PRACTICE WRITING YOUR LETTERS

ABSTRACT

As today's kids and adults move into the digital world, they will have to start writing on the computer with a mouse or finger. With this application, I hope to help them write their letters easier.

Kimberly Cable

Belleview University, DSC 680 – Applied Data Science February 2022

Practice Writing Your Letters

Topic

Learning to write your letters with a mouse is hard. This application aims to help those who would like help writing the alphabet with their mouse.

Business Problem

As today's kids and adults move into the digital world, they will ultimately have to start writing with a mouse or finger. With this application, I hope to help kids and adults, write with their mouse and/or finger.

Data Explanation

Number of Classes: 52
Number of Data Points: 1300
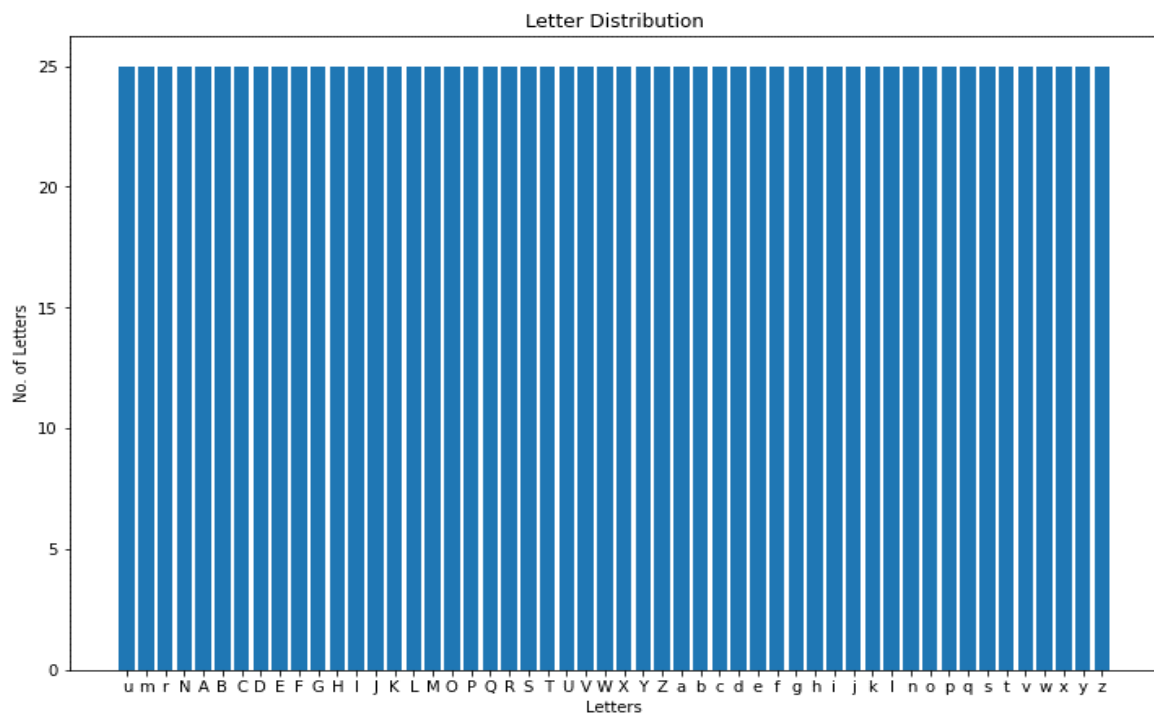
Distribution of Letters in Training Data

The data will be created from the application. The application will have a training button that will allow the user to write all the letters in the English alphabet to help establish a good training dataset for the model to use. Currently, there are 52 letters, 26 uppercase, and 26 lowercase. Each letter is in the training dataset 25 times. Below is an example of some letters in the training dataset.

Sample of the Training dataset
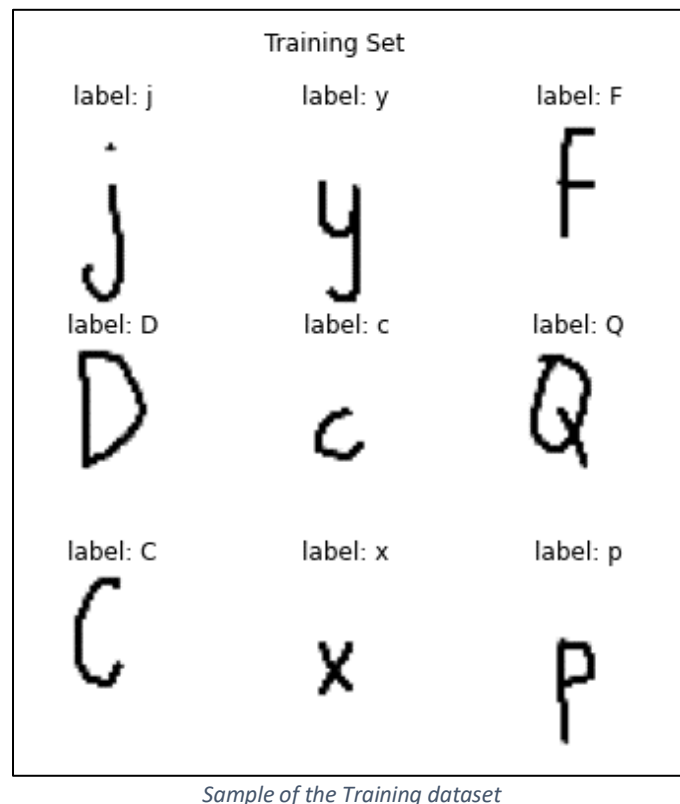
Methods

A deep learning method using TensorFlow's Keras Sequential class. Letter images from the application will be used as the training dataset for the model. The model will use three layers along with data augmentation and dropout layers to prevent the training data from overfitting. Once the model is trained, the user can then use the application to practice their letters. The application will prompt the user to write a specific letter (upper and lower case), then the application will let the user know if they wrote the correct letter properly.

Analysis

The model was created using the Sequential class from TensorFlow's Keras. Six layers were used plus some dropout layers to avoid overfitting the data (where the model only works for the training data and not any new data).

```
Model: "sequential"
_____
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 48, 48, 32)        320

 max_pooling2d (MaxPooling2D  (None, 24, 24, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 22, 22, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 11, 11, 64)       0
 2D)

 dropout (Dropout)           (None, 11, 11, 64)        0

 conv2d_2 (Conv2D)           (None, 9, 9, 64)          36928

 max_pooling2d_2 (MaxPooling  (None, 4, 4, 64)         0
 2D)

 dropout_1 (Dropout)         (None, 4, 4, 64)          0

 flatten (Flatten)           (None, 1024)              0

 dense (Dense)               (None, 53)                54325

=================================================================
Total params: 110,069
Trainable params: 110,069
Non-trainable params: 0
_____
```

Data augmentation was also used to help generate more training data as the application only uses training data that the user has provided. The augmented images allowed the dataset to be more diverse in the training data by shifting the images by 10% in width and height and flipping the images.

The model was run with the training dataset split into two: a training dataset and a validation dataset.

With the two datasets, the model could be tested on unseen data to see how well the model predicted letters with the new validation data.

```
Epoch 23/100
103/104 [============================>.] - ETA: 0s - loss: 0.3455 - accuracy: 0.9039
Epoch 23: val_loss did not improve from 0.09068
104/104 [=============================] - 1s 13ms/step - loss: 0.3453 - accuracy: 0.9029 - val_loss: 0.1365 - val_accuracy: 0.9462
Epoch 24/100
100/104 [============================>..] - ETA: 0s - loss: 0.3546 - accuracy: 0.8940
Epoch 24: val_loss did not improve from 0.09068
104/104 [=============================] - 1s 12ms/step - loss: 0.3472 - accuracy: 0.8971 - val_loss: 0.0950 - val_accuracy: 0.9692
Epoch 24: early stopping
```

Training/Validation statistics

After running the model with our training and validation data, the validation loss was 0.0950 and the validation accuracy was 96.92% whereas the training loss was 0.3472 and the training accuracy was 89.71%.
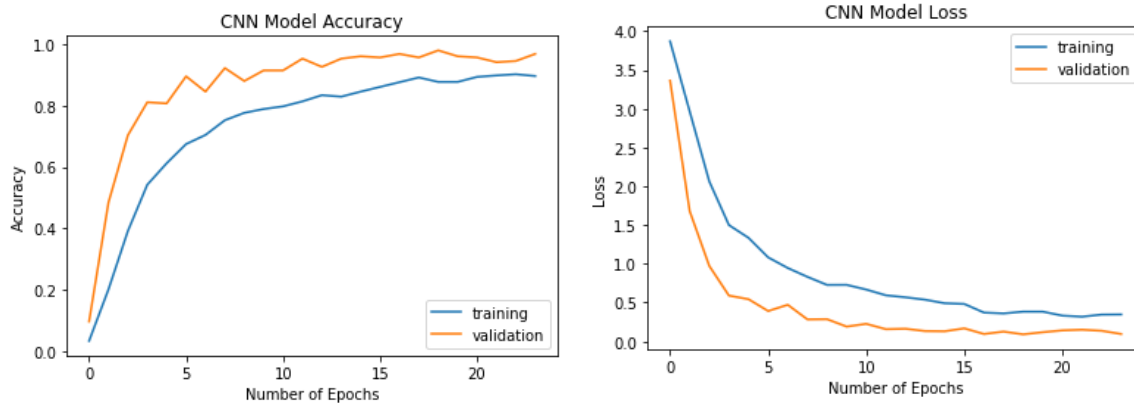
Looking at the graphs above, the model did a good job of not overfitting the data and the validation data did better than the training data.
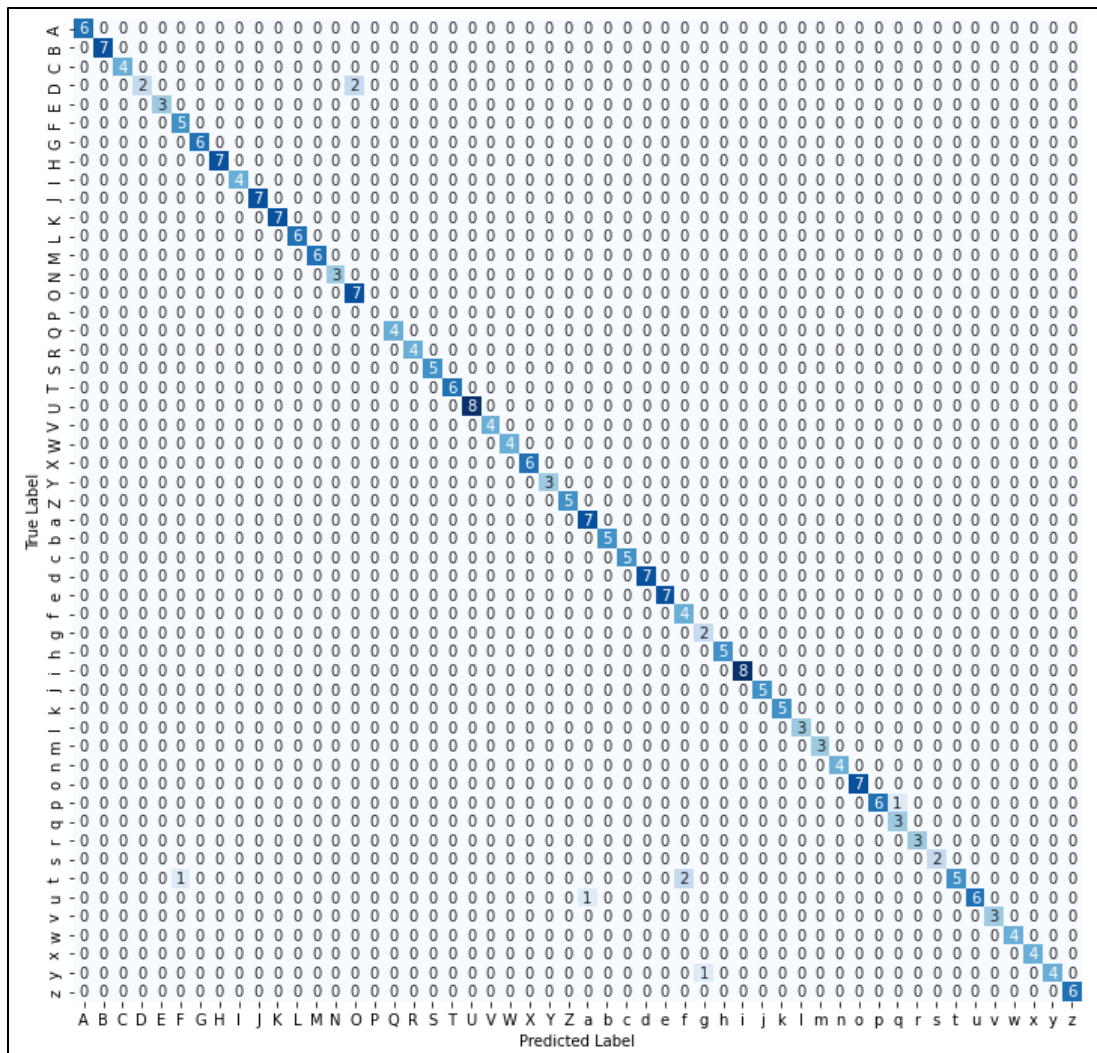
Confusion Matrix

Further evaluation shows using the confusion matrix, above, the model did have some difficulties predicting letters such as the letter "D" was predicted as an "O" and the letter "f" was predicted as a "t". With more training data this could ultimately be corrected but a 96% validation accuracy value is a very high percentage for this model.

Conclusion

This application helps adults and children learn how to write letters of the English alphabet with their mouse. With enough training data, I believe this is a great tool for both children and adults to practice writing their letters with their mouse.

Assumptions

The letters that are being trained are the proper way a student is taught to write letters. Personal ways of writing a particular letter such as writing the letter "z" with a slash may not be interpreted correctly. Personal letter flourishes may also not be recognized.

Limitations

The application is limited, currently, to the author's own writing of the letters therefore other's writing styles may not be recognized in the application. As more people add to the training dataset the better the model will be able to handle different writing styles.

Challenges

With any learning model like this, getting the model to have a low loss value and a high validation value with the training/validation data will be a challenge. Also, creating enough training data for the model to use effectively will take time.

Future Uses/Additional Applications

Future versions of this application will be to combine the letters into words and then eventually into short sentences. Other uses of this application can be to learn the alphabet of other languages or even learn to draw small characters such as animals.

Recommendations

This application needs a computer with some memory if you are going to train the data. For just practicing your letters a normal computer is sufficient. If the retraining button fails to run, you may need to go into Jupyter Notebook and run the training notebook.

Implementation Plan

To implement this application into production, a hosting environment able to handle TensorFlow, Python 3, Flask, HTML, and CSS is needed. The training button in the application should be commented out as training the model should only be done outside of the production environment.

Ethical Assessment

Ethically, the application may be used inappropriately. Training images that are not the correct letters or inappropriate images could be entered into the training part of the application and the application is not able to filter them out.

References

(2018). In F. Chollet, *Deep learning with Python.* Manning.

Busteed, D. (n.d.). *Build a Handwriting Recognition App with Python, Tensorflow, and Flask*. Retrieved from https://www.youtube.com/watch?v=lpWIf-iH7DA

Purwoko, P. (n.d.). *How to Deploy Digit Recognition Model Into Drawing App.* Retrieved from Analytics Vidhya: https://medium.com/analytics-vidhya/how-to-deploy-digit-recognition-model-into-drawing-app-6e59f82a199c
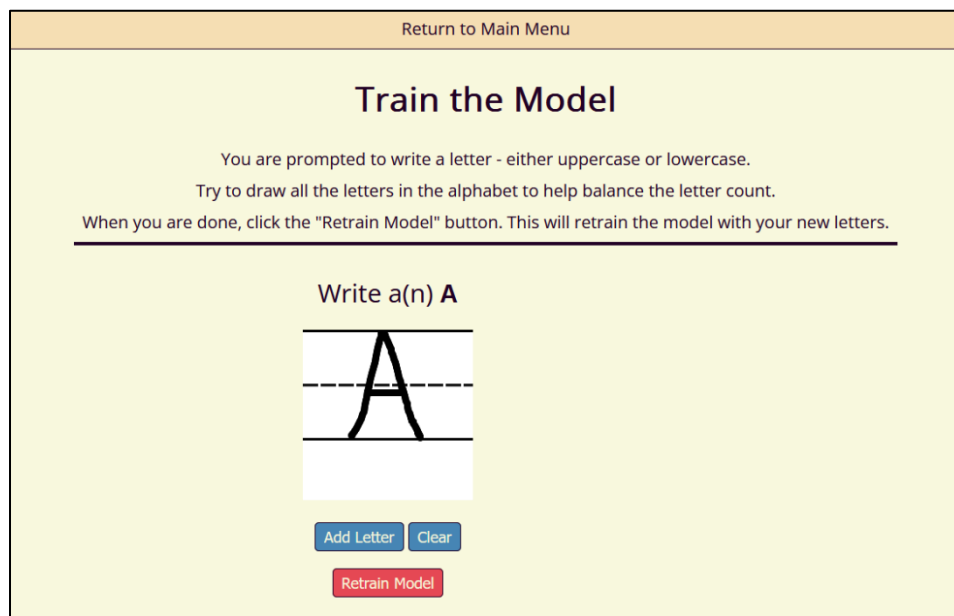
Appendix

Learn Your Letters Interface
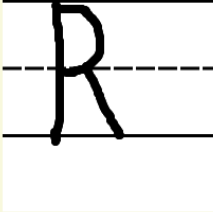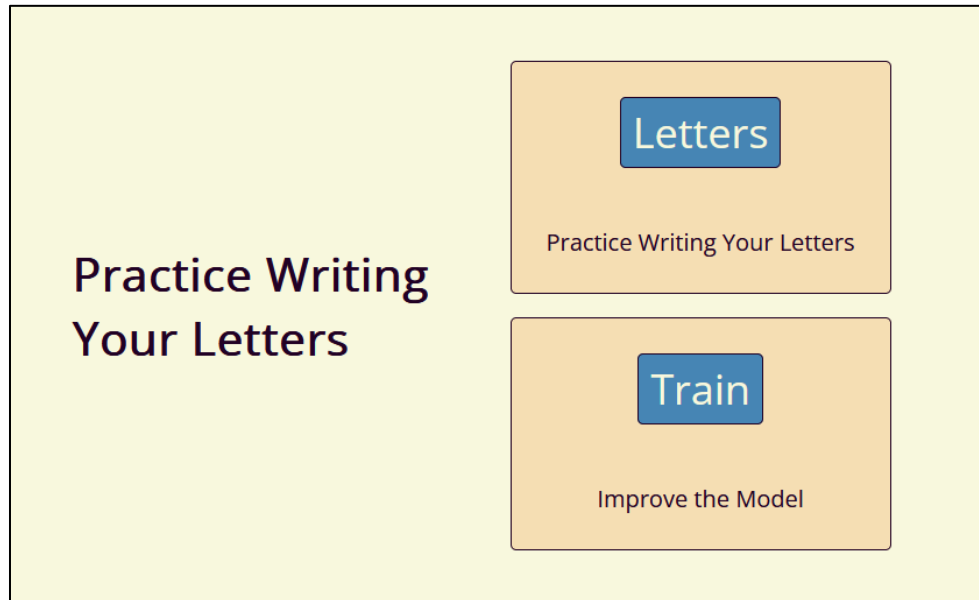

Home Page

Adding letters to the training data:

Train the Model Page

Practicing your letters:

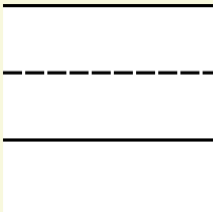## Practice Writing Your Letters

Write a(n) **R**

Check   Clear

Practice Your Letters Page

Writing a correct letter:

Practice Writing Your Letters

Write a(n) **V**

Correct!

You were supposed to write: **R**

I thought you wrote: **R**

Check   Clear

Practice your Letters Page

Writing an incorrect letter:

Practice Writing Your Letters

Write a(n) **V**

Maybe I was mistaken. Try again!

You were supposed to write: **V**

I thought you wrote: **A**

Check   Clear

Practice your Letters Page