Developing a Custom OCR Model

C. Van Nipper
Dpt. Electrical and Computer Engineering
Mercer University
Macon, Georgia, United States of America
11023852@live.mercer.edu

***Abstract*—Optical Character Recognition (OCR) models, while some of the simplest Neural Networks to develop and train, have proven to have many useful applications. They are often used to automate tedious processes that otherwise would have required human attention to accomplish. However, many models suffer with reading handwritten characters, especially when the users happen to have poor or unusual handwriting. To fix this problem, I propose creating a custom OCR model in which handwritten data created by this user is added to the training set and used to train the model. In this study, the effect of adding user-created handwritten data to the training and testing sets will be studied. Specifically, I am interested in utilizing the ratio of diverse data to user-created data as the independent variable, while examining the overall accuracy of the model as the dependent variable.**

After designing and training a variety of models, each utilizing different amounts of user-created training data, it was determined that for this structure of Neural Network, a model trained on roughly 20% user data and 80% standard data was the most effective at identifying a user's handwriting, granting an accuracies of roughly 85%. These levels of accuracy are much higher than accuracies observed in models with the same architecture and structure that were not trained with user-created data. The principle that OCR models have potential to grant much higher accuracies if they are trained on small amounts of user-created data can be useful in the pursuit of improving the effectiveness of OCR models that are used worldwide.

***Index Terms*—Neural Networks, Optical Character Recognition**

I. Introduction

Whether banking, healthcare, criminal justice, or otherwise, Optical Character Recognition (OCR) models are widely used to interpret words and phrases from images. In recent years, OCR has become a tool that is critical for the automation of processes that were tediously done by humans in the past. OCR is one of the most simple tasks a Neural Network can accomplish, and it does not require complex model architectures to be effective [1]. However, there are downsides. Many OCR models are not effective at reading the handwriting of users that they have not been trained on, especially if the user's handwriting is unusual or messy to read. An example is the Apple iPhone's OCR model, which excels at reading digital text, but often outputs inaccurate characters when attempting to read handwritten text from an image. This is especially true when the text is written by a user with unusual or messy handwriting. For this reason, in this study, I will attempt create a custom OCR model which excels at reading my own handwriting. To do this, I will train a model using a mixture of data from the EMNIST ByClass data set and handwritten data created by myself.

II. Project Goals

The custom model will utilize varying amounts of user data, and will have 62 classes of outputs, or characters (0-9, a-z, A-Z). To measure the effectiveness of the model, the following goals have been set:

A. Model Accuracy

The desired accuracy for the custom OCR model is in the range of 80% to 90%. This level of accuracy is standard for OCR models reading digital text with 62 classes of characters [2]. Accuracy will be calculated by the following equation:

$$\text{accuracy} = \frac{\text{\# correctly predicted characters in sentence}}{\text{\# total characters in sentence}} \quad (1)$$

B. Ease of Use

To make use of the model easier, it should be able to take an image of a handwritten sentence as input and output a digital string that represents the model's best prediction of what the sentence should be.

C. Optimal User Data

The ratio of of user data to EMNIST ByClass data used will be used as a parameter for this study. Six models will be trained, each using different concentrations of user data. This will enable me to determine if there is an optimal ratio of user data to EMNIST data which grants the highest accuracy, and if so, what that ratio is.

III. Methods

A. Data Preparation

All images of characters in both the training and testing sets are 28 pixel by 28 pixel, grayscale images. 30,000 images from the EMNIST ByClass dataset will be used, and I created 1,240 images of user data to be used. This number is expanded to 9,920 user images via data augmentation.

B. Model Architecture

The model architecture for this study is a Convolutional Neural Network (CNN) with the following layers:

1) Input layer (28 pixels by 28 pixels).
2) Convolutional layer (kernal size 5 pixels by 5 pixels).
3) Max Pooling layer.

4) Convolutional layer (kernal size 5 pixels by 5 pixels).
5) Max Pooling layer.
6) Dense layer (256 nodes).
7) Dense layer (128 nodes).
8) Dense Output layer (62 nodes).

This particular structure is derived from a special kind of CNN called *LeNet*, an structure proposed by Yann LeCun (the inventor of the CNN) to be effective at identifying handwritten characters [3].

C. Training Details

Six models will be trained using the architecture/structure mentioned above.

- Model 1: No user data, 100% EMNIST data.
- Model 2: 5% user data, 95% EMNIST data.
- Model 3: 10% user data , 90% EMNIST data.
- Model 4: 15% user data, 85% EMNIST data.
- Model 5: 20% user data, 80% EMNIST data.
- Model 6: 25% user data, 75% EMNIST data.

Each model will be trained with 10 epochs.

IV. Results

To measure the accuracy of differently trained models, three tests were used. Accuracy was measured and compared using the following tests:

- Test 1: Validation data (mix of EMNIST and user characters).
- Test 2: "example1.png", an image example of a sentence written in my handwriting.
- Test 3: "example2.png" another image example of a sentence written in my handwriting.

After running the three tests, the following results were observed:

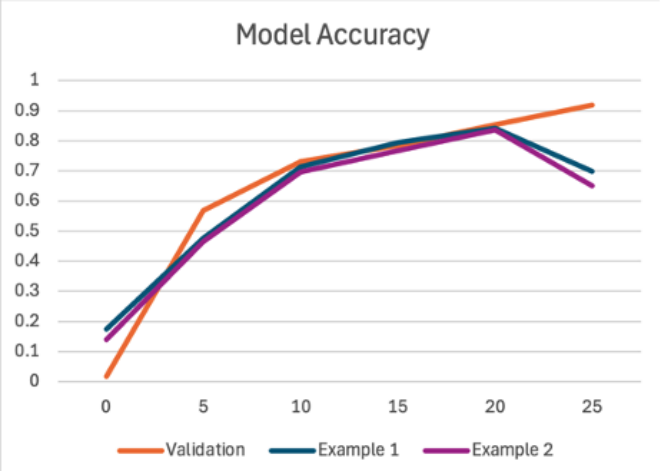

Fig. 1. User data used to train the model vs. its accuracy for each test.

As can be seen in the Figure above, for the validation test, as the percentage of user data increased, so did the overall accuracy of the model. However, for examples 1 and 2, we observe a slight decrease in accuracy at 25% user data. The increase of accuracy for the validation test is likely due to overfitting in the training set. Because of this, it can be inferred that the model trained with 20% user data and 80% EMNIST data is the most effective of the six models tested in this study.

V. Future Directions

While all the desired goals were met in this study, it is clear that there are several potential future directions that could be pursued if I were to continue with this study. While collecting data, I noticed that some characters tended to be less accurate than other characters. For example, it was difficult for the model to distinguish between the characters 'o', 'O', and '0'. Accuracy for characters that look similar could be improved by including an increased concentration of these difficult characters in the training set. Also, a clever usage of heuristics, or special rules, could help with improving the final accuracy for the model. An example of one of these rules is alphanumeric consistency, which says if a character begins and ends with a digit, all the characters in between should also be digits.

Another potential direction for the study would be training a new model with a new structure to recognize all characters, including special characters. This would make the task more complicated, but would ensure the could read any handwritten sentence with reasonable accuracy.

VI. Conclusion

After designing, training, and testing various models using varying ratios of user data to standard data, I was able to create an OCR model which is significantly better at identifying my own handwritten characters than other models. Applications of this study may extend to the utilization of personally-trained models in smartphones or other devices instead of using one universally-trained model. This could potentially be done by prompting the user to enter examples of their own handwriting for the device's model to be trained on. This study has shown that this new method of creating custom models on a user-by-user basis significantly improves accuracy, and while applying this principle to the real world may take some front-end effort, it would surely be worth it in the long run.

References

[1] S. N. Srihari, A. Shekhawat, and S. W. Lam, "Optical character recognition (OCR)", *Encyclopedia of Computer Science*, pp. 1326–1333, Jan. 2003.
[2] M. Sabourin, and A. Mitiche, "Optical character recognition by a neural network", *Neural Networks*, vol. 5, no. 5, pp. 834-852, 1992.
[3] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional neural networks and applications in vision", *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pp. 253-256, 2010.