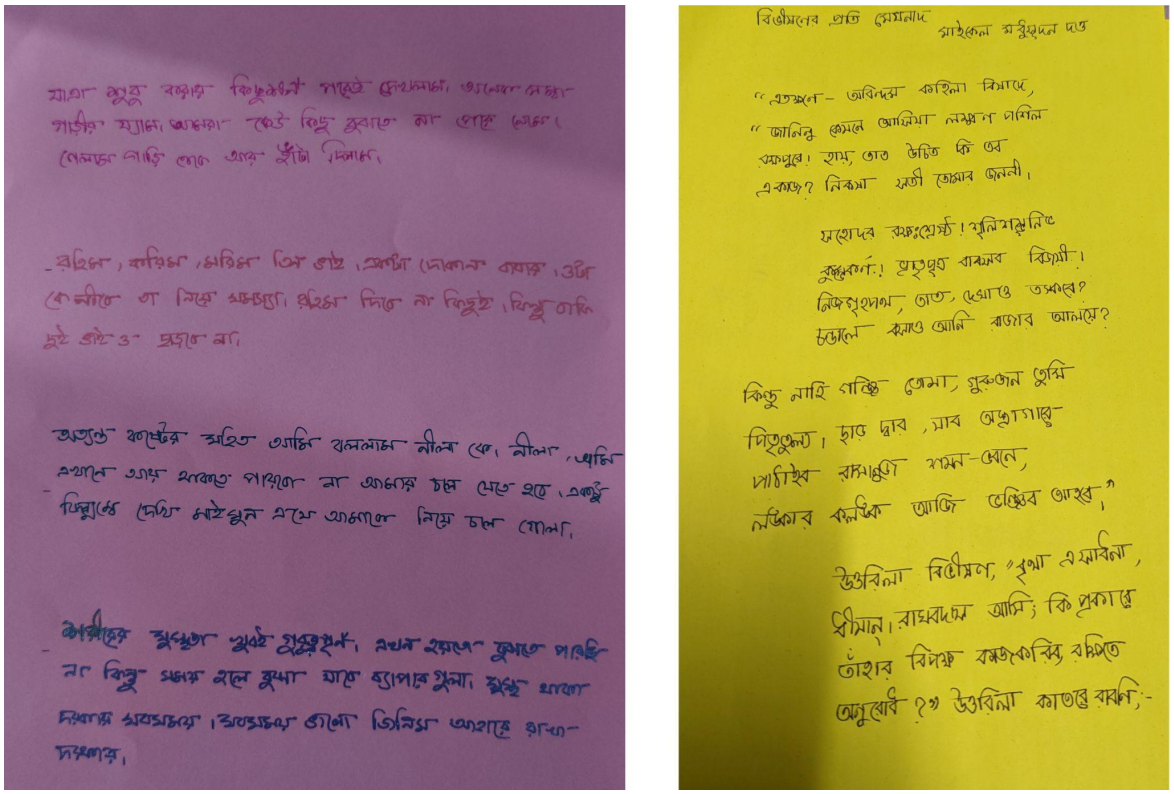


Figure 5.1: Customized dataset for written Bangla Texts.

Annotation is done on 2778 word images with these 90 labels/characters. An example of annotation is shown in figure 5.2.

5.4 Evaluation Metrics

For measuring accuracy or to find the system's detection and recognition rate, some metrics like IoU, Precision, Recall, F1-score, accuracy are used. For finding the accuracy two types of approaches have been used. One is micro-averaging and the other one is micro-averaging. They are used to evaluate model performances.

IoU: It is the ratio of the intersection of the Ground Truth and the Machine Predicted Value to the union of the Ground Truth and the Machine Predicted Value.

True Positive (TP): True Positive refers to an image object that is correctly identified (IoU>threshold) and categorized.

False Negative (FN): A false negative indicates that something that is present was not detected; something was overlooked.

False Positive (FP): False Positive occurs when an object in an image is correctly identified (IoU>threshold) but incorrectly categorised.

Table 5.1: Each class, labels and their occurrences for proposed dataset.

Class	Label	Occ.
অ	1	222
আ	2	122
ই	3	13
ঈ	4	38
উ	5	18
ঋ	6	15
এ	7	150
ঐ	8	15
ও	9	77
ঔ	10	16
ক	11	532
খ	12	117
গ	13	102
ঘ	14	27
ঙ	15	13
চ	16	78
ছ	17	101
জ	18	161
ঝ	19	23
ঞ	20	13
ট	21	102
ঠ	22	37
ড	23	45
ঢ	24	34
ণ	25	48
ত	26	376
থ	27	86
দ	28	161
ধ	29	85
ন	30	511
প	31	203

Class	Label	Occ.
ফ	32	32
ব	33	399
ভ	34	62
ম	35	301
য	36	74
র	37	514
ল	38	248
শ	39	109
ষ	40	43
স	41	243
হ	42	192
ড়	43	42
ঢ়	44	2
য়	45	207
ৎ	46	20
ঃ	47	4
ং	48	31
ঁ	49	18
০	50	21
১	51	22
২	52	23
৩	53	21
৪	54	25
৫	55	24
৬	56	25
৭	57	21
৮	58	19
৯	59	18
া	60	1236
ি	61	454
ী	62	93

Class	Label	Occ.
ে	63	940
ু	64	244
?	65	0
র-ফলা	66	71
য-ফলা	67	102
ক্ষ	68	30
ন্ত	69	26
ত্র	70	32
ঙ্গ	71	14
স্থ	72	17
ম্ব	73	13
ক্ত	74	26
স্ত্র	75	10
ন্দ	76	20
জ্ঞ	77	14
দ্ধ	78	23
ব্র	79	8
ৌ	80	4
ত	81	3
ষ্ট	82	23
ণ্ণ	83	10
ল্ল	84	13
ম্প	85	8
।	86	44
ূ	87	21
ৃ	88	22
ৈ	89	9
ো	90	101

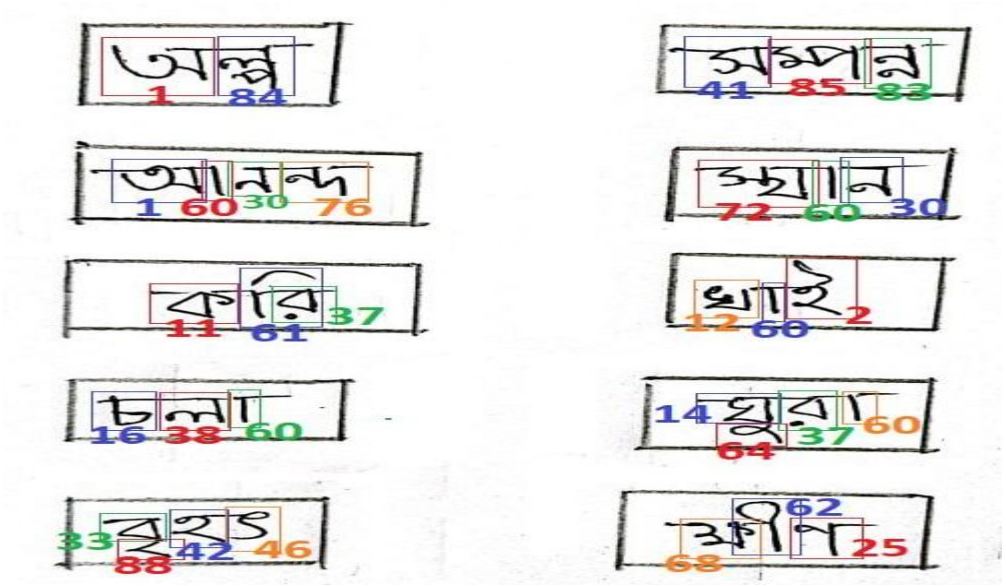


Figure 5.2: Sample annotated images of dataset.

Table 5.2: Sample confusion matrix of a multiclass classification model with 25 images.

		True/Actual		
		A	B	C
Predicted	A	4	6	3
	B	1	2	0
	C	1	2	6

Precision: Precision is defined as the number of correct results divided by the total number of correct results. With higher precision, the algorithm produces more relevant results than irrelevant ones. In the case of binary classification,

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (5.1)$$

For multiclass classification, Precision of class A is the number of correctly predicted A images (4) out of all predicted A images (4+6+3=13) which amounts 4/13= 30.8%. This means only 30.8% images predictor classifies as A are actually A.

Recall: Recall is calculated by dividing the number of right results by the number of results that should be returned. In the case of binary classification,

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (5.2)$$

For multiclass classification, Recall for class A is the number of correctly predicted A images (4) out of number of actual A photos (4+1+1=6) which amounts 4/6=66.7%. This means that the classifier classified 66.7% of the A photos as A.

F-measure: The F measure (F1 score or F score) is a test accuracy metric that is defined as the weighted harmonic mean of the test's precision and recall.

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5.3)$$

Accuracy: Accuracy is how close or far off a given set of measurements are to their true value. It is the measure that tells how much accurate the result is. It is expressed as:

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \quad (5.4)$$

For word segmentation cases, True Negative is ignored to calculate each class accuracy.

In classification based approaches precision, recall and F1 score is used to evaluate performance of a trained machine learning model. Multiclass classification formula is used to calculate these values.

5.5 Results

The results of word detection and Character Segmentation and Detection by using all the experimented methods of optical character recognition of handwritten Bangla documents are presented in results section.

4.5.1 Experimental Results of Word Detection

Total 40 customized Bangla documents images are used to detect words. Morphological operation is used to detect words from the images.

A. Qualitative Results of Word Detection

Figure 5.4 depicts two sample photos in which words are discovered using morphological operations. Cropped detected words are later used in the character recognition stage. The majority of the terms are accurately identified. Some words are undetectable. In other circumstances, a single detection box is drawn across many words.

B. Quantitative Results of Word Detection

A total of 40 images are used to test word detection from images. A total of 3979 bounding boxes are drawn on those images. Among those bounding boxes, 3681 boxes were drawn on actual words (True Positive). 296 boxes are drawn on image areas that do not contain a complete word (False Positive). 31 words in test images are not detected (False Negative).

Table 5.3: Performance evaluation of word detection from images.

	TP	FP	FN	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
Value	3681	296	31	92.55	99.16	95.74	91.84

Here, True-Negative(TN) has been neglected as this doesn't add any significant meaning to the measurements.

4.5.2 Experimental Results of Character Recognition

The purpose of the following section is to show both models' performance on character recognition. Some of the Bangla Characters look very similar to one another. Also, most of the compound characters are direct composition of two or more characters. As a result, the recognition can often be misleading. For example, in figure-5.3 ,the character '঺' and 'ট' look very similar. Also, the character 'ষ্ট','ষ' and 'ট' have very high similar looks.

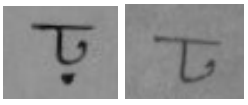


Figure 5.3: Similarity between Bangla Characters

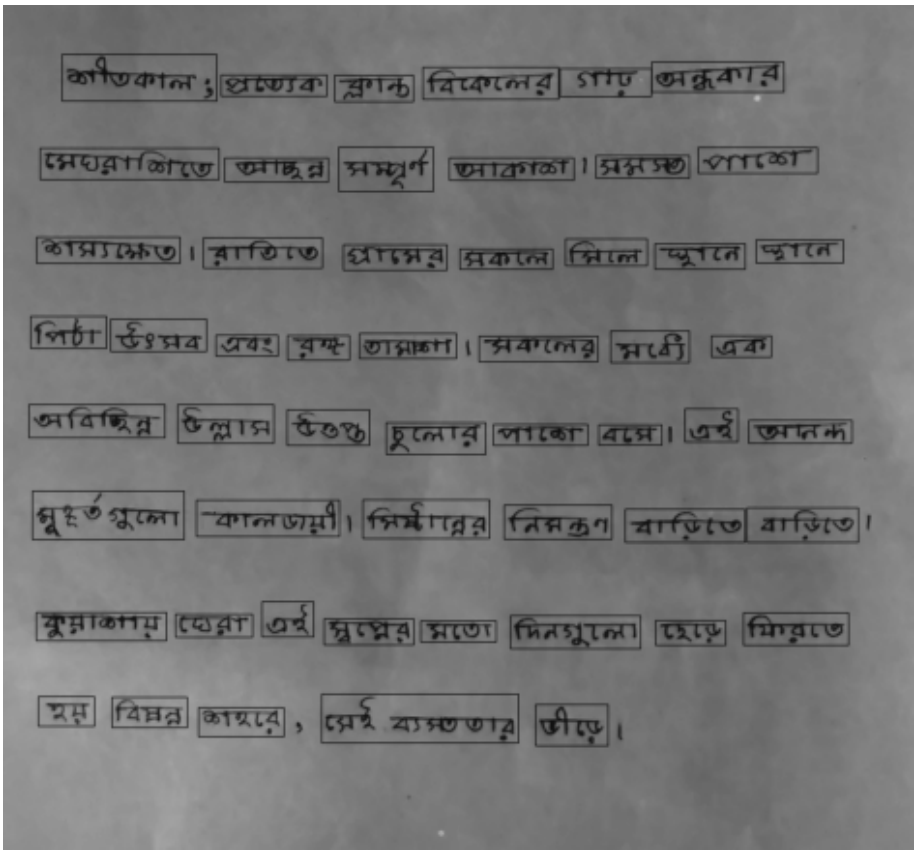


Figure 5.4: Sample images with detected words.

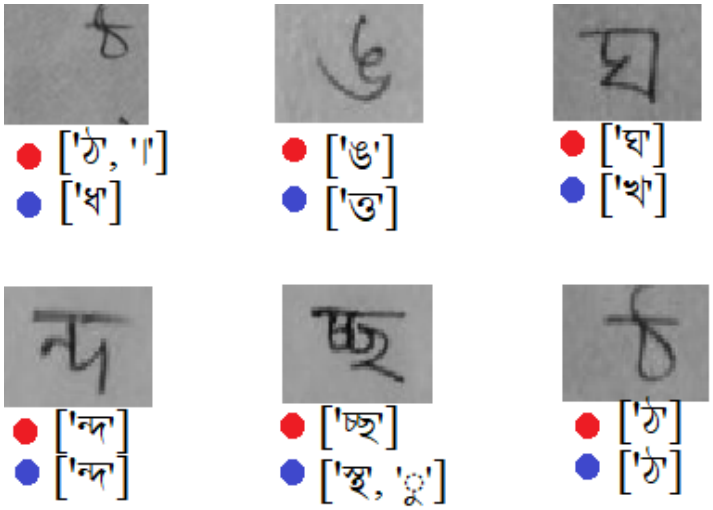


Figure 5.5: Output Result for Character Recogntion (Red- Faster-RCNN,Blue -SSD)

A. Qualitative Results and Analysis of Character Recognition:

The figure 5.5 shows the output for images with single character. The output in blue is for SSD MobileNetV2 and output in red is for Faster-RCNN with ResNetV2. As stated before, the character 'ঘ' and 'খ' has very high similarity in looking which is why SSD outputs 'ঘ' as 'খ'. But Faster-RCNN gives the correct output.

B. Quantitative Results and Analysis of Character Recognition:

A confusion matrix has been shown here to have an understanding of both model's performance. The test was done with 213 images with single characters. The class distribution of the test set with single characters is shown in figure 5.6. The column values of the confusion matrix mean the model's predicted Output and the row values means actual Output.From this matrix, the most similar looking characters can be easily detected. In that case, both the model can be trained with more training images of thos characters to increase performance.Also, the characters on which the model is performing extremely well can be seen. The diagonal values of the matrix returns the True Positive values for the characters. Any other values beside the diagonals mean False Positives meaning that the model was expected to give different output. The first column for each row defines False

Negatives meaning there was no output at all. Figure 5.7 is the confusion matrix for SSD and figure 5.8 is the confusion matrix for Faster-RCNN.

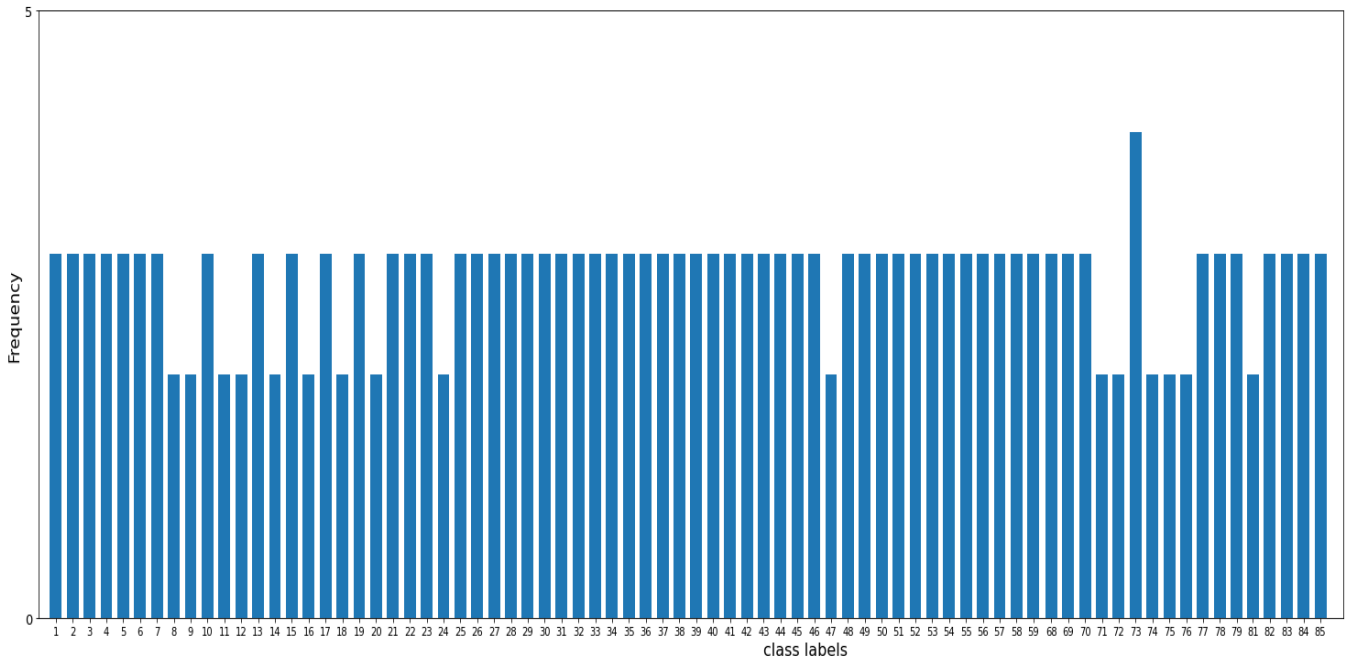


Figure 5.5: Class Distribution for test images of Single Characters

5.5.3 Experimental Results of Character Detection and Recognition

The following sections provide the quantitative and qualitative results related to character detection and recognition. For this purpose, SSD with MobileNetV2 and FasterRCNN with ResNetV2 model is performed. Various performance metrics are calculated on the test data.

A. Qualitative Results and Analysis of Character Detection and Recognition

Figure 5.8 shows some sample outputs from test data where bounding boxes and the predicted class of each bounding box for each word image are shown. From the figure, bounding boxes and Blue marked classes output are obtained initially from the character detection and classification model named SSD with MobileNetV2. Above them, Red marked classes outputs are shown which are obtained from using the Faster RCNN with ResNetV2 model.The characters that are similar to some other characters are misclassified sometimes.

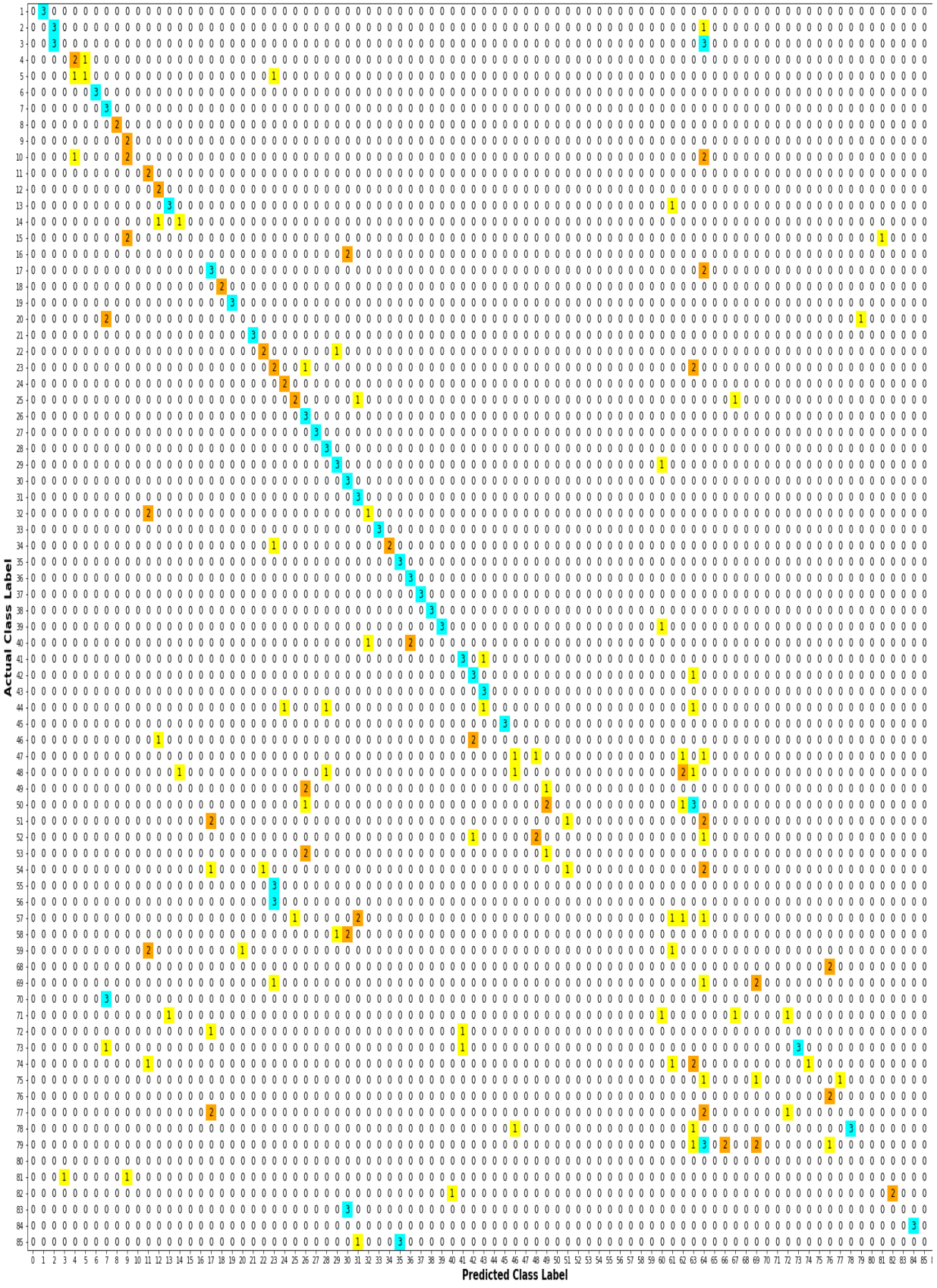


Figure 5.6: Confusion Matrix for SSD during Character Recognition

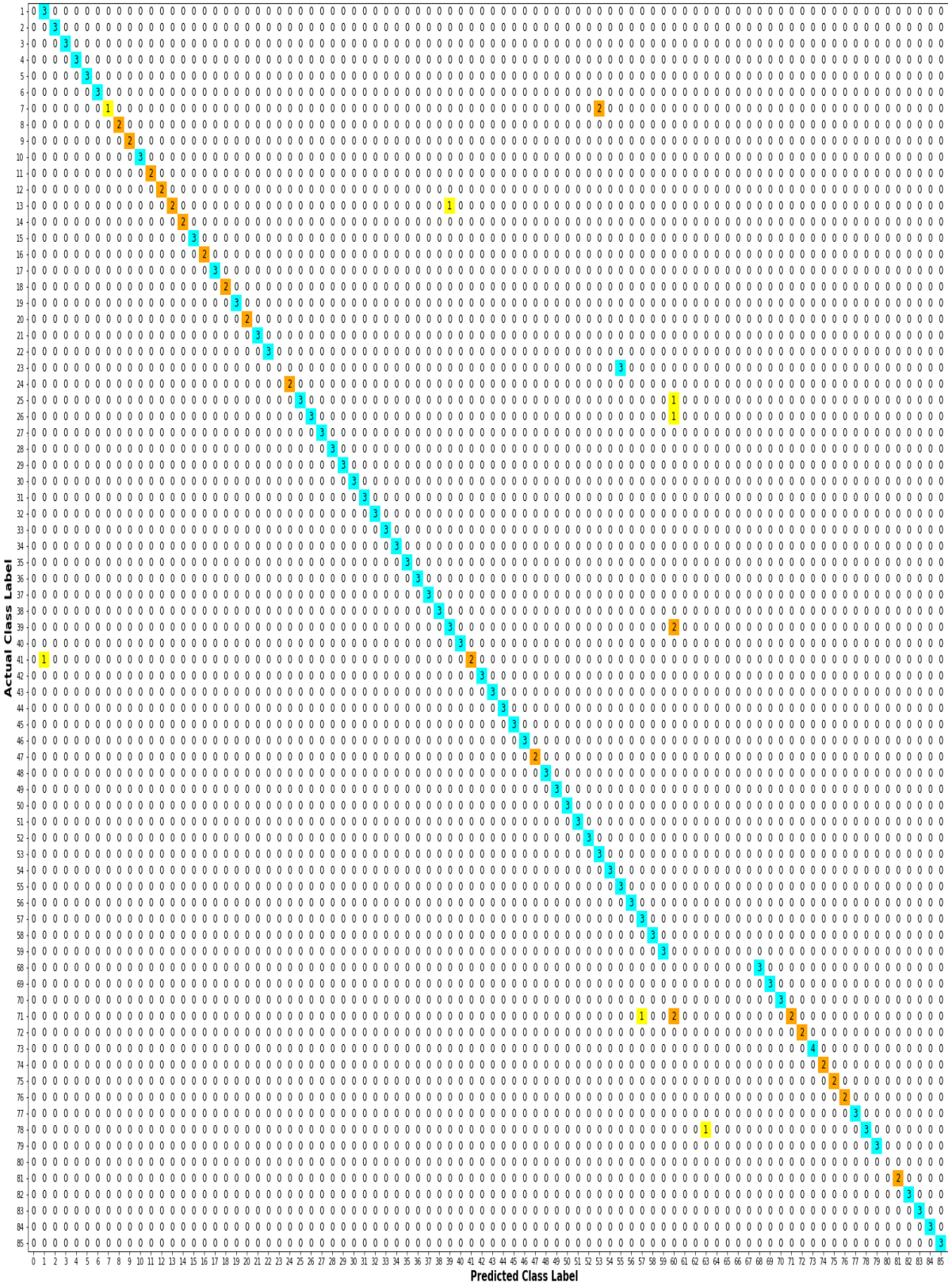


Figure 5.7: Confusion Matrix for Faster-RCNN during Character Recognition

B. Quantitative Results and Analysis of Character Detection and Recognition

Figure 5.9 shows the classss Distribution of each class for the test set. Total 274 images of multiple characters were selected as test images. Figure 5.10 shows the precision of each class for the SSD and Faster RCNN model. For a class. For a class, Precision is the ratio between the number of true predicted occurrences of that class and the number of all predicted occurrences of that class. If precision is higher for class A, it denotes that the maximum number of images the predictor classifies as A is actually A. From Figure 4.10, class 'ছ' / label '17 has high precision for the both models. Also, Class 'অ'/label '1' has good precision in both models. Average precision of SSD with MobileNetV2 model is 76.76. For Faster RCNN with ResNetV2 model, average precision is 89.82.

Figure 5.11 shows the recall of each class for the sequential model and hybrid model. For a class, Recall is the ratio between the number of true predicted occurrences of that class and the actual number of occurrences of that class. From Figure 4.11, class 'ঐ'/label '3' has very low recall value for SSD but very high value for Faster-RCNN.

Average macro recall is 78.25 for SSD with MobileNetV2 model and 92.99 for Faster RCNN with ResNetV2 model.

Figure 5.12 shows the F1-score of each class for both models. F1-score denotes the harmonic mean of precision and recall value. F1-score is an important measure because it combines both precisions and recalls to evaluate a model. Macro-F1 score for SSD is 73.61 and Faster RCNN model is 89.92.

From these each class precision, recall and F1-score value, Macro average precision,recall and F1-score are calculated for both SSD and Faster RCNN models. Faster RCNN model's overall performance is a lot better than SSD model based on the The comparison between these measurements is shown in figure 5.13.

Although Faster-RCNN is giving a much better result than SSD, SSD is relatively much faster than Faster RCNN. For real time usage, where time is an issue, SSD may be used and where Accuracy is more important, Faster RCNN can be used.

Figure 5.8: Some sample output from test data for classification model.

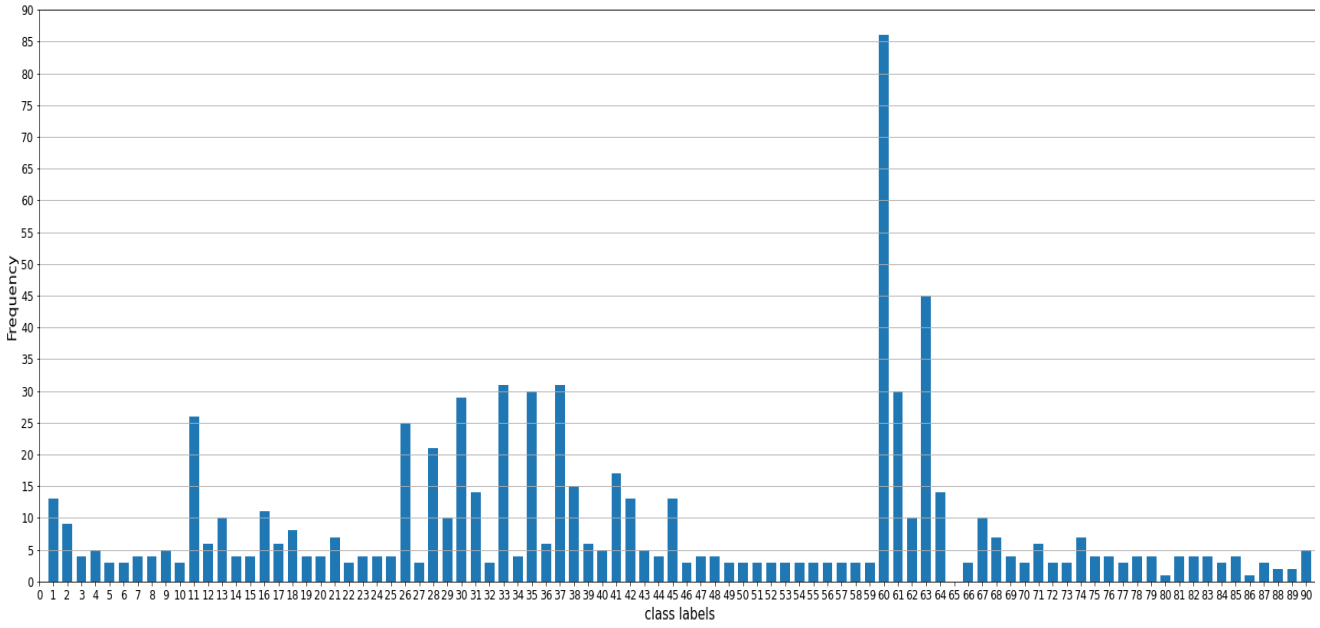


Figure 5.9: Class Distribution of Test Images with multiple characters

Precision, Recall and F1-score are found to be 76.76%, 78.25% and 73.60% respectively for SSD with MobileNetV2. For Faster RCNN with ResNetV2,Precision, Recall and F1-score are found to be 89.82%, 92.99% and 89.92% respectively.To improve the performance of the system some post-processing method of word correction can be introduced. Despite the difficult nature of the input data, experimental results suggest that this strategy is promising.

5.5.4 Performance Evaluation in Word Level

Levenshtein Distance is used to measure the performance of the model in word level on 274 test images. Here 100% accuracy means the predicted word and the annotated labels were exactly same. 75% accuracy means if there were 4 annotated labels in the actual output, model correctly predicted 3 of the labels. Table 5.2 shows the Levenshtein Distance for the test images. While 47.08% of the test images had a Levenshtein Distance of 0 for SSD, and 64.23% of the test images had a Levenshtein Distance of 0 for Faster-RCNN. The rest of the images had more than 0.

5.6 Displaying Result

The result is displayed by processing and rearranging identified characters based on their 'outcome is achieved.

4.6.1 Rearranging

Characters are rearranged based on their co-ordinate value to form a word. Sub-characters such as ‘ে’ and ‘ি’ are written ahead of their corresponding character, but they are typed after it. Again, ’আ’ was not a part of the labeled classes. Whenever ‘অ’ and ‘া’ are found together, they are replaced with ‘আ’. When reordering the characters, this type of exceptions are taken into account. Similarly, all of the words are ordered according to where they appear in the input image. After combining the words, a paragraph is revealed. Figure 5-14 shows the output for both model for same input .

5.7 Conclusion

Following the preprocessing phase, words are detected using morphological transformation. Detected words are segmented and detected using MobileNetV2 and Faster RCNN. At first, a dataset with 2426 images each having a multiple characters is trained for both of the models.

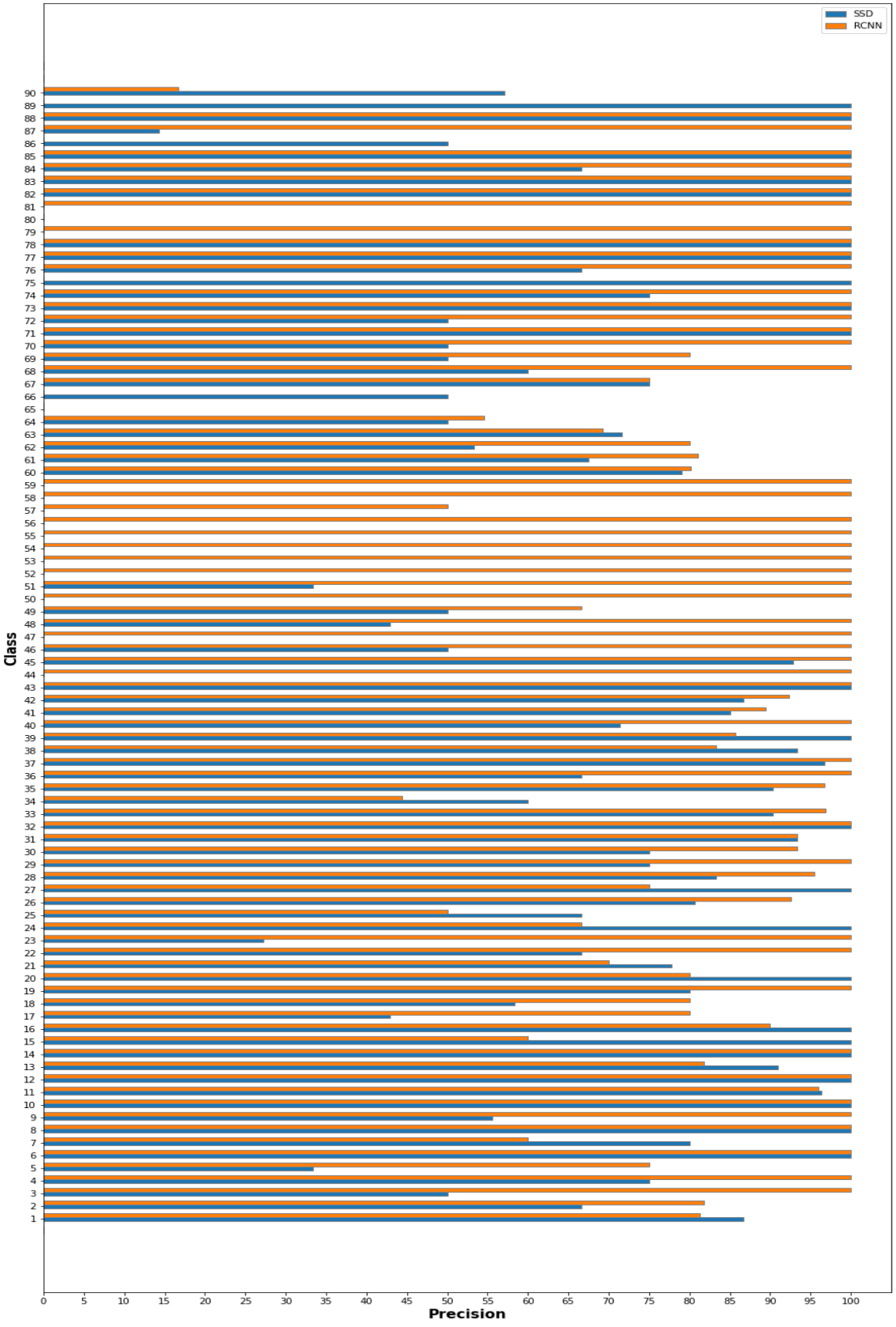


Figure 5.10: Precision of each class for SSD and Faster RCNN model.

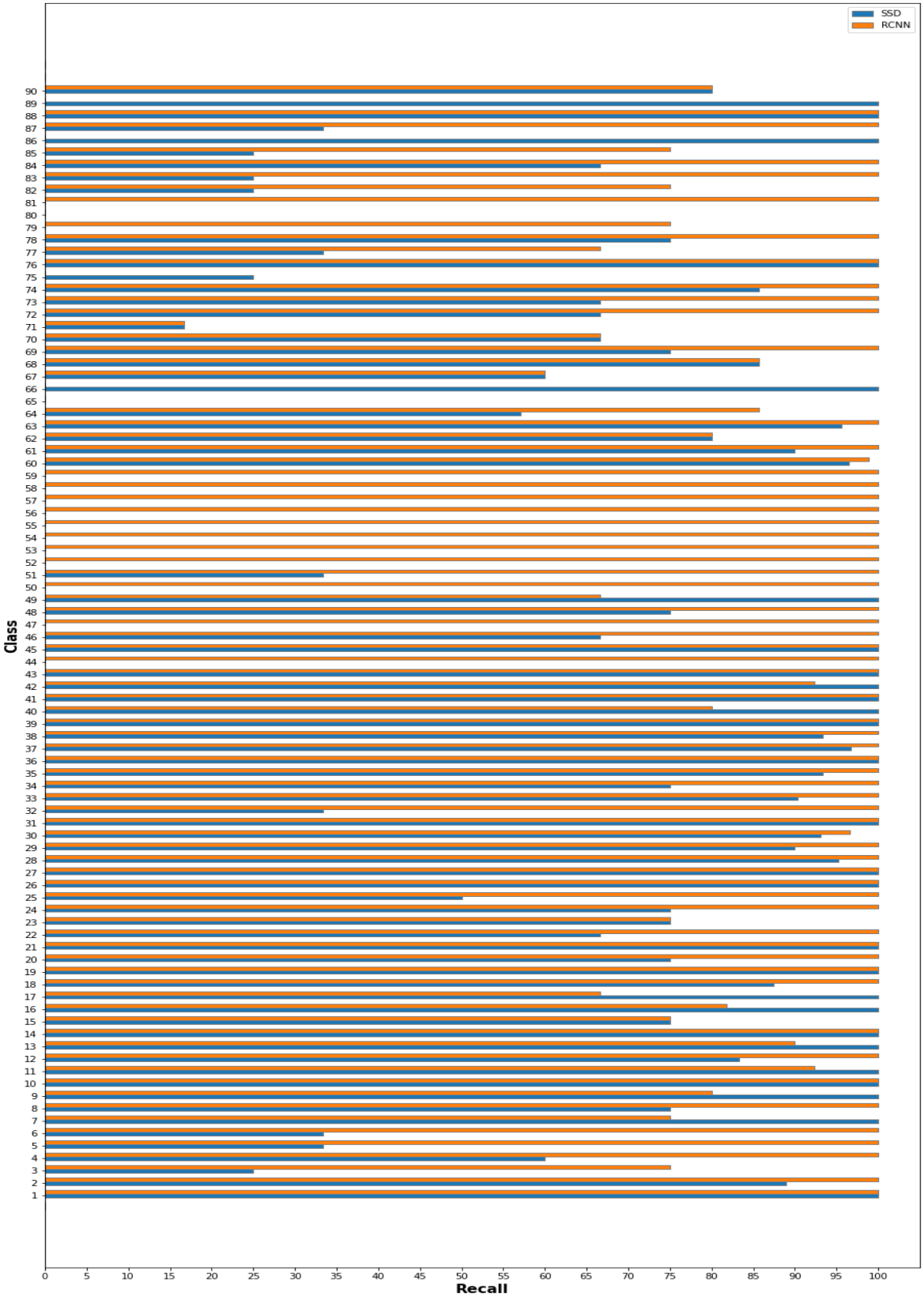


Figure 5.11: Recall of each class for SSD and Faster RCNN model.

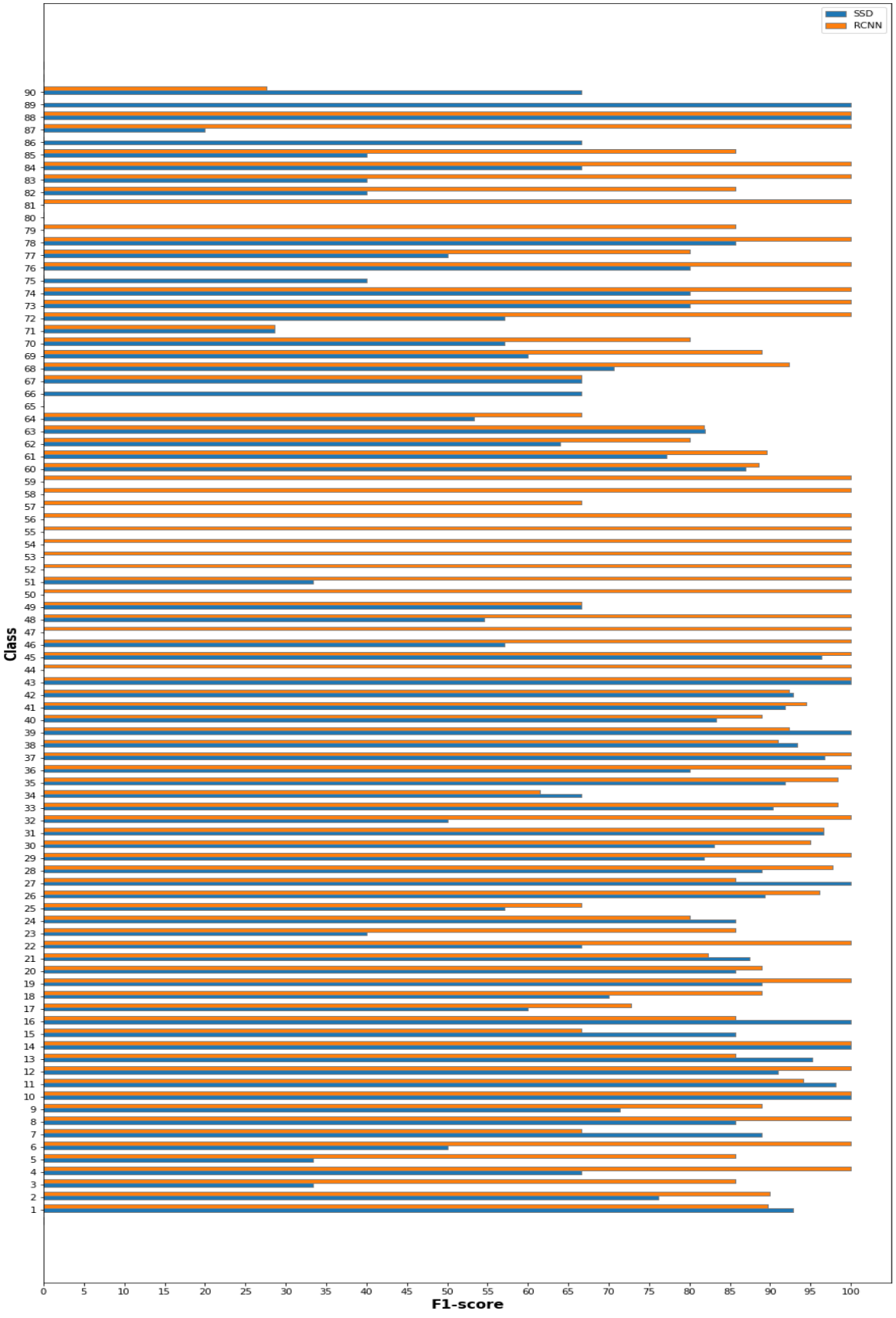


Figure 5.12: F1-score of each class for SSD and Faster RCNN model.

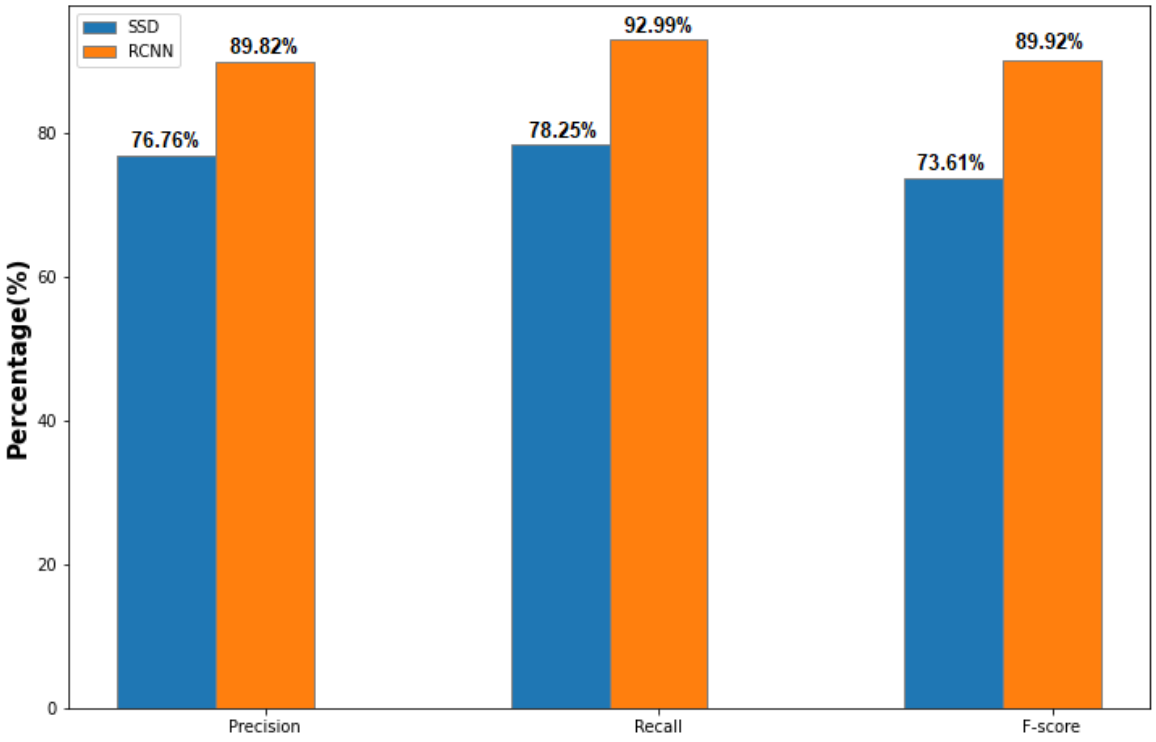


Figure 5.13: Precision, recall and F1-score of character detection and recognition.

Table 5.4: Performance evaluation in word level(Total 274 Images)

Minimum Accuracy (%)	No. of image (ssd)	Percentage of Images (ssd)	No. of image (rcnn)	Percentage of Images (rcnn)
100	129	47.08%	176	64.23%
75	30	10.95%	33	12.04%
50	44	16.06%	38	13.87%
25	13	4.74%	19	6.93%
0	58	21.17%	8	2.92%

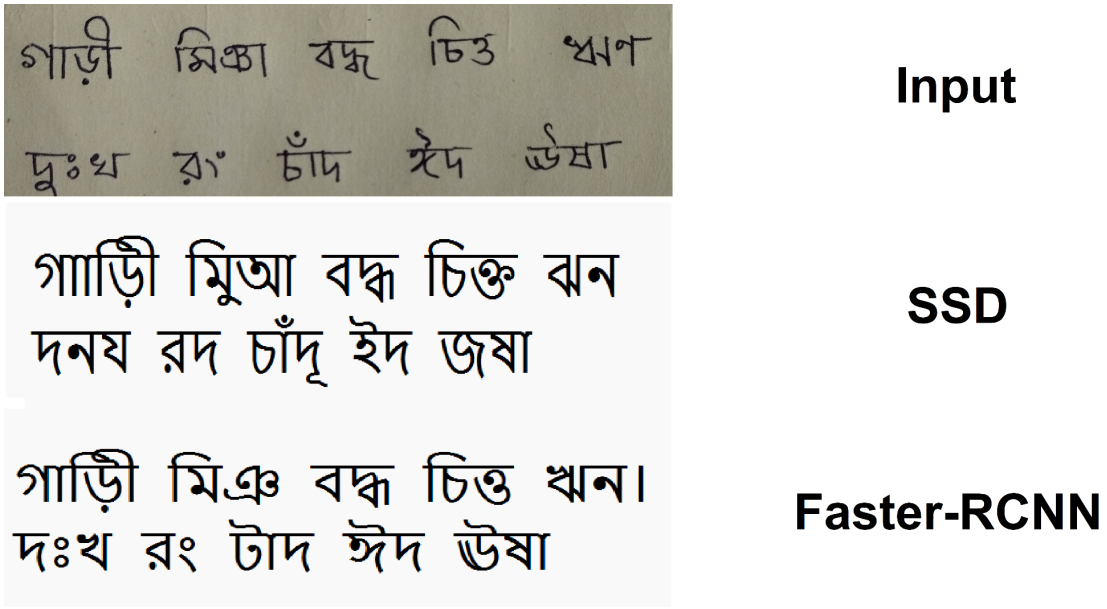


Figure 5.14: Final Output for both model for same input

CHAPTER VI

Conclusion

6.1 Summary

Optical Character Recognition systems are one of the Computer Vision applications that have transformed education, banking, and other business fields. OCR technology is widely employed in the banking, healthcare, communication, and education industries. In all government and business offices in Bangladesh, OCR for Bangla language can reduce the hidden cost and time for manual data entry. However, the OCR findings must be re-edited because they are not completely accurate. An optical character recognition system for handwritten Bangla papers has been created in this thesis. For character detection and recognition from words, a bespoke dataset was generated. Some preprocessing processes, including as the Hough transform and morphological transformation, were used to separate words from documents. As a result, the suggested system used an one shot detector and Faster RCNN to segment and classify characters from words.

6.2 Limitations

The limitations of developing OCR for handwritten Bangla documents are as follows:

- There are 50 normal characters, 10 number characters and 171 compound characters in Bangla language. It is hard to classify each character correctly because of the vast amount of characters. Although, this research included 17 of the most used compound characters, there are still vast amount of compound characters and punctuation marks are left to be included.
- In Bangla language, there is no space between characters in each word. General image processing techniques can’t segment handwritten characters correctly. Though, Deep learning based segmentation technique proposed by this thesis

overcomes this limitation. But, this system will struggle if there is hardly any gap between different lines and different words.

- For creating a generalized system, the size of the dataset plays a huge role. For developing a complete OCR for handwritten Bangla documents, the dataset size needs to be huge. Although, a custom dataset has been created, dataset that is of large size can't be made.
- This OCR performed poorly for some real world handwritten Bangla documents. As handwritten texts have much more diversity, small dataset is playing a big role here.
- In the dataset annotation step, 'ো' was identified as a distinct class. But it's actually a combination of the classes 'ে' and 'া'. Declaring the distinct class made the system's performance slightly worse. Performance can be improved by avoiding this class.

6.3 Future Work

The following tasks can be performed in the future:

- A large dataset of character segmentation and classification can be built which has diverse handwriting.
- Currently, proposed dataset has 90 labels for characters. Number of labels can also be increased by including more compound characters.
- The dataset can be trained with proposed SSD model and Faster RCNN model for detection and classification.
- Both the model's result can be ensemble to have a more accurate performance.
- R-CNN, SPP-net, YOLO and other object detection models can be used to get a comparison analysis of results.

6.4 Conclusion

An optical character recognition system for handwritten Bangla documents is developed, which takes a handwritten text image as input and turns it to editable text as output. The image of a handwritten paper is split into lines and words using the Hough Transform and Morphological Transformation. To detect and distinguish letters from word pictures, the SSD-MobilenetV2 detection model and Faster RCNN-ResNetV2 is presented. The

suggested method can segment and recognize handwritten characters from photographs. Hopefully, optical character recognition for handwritten Bangla papers can aid in the transition to a more digital Bangladesh.

REFERENCES

[1] A. Bishnu and B. Chaudhuri. “Segmentation of Bangla handwritten text into characters by recursive contour following”. In: *Proceedings of the Fifth International*
Conference on Document Analysis and Recognition. ICDAR’99 (Cat. No. PR00318).
IEEE. 1999, pp. 402–405.

[2] U. Pal, A. Belaïd, and C. Choisy. “Water reservoir based approach for touching numeral segmentation”. In: *Proceedings of Sixth International Conference on Document Analysis and Recognition*. IEEE. 2001, pp. 892–896.

[3] V. Sharda and A. C. Kishan. “Skew Detection and Correction in Scanned Document Images.” PhD thesis. 2009.

[4] E. A. Sekehravani, E. Babulak, and M. Masoodi. “Implementing canny edge detection
algorithm for noisy image”. In: Bulletin of Electrical Engineering and Informatics 9.4
(2020), pp. 1404–1410.

[5] D. Duan, M. Xie, Q. Mo, Z. Han, and Y. Wan. “An improved Hough transform for line detection”. In: *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*. Vol. 2. IEEE. 2010, pp. V2–354.

[6] Y. Lu and M. Shridhar. “Character segmentation in handwritten words—an overview”. In: *Pattern recognition* 29.1 (1996), pp. 77–96.

[7] S.-W. Lee, D.-J. Lee, and H.-S. Park. “A new methodology for gray-scale character segmentation and recognition”. In: *IEEE transactions on pattern analysis and machine intelligence* 18.10 (1996), pp. 1045–1050.

[8] N. Dave. “Segmentation methods for hand written character recognition”. In: *International journal of signal processing, image processing and pattern recognition* 8.4 (2015), pp. 155–164.

[9] J. Zhou and D. Lopresti. “Extracting text from WWW images”. In: *Proceedings of the Fourth International Conference on Document Analysis and Recognition*. Vol. 1. IEEE. 1997, pp. 248–252.

[10] L. Likforman-Sulem, A. Hanimyan, and C. Faure. “A Hough based algorithm for extracting text lines in handwritten documents”. In: *Proceedings of 3rd international conference on document analysis and recognition.* Vol. 2. IEEE. 1995, pp. 774–777.

[11] H. Wang and J. Kangas. “Character-like region verification for extracting text in scene
images”. In*: Proceedings of Sixth International Conference on Document Analysis and Recognition*. IEEE. 2001, pp. 957–962.

[12] M. Z. Alom, P. Sidike, M. Hasan, T. M. Taha, and V. K. Asari. “Handwritten Bangla character recognition using the state-of-the-art deep convolutional neural networks”. In: *Computational intelligence and neuroscience* 2018 (2018).

[13] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu, and A. Y. Ng. “Text detection and character recognition in scene images with unsupervised feature learning”. In: *2011 International Conference on Document Analysis and Recognition.* IEEE. 2011, pp. 440–445.

[14] S. Roy, N. Das, M. Kundu, and M. Nasipuri. “Handwritten isolated Bangla compound
character recognition: A new benchmark using a novel deep learning approach”. In: *Pattern Recognition Letters* 90 (2017), pp. 15–21.

[15] A. Fardous and S. Afroge. “Handwritten isolated Bangla compound character recogni-
tion”. In: *2019 International Conference on Electrical, Computer and Communication*
Engineering (ECCE). IEEE. 2019, pp. 1–5.

[16] Ø. D. Trier, A. K. Jain, and T. Taxt. “Feature extraction methods for character recognition-a survey”. In: *Pattern recognition* 29.4 (1996), pp. 641–662.

[17] A. Dutta and S. Chaudhury. “Bengali alpha-numeric character recognition using curvature features”. In: *Pattern Recognition* 26.12 (1993), pp. 1757–1770.

[18] B. Purkaystha, T. Datta, and M. S. Islam. “Bengali handwritten character recognition using deep convolutional neural network”. In: *2017 20th International conference of computer and information technology (ICCIT).* IEEE. 2017, pp. 1–5.

[19] T. Ghosh, M. M.-H.-Z. Abedin, S. M. Chowdhury, Z. Tasnim, T. Karim, S. S. Reza, S. Saika, and M. A. Yousuf. “Bangla handwritten character recognition using MobileNet V1 architecture”. In: *Bulletin of Electrical Engineering and Informatics* 9.6 (2020), pp. 2547–2554.

[20] T. Islam and R. I. Rasel. “Real-time bangla license plate recognition system using faster r-cnn and ssd: A deep learning application”. In: *2019 IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON)*. IEEE. 2019, pp. 108–111.

[21] A. Chaudhuri, K. Mandaviya, P. Badelia, and S. K. Ghosh. “Optical character recognition systems for English language”. In: *Optical Character Recognition Systems for Different Languages with Soft Computing*. Springer, 2017, pp. 85–107.

[22] M. B. Bora, D. Daimary, K. Amitab, and D. Kandar. “Handwritten character recognition from images using CNN-ECOC”. In: *Procedia Computer Science* 167 (2020), pp. 2403–2409.

[23] S. Abdullah, M. M. Hasan, and S. M. S. Islam. “YOLO-based three-stage network for Bangla license plate recognition in Dhaka metropolitan city”. In: *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*. IEEE. 2018, pp. 1–6.

[24] S. Acharya, A. K. Pant, and P. K. Gyawali. “Deep learning based large scale handwritten Devanagari character recognition”. In: *2015 9th International conference on software, knowledge, information management and applications (SKIMA)*. IEEE. 2015, pp. 1–6.

[25] T. C. Wei, U. Sheikh, and A. A.-H. Ab Rahman. “Improved optical character recognition with deep neural network”. In: *2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA)*. IEEE. 2018, pp. 245–249.

[26] K. O’Shea and R. Nash. “An introduction to convolutional neural networks”. In: *arXiv*
preprint arXiv:1511.08458 (2015).

[27] Prijono, B., 2018. Student notes: Convolutional neural networks (cnn) introduction. *Indoml. com.[Online]. Available: https://indoml. com/2018/03/07/student-notes-convolutional-neuralnetworks-cnn-introduction*.

[28] D. L. Elliott. *A better activation function for artificial neural networks.* Tech. rep. 1993.

[29] Agarap, A.F., 2018. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

[30] Seidaliyeva, U., Akhmetov, D., Ilipbayeva, L. and Matson, E.T., 2020. Real-time and accurate drone detection in a video with a static background. *Sensors*, *20*(14), p.3856.

[31] Wang, C.F., 2018. A basic introduction to separable convolutions. *Towards Data Science*, *13*.

[32] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. "Ssd: Single shot multibox detector". In: *European conference on computer vision*. Springer. 2016, pp. 21–37.

[33] D. Bradley and G. Roth. "Adaptive thresholding using the integral image". In: *Journal*
of graphics tools 12.2 (2007), pp. 13–21.

[34] W. Rong, Z. Li, W. Zhang, and L. Sun. "An improved CANNY edge detection algorithm". In: *2014 IEEE international conference on mechatronics and automation*.
IEEE. 2014, pp. 577–582.

[35] P. V. Hough. *Method and means for recognizing complex patterns*. US Patent 3,069,654.1962.

[36] Yujian, L. and Bo, L., 2007. A normalized Levenshtein distance metric. *IEEE transactions on pattern analysis and machine intelligence*, 29(6), pp.1091-1095.

[37] N. Das, K. Acharya, R. Sarkar, S. Basu, M. Kundu, and M. Nasipuri. "A benchmark image database of isolated Bangla handwritten compound characters". In: *International Journal on Document Analysis and Recognition (IJDAR)* 17.4 (2014), pp. 413–431.

[38] S. Alam, T. Reasat, R. M. Doha, and A. I. Humayun. "Numtadb-assembled Bengali handwritten digits". In: *arXiv preprint arXiv:1806.02452 (2018)*.

[39] https://github.com/pritomsaha/Context-sensitive-Bangla-spell-checker

[40] Nibaran Das,Kallol Acharya,Ram Sarkar,
"A_benchmark_image_database_of_isolated_Bangla_handwritten_compound_characters"

[41] Anthony Adole;Eran Edirisinghe;Baihua Li;Chris Bearchell; (2020). *Investigation of Faster-RCNN Inception Resnet V2 on Offline Kanji Handwriting Characters . Proceedings of the 2020 International Conference on Pattern Recognition and Intelligent Systems, (), –.* doi:10.1145/3415048.3416104

[42] Christian Szegedy,Sergey Ioffe,Vincent Vanhoucke,Alex Alemi; "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning"

[43] Shaoqing Ren,Kaiming He, Ross Girshick,Jian Sun;(2012)Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks