Document Classification Using Convolutional Neural Networks

M. M. Diamantino *H. Pedrini*

Relatório Técnico - IC-PFG-19-07

Projeto Final de Graduação

2019 - Julho

UNIVERSIDADE ESTADUAL DE CAMPINAS

INSTITUTO DE COMPUTAÇÃO

Contents

Document Classification Using Convolutional Neural Networks

Matheus Mortatti Diamantino[*] Helio Pedrini[†]

Abstract

In this work, we explore different architectures of deep convolutional neural networks applied to the problem of document image classification, without the need of using OCR techniques or others to extract information about the document, letting the CNN to learn how to interpret and extract information about the images by itself. The popular document image dataset called RVL-CDIP is used to train this model to compare results to other approaches in this field (see [1, 2]) and also to compare results to different architectures than the ones used on them. We export weights from the VGG16 ImageNet network with inter-domain transfer learning and apply region based training with intra-domain transfer learning to train a model to classify documents by looking only at headers, footers and other sections of a document. We also use a real use case company document dataset to train VGG16. Since RVL-CDIP is a fairly complex dataset, the goal was to train a model that would be used in a real world situation with a useful dataset for their context. With this dataset, a 99% accuracy rate was achieved using data augmentation techniques.

1 Introduction

Document classification is a widely explored field in computer vision and machine learning. Manually classifying and separating documents can become a cumbersome task for companies that manage thousands of documents regularly, ranging from pay slips to employee information to different kinds of contracts. This motivates the importance of the problem of automatic document classification, so that companies can focus their employee time on more important tasks than organizing and labeling their documents.

Different approaches to the problem have been tried, especially in the computer vision area, for example, in the reference *Semi-Structured Document Image Matching and Recognition* [1], where the authors used an adaptation of object recognition for image documents. This technique involves the interest point extraction SURF, interest point selection to decrease the high number of points and an adapted Random Sample Consensus algorithm for separating correct point matches from bad ones. However, with the rise of Machine Learning techniques and, more specifically, Convolutional Neural Networks, we are now able to classify images by letting the network learn how to extract the right features from them to make the classification decision.

[*] Institute of Computing, University of Campinas, 13083-852, Campinas-SP, Brazil.

[†] Institute of Computing, University of Campinas, 13083-852, Campinas-SP, Brazil.

With this new way of classifying images, it is now possible to create a model that is flexible enough to classify all kinds of images, as seen by the popular use of the ImageNet database [3] to train networks for a general image classification problem. This dataset has 1000 (one thousand) different image classes and different Convolutional Neural Networks that are trained with it achieve over 95% accuracy.

This project specifies the general image recognition problem with CNNs to classify business documents for the main proposed usage in companies. We use the RVL-CDIP dataset to achieve a general purpose document classifying ranging from forms to handwritten documents. Moreover, we collaborate with a company to provide them with a specific model that is trained with their document dataset. In addition, we develop a simple tool to make it easier to use the model, where the user can choose the file to classify and the tool will classify it and place it in a directory with the label name.

2 Objectives

This project has the main objective of exploring different architectures for the problem of document image classification. We aim to reproduce the state-of-the-art work *Document Image Classification With Intra-Domain Transfer Learning and Stacked Generalization of Deep Convolutional Networks* [4], where the authors achieved 92.21% accuracy on the RVL-CDIP dataset by using transfer learning technique with the VGG16 [5] architecture to train a model using the whole document image and then train other models reusing weights from the first one over specific sections of the image, such as headers and footers.

We also aim to evaluate the architectures with real-world usage examples. For that, we partnered with the *Dom Rock* company [6] to use their document image database to train a model specific tailored for their needs. The approaches analyzed often use a general purpose document dataset, with classes ranging from forms to handwritten letters to news, but in the context of a more specific (and more realistic) usage we see not only a much more restrict number of classes, but also documents with a lot more in common among them.

3 Background

In this section, we briefly review some concepts that were used to develop this work, as well as discuss the steps we took to reach the final result.

3.1 Convolutional Neural Networks

A Convolutional Neural Network is a type of machine learning architecture that can take an image as the input and apply layers of filters that are learnable through the process of training the model through back propagation. That means that it can learn to filter relevant information from the input image and use that to find the classification result for that image.

This type of deep learning architecture yields better results for images compared to traditional neural networks because of the 2D nature of it. While Neural Networks take a 1D array of values to classify, Convolutional Neural Networks take a 2D input. That means that it can better analyze relationships between neighboring pixels, which is of great importance for the temporal and spatial nature of images. For example, if we are trying to train a model to identify whether or not an image has blur, we need to know how much a pixel value has changed compared to its neighboring pixels. If an image has excessive blur, then there will be a high amount of pixels which has a similar value to their neighbours. To a learnable network to detect that with high accuracy, there is no way of doing that if the image is transformed in a 1D array because each pixel now lost its relation with their original neighbors.

3.1.1 Architecture

As shown in Figure 1, the overall architecture of a convolutional neural network is comprised as the input layer, a series of filtering layers followed by a pooling layer, a few fully connected layers and, finally, the output layer.

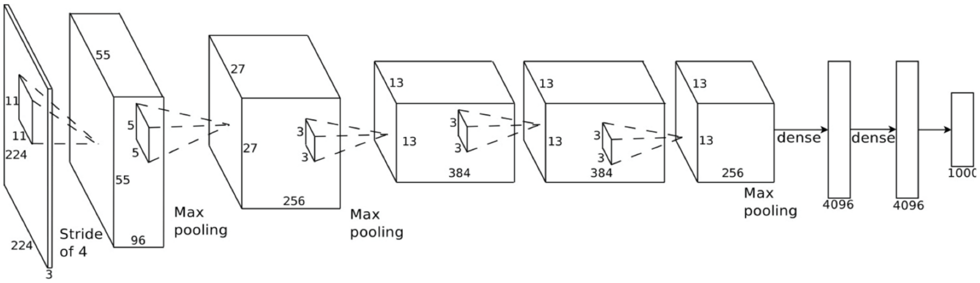


Figure 1: Example of a Convolutional Neural Network [7].

The input image can be either a one-channel binary image or a multiple channel one (usually RGB). This means that this type of network is able to take a three-dimensional input, rather than just a two-dimensional one. The subsequent filtering and pooling layers are the one that will extract information from the image so that the fully connected layers can take that information and learn to make the classification decision based on it.

3.1.2 Filters

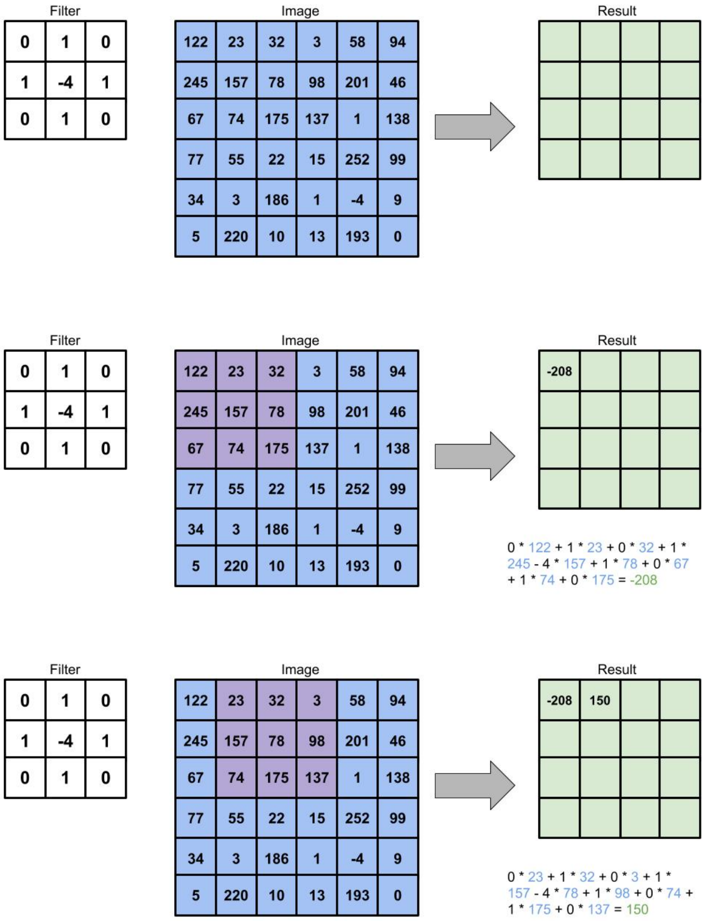
The learnable filters present in a Convolutional Neural Network work in a similar fashion from Image Filters present in the Computer Vision field. As illustrated in Figure 2, a filter (or kernel) is a matrix of arbitrary size that is applied to another matrix to get a new matrix as output. A filter F applies to an image IMG by doing the following operation:

$$R[i,j] = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} F[x,y] * IMG[x+i, y+j] \tag{1}$$

$$0 \le i < Size_x(IMG) - N, 0 \le j < Size_y(IMG) - N \tag{2}$$

where N is the size of the filter. By doing this, we will have a resulting matrix R of size $Size_x(IMG) - N$ by $Size_y(IMG) - N$. The sum described in Equation 1 can have an arbitrary step size of k so that the values for i and j are $0, k, 2k, 3k...N$. This step size is called Stride.

There are several known filters that can extract different information from the image. For example, the filter illustrated in Figure 2 is a Laplacian operator that will compute the second derivative of the image pixel values. This means that it is computing how drastic a change of pixel value is when compared to its neighbors, and with that it is possible to measure blur amount and detect edges for example.



0 * 122 + 1 * 23 + 0 * 32 + 1 * 245 - 4 * 157 + 1 * 78 + 0 * 67 + 1 * 74 + 0 * 175 = -208

0 * 23 + 1 * 32 + 0 * 3 + 1 * 157 - 4 * 78 + 1 * 98 + 0 * 74 + 1 * 175 + 0 * 137 = 150

Figure 2: Example of a filter application.

Filters are an effective way to extract information from an image. However, instead of using hard-coded values for the filter, a convolutional neural network will change its value depending on its goal, learning to extract relevant information for a specific problem that the model is trying to solve.

3.1.3 Pooling

A pooling layer works similarly to the filtering layer. It also goes through sections of the input applying operations to it, but instead of multiplying each filter values with one input value, it takes the whole section and takes either its average value or maximum/minimum value. When the pooling layer does the former, it is called an Average Pooling Layer, and when it does the latter it is called a Max/Min Pooling Layer. There are other types of pooling operations, but these two are the most commonly used in CNNs.

A pooling layer has the function of not only reducing the complexity of data, but also to reduce noise and variance from the data and to extract dominant features. However, an average pooling layer will not reduce noise or extract dominant features, but rather smooth out data values since it is taking the average values of neighboring pixels. For this reason, max/min pooling is most frequently used.

The decision of whether to use minimum or maximum pooling comes down to the nature of the data. If the image is mostly dark, taking the maximum values will extract bright areas, whereas taking the minimum will make the image even darker. The opposite happens when an image is mostly white, so the architecture of the CNN needs to make use of one or the other depending on the nature of the problem.

When we combine multiple filtering and pooling layers, we are trying to extract different image features with each of them. Usually, the layers at the beginning of the model tend to extract low-level features from the image such as edged, color and gradient orientation. When the data reaches later steps in the model, it starts abstracting the data more and more so that it removes information that is not relevant for the classification itself, extracting more high level features from the data.

3.1.4 Fully Connected Layers

Having fully connected layers is useful for taking the extracted data from the convolutional layers and apply the actual classification based on it. The final image output is flattened into a 1D vector and fed into the feed forward neural network and then a back propagation algorithm is applied for learning the correct weights for classification.

3.2 Transfer Learning

Transfer Learning is the process of training a model for a problem using as a base weights that were developed by training that model over a dataset for a different problem. For this technique to have a good result, the base weights needed to either be used for a similar problem or have been developed to extract and classify general properties of the data. As an example, take a convolutional neural network that was trained to detect and classify a number of different images as to what is present in it, such as a chair, a dog or a lizard.

As discussed previously, it is often the case that the first layers of a CNN learns general, low-level features of an image, such as edge detection. Because of that, if we were now to reuse that network to classify documents, the model could reuse those learned features to solve this problem.

The last example is referring to a very popular dataset called ImageNet [3] that has been largely used to train a number of different network architectures. A few of them are VGGNet [5], ResNet [8] and GoogLeNet [9].

An architecture trained on the ImageNet dataset was proposed for this project to greatly reduce training times and increase accuracy since the model now can reuse useful learned filters instead of learning them from scratch.

3.3 Ensemble

Ensemble is a machine learning technique that combines different model results for the same problem to create a new result based on them. It is common in the machine learning field to train multiple model architectures with the same dataset to provide different learning results. However, it can be difficult to train a model to have a good performance on all aspects of a dataset, so combining different models to get the result can be crucial to improve accuracy when each model performs better in areas that others do not.

There are several different ensemble methods, but the one that is used in this work is the Stacked Generalization Method, where we take the final results of multiple models and train a new model based on those results.

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & y_{32} & x_{33} \end{bmatrix} \qquad Y = \begin{bmatrix} y_{11} \\ y_{21} \\ y_{31} \end{bmatrix} \tag{3}$$

As illustrated in Equation 3, say we have X as the training dataset for the Stacked Generalization Model. Each column represents the classification results of each models, whereas the Y matrix is the true values for each classification. It is important to notice that the dataset used to train the models needs to be different than the dataset used to train the ensemble model to avoid over-fitting. In the case of this project, the validation dataset for the main models was used to train the ensemble one and the test dataset was used for final validation.

3.4 Interpreting a Convolutional Neural Network Model

Understanding exactly what each layer of a machine learning model is doing is not a simple task, but it is crucial to do it so that not only it is possible to tune the model to improve accuracy, but to analyze what aspects of the data the network is focusing on, so that we can "debug" the model the same way we usually do a normal program.

With Convolutional Neural Networks, since the input is usually an image, it is easier to extract data that is more easily interpreted by a human. The simplest way of doing this is to plot the output of each filter as a 2D image, and that already gives us great insight on what each filter is extracting from the image. However, it is possible to create a heat map for each layer to better visualize in which regions of the image the filters are focused on and, to do that, we can use GradCAM [10] (Gradient Weighted Class Activation Map). In later sections, this technique will be used to interpret the output of the networks trained for this project.

It is important to point out that the code used to implement this technique was developed by Rawlani [2] and the paper [11] written by the same person also helped understand how Interpretability works.

3.5 RVL-CDIP

RVL-CDIP [12] is an extensive dataset with 400,000 grayscale document images divided into 16 classes, with 25,000 examples for each class. It is divided into 320,000 images for training, 40,000 images for validation and 40,000 for final testing. The classes present in this dataset are illustrated in Figure 3.

4 Work Development

In this section, we present the main stages associated with the development of our proposed method.

4.1 Choosing Different Approaches

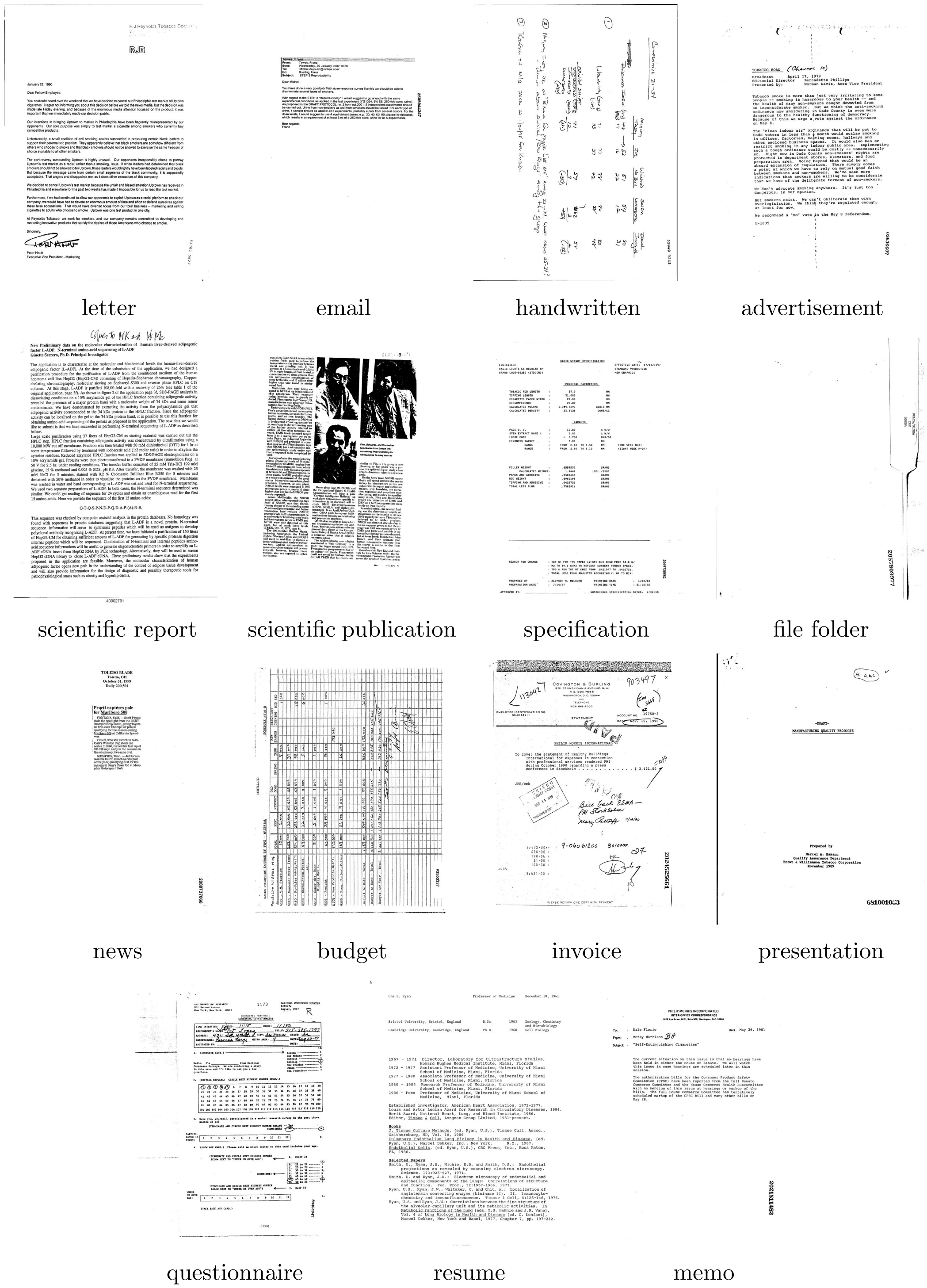
At the start of the project, research on the topic was conducted to know previous work on the topic of document classification. From the most recent approaches, it was found that, although there was still some focus on Computer Vision techniques, such as interest point extraction, most of the new studies in this topic are in the Machine Learning field, more specifically using Convolutional Neural Networks. For this reason, the effort made in this project was focused on this field.

4.2 First Try - NIST Tax Form Dataset

Initially, we developed a partial reproduction of the paper *Convolutional Neural Networks for Document Image Classification*. It creates a small CNN architecture consisting of an input layer of a downsampled, normalized image of 150×150 resolution, first convolutional layer with 20 kernels of size 7×7 each followed by a 4×4 Max Pooling layer, second convolutional layer with 50 kernels of size 5 followed by another 4×4 Max Pooling layer. Finally, two fully connected layers with 1000 neurons were followed by the output layer that consisted of logistic regression with softmax function that produced the probability of each class. This network is illustrated in Figure 4.

The described architecture was trained with the *NIST Tax Form Dataset* [13], a collection of 5590 tax-form images from the National Institute of Standards and Technology, categorized into 20 different classes (a few examples are shown in Figure 5).

It achieved 100% accuracy, as well as our model with the same architecture, which will be further discussed in the next sections. It also trained on another dataset that shared similarities with the RVL-CDIP, achieving accuracy of 65.35%. However, this part of the work was not reproduced.



letter email handwritten advertisement

scientific report scientific publication specification file folder

news budget invoice presentation

questionnaire resume memo

Figure 3: Examples of each class from the RVL-CDIP dataset.

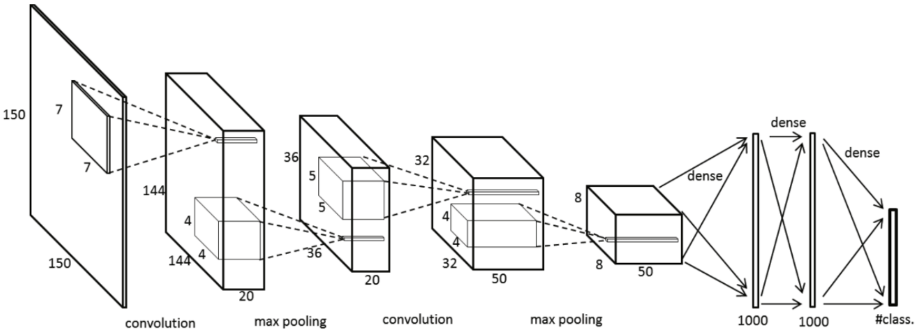
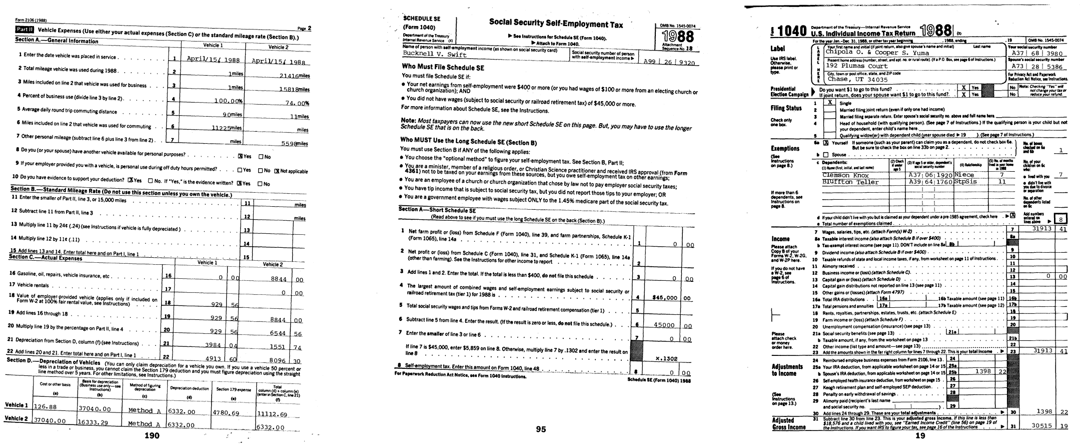


Figure 4: Convolutional Network Architecture used in the early stages of the project.



(a) (b) (c)

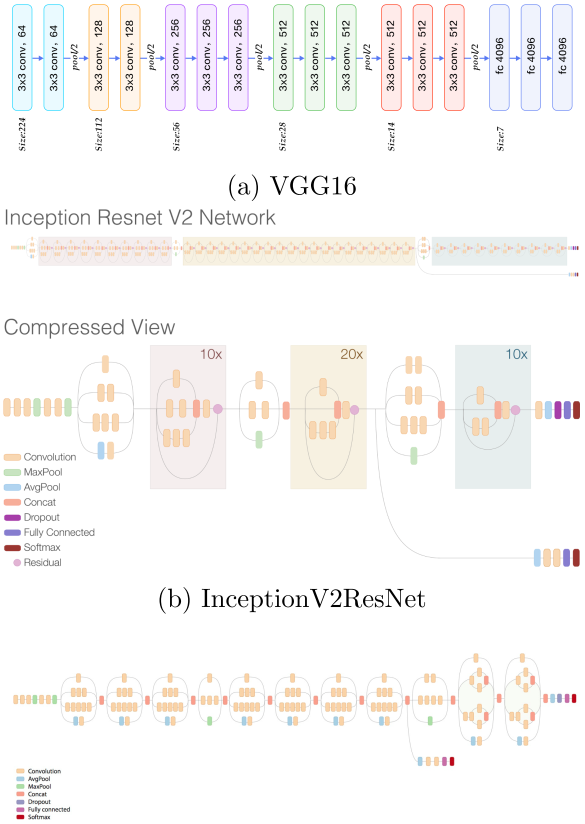
Figure 5: Examples of each class from the NIST Tax Form dataset.

4.3 Transfer Learning Usage

Since the accuracy rates in the last section were too high, we started looking for a more complex dataset that featured not only classes with more variety, but also with greater intra-class variation.

The RVL-CDIP dataset was found and used in the work *Document Image Classification With Intra-Domain Transfer Learning and Stacked Generalization of Deep Convolutional Networks* [4]. In that work, it is stated that the ImageNet trained network VGG16 performed better for the document classifying problem. Thus, the next part of the project was to verify that this was true. For that, three different pre-existing networks were trained on the RVL-CDIP dataset with the Transfer Learning technique: VGG16, InceptionV3 and InceptionV2ResNet. Each network structure is illustrated in Figure 6.

Although our goal was to increase the number of networks used, hardware limitations prevented us from doing that since it is common for ImageNet trained networks to be very large in size and requiring a lot of VRAM on the GPU. AS illustrated in Figure 6, except for the VGG16 network, the networks are several layers long and normal consumer-grade GPUs can be not enough to fit them in.



(a) VGG16

(b) InceptionV2ResNet

(c) InceptionV3

Figure 6: Structures of each network used for transfer learning.

4.4 Final Network - Stacked Generalization Ensemble

After analyzing which network worked best for the field, the next step was to continue following the work in the paper described in the last section. The authors trained a holistic network over the RVL-CDIP dataset using transfer learning with VGG16 and then used the resulting weights to train four other networks using different parts of the input image: header, footer, left body and right body, as illustrated in Figure 7.

They improved training times since the new networks were using weights that were developed for the same problem. In addition, training different networks on specific sections of the image and then combining the results of each one for a final classification could improve accuracy, in theory.

For that, a Stacked Generalization Ensemble technique was used, as described in previous sections. A 3-layer fully connected neural network (256-256-16) was used to train over

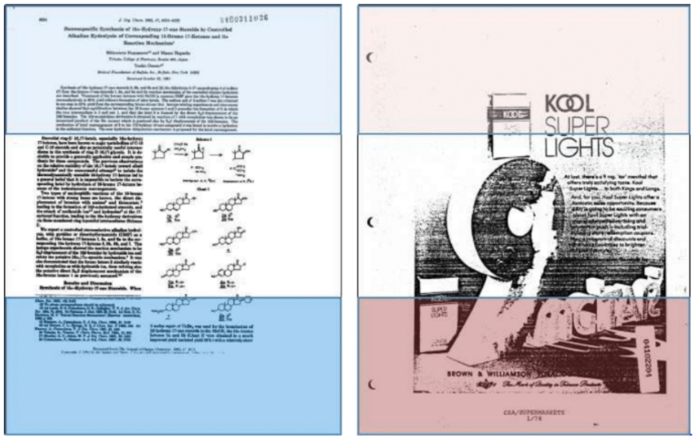


Figure 7: Example of extracting the header and footer of a document image (taken from [4]).

the results of the previous five networks and got results with 92.21% accuracy, a state-of-the-art result for this problem.

4.5 Dom Rock's Dataset

Dom Rock [6] is a company that provided us document images related to their business, such as invoices and statements. Thus, we were able to work with real world data and create a model that will have practical usage for a business that needs their documents classified and organized.

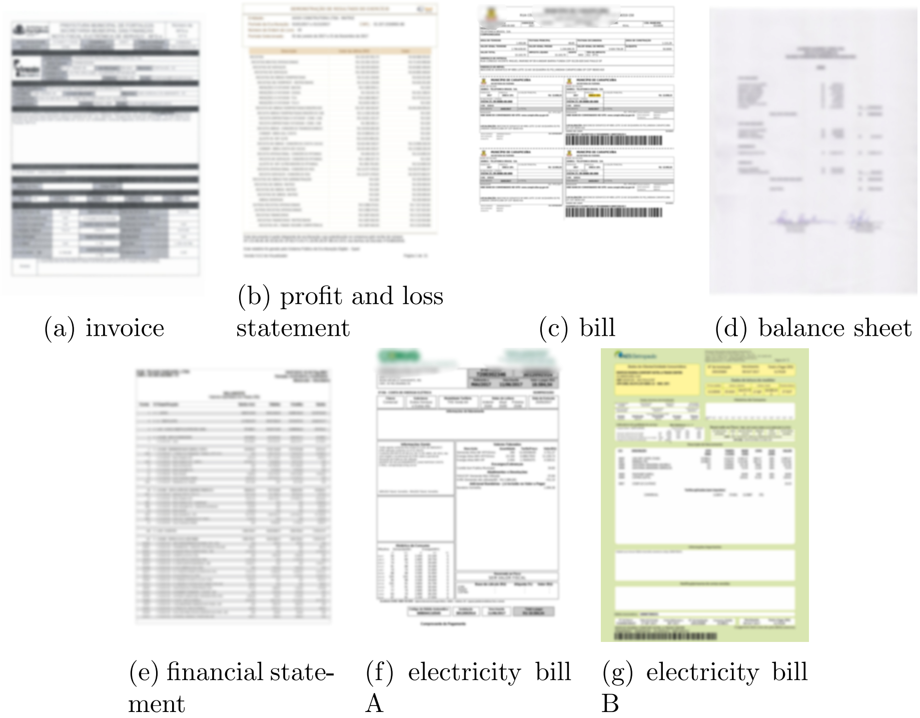
As illustrated in Figure 8, the dataset is divided into eight classes and every image from it shown in this paper will have to be blurred for privacy purposes.

The data was provided in a non preprocessed state, meaning we had full pdf files with multiple pages and unorganized classes. The first step was to separate all pdf files in individual images and organize the data properly into the final classes.

Then, it was noticed that the dataset had two problems: lack of data and unbalanced classes. Transfer learning is often used on datasets that are not large enough to train a network from scratch, however, this dataset had less than 1000 examples and some classes had less than 10. To address that, a data augmentation technique was used to both expand the number of examples and to balance the classes.

A tool was created to count the number of examples on each class and augment each of them based on it, by creating more examples of classes that had fewer examples. The new images were slightly rotated and zoomed in or out to simulate bad scanning of documents and make the network more resilient to bad data. The result was over 4000 examples, with each class having less than 100 images of difference in size.

The next step for the project, then, was to use this dataset to train the VGG16 network using transfer learning, and if necessary apply ensemble techniques to try and improve the results.



(a) invoice

(b) profit and loss statement

(c) bill

(d) balance sheet

(e) financial statement

(f) electricity bill A

(g) electricity bill B

Figure 8: Examples of Dom Rock's dataset.

4.6 RVL-CDIP Simple Version

Since the main idea of this project was not only to have a proper document classifying model, but to apply it in real case scenarios, such as business-type documents, we decided that the next and last step of this project was to remove from the RVL-CDIP dataset the document classes that would not be applicable or used in a company such as Dom Rock. Therefore, the following classes were chosen to train a new model: (a) form; (b) email; (c) specification; (d) budget; (e) invoice; (f) questionnaire; (g) resume; (h) memo.

4.7 Document Classification Tool

Since there was a real need for a tool that classified documents, one was created. It simply takes a document, separates each of its pages into one image and inputs each one into the model.

Then, it combines the answer to come up with a final one by summing up the probability result vector of each classification and then taking the argmax of the final probability vector. It then places the original document inside a folder with the classification label name inside the folder specified in the user interface (UI). The resulting tool is illustrated in Figure 9.

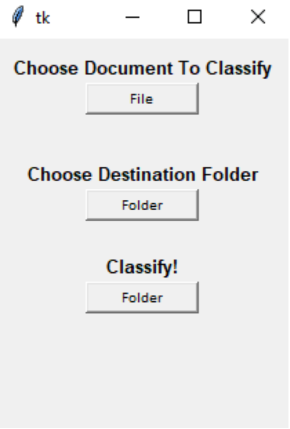


Figure 9: Simple UI for the document classification tool.

5 Results

All experiments were performed using a notebook computer equipped with an Intel Core i7-6700HQ processor (4 cores) running at 2.60GHz and 16GB of RAM, with Ubuntu 16.04 and a Nvidia GTX 1060 GPU with 3 gigabytes of VRAM.

5.1 NIST Tax Form Dataset

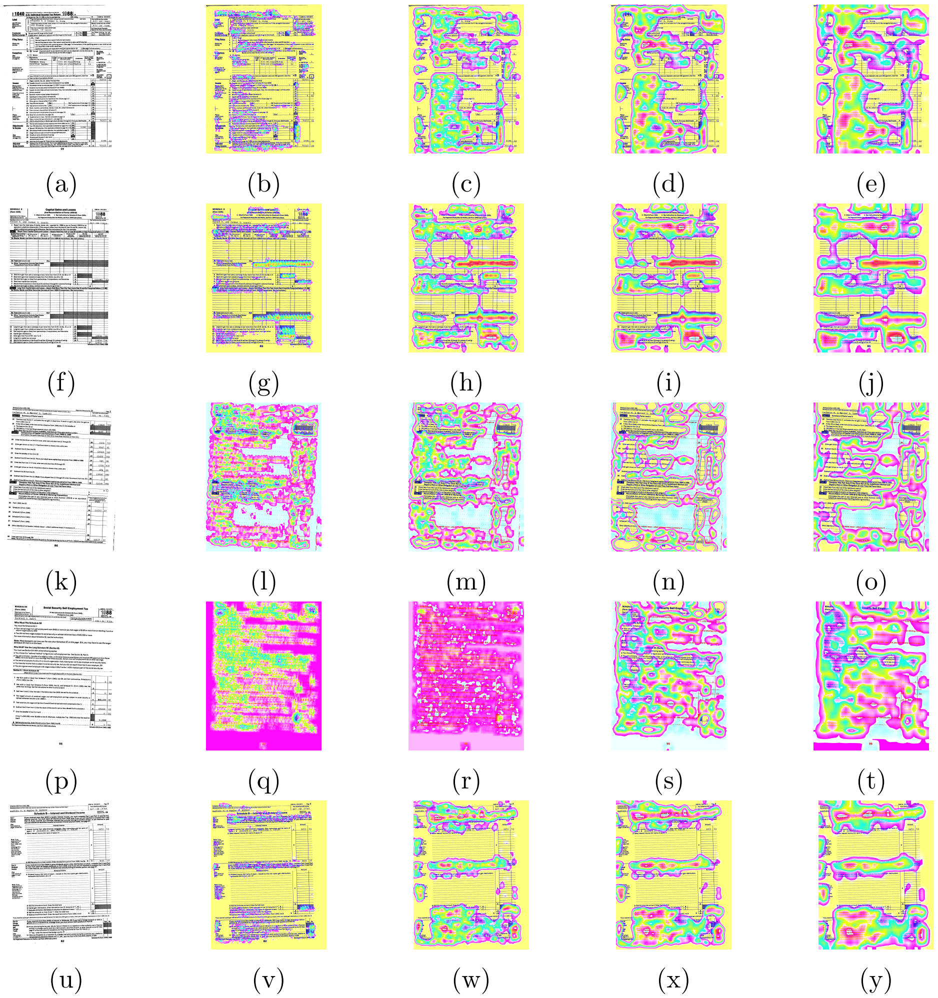
For the model that was trained with the NIST Tax Form Dataset (see figure 4), we obtained a result with 100% accuracy across the training, validation and test dataset. In the machine learning field, it is hard to obtain a perfect score such as this, so further analysis is required to understand what the network is extracting from the image in order to have achieved this result. For that, we will extract a heatmap image from each layer of the network that will illustrate what were the focus points that the network used to classify the image.

Upon analyzing Figure 10, we can see that the network starts by focusing on very specific points of the document and does a very good job of extracting the points where not only there is text, but where there are key structural points to classify the image, such as the header of the document.

As we progress through each layer, the network extracts higher level features from the document by focusing on larger areas. We can see that it expands focus on what makes the document structure, such as text areas and lines that separates document sections.

Almost every image present in this dataset has their document type written somewhere on its header. Notice how on every image shown in Figure 10, the area where the type is written is highlighted by a heat signature. This shows that the model was efficient on figuring out what were the most important aspects of the document so that it could classify it as intended.

It is also interesting to point out that, in some cases, the focus points occur on the negative spaces of the document (areas where there is no written text of drawings), such as in Images 10k to 10o.


(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l) (m) (n) (o)

(p) (q) (r) (s) (t)

(u) (v) (w) (x) (y)

Figure 10: Heatmap output of each layer of network from Figure 4, from the input image (left) to the last convolutional layer output (right). Each row is a different document type.

5.2 Transfer Learning Network Comparison

Three networks with ImageNet weights were trained over the RVL-CDIP dataset. They were put through 20 epochs each with a batch size of 100, which translates to a training with 320,000 images. For the results seen in Table 1, a 40,000 image validation dataset was used.

Table 1 shows that the VGG16 network performed better than the other two by at least 2 percentual points in the precision metric. Both Inception networks are larger and more complex than the VGG16 and that might have contributed to this result. While more complex networks might work better for when there is a great variety in the dataset, such as in ImageNet, for a document image dataset that might translate in slightly greater overfitting.

	Precision	Recall	F1-Score
VGG16	88	87	87
InceptionV2ResNet	86	86	86
InceptionV3	84	84	84

Table 1: Results for each architecture.

Documents often have structural similarities among them and simple feature detection filters would extract enough information for classification. Larger networks will over extract features and obtain too specialized in the dataset that it was trained on, increasing the chance of overfitting.

From this result, it is possible now to confirm the assumptions of previous works that state that the VGG16 network often perform better on the document classification problem.

5.3 VGG16 Holistic Training and Intra-Domain Transfer Learning

As detailed in previous sections, to reproduce the work done in *Document Image Classification With Intra-Domain Transfer Learning and Stacked Generalization of Deep Convolutional Networks* [4], it was necessary to train the VGG16 network with the RVL-CDIP dataset and then use the resulting weights to train four other networks that focused in different sections of the document: header, footer, right body and left body. This process is what is called Intra-Domain Transfer Learning.

All document images were preprocessed by first extracting the area of the document that it needed to train on. Then, we resize the image to a resolution of 150×150 to simplify the training reducing training time. This did not compromise training because we are interested in the overall structure of the document and not on what is actually written on them. However, the work being reproduced used a resolution of 224×224 in training, but that was not possible with the hardware in our disposal so it was necessary to reduce the resolution for training.

	Our Work	Das et al. [4]
Holistic	88.1	91.1
Right Body	81.3	82.2
Left Body	86.1	85.2
Header	81.4	86.0
Footer	78.7	81.2
Ensemble	60.0	92.2

Table 2: Results for VGG16 trained over RVL-CDIP dataset.

As shown in Table 2, the work done in this work did not match the one performed by Das et al. [4]. There are a few aspects that can explain this result.

Firstly, looking at the holistic model, our results were 3 percentage points behind. That is considered a good result because of the fact that the training aspects of the reproduced work was not disclosed, such as batch size and epochs. The resolution of the input image also might have influenced to the difference in result. The same argument is applied to the intra-domain transfer learning models, since we see similar results ranging from 0.9 to 4.6 percentage points behind.

The holistic model had an accuracy of 99.54% on the training dataset. Comparing this result with the validation and test dataset of 88%, we see a massive drop. That shows as the level of overfitting we are dealing with and how there is a fine line between extracting detail that will lead to a more general classification result and one that will only serve to classify documents similar to the ones already seen in training. It makes sense since documents can be very similar to one another and extracting fine details can be crucial to obtain the correct classification, however, it can easily lead to overfitting.

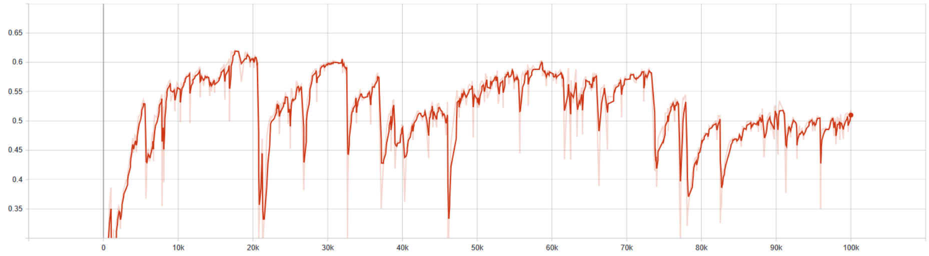


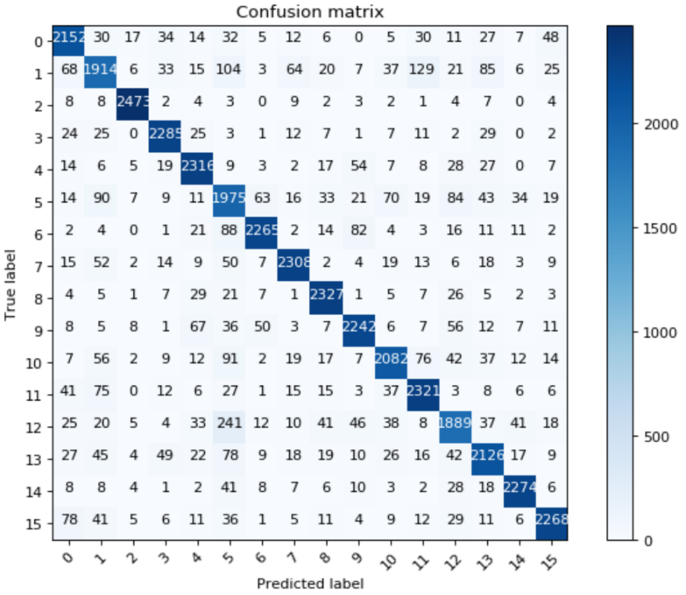
Figure 11: Accuracy progress during training of our stacked generalization ensemble model.

However, more interestingly is the result for the ensemble model. Our results were 32.2 percentage points behind, which is very surprising given that the architecture was the same and the results from the five networks used were not that much different. We can see in Figure 11 that the accuracy during training was very unstable, as it dropped as much as 20 percentage points in between training epochs.

The best results that were obtained was using a batch size with the same size as the training dataset. That was expected, since a higher batch size means more chances that the model will reach the lowest loss value in its loss function, with the downside of a slower training. 100,000 epochs were used to obtain this result.

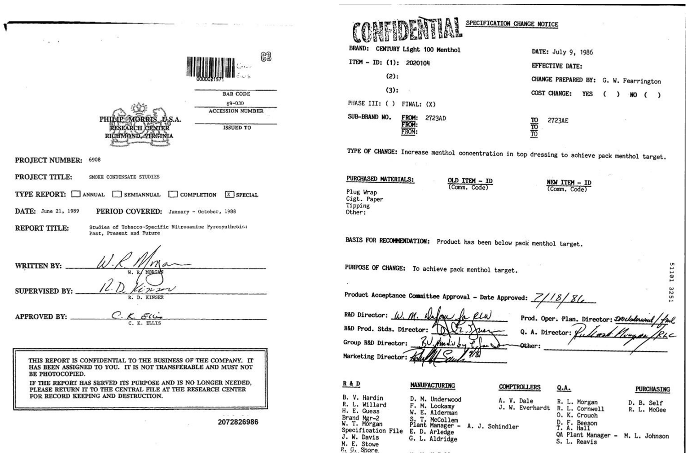
Upon further analyzing the confusion matrix from the holistic model (Figure 12), it is possible to spot a the classes that the model got confused the most. The ones that stand out are Scientific Report and Form, with 70% and 80% results on the precision metric, respectively.

Both classes get confused with each other a considerable amount of times. Figure 13 shows similarities between them in an example. We can observe that both can have lines that have the purpose of handwritten input and signatures that usually have very curved lines. That could have been one of the reasons why the model had a hard time differentiating between both.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2152	30	17	34	14	32	5	12	6	0	5	30	11	27	7	48
1	68	1914	6	33	15	104	3	64	20	7	37	129	21	85	6	25
2	8	8	2473	2	4	3	0	9	2	3	2	1	4	7	0	4
3	24	25	0	2285	25	3	1	12	7	1	7	11	2	29	0	2
4	14	6	5	19	2316	9	3	2	17	54	7	8	28	27	0	7
5	14	90	7	9	11	1975	63	16	33	21	70	19	84	43	34	19
6	2	4	0	1	21	88	2265	2	14	82	4	3	16	11	11	2
7	15	52	2	14	9	50	7	2308	2	4	19	13	6	18	3	9
8	4	5	1	7	29	21	7	1	2327	1	5	7	26	5	2	3
9	8	5	8	1	67	36	50	3	7	2242	6	7	56	12	7	11
10	7	56	2	9	12	91	2	19	17	7	2082	76	42	37	12	14
11	41	75	0	12	6	27	1	15	15	3	37	2321	3	8	6	6
12	25	20	5	4	33	241	12	10	41	46	38	8	1889	37	41	18
13	27	45	4	49	22	78	9	18	19	10	26	16	42	2126	17	9
14	8	8	4	1	2	41	8	7	6	10	3	2	28	18	2274	6
15	78	41	5	6	11	36	1	5	11	4	9	12	29	11	6	2268

Figure 12: Confusion matrix result for holistic model.



C3

0000021571

BAR CODE

89-030
ACCESSION NUMBER

ISSUED TO

PHILIP MORRIS, U.S.A.
RESEARCH CENTER
RICHMOND, VIRGINIA

PROJECT NUMBER: 6908

PROJECT TITLE: SMOKE CONDENSATE STUDIES

TYPE REPORT: □ ANNUAL □ SEMIANNUAL □ COMPLETION ☒ SPECIAL

DATE: June 21, 1989 PERIOD COVERED: January - October, 1988

REPORT TITLE: Studies of Tobacco-Specific Nitrosamine Pyrosynthesis: Past, Present and Future

WRITTEN BY: W.R. Morgan
W. R. MORGAN

SUPERVISED BY: R.D. Kinser
R. D. KINSER

APPROVED BY: C.K. Ellis
C. K. ELLIS

THIS REPORT IS CONFIDENTIAL TO THE BUSINESS OF THE COMPANY. IT HAS BEEN ASSIGNED TO YOU. IT IS NOT TRANSFERABLE AND MUST NOT BE PHOTOCOPIED.

IF THE REPORT HAS SERVED ITS PURPOSE AND IS NO LONGER NEEDED, PLEASE RETURN IT TO THE CENTRAL FILE AT THE RESEARCH CENTER FOR RECORD KEEPING AND DESTRUCTION.

2072826986

(a) Scientific Report

CONFIDENTIAL SPECIFICATION CHANGE NOTICE

BRAND: CENTURY Light 100 Menthol

ITEM - ID: (1): 2020104
(2):
(3):

DATE: July 9, 1986

EFFECTIVE DATE:

CHANGE PREPARED BY: G. W. Fearrington

COST CHANGE: YES ( ) NO ( )

PHASE III: ( ) FINAL: (X)

SUB-BRAND NO. FROM: 2723AD TO 2723AE
FROM: TO
FROM: TO

TYPE OF CHANGE: Increase menthol concentration in top dressing to achieve pack menthol target.

PURCHASED MATERIALS:	OLD ITEM - ID (Comm. Code)	NEW ITEM - ID (Comm. Code)
Plug Wrap		
Cigt. Paper		
Tipping		
Other:		

BASIS FOR RECOMMENDATION: Product has been below pack menthol target.

PURPOSE OF CHANGE: To achieve pack menthol target.

51101 3251

Product Acceptance Committee Approval - Date Approved: 7/18/86

R&D Director: W. M. Dufour for RLW
R&D Prod. Stds. Director:
Group R&D Director:
Marketing Director:

Prod. Oper. Plan. Director:
Q. A. Director:
Other:

R & D	MANUFACTURING	COMPTROLLERS	Q.A.	PURCHASING
B. V. Hardin R. L. Willard H. E. Guess Brand Mgr-2 W. T. Morgan Specification File J. W. Davis M. E. Stowe R. G. Shore	D. M. Underwood F. M. Lockamy W. E. Alderman S. T. McCollem Plant Manager - A. J. Schindler E. D. Arledge G. L. Aldridge	A. V. Dale J. W. Everhardt	R. L. Morgan R. L. Cornwell O. K. Crouch D. F. Beeson T. A. Hall QA Plant Manager - M. L. Johnson S. L. Reavis	D. B. Self R. L. McGee

(b) Form

Figure 13: Comparison between Scientific Report and Form examples.

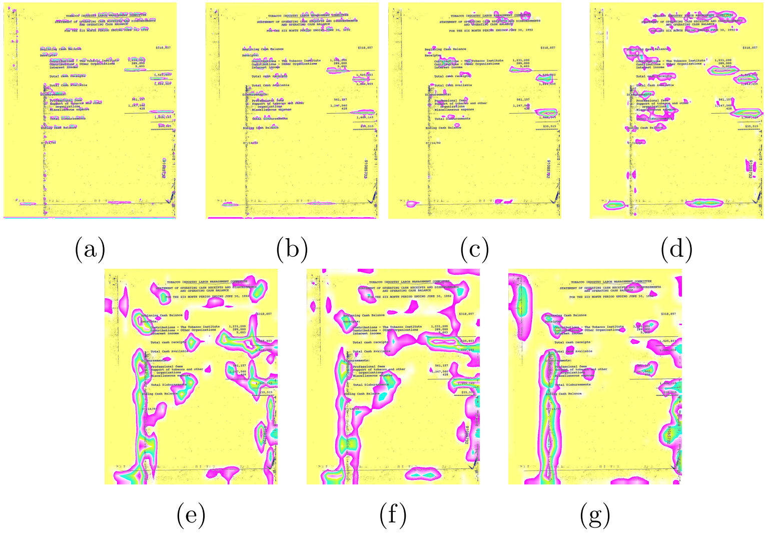
5.4 RVL-CDIP Simple Version Training

With the intent of simplifying the problem to the subset of business related documents, a model with half the number of classes from RVL-CDIP was trained and the results can be seen in Table 3. The results obtained for the validation dataset was 4 percentage points higher than the previous and for the test they were 3 percentage points higher.

From Figures 14 and 15, we can compare the outputs from a budget document example that was classified wrongly in the network trained with the full RVL-CDIP dataset and

	Precision	Recall	F1-Score
Validation	91.5	91.5	91.5
Test	91.4	91.4	91.4

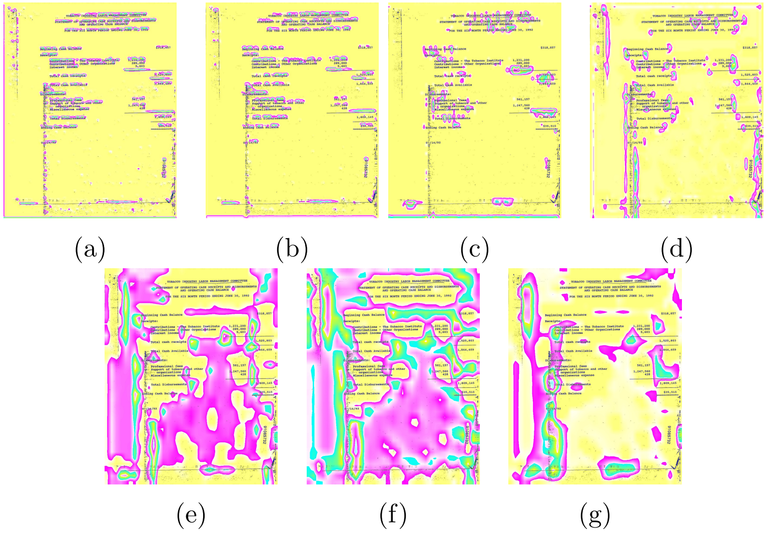
Table 3: Results for VGG16 trained on the simple version of the RVL-CDIP dataset.



(a) (b) (c) (d)

(e) (f) (g)

Figure 14: Examples of network output from the full RVL-CDIP dataset training.



(a) (b) (c) (d)

(e) (f) (g)

Figure 15: Examples of network output from the simple RVL-CDIP dataset training.

that the network trained with the simple version of the dataset correctly classified. The differences are very subtle, but we can see some few differences in areas of focus between them, as well as those areas being slightly larger on the simple version.

Since the differences are minor, one can theorize that the real differences is in how the later fully connected layers interpret the extracted data to make the classification. With a more limited number of classes, the classification is made simple, so it is safe to say that the bulk of the performance difference is originated in that area of the network.

5.5 Dom Rock's Dataset Training

Upon training VGG16 with the Rom Rock's dataset without data augmentation, we had an accuracy of 95%. That is a good result, however, if we weight in the fact that the classes were very unbalanced and there were not enough examples to train this model on, it is safe to say that this model was not robust enough to be used in a real world scenario, since there probably were several document examples that it would struggle to classify properly since it has seen so few examples.

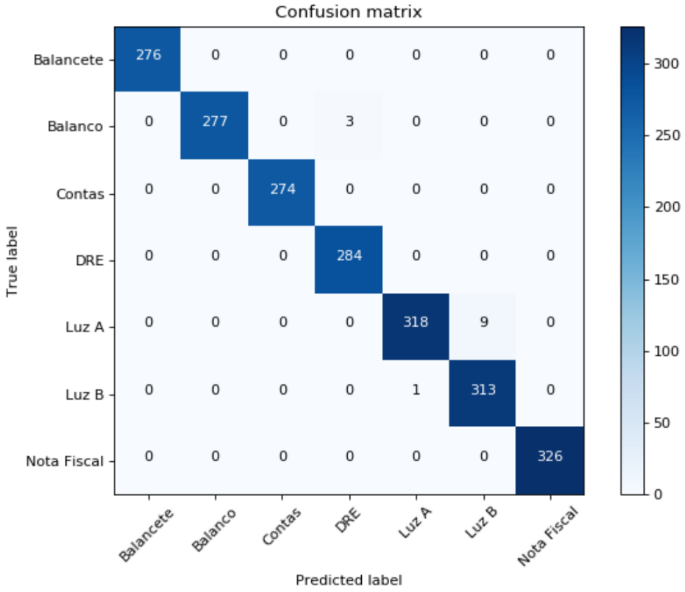
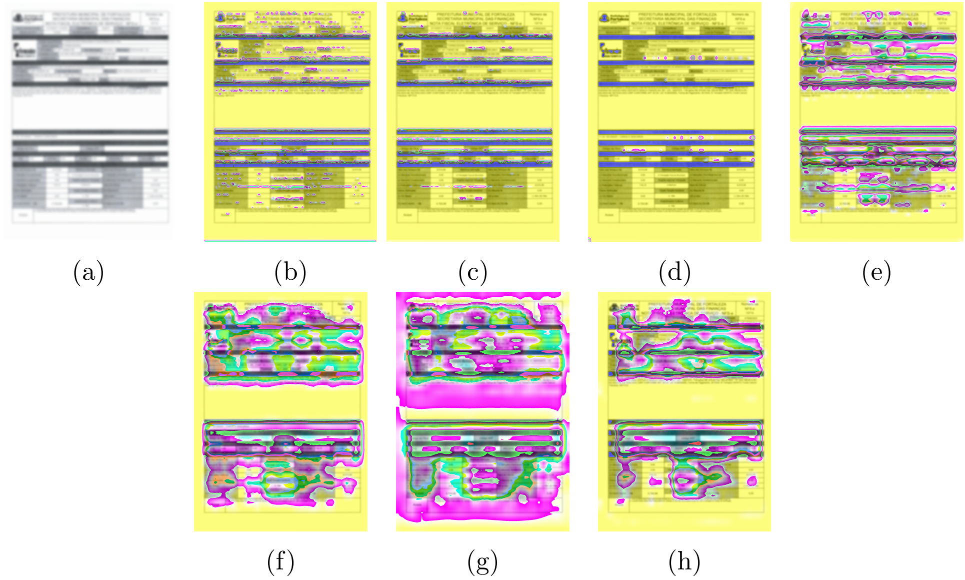


Figure 16: Confusion matrix result for Dom Rock's test dataset.

After augmenting the data, the results were much better, scoring over 99% on both validation and test dataset. It is still possible to argue that the model might not perform great if we give it an example different enough from what it has seen because it is probably too specialized on classifying the examples in the dataset. However, with data augmentation we enforced the model to learn to classify no matter how bad the image might be, even if it cuts some section of the document due to bad scanning. Figure 16 shows the confusion matrix for the test results and one can verify that the only errors that occurred were between electrical bills A and B and profit and loss statement with balance sheet. Those errors were expected since the two types of electrical bills are very similar to each other and the profit and loss statement and balance sheet also have very similar structure.



(a) (b) (c) (d) (e)

(f) (g) (h)

Figure 17: Examples of network output with an invoice input.

Figure 17 shows an example of a heatmap output from the first 7 filters from the network. It shows clearly that the model has learned to extract the layout of the document, seeing that it focuses on the horizontal shapes defined by the document and expands out from it to extract more general features from the layout, such as the area where the text is located.

6 Conclusions and Future Work

This work proposed to take current document image classification techniques and apply them to the more specific use case of business-like documents and it has been successful at that. While it is good to have a model that is versatile enough to classify a wide range of documents, it will not be often that it will be needed to classify both an advertisement and a scientific report. By shrinking the applicable data that it is used, we were able to create models that were more efficient and specialized in one very useful use case. A simple tool was created for people with zero knowledge about the underlying system to use it easily, giving this project a practical application outside the research world.

Different machine learning and data processing techniques were applied to achieve the results. Topics such as transfer learning, data augmentation, ensemble and convolutional neural networks were learned during the process of building this project.

The field of document classification and analysis keeps expanding, with Amazon's Textract [14] being one of the latest advancements in the field. Although that work is related to the classification problem, analyzing and extracting document data are also relevant to create a useful tool in real scenarios.

In addition, upon gathering more data from Dom Rock's archives, it will be possible to create a more robust model to solve the problem. It will also stimulate the use of more advanced techniques seen in this work such as stacked generalization ensemble with region specific training. Since the dataset was small, the results with only a holistic model were high enough that those techniques we not needed.

References

[1] O. Augereau, N. Journet, and J.-P. Domenger, “Semi-Structured Document Image Matching and Recognition,” in *Document Recognition and Retrieval XX*, vol. 8658. International Society for Optics and Photonics, 2013, p. 865804.

[2] H. Rawlani, “Visualizing VGG16 Convolutional Neural Network using Keras,” 2019, https://github.com/himanshurawlani/convnet-interpretability-keras.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in *IEEE Computer Vision and Pattern Recognition*, 2009.

[4] A. Das, S. Roy, and U. Bhattacharya, “Document Image Classification with Intra-Domain Transfer Learning and Stacked Generalization of Deep Convolutional Neural Networks,” *CoRR*, vol. abs/1801.09321, 2018.

[5] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” in *3rd International Conference on Learning Representations*, San Diego, CA, USA, May 2015.

[6] “Dom Rock,” 2019, https://www.domrock.com.br.

[7] “Convolutional Neural Network Course,” 2019, https://brilliant.org/wiki/convolutional-neural-network/.

[8] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going Deeper with Convolutions,” *CoRR*, vol. abs/1409.4842, 2014.

[10] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, “Grad-CAM: Why Did You Say That? Visual Explanations from Deep Networks via Gradient-based Localization,” *CoRR*, vol. abs/1610.02391, 2016.

[11] “Visual Interpretability for Convolutional Neural Networks,” 2019, https://towardsdatascience.com/visual-interpretability-for-convolutional-neural-networks-2453856210ce.

[12] A. W. Harley, A. Ufkes, and K. G. Derpanis, "Evaluation of Deep Convolutional Nets for Document Image Classification and Retrieval," in *International Conference on Document Analysis and Recognition*, 2015.

[13] D. Dimmick, M. Garris, and C. Wilson, "Nist Structured Formsreference Set of Binary Images (SFRS)," 1991, http://www.nist.gov/srd/nistsd2.cfm.

[14] "Amazon Textract," 2019, https://aws.amazon.com/pt/textract/.