

# Progress Report on Second Iteration

Yi Liu

## 1. Differentiation between Microfilm and Scanned images

In the first iteration of this project, we proposed to adopt the state-of-art image classification model, ResNeXt, using transfer learning, to classify the digitization type of documents. In addition, to train the model, we created a labeled database containing 1,200 images from the Civil War dataset. In the database, there are 600 scanned document and 600 scanned documents from microfilm.

In the second iteration, we continue to fine-tune the model for a better classifier. Based on the observation during the labeled database construction, the ratio of the number of digitized materials from microfilm to those from the scanning process is about 1 to 16. Hence, there are three metrics to evaluate the fine-tuned classifier.

Further, three evaluation metrics are (1) training performance, (2) validation performance, and (3) prediction performance. First, the training performance is the classification performance on the training set, which is 90% of the 1200-database. This metric represents the ability of the classifier on classifying data has been seen by the model. Second, the validation performance is the classification performance on the validation set, which is the rest 10% of the 1200-database. And this metric validates the training process to compute an expected prediction performance using a small set of labeled unseen-data. And third, the prediction performance is an evaluation of the entire Civil War collection. Based on the previous observation, the entire Civil War is expected to have about 2,256 document images digitized from microfilm. Hence, by comparing the predicted ratio of microfilmed and scanned document images, the strength of the classifier can be observed.

In the experiment, at which time to stop the training process and save the trained weights of the classifier is based on the training performance and validation performance. The general idea is to stop the training when both training and validation performances are good (i.e., the harmonic mean of training and validation F1 scores is greater than 99%). At the same time, we want to avoid overfitting and underfitting. Overfitting means the training performance is higher than the validation performance. Hence, the classifier could suffer from picking up noise when overfitting occurs. Underfitting means the validation performance is higher than the training performance. This is where the prediction could be biased. Considering the harmonic mean of two metrics has high response if two metrics have high values, and, at the same time, they are close to each other. Therefore, we compute the harmonic mean of the training and validation performance to decide the stopping point of the training.

Shown in Figure 1-4, the model started to converge usually after around 30 epochs. And both training and testing performances on the accuracy, precision, recall and F1 score are very promising. After convergence, the best epoch is the 44<sup>th</sup> training epoch, where the training accuracy is 98.52%, and the validation accuracy is 100%. Hence, the 44<sup>th</sup> epoch is stored for analyzing prediction performance.

In the prediction performance analysis, the stored classifier made predictions on the entire Civil War collection. Table 1 shows the prediction results. The prediction ratio of microfilmed document images to scanned document images is roughly 12:1. Hence, the classifier is more generous in classifying a document

image as digitized from microfilms than the expectation (i.e., 16:1). Figure 5-8 shows 4 types of typical mis-classifications.

The four types of “problematic” document images, are: (1) one that is largely “blank” (e.g., Figure 5); (2) one that has poor contrast quality (e.g., Figure 6); (3) one that is a picture of a physical item (e.g., a coin in Figure 7); and (4) one that is a graphical photo (e.g., a portrait photo in Figure 8). We suspect that there are two possible reasons. First, for type (1), there is little information for the classifier to make prediction since the document image contains largely background pixels. Second, for type (2), the poor quality could weaken the visual features that are required for the classifier to make the prediction. Third, these four types are rare or missing from the training database. Hence, the classifier was not trained sufficiently to make predictions.

Therefore, for future iterations of this project, two options could effectively improve the performance further. First, we can expand the training database to include more examples of the four-type document image to increase the variety. Second, we can apply a pre-processing step to normalize the document image quality for the collection before the prediction stage.

## 2. Conclusion

We found that classification performance for the digitization type differentiation to be promising. There are some mis-classified cases. However, the problem could be fixed by increasing the variety of the training database and applying pre-processing techniques. Further, although the microfilmed photo was not included in the training database, the classifier was able to correctly predict such photo as microfilmed material, shown in Figure 9. This suggests that the model has the generality to apply on a large collection for digitization type prediction.

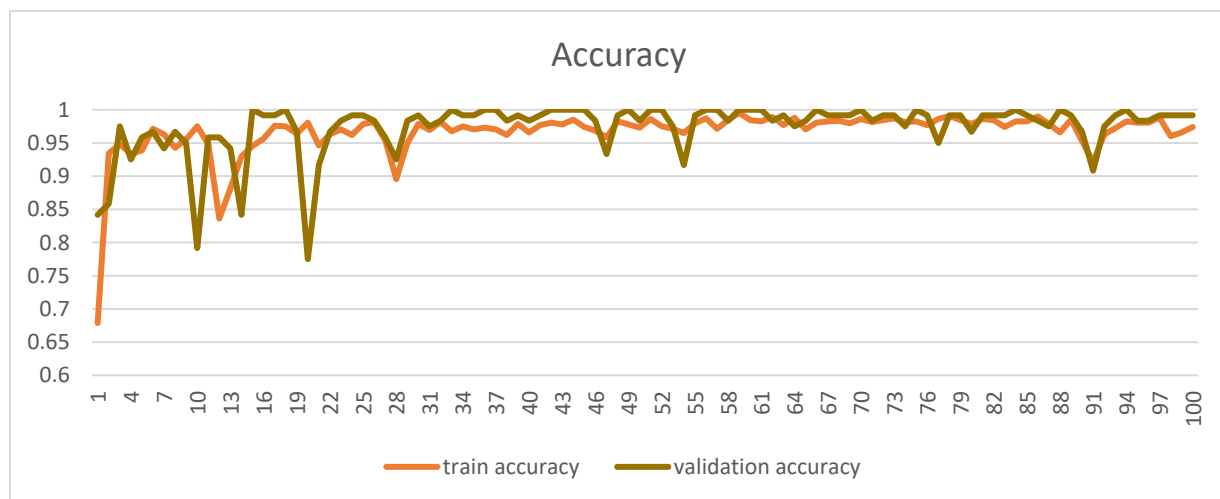


Figure 1 Training and validation accuracies

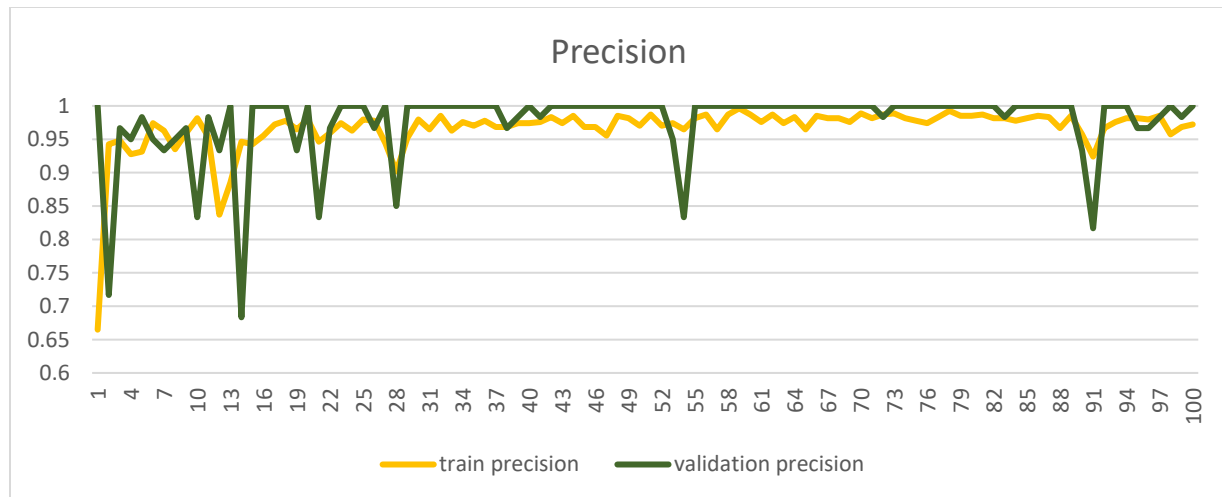


Figure 2 Training and validation precisions

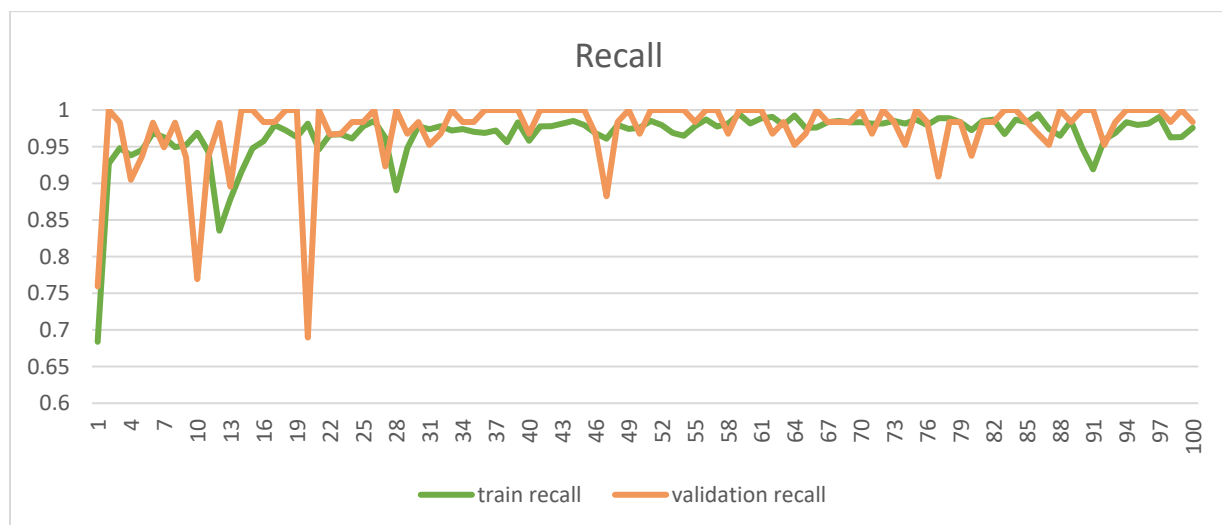


Figure 3 Training and validation recall

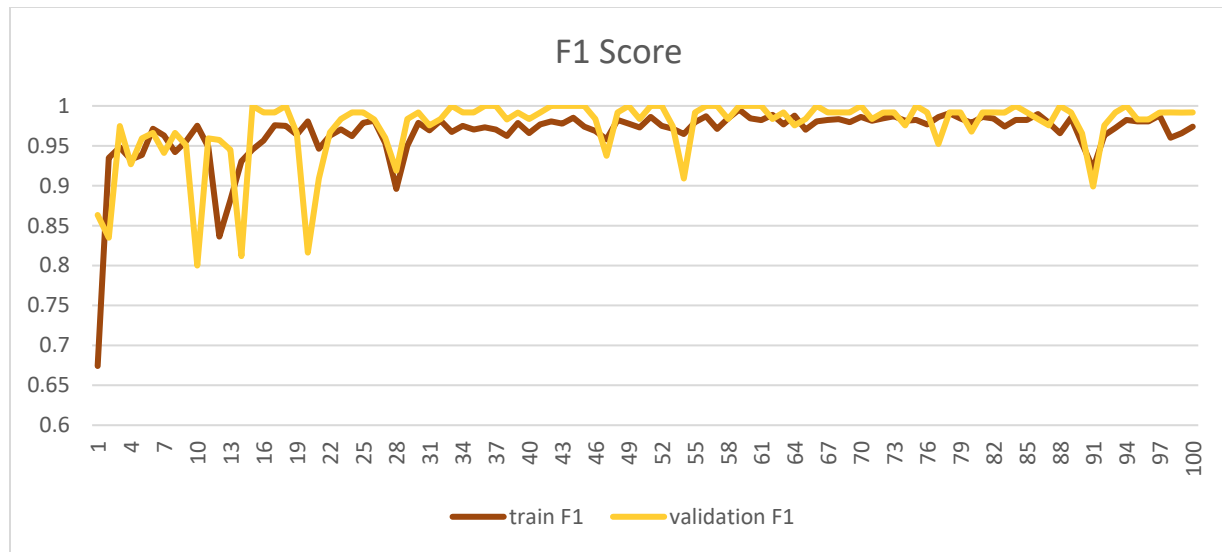


Figure 4 Training and validation F1 score

Table 1 Prediction Results		
Total	Predicted Microfilmed Documentation	Predicted Scanned Documentation
36103	2834	33269

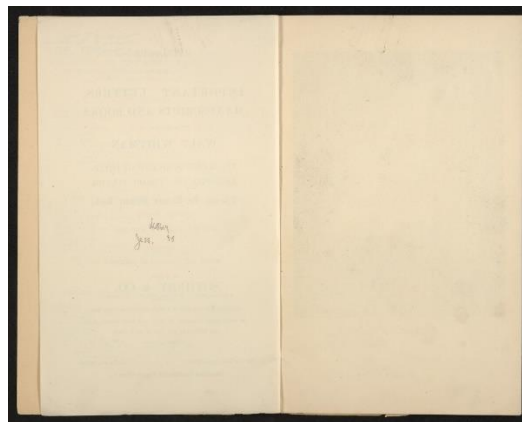


Figure 5 Type (1) mis-classification: "blank" document image

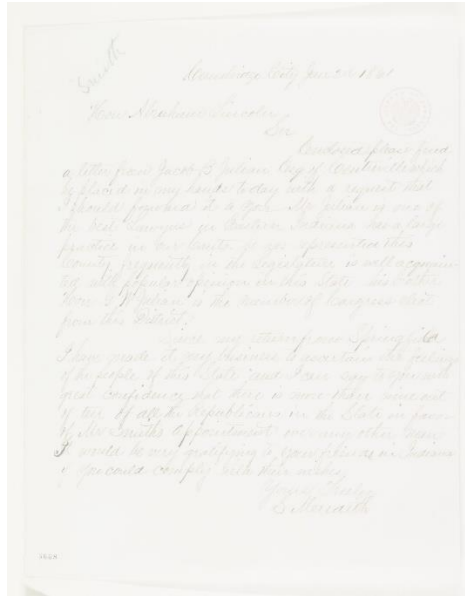
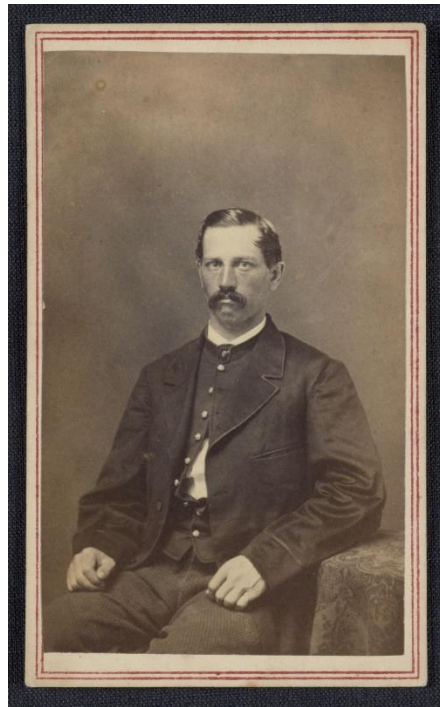


Figure 6 Type (2) mis-classification: poor contrast quality



Figure 7 Type (3) mis-classification: item images



*Figure 8 Type (4) mis-classification: graphical images (photo)*



*Figure 9 Microfilmed frame-photo being correctly classified*