April 2019

Offline Bangla Handwriting Recognition with Sequential Detection of Characters/Diacritics

Nishatul Majid
Boise State University

Offline Bangla Handwriting Recognition with Sequential Detection of Characters/Diacritics

Abstract

This presents an offline handwriting recognizer for Bangla script. In spite of being a major script, very little progress has been made in this field for Bangla. Here, we present a handwriting recognition unit with sequential detection of characters/diacritics. A faster R-CNN was used to spot the graphemes from word images and the results were merged to form a transcription. Transfer learning and data augmentation techniques were applied to increase the speed and accuracy of the process. We achieved a WER and CER of 21.5% and 8.9% respectively, which is the first reported transcription result for Bangla script.

OFFLINE BANGLA HANDWRITING RECOGNITION WITH SEQUENTIAL DETECTION OF CHARACTERS/DIACRITICS

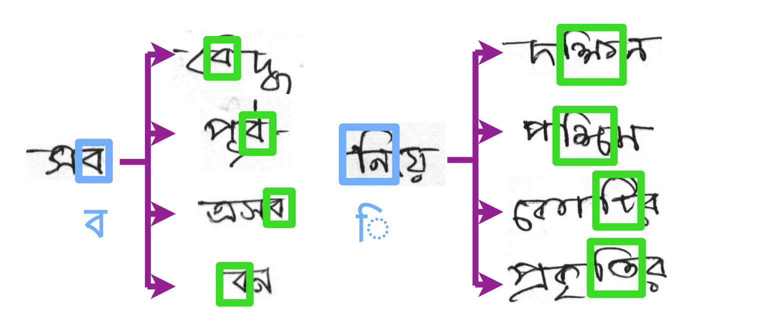
NISHATUL MAJID, ADVISOR - DR. ELISA H. BARNEY SMITH
ELECTRICAL AND COMPUTER ENGINEERING, BOISE STATE UNIVERSITY

BOISE STATE UNIVERSITY

GRADUATE STUDENT SHOWCASE, 2019

Overview

- The objective is to **recognize offline Bangla** handwritten text from images.
- We are using samples of characters/diacritics from a labeled word image to train two networks to find them in a new text sample.
- This is done using **deep learning** with a **Faster R-CNN.**
- The pre-trained **VGG-16** net was used with **transfer learning** to accelerate training.
- **Individual networks** were prepared for **character** and **diacritics** detection.

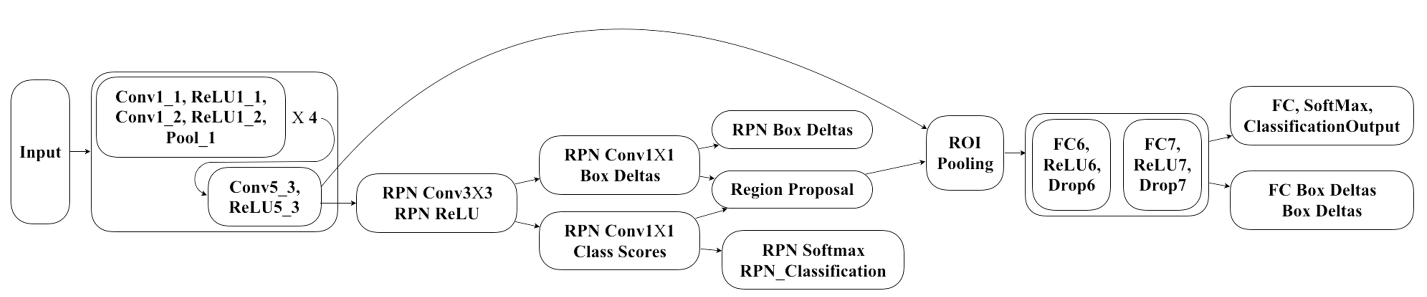


Character Detection Diacritic Detection

- A word recognition unit combined the detection results into transcription.
- This method could be applied to **other Indic scripts.**

Faster R-CNN with VGG-16

- A **Faster R-CNN** network was prepared for the character/diacritic detection.
- The network weights from the **pre-trained VGG-16** net was transferred into to a DAG (Direct Acyclic Graph) architecture.



- Two separate networks were prepared for characters and diacritics, named **C-Net** and **D-Net** respectively.
- Faster R-CNN works almost **close to real-time**, but it is slow to train.

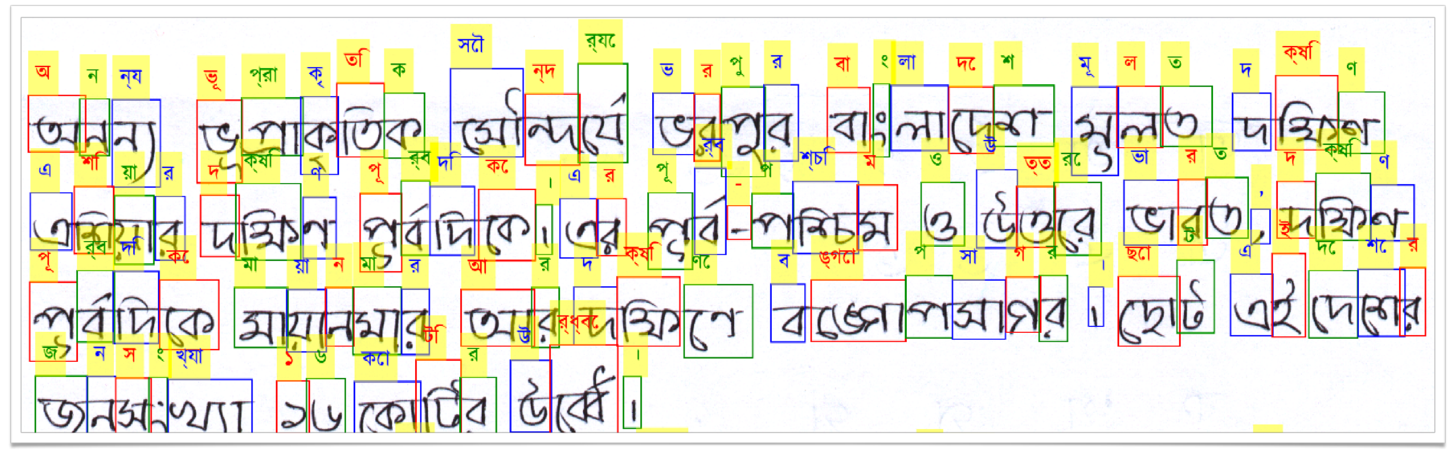
Training Parameters

- Image were **resized** to **600 pixels** at their smallest dimension during training.
- Stochastic Gradient Descent with Momentum (**SGDM**) was used for training.
- **Learning rate** was set to **0.001** with number of **maximum epochs** to **10**.
- **Overlap ratios** up-to **0.6** were used for **negative training**.
- **Number of region proposals** to randomly sample from was set to **64**.
- Increasing the number of epochs, regions or decreasing the learning rate usually makes the training process better but slower.

The BSU Bangla Dataset

The **BSU Bangla Dataset** is the only publicly available dataset that can be used for this approach. This dataset contains -

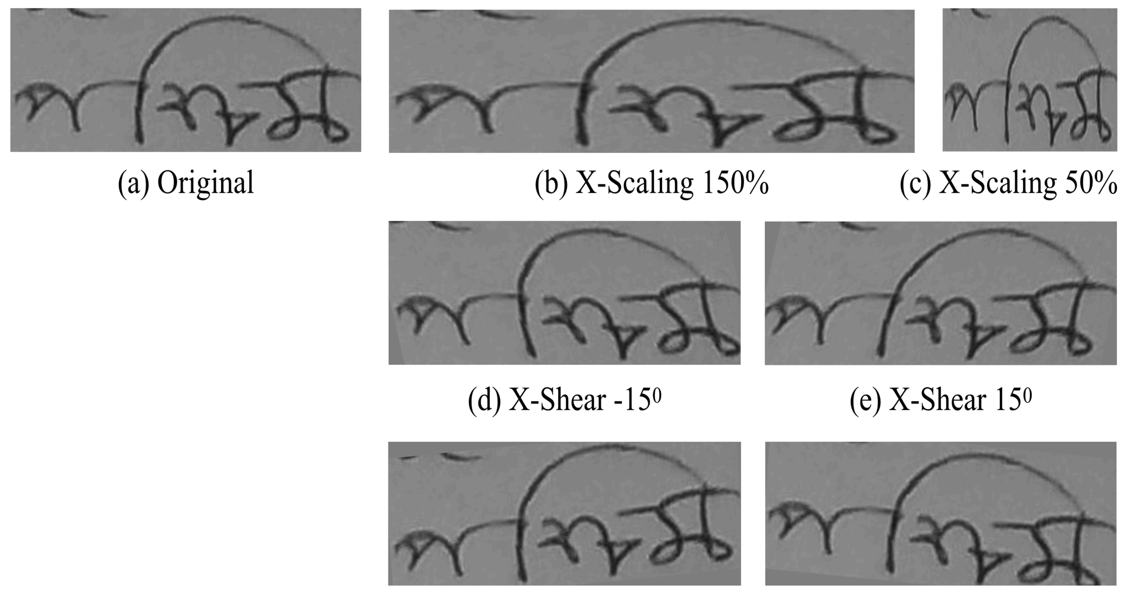
- Handwriting essay samples from more than **150 participants**.
- This essay contains **almost all basic characters, vowel diacritics and several high frequency conjuncts**.
- All the pages were **tagged with associated ground truth** for characters, words and lines.
- Of the **15,656 word images** from the dataset, **80%** were used for **training**, **10%** for **validation** and **10%** for **testing**.



Data Augmentation

Data augmentation is a widely known trick to mitigate the problem of training with smaller datasets. Here we applied a combination of three basic yet very effective techniques -

1. **Shearing** along the X-axis (between -5⁰ to 5⁰),
2. **Rotation** (between -5⁰ to 5⁰) and
3. **Scaling** along the X-axis (between 50 - 150% of the original image width)



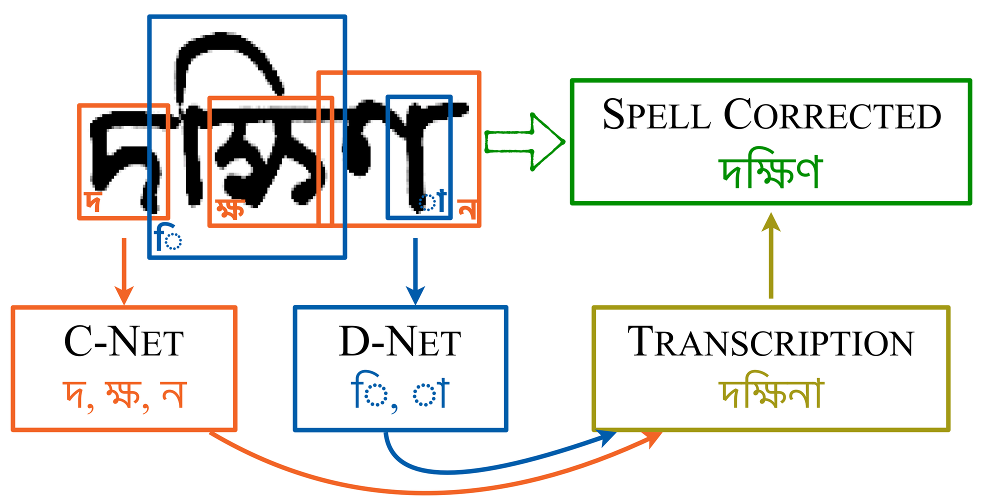
(a) Original (b) X-Scaling 150% (c) X-Scaling 50%
(d) X-Shear -15⁰ (e) X-Shear 15⁰
(f) Angle -5⁰ (g) Angle 5⁰

The amounts were drawn randomly. For each word, we generated three additional images with a combination of these three techniques, thus quadrupling the whole training set.

Word Transcription Unit

The C-Net and D-Net recognition results are combined into a word transcription unit.

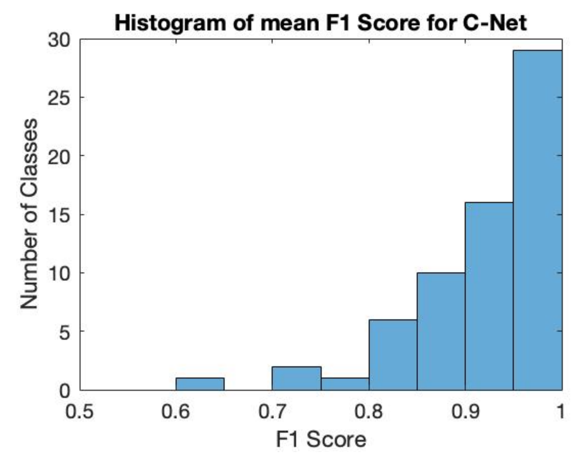
- When detections overlap, the largest bounding box exceeding the threshold was kept.
- Basic script rules were implied to eliminate false detections.
- A basic spell checker was used to further improve the accuracy.

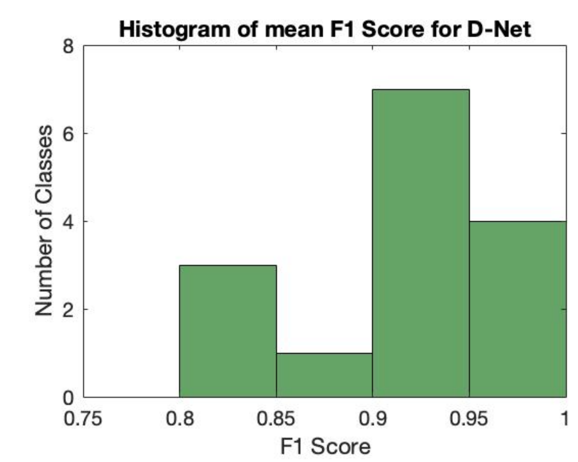


Results

Performance Scores for all Networks

Network	Performance Parameters	Score
C-Net	mAP	0.8713
	F1 Score	0.8961
D-Net	mAP	0.9034
	F1 Score	0.9317
Word Recognizer	Precision	0.8825
	Recall	0.8942
	mAP	0.8842
	F1 Score	0.8996
	WER	0.2436
	CER	0.1120
	WER (after spell check)	0.2152
	CER (after spell check)	0.0891





Conclusion

- This is the very **first transcription level work for Bangla** script.
- This approach of sequential detection of characters/diacritics can be **extended to other Abugida scripts** like Assamese, Devanagari, Gurmukhi, Gujarati etc.
- Using **more data** and increasing/improving the **augmentation** can further **improve the performance**.

- We are grateful for the high-performance computing support of the **R2 Compute Cluster** provided by **Boise State University's Research Computing Department** for this research.