Bengali.AI Handwritten Grapheme Classification

Big Data Analysis Final Project Group 4

MAY 5, 2020

Xiaosong Wen, Manyang Sun, Yueqin Liang, Yu Zhai

Table of Contents

1. Introduction and Background

Bengali is the 5th most spoken language in the world with 228 million native speakers and another 37 million as second language speakers. It is the national and official language of Bangladesh and the second most popular language in India. Automatic handwriting character recognition has many academic and commercial interests. Although extensive work has been done for English handwritings, Bangla Handwritten character recognition remains largely unsolved due to the presence of many ambiguous handwritten characters and excessively cursive Bangla handwriting. Bengali words consist of three kinds of graphemes: grapheme root, vowel diacritic, and consonant diacritics. Our project is to classify these three components of handwritten Bengali with the image of the handwritten Bengali graphemes from Kaggle (Bengali.AI, 2020).

The main challenge in handwritten character recognition is to deal with the enormous variety of handwriting styles by different writers. Furthermore, some complex handwriting scripts comprise different styles for writing words. In this project, a set of deep learning neural networks will be discussed and their performance on the application of handwritten Bengali recognition is systematically evaluated.

2. Dataset Overview

The provided training dataset contains 200,840 individual hand-written Bengali characters (graphemes), each of which is a greyscale 137-pixel by 236-pixel image. It is worth noting that Bengali’s alphabet is made up of 168 grapheme roots, 11 vowels, and 7 consonants.

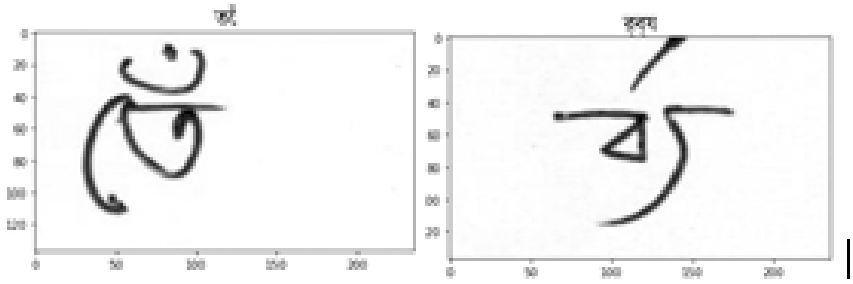


Figure 1 Two example of the Bengali characters.

Also, after sorted the number of occurrences of each component and make the plots blow, we discovered that some of the components has relatively higher distribution compared with others.

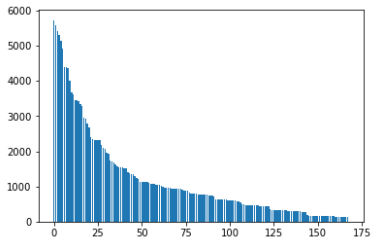


Figure 2 distributions of root

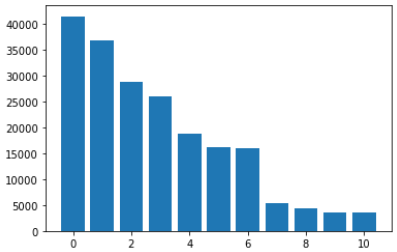


Figure 3 distributions of vowel

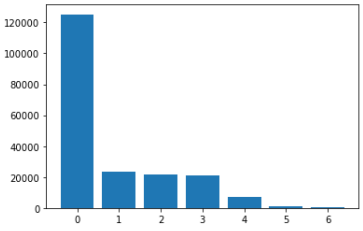


Figure 4 distributions of consonant

There are around 168*11*7 = 12,936 possible graphemes of which roughly 1,000 are represented in the training set. The test set will include some graphemes that are not included in the training set but have no new grapheme components. Moreover, the complete test set will contain the same size and number of images as the training set. All the classes of grapheme root, vowel and consonant of each Bengali character are given in the training set that act as target variables, whereas only Bengali character images are provided in the test set.

	A image_id	# graphe...	# vowel_...	# conson...	A graphe...
1	Train_0	15	9	5	ক্ট্রো
2	Train_1	159	0	0	হ
3	Train_2	22	3	5	খ্রী
4	Train_3	53	2	2	টি
5	Train_4	71	9	5	ধ্রো
6	Train_5	153	9	0	স্পো
7	Train_6	52	2	0	ঞ্জি
8	Train_7	139	3	0	ষী
9	Train_8	67	0	0	থ
10	Train_9	64	7	1	তেঁ

Figure 5 examples of training data

3. Tool and Environment

In this project, we used Python 3.6 and TensorFlow 2.1.0 on Keras API with Linux operating system.

4. Methodology

This project is a multi-task classification and detection problem. As it is related to the recognition and detection, we could possibly use CNN or GAN to solve this problem. We have done some surveys and list three solutions we may use in this project.

4.1 Multi-Output CNN

As we mentioned, all the Bengali Graphemes contain three components. So, we can first use CNN to extract the information from the images and use three different fully

connected layers to classify different components. We choose the VGG16 network among with 3 output for classification of the three components.

4.1.1 VGG16

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition” (Simonyan 2014), The architecture is showed below:

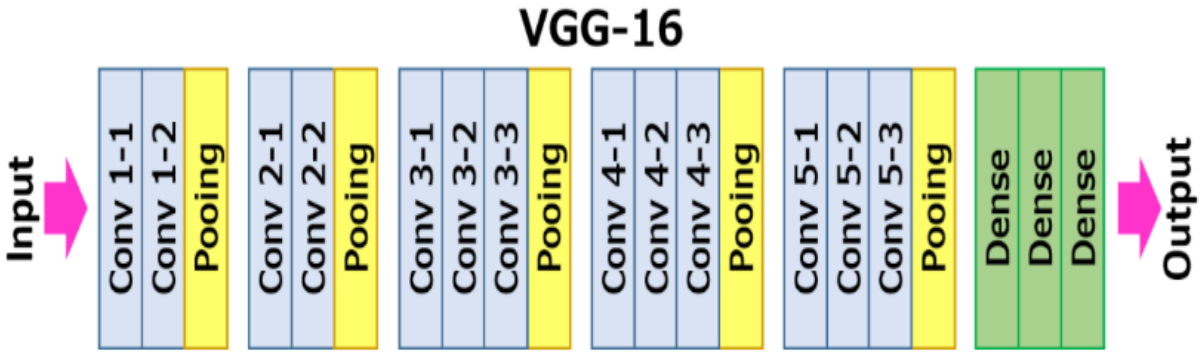


Figure 6 Architecture for VGG16

4.1.2 Network Model:

Our final model is

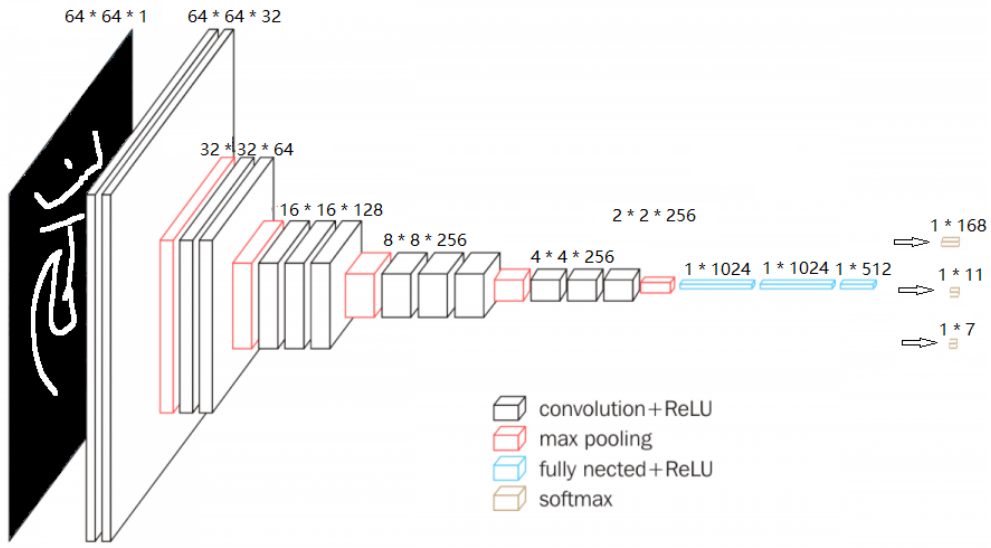


Figure 7 Multi-output CNN

4.2 EfficientNet

EfficientNet is a convolutional neural network (CNN) provided by Google in last May, which brings an accuracy improvement of up to 6% while on the order of 5–10x more efficient than most current CNN's (Mingxing Tan, 2019) .

4.2.1 Basic of EfficientNet

EfficientNet uses model scaling, including width scaling, depth scaling and resolution scaling, to improve the performance. Google gave the best coefficient for the scaling by AutoML. In this project, we are planning to use EfficientNet-b7 as our main structure.

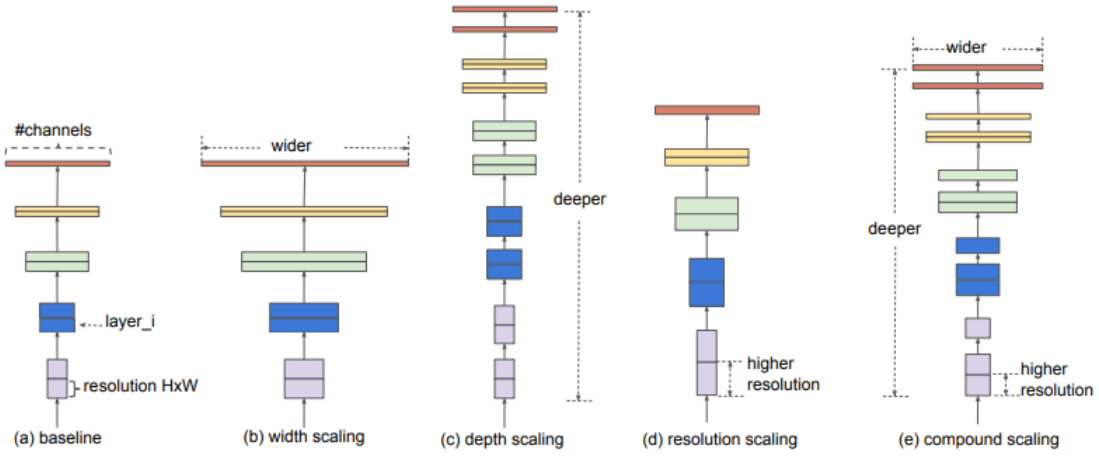


Figure 8 Scaling in EfficientNet

4.2.2 Building neural network model

The basic unit of EfficientNet is a mobile inverted bottleneck convolution layer (SANDLER 2018) (Figure 4)

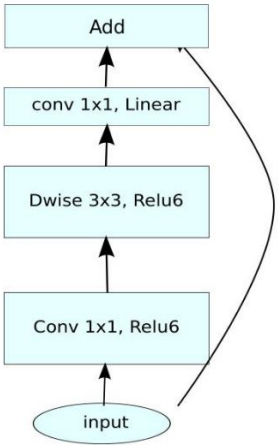


Figure 9 Structure of the EfficientNet

Standard EfficientNet model (figure 5) from with Generalized mean pool was used in the project.

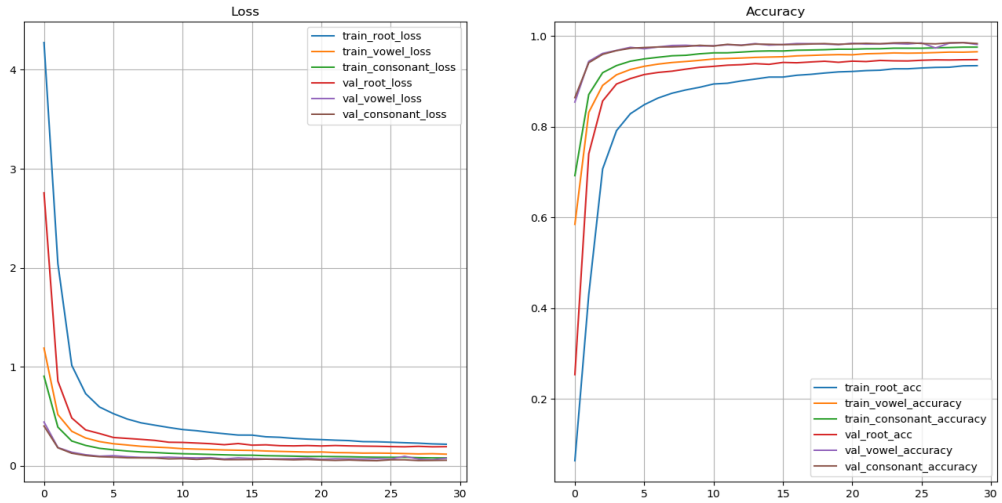
Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

Figure 10 structure of MBConv layer.

5. Result

We used ‘Categorical crossentropy’ loss function with different weight for different components to evaluate the output of the model.

5.1 Performance of Multi-Output CNN



5.2 Performance of EfficientNet

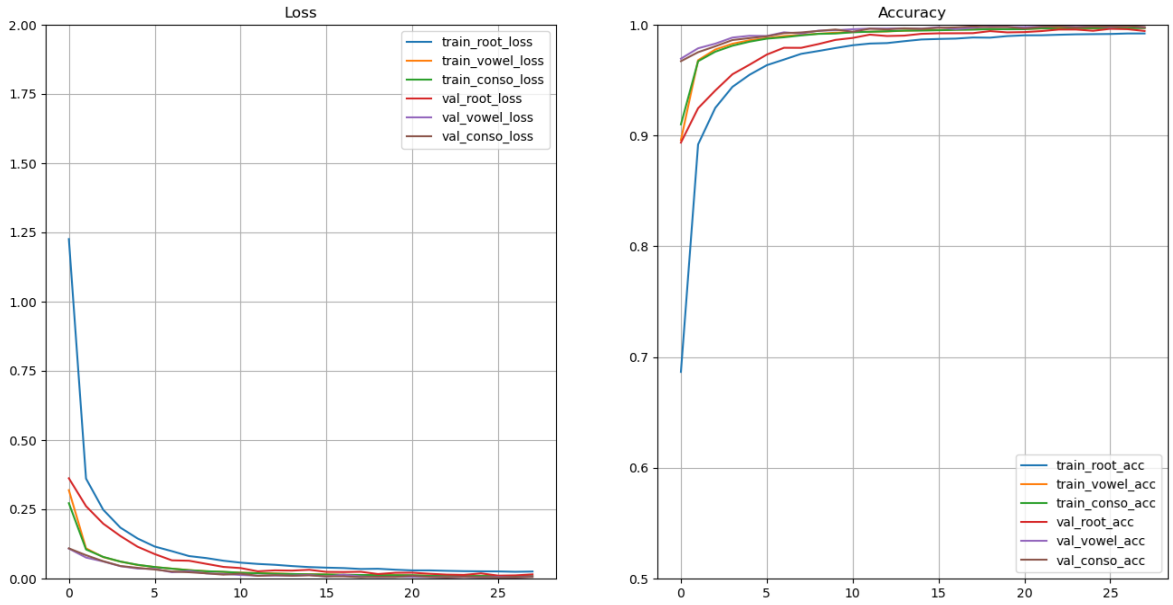


Figure 11 loss and accuracy vs epoch

6. Conclusion

Both the VGG16 and the EfficientNet achieve high accuracy for the classification of the hand-written Bengali characters. The VGG16 model achieved 97% of macro-averaged recall on the validation set. The EfficientNet model could perform 99% macro-averaged recall on the validation set. The EfficientNet model achieved both higher accuracy and better efficiency over the VGG16.

7. Reference

1. Bengali.AI. (2020, Jan). Bengali.AI Handwritten Grapheme Classification. Retrieved April 2017 from https://www.kaggle.com/c/bengaliai-cv19/overview
2. Mingxing Tan, Quoc V. Le, (2019) EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks
3. HOWARD, Andrew G., et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.
4. SANDLER, Mark, et al. Mobilenetv2: Inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018. p. 4510-4520.
5. Karen Simonyan, Andrew Zisserman. VGG: Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, 2014.