Initial Blog Posts

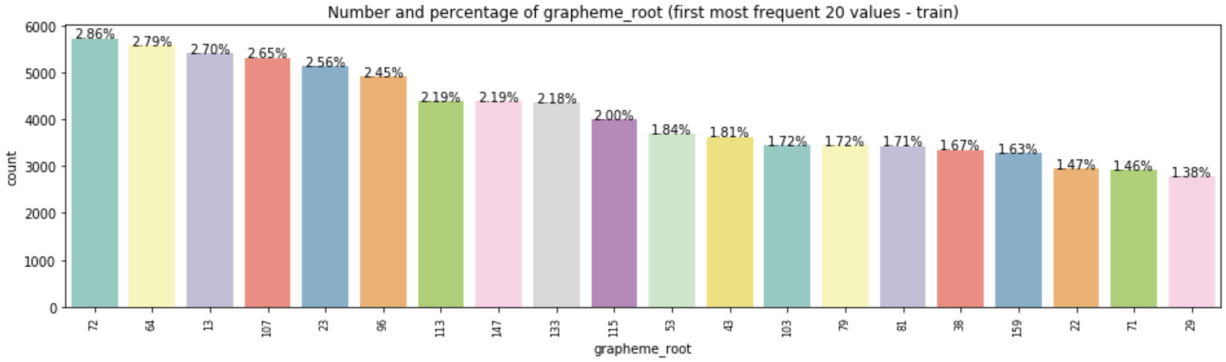
Overview:

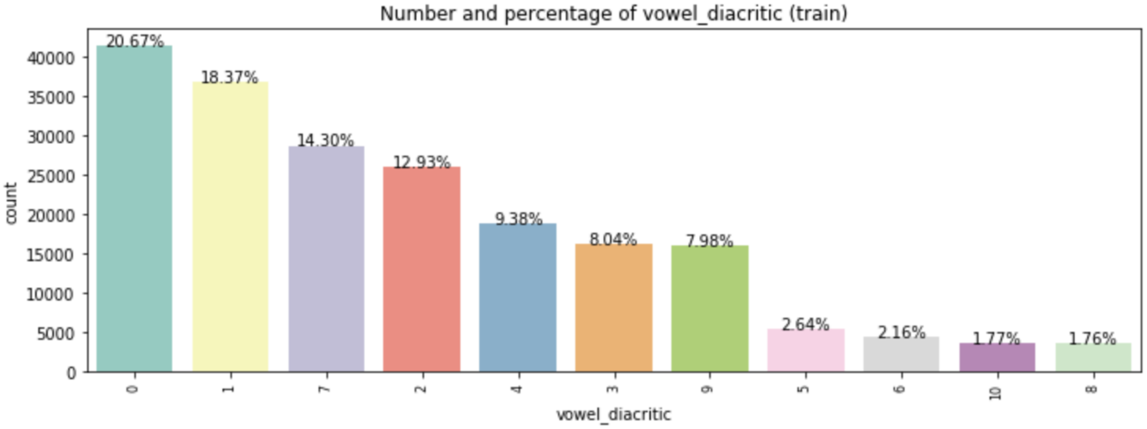
For the initial blog post, we performed EDA and built the baseline model. The accuracy scores are 88.5% for root, 94.4% for vowel and 96.0% for consonant. The major resources we used are Kaushal Shah's Kaggle notebook for modelling and Gabriel Preda's Kaggle notebook for EDA. It is worth noting that Kaushal Shah's model fitting utilizes data augmentation for creating more training data.

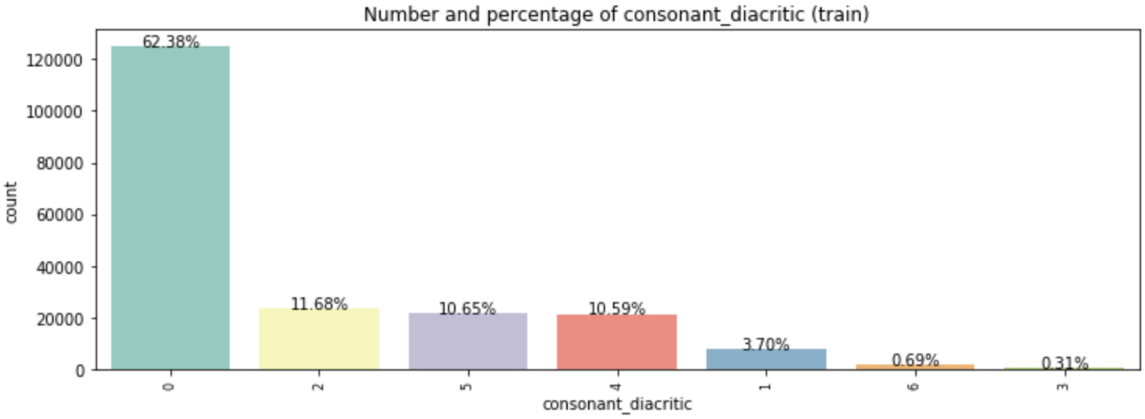
EDA:

The dataset contains in total 168 classes of Grapheme Root, 11 classes of vowel diacritics, and 7 classes of consonant diacritics. As we are given handwritten images, we aim to find the root, vowel and consonant in each image by building a Neural Network.

Below are three plots that show the distributions of class values in the dataset. And we plotted the most frequent 20 values of grapheme root and all values of vowel as well as consonants diacritics. Grapheme root and vowel diacritics are distributed in a smooth trend, but we can see a big gap in the use of consonants diacritics after class_0.





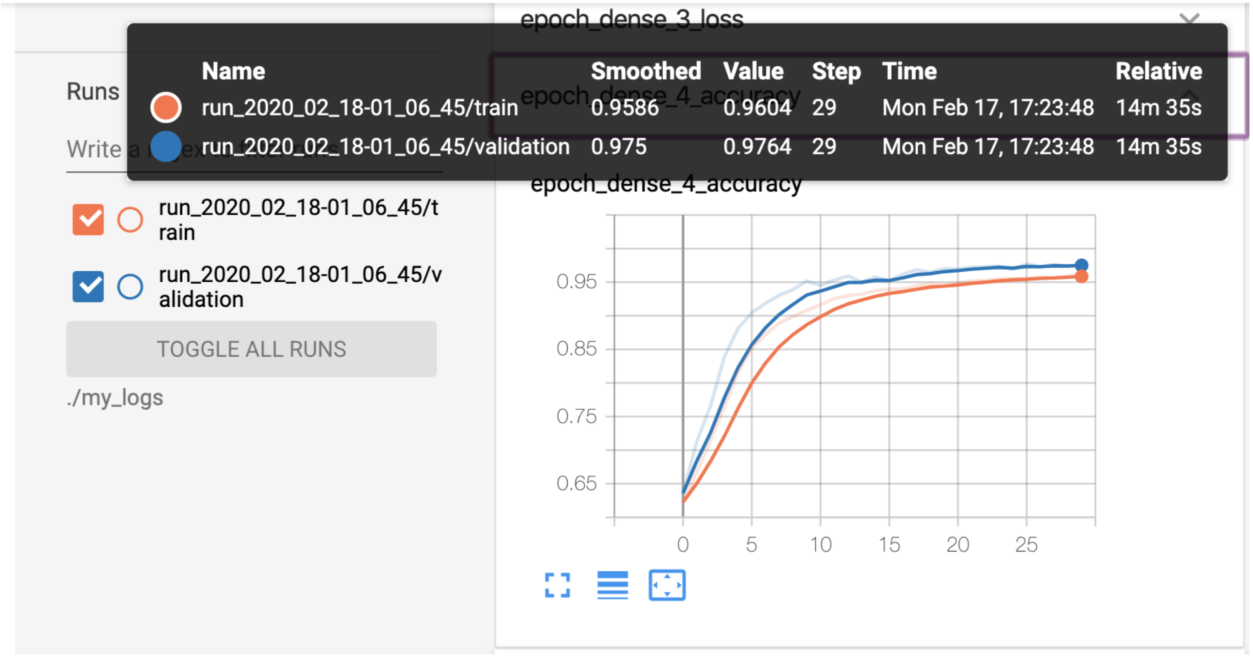


Building Baseline model:

The goal is to get the baseline model code run without errors, and hopefully get a high initial accuracy score. We adapted Kaushal Shah's Kaggle notebook for building the baseline model. We loaded and saved the data into the working space. Then we resized the images by centre cropping the region of interest and now they are 64 by 64 squares. We did not modify the learning rates at the stage, so each training file has a relatively similar accuracy score. For each of the training files, we train all the roots, vowels and consonants using the same model except for the output layers. We applied Conv2D, batch normalization and dropouts in the model. The current accuracy scores we achieved are 88.5% for root, 94.3% for vowel and 96.0% for consonant.

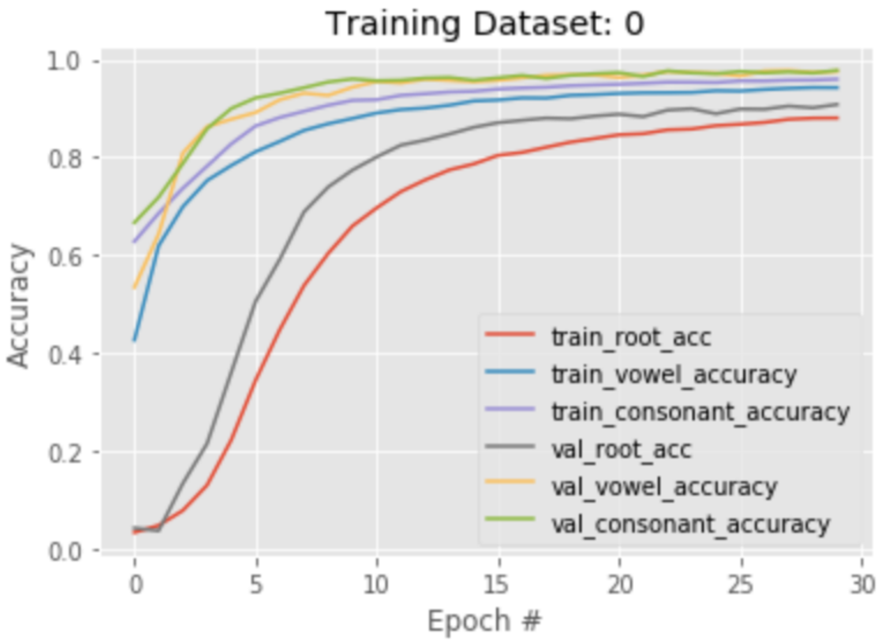
Tensorboard:

Since each training file shows similar scores, in this section, we only used the first training file as an example to visualize how the model performed. For each component of the character, we got a tensorboard that shows the accuracy and the loss. The figure below is one of the tensorboard visualizations.



	Name	Smoothed	Value	Step	Time	Relative
●	run_2020_02_18-01_06_45/train	0.9586	0.9604	29	Mon Feb 17, 17:23:48	14m 35s
●	run_2020_02_18-01_06_45/validation	0.975	0.9764	29	Mon Feb 17, 17:23:48	14m 35s

As we can see, the accuracy for epoch_dense_4, here represents consonant training is 96%, and is 97.5% for the validation. There is no overfitting or underfitting during the training process. This shows that we have reached our goal for a relatively high accuracy score, but we will still improve it later on.



The plot above shows the training histories of all three components (root, vowel, and consonant). We can see that overall the accuracy score of recognizing consonants is higher than the other two. The root group only reaches 88% and it still has large room for improvements.

Next Step:

From this point, we will:

1. Perform confusion matrix to evaluate predictions
2. Tune hyperparameters to improve the model accuracy
3. Implement data augmentation to diversify the input data and improve the model accuracy
4. Modify DL model layers to improve accuracy

Links/documentation for the team's baseline Kaggle entry:

1. **https://www.kaggle.com/kaushal2896/bengali-graphemes-starter-eda-multi-output-cnn**
2. **https://www.kaggle.com/gpreda/bengali-ai-handwritten-grapheme-getting-started**