DocuSort: Document Classification

Emily Turner
Department of Computer Science
Middle Tennessee State University
Murfreesboro, United States
ert3c@mtmail.mtsu.edu

Dylan Fox
Department of Computer Science
Middle Tennessee State University
Murfreesboro, United States
djf3f@mtmail.mtsu.edu

Anthony Ghebranious
Department of Computer Science
Middle Tennessee State University
Murfreesboro, United States
ang5v@mtmail.mtsu.edu

Tyler Christian
Department of Computer Science
Middle Tennessee State University
Murfreesboro, United States
tkc2s@mtmail.mtsu.edu

Munayfah Albaqami
Department of Computer Science
Middle Tennessee State University
Murfreesboro, United States
maa9e@mtmail.mtsu.edu

***Abstract*—Time is of the essence in all workplaces. Tasks should be performed optimally to ensure that the most amount of work is completed in the least amount of allocated time; however, complications arise when a task is tedious, complex, or intricate with details. As a result, productivity is lost and less work is completed. This situation can be especially common when sorting documents. Undergoing a task such as sorting documents requires classification, efficiency, and, most importantly, accuracy. For humans, this task becomes increasingly difficult over time, which can lead to sorting mistakes or allocating too much time to sorting when there are tools to provide significant aid. One such tool is the use of Convolutional Neural Networks (CNN). From using this too, an effective evaluation of classifying documents while simultaneously achieving validation without overfitting data was achieved. The network built was able to identify and sort documents from sixteen variants despite implementing a small data set. With further implementation, this work would be able to be interfaceable with other systems, simplifying work. Two models were created to complete the task one of which was a Convolutional Neural Networks (CNN) without transfer learning and the other was a CNN with transfer learning. We used 1,250 images from different classes as our dataset. The images were read into the computer as pixels. The two models performed differently we found that the simple CNN displayed extreme levels of overfitting while the CNN with transfer learning had minimal overfitting.**

***Index Terms*—convolutional neural networks, CNN, document classification, document sorting**

I. Introduction

Everyone has documents whether they are invoices or records. Sometimes keeping track of those documents can be tough. On top of that, every document has many different elements on it. Oftentimes we just need to find something specific on a document and we can't find the document. Or the document is long and we aren't able to find it. Oftentimes it is very hard to organize all the data from those documents resulting in important information not being found or some of the invoices not being paid.

Much of the data being used is from real documents. By using real documents we can ensure that the neural net performs the task on data that a user might actually have.

This project extracts the data from all the documents in a data set. After extracting all the data it then categorizes it and sends it to a spreadsheet for easier viewing. The goal is to perform this task using a neural net that can do all this on its own with little to no human aid. Having a tool like this will make it much easier to pay all invoices on time or find the information the moment you need it. The neural net will allow you to see all your different documents on a single document instead of searching through the many documents that you have. This neural net will organize everything in a spreadsheet which allows the user to know exactly where their money is going or know important details without looking through every one of your documents. When looking at invoices in particular it makes it easier for them to cut out recurring payments that may be unnecessary when on a tight budget.

II. Background

One motivation of our work was to create a product that would be applicable to other systems, simplifying work. Our aim is to create one CNN that will take in various document images (with noise) and classify those documents based on pixel intensities. If multiple images have pixel intensities in the same areas, then those documents most likely belong together. While such a goal would yield practical, diverse results, this means that challenges are also practical and diverse. For example, even after creating categories for documents, each category will still have great variability. One document could even appear as another (an email could look like a memo, or an invoice could look like a receipt). This would mean that generalization is a core challenge in this project.

Such ambitions and challenges were experienced by Harley, Ufkes, and Derpanis when also sorting documents, who also used CNNs to sort documents. Similar to Harley, Ufkes, and Derpanis, we used two CNNs, one as a small, holistic CNN

while the other is a “container” CNN, which holds the smaller CNN [1].

Another similar work includes the work of Cheng et al., where a CNN was also used to classify documents. However, the process of doing so was different. Instead of using images of various documents, textual relationships in documents are stored by additionally using a “long short-term memory recurrent network to obtain the high-level abstract representation[s]” of text throughout a document [2]. Our network is different in that we do not use Natural Language Processing (NLP) to feed data into our CNN since we are using images. This would mean that preserving spatial relationships are not as important in Cheng et al.’s work since sequential relationships are more important. In addition, intentionally feeding noise into their CNN could mitigate progress since the CNN must rely on not editing data while our network must rely on intentionally creating noise so we can pass the most amount of the most various document images into our network.

Last, Muhammad Zeshan Afzal also created a similar work using a method that is closely related to our method. They used a database of approximately 1.2 million RGB images. A major difference between our methods is that Afzal downsizes the images to 227 x 227 in order to ”lower the computational complexity of the system” [3]. Another difference between the two CNNs is our CNN doesn’t focus so much on the channel of the images, Afzal converts any grey samples in the data set to a three-channel that way it can conform to an RGB format.

III. Methods

A. Data

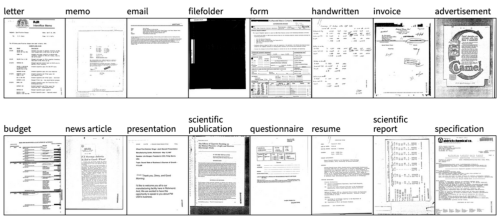

Fig. 1. The different classes of images we used in our dataset.

Our dataset for this problem is a subset of the RVL-CDIP (Ryerson Vision Lab Complex Document Information Processing) dataset, which consists of 400,000 grayscale images in 16 classes (shown below), with 25,000 images per class. Although we, initially, hoped to use the original dataset in its entirety, its large size led us to downsize. In the original dataset, the images were scattered across many folders and subfolders, so to help reduce the amount of space being used, we transferred all of the images into a single folder, and the images were given simpler names, like img1.tif.

However, this created a new obstacle because the original data labels were assigned to their corresponding images based on their path names, which were now null and void. To remedy this problem, we hand labeled just 1,250 of the images from the original dataset, which were chosen at random. Despite yielding a considerably smaller dataset, our efforts helped simplify the process of reading in the data images and their labels. However, because we chose the images to use at random, there was added bias due to some classes being underrepresented in the training data but being overrepresented in the testing data.

B. Our Approach

We decided to use our smaller dataset as an opportunity to test the efficacy of applying transfer learning to a CNN. In order to test the performance of transfer learning on a CNN, we, first, created a simple CNN that would serve as a baseline for gauging performance. Then, we created a CNN that used transfer learning, and we gathered data about its performance in order to compare it with our baseline.

The deep CNN we used as our base network for transfer learning was the VGG16 model. This model was originally proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition” (about VGG16). This high-performance, pre-trained model was developed for image classification tasks. It achieved 92.7

We anticipated that the CNN that used transfer learning would yield greater training and testing accuracy than the simple CNN by itself. However, we didn’t expect either of our two models to perform better than those used in the Ryerson University study, which were trained on the full RVL-CDIP dataset.

C. Image Classification

Both of our networks took in images of size 224×224. The images were read into the computer as pixels, and in order to preserve the spatial relationship of those pixels, the CNNs read in the image data as 2D tensors rather than trying to flatten the representation of the image into a one dimensional tensor. Then, they use kernels to scan patches of the image in order to identify where specific features might be present. After that, the network uses pooling to actually detect those features regardless of where they appear in the image, which eliminates translational variance.

IV. Results

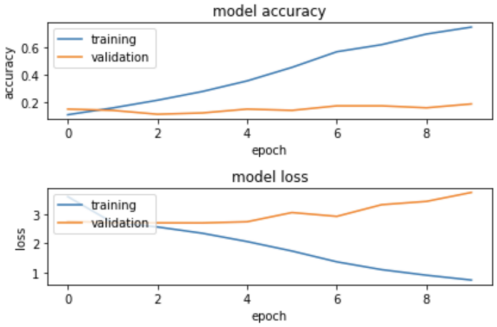

Fig. 2. CNN without transfer learning

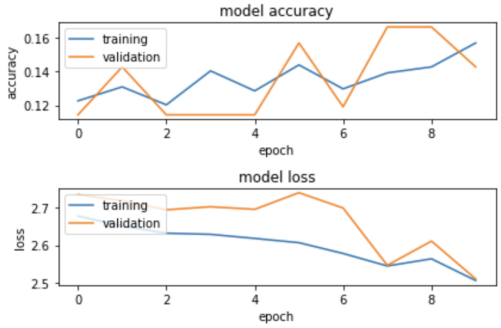

Fig. 3. CNN with transfer learning

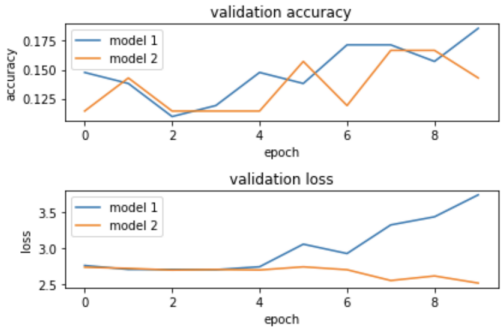

Fig. 4. Comparison of the accuracy and loss between the 2 models

After running both models, the simple CNN displayed extreme levels of overfitting while the CNN that used transfer learning had minimal overfitting. The simple CNN displayed 76.44% categorical accuracy for the training data but only 18.57% categorical accuracy for the validation data. Its categorical for the testing data was just 5.50%. This indicates that the simple CNN was not able to generalize for new data, it simply memorized the training data. In contrast, the CNN with transfer learning had 15.71% categorical accuracy on the training data and 14.29% accuracy on the validation data. However, the categorical accuracy for the testing dataset was still quite low at only about 4%. Despite having lower categorical accuracy on the validation and test data, the CNN with transfer learning was able to avoid overfitting, and was, therefore, able to generalize.

V. Discussion

Despite not achieving the results we expected, the CNN with transfer learning was able to generalize for the problem, and it avoided simply memorizing the training data like the simple CNN. In the future, it would be wise to ensure that the dataset being used is large and has more equal representation for all classes. The bias present in our dataset was likely the cause of both models performing poorly on the testing data. In spite of our dataset's shortcomings, we were able to develop two models that trained successfully, and the CNN with transfer learning was more successful at classifying images it had never seen before.

References

[1] A. U. Adam W. Harley and K. G. Derpanis, "Evaluation of deep convolutional nets for document image classification and retrieval."

[2] Z. Y. Y. Cheng and Q. Z. M. Wang, "Document classification based on convolutional neural network and hierarchical attention network," *Invent. Math.*, 2018.

[3] A. Parusiński and P. Pragacz, "Deepdocclassifier: Document classification with deep convolutional neural network," *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 2015.