March 2021

Lyrasis Project Update

University of Connecticut

Content by:
Dr. Joseph Johnson,
Rohan Parikh,
Joey Bechard,
Neha Ande

March Updates

Summary

The primary goal set by the UConn Lyrasis researchers over the past several weeks has been to increase data collection and begin the initial phases of text recognition on journal page characters. Using a data collection tool built by Joey, we have taken a combined amount of over 25,000 individual screenshots of letters throughout Volumes 40 to 43 of the initial image set. These screenshots are automatically categorized into groups by letter type allowing us to begin developing prediction models. Keras and Jupyter notebooks have been implemented to develop Convolutional Neural Networks (CNNs) on 7 base letters (a, e, n, s, t, i , l).

After various iterations and layer implementations, a prediction accuracy on the test dataset has ranged between 85% and 95%. As we continue to increase the size of the dataset, this accuracy is expected to increase. Furthermore, other methods such as K-Fold Validation are being implemented to artificially introduce variation in the data for increased accuracy.

Predictive Model Development

Three initial models have been created by the UConn Lyrasis researchers to make predictions on 5 to 7 letters for which the most screenshots have been collected.

Model 1

The first model consists of 3 convolutional layers with depths of 32, 64, and 64 respectively. Each convolutional layer is followed by a relu activation layer and maxpooling filters. In the final convolution, the data is flattened, parsed through a 128 deep Dense layer and outputted through a 7 deep Dense layer for the 7 classification classes (a, e, n, s, t, i , l).

After training on these 7 letters , this model implemented Stratified K-Fold Validation (10=folds) on the validation dataset for an average validation accuracy of ~85%. The letter dataset has a minimal number of screenshots of all the other letters due to their lower frequency of being used. However, running this model on all 26 letters (lowercase) with the same stratification gave an average validation accuracy of 77.17%.

```
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu',
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.Flatten())
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dense(7, activation='softmax'))
```

Predictive Model Development

Three initial models have been created by the UConn Lyrasis researchers to make predictions on 5 to 7 letters for which the most screenshots have been collected.

Model 2

The second model consists of 4 convolutional layers with depths of 32, 32, 64, and 64 respectively. Each convolutional layer is followed by a relu activation layer and maxpooling filters. In the final convolution, the data is flattened with a dropout of 0.5, then parsed through a 64 deep Dense layer and outputted through a 5 deep Dense layer for the 5 classification classes (a, e, n, s, t). With an initial test on the validation data, it has an accuracy of ~86% across the 5 letters chosen.

```
model = Sequential()
model.add(Conv2D(32, (3, 3), input_shape=input_shape))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(64))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(5))
model.add(Activation('softmax'))
```

Predictive Model Development

Three initial models have been created by the UConn Lyrasis researchers to make predictions on 5 to 7 letters for which the most screenshots have been collected.

Model 3

The final model consists of 3 convolutional layers with depths of 256, 512, and 1028 respectively. Each convolutional layer is followed by a relu activation layer and maxpooling filters. In the final convolution, the data is flattened with a dropout of 0.5, then parsed through a 64 deep Dense layer and outputted through a 5 deep Dense layer for the 5 classification classes (a, e, n, s, t). With an initial test on the validation data, it has an accuracy of ~95% across the 5 letters chosen.

```
model = Sequential()
model.add(Conv2D(256, (3, 3), input_shape=input_shape, kernel_regularizer=l2(0.0005), padding="valid"))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(BatchNormalization(axis=-1))

model.add(Conv2D(512, (3, 3), kernel_regularizer=l2(0.0005), padding="same"))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(BatchNormalization(axis=-1))

model.add(Conv2D(1024, (3, 3), kernel_regularizer=l2(0.0005), padding="same"))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(BatchNormalization(axis=-1))

model.add(Flatten())
model.add(Dense(64))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(BatchNormalization())
model.add(Dense(5))
model.add(Activation('softmax'))
```

Predictive Model Development

Three initial models have been created by the UConn Lyrasis researchers to make predictions on 5 to 7 letters for which the most screenshots have been collected.

Comparison Table

Model #	Layers	Accuracy	K-Fold Validation
1 (7 Letters)	3 (32,64,64)	85%	Yes
1 (26 Letters)	3 (32,64,64)	77.17%	Yes
2 (5 Letters)	4 (32, 32, 64, 64)	86%	No
3 (5 Letters)	3 (256, 512, 1024)	95%	No

Next Steps

1. We will continue to grow the 26 letter dataset to approximately 50,000 characters, lowercase and uppercase combined over the next months.

2. Implementing a feature to save full lines of text and stringing characters together to save various sentences. This will be used to create CV models that can determine which letters are present in a given sentence (based on the classification model).

3. Improving and retraining the current CNN models to improve accuracy on all 26 letters and bring the average closer to 99%.