Universiteit Leiden
The Netherlands

Opleiding Informatica

Visual classification of e-discovery images with neural networks

Marc van Duyn

Supervisors:
Suzan Verberne & Johannes C. Scholtes

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

August 22, 2018

Abstract

We investigate whether transfer learning can be applied to images related to eDiscovery, using a deep convolutional network named inceptionV3, trained on a large dataset of general objects for classification for the image net recognition challenge. We use transfer learning to retrain the model because we were not capable to conventionally train a neural network with the insufficient amount of labeled data. Also, we use multiple augmentation techniques to enlarge our training dataset. We investigate which configuration of techniques results in the best performing model and shows the most satisfactory results for classification of eDiscovery images, keeping in mind the new target categories contrast significantly from the original target classes. Nevertheless, we have obtained a top-1 accuracy of 75.3%. Also, the top-5 error rate was 2.9%. This allowed us to give a general advice on which techniques to use and whether or not transfer learning can be applied for recognition of categories very distinctive from the original tasks. We show the reader how we set up our models and show some examples of how we implemented them in our applications.

Contents

Chapter 1

Introduction

With the recent general data protection regulation(GDPR) law of the European union [Eur16] there is an increasing demand for the need of detecting and protecting personal data. Tools, such as ZyLAB ONE eDiscovery, that help users with the electronic aspect of identifying, collecting and producing electronically stored information(ESI) in response to a request for production in a lawsuit or investigation help organizations to become GDPR compliant. This process of analyzing data as a response to a litigation, government investigations, or Freedom of Information Act requests, where the information sought is in electronic format (often referred to as electronically stored information or ESI) [Cas], is called eDiscovery. ZyLAB is a global leader in the development of award-winning search, text mining, machine learning, and other artificial intelligence technologies. ZyLAB ONE integrates artificial intelligence and data science to accelerate the eDiscovery process. Visual classification of eDiscovery related images will greatly improve this process because many documents contain visual data. With automatic visual classification through deep learning techniques, it wants to enable its users to visual classify large volumes of documents in an automatic manner to help them in their eDiscovery process.

The main part of this research is the evaluation whether features extracted from an eDiscovery image data set to retrain a pre-trained convolutional neural network, can successfully be re-purposed in recognizing eDiscovery related images. We are using the Inception model V3 [SVI+15] as our pre-trained model in conjunction with the tensorflow framework [ABB+15] for retraining and evaluating. We chose this model because it proved to have high accuracy and of the publicly available source files [SVI+15]. The technique we are using to re-purpose an already trained model for our own categories is called transfer learning. This is a technique that reuses the feature extraction capabilities from a model already trained on related tasks and reusing it for our own categories by building a new classification layer on the existing model.

The goal of this research is to develop a neural network able to classify images related to eDiscovery. In order to reach this goal, we will do experiments where we will alter our training dataset with different techniques to construct the best performing model. The research question of this thesis is: What is the basic recognition quality of eDiscovery images from a retrained Inception V3 model by retraining it on an eDiscovery related

1.1 Thesis Overview

- Chapter 1 Contains the introduction, overview of the thesis.
- Chapter 2 Discusses related work.
- Chapter 3 Outlinge of the research.
- Chapter 4 Overview of obtained results.
- Chapter 5 Some concrete implementations outlined.
- Chapter 6 Conclusion of the obtained results.

data set?

The research question of this thesis is: *"What is the basic recognition quality of eDiscovery themed images from a retrained Inception V3 model by retraining the model on an eDiscovery related data set?"*

The author of this bachelor thesis would like to thank Suzan Verberne supervisor of Leiden University and the supervisors Johannes C.Scholtes and Jeroen Smeets of ZyLAB technologies for their valuable help.

Chapter 2

Related Work

2.1 Convolution neural networks

Convolutional neural networks(CNN) have been researched for a long time in computer vision. These networks go back to the 1980s and have proved they could be used for a wide variety of visual classification tasks. The main idea is that, through several layers of filtering and subsampling, a network can learn to distinguish between local and global features of any image. A filtering layer can help to detect local features, where the same filter applied to a heavily sub-sampled image can help in recognizing global characteristics. This allows a CNN to recognize the same objects of a figure with similar characteristics in an unknown location inside an image, whereas a normal neural network is only capable of recognizing absolute positions of pixels which are not relative to an environment.

Early developments showed that these networks successfully recognize digits from handwriting([LBD+89]). As a result, these networks were used for the recognition of handwritten zip code digits provided by the U.S. postal service. Recent developments in CNN models have caused that they are now capable of classifying with high accuracy datasets with millions of images. Since then convolution neural networks have been widely used in large-scale image recognition tasks, such as the ImageNet recognition challenge [DDS+09]. A network named GoogleNet (Inception V1) was a big milestone in the research in the visual classification with CNNs. This model was the first to introduce the inception deep convolutional architecture [SLJ+15]. Later this architecture was refined to improve accuracy, first with the introduction of batch normalization, which came to be known as Inception V2 [IS15] and after that with additional factorization ideas which will be referred to as Inception V3 [SVI+15].

2.2 Transfer learning

Transfer learning is a well-known technique in machine learning, with the earliest cited work coming from Lorien Pratt who did a research about the discriminability-based transfer (DBT) algorithm in 1993 [Pra93]. This research proved that the DBT algorithm demonstrated significant learning speed improvement over randomly initialized neural networks without compromising accuracy. Learning with pre-made models has also a long history with other machine learning concepts. Caruana [Car98] with the book Learning to learn describes multi-task learning, which is a technique whose main goal is to improve generalization performance by leveraging the domain-specific features contained in the training signals of related tasks. A year earlier Thrun [Thr96] did a study where he investigated whether learning with already obtained knowledge can be used on learning new things. He showed that learning approaches with past knowledge produce more accurate results from less training data, by their ability to transfer knowledge across. This technique of learning across task using deep representations has been a well known([RBLP07], [MDG$^+$12]). But most of the time this was only applied to relatively small datasets, such as CIFAR and MNIST, whereas large datasets show significant lesser results [LRM$^+$12]. Only recently researchers managed to use transfer learning on deep convolutional networks applied with very large data sets where the results proved satisfactory [DJV$^+$13]. This has enabled us to leverage a large labeled database to retrain a CNN and learn features using its representational and generalized ability to perform semantic visual recognition's tasks.

2.3 Data

To evaluate and train the model we will use a large data set with eDiscovery related images. In the process of making this dataset, we will make use of different augmentation techniques. These techniques are a common practice in deep learning [SSP]. Image augmentation has proved to be highly effective to artificially increase the size of the data set. [Die15]

2.4 eDiscovery

Automated eDiscovery processes to prioritize and select documents for review have shown to be highly effective and great cost savers in comparison with exhaustive manual reviewing [GC10]. Data retrieved and examined for forensic evidence from laptops, smartphones, tablets, flash drives, smart watches and other electronic devices are rapidly changing the legal landscape. Also the general data protection regulations(GDPR) of the European union [Eur16] increases the demand for the need to detect and protect personal data. Counseling from forensic examiners which offer end-to-end eDiscovery services, including collection, reviewing and production of documents with the inclusion of different forensic tools are now highly recommended for organizations working with personal data [Dea17].

The images related to eDiscovery contain mostly text and symbolic characteristics, for example, id-cards have citizen service numbers and symbols unique to the characteristics of the specific card. Text recognition in natural images is a challenging problem. But with the power of large multi-layer neural networks, CNN's have proven to be highly effective in detecting text and recognizing characters [WWCN12]. Also, the past developments showed that extracting text and symbols from images could be done with neural networks. Segmentation and classification of text out of images with the goal of converting the image to text, have been archived with a layered feed-forward neural network [ITW93].

Chapter 3

Methods

3.1 Tools

The research is mainly done with tensorflow, "an open source machine learning framework for everyone" [ABB+15], and the Python programming language. These tools are used purely for the creation of the models. For the creation of datasets we made an image scraper with the python library beautifullsoup. Beautifullsoup allowed us to parse the HTML pages and extract the image links. We collected all these links and downloaded them individually. Further, we applied images augmentation and preprocessing with imgaug, a python library that helps with augmenting images for machine learning projects. In this research we will also describe how we implemented the created models, this was done with the programming language C# and with Flask, "Flask is a microframework for Python based on Werkzeug, Jinja 2 and good intentions." [Ron10].

3.1.1 Data collection

We constructed an URL with urllib, a python library that provides a high-level interface for fetching data across the World Wide Web. The library can only open URLs, thus search and retrieve operations needs to be delegated. To parse and retrieve the image links from the HTML source, we use beautifullsoup, a Python library for pulling data out of HTML and XML files. It comes with built-in functionality for navigating, searching, and modifying the parsed contents. After locating the image links, we filter on JPG format files and download them individually.

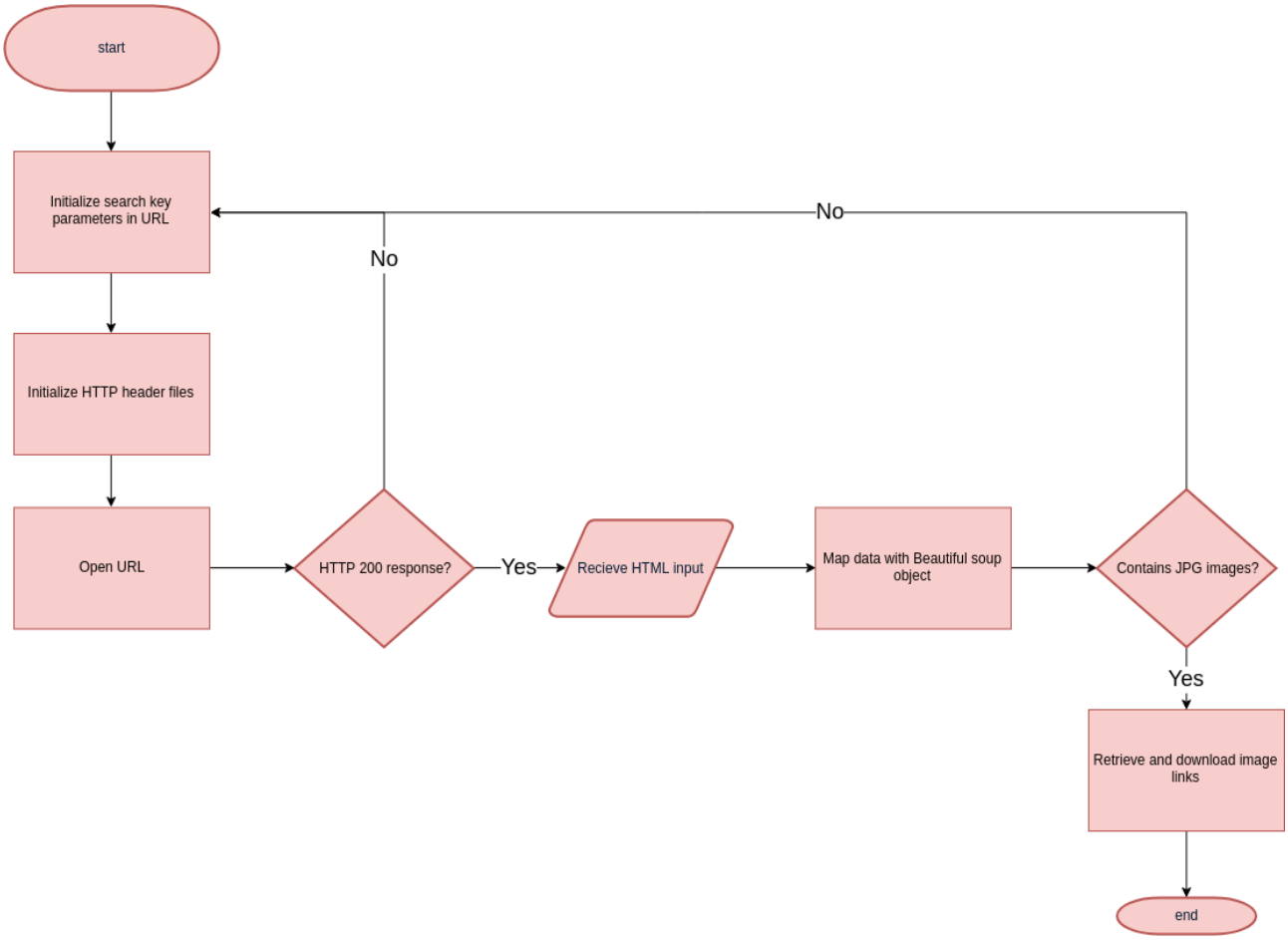


Figure 3.1: Data collection process

3.2 Inception V3 model

The model is a convolution neural network, which implies that it makes the explicit assumption that the input is an image, and has an input shape of $299 \times 299 \times 3$. The inception architecture has its namesake from the use of mini models for decision-making tasks inside the bigger model [SLJ+15]. These tasks define which type of convolution the model is going to make at each layer, with convolutions being a building block of a layer doing computational operations. Additionally, this architecture allows the model to recover both local features via smaller convolutions and high abstracted features with larger convolutions. This idea of making decisions on which convolution to make is a solution to the problem of having salient parts in an image varying in size.

Figure 3.2: Images of passport's occupying different space sizes, showing the variation in information locations.

By filtering with multiple sizes operating on the same level, would allow the model to chose the right kernel size for the convolution operation to be applied. For example inception V1 applied it inceptions with 3 different sizes of filters ($1 \times 1, 3 \times 3, 5 \times 5$) with additional max pooling. Then the results would be concatenated and sent to the next inception module in the architecture. Also, the use of dimension reduction, by adding a 1×1 filter helped the inception model being less computationally expensive than comparable models.

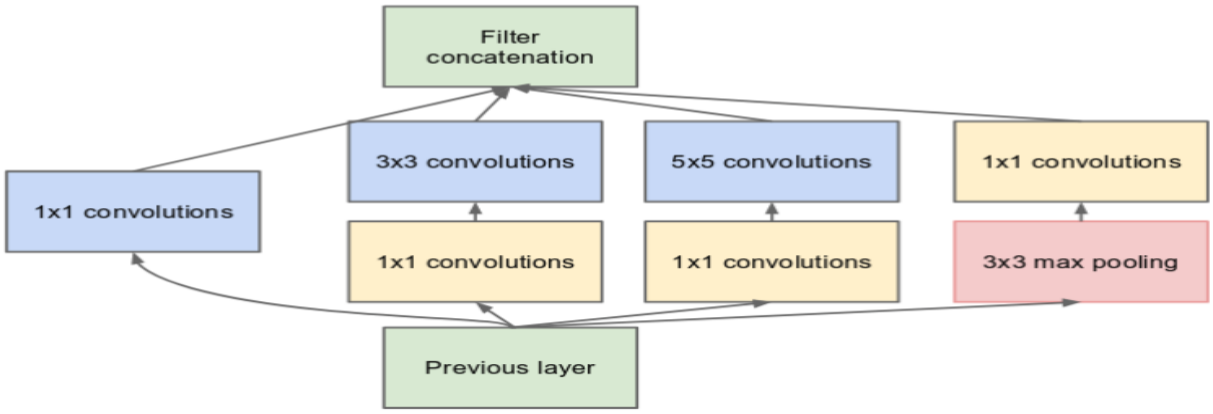


Figure 3.3: inception v1 module [Chu17]

Inception V3 and inception V2 where upgrades on the original V1 architecture focused on increasing the accuracy and reducing the computational complexity. This was achieved by changing the representational

bottleneck, whereas V1 reduced the dimension too much, causing loss of information. Also, V2 and V3 made use of smart factorization methods, making the convolutions more efficient and improving the computational complexity. V3 incorporated all the features of V2 with additional improvements contributing to the overall accuracy of the model. It improved the training performance by using an RMSProp optimizer, to speed up batch learning, and batch normalization in the auxiliary classifiers to prevent zero activation and vanishing gradients. Also, it added a 7×7 factorized filter to the inception modules, improving the overall accuracy. Label smoothing was also applied, which resulted in a more evenly distribution of the confidence, which resulted in fewer cases where the network becomes too confident about a class preventing over-fitting on training data.

The inception V3 model is publicly available. Additionally, it is one of the best performing models on the image net classification challenge. These were determining factors for us to use this model.

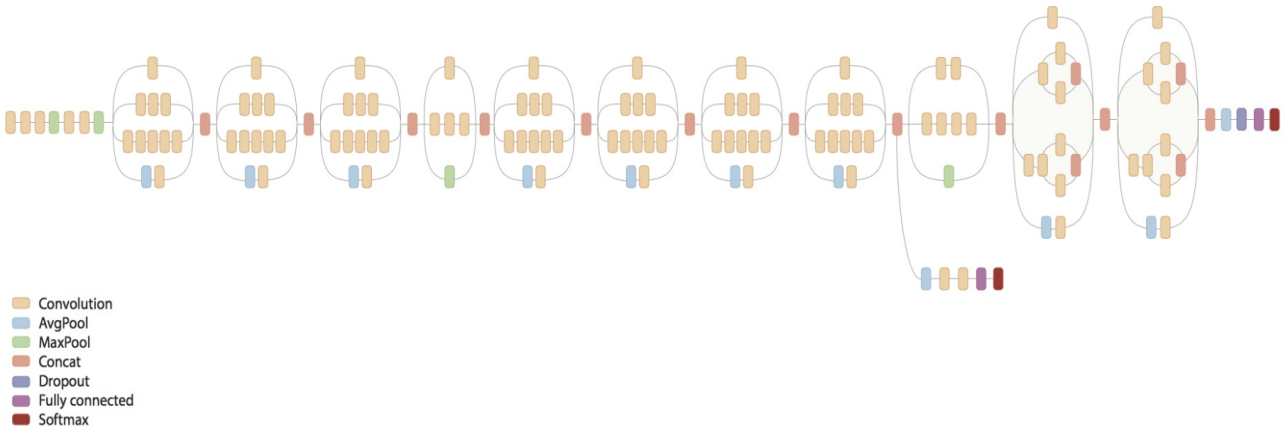


Figure 3.4: inception v3 structure [Chu17]

3.3 Data

ZyLAB provided a data set containing a total of 430 images distributed over 3 categories. The final model needed tot classify a total of 17 categories. These categories where: If the task at hand was to train a model to

Class	Description
Check	Document that orders a bank to pay a specific amount of money from a person's account to the person in whose name the cheque has been issued.
Credit card	Payment card issued to users (cardholders) to enable the cardholder to pay a merchant for goods and services based on the cardholder's promise to the card issuer to pay them for the amounts so paid plus the other agreed charges [Cas].
Driver license	Official document permitting a specific individual to operate one or more types of motorized vehicles, such as a motorcycle, car, truck, or bus on a public road.
Expense form	Template to list expenditures.
Facebook	Social media platform.
ID card	An identity document (also called a piece of identification or ID, or colloquially as papers) is any document which may be used to prove a person's identity.
Mainframe terminal	A computer terminal is an electronic or electromechanical hardware device that is used for entering data into, and displaying or printing data from, a computer or a computing system [Law98].
Money (coins and cash)	Money is any item or verifiable record that is generally accepted as payment for goods and services and repayment of debts in a particular country or socio-economic context [Mis07].
News paper	Document that is periodical published containing articles with information about current events.
Office Memo	Template to share important information with members inside an organization.
Passport	A travel document, usually issued by a country's government, that certifies the identity and nationality of its holder primarily for the purpose of international travel [CC08].
Presentations	The process of presenting a topic to an audience. It is typically a demonstration, introduction, lecture, or speech meant to inform, persuade, inspire, motivate, or to build good will or to present a new idea or product.
Whatsapp	Messaging application.
Word processor	Computer program or device that helps user with creating documents, style editing, formatting and with the output of text.
Spreadsheet	Application that analysis and stores data in tabular form.
Email	Electronic mail is a method for exchanging messages between electronic devices.

classify 3 categories, the 430 images would be more than enough. But we had to create a model capable of successfully distinguishing the 17 categories. In order to create the machine learning models, we had to obtain enough images to cover all the categories. It is self-evident that the more data a machine learning algorithm has access to, taking into account overfitting and data quality, the more effective it is going to be. Therefore it is of great importance to have a good quality data set with enough images to create an accurate model. The image scraper enabled us to get approximately a set of 80 images, specified on 'jpg', per individual search key. The images we collected were of poor quality. So in order to create a data set of good quality, we had to review

every image and weed out low quality images. This not only meant weeding out images of low visual quality but also images that where not corresponding to the visual characteristics of the target class. For example scraped pictures of emails contained also emblems of email services such as gmail or hotmail. The resulting data set still contained examples of low quality but good enough to use for training the models.

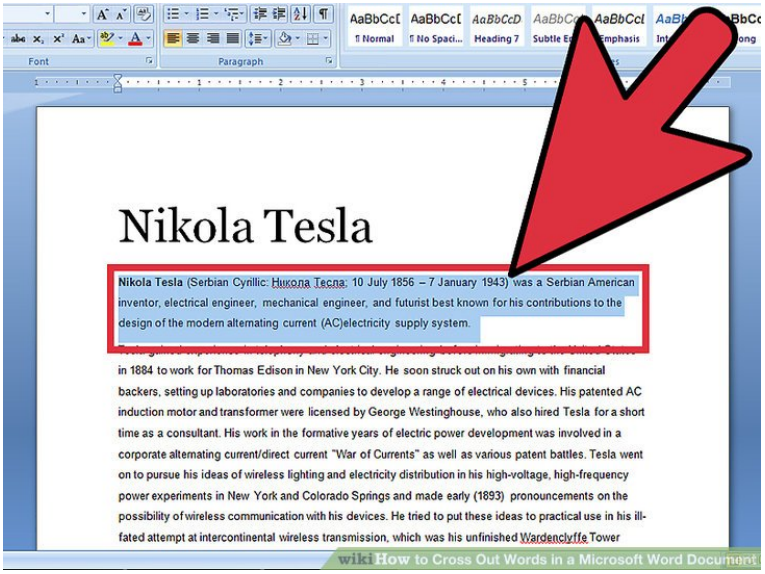


Figure 3.5: Example of low quality image we obtained, where custom markings are applied

For every category, we tried to obtain a diversified set of images. This not only applied to the uniqueness, for example, the visual distinctiveness of the image in comparison with other images contained in the data set, but also the differences in characteristics related to factors such as country, nationality and object shapes. For example, a driver license from Germany has different characteristics in comparison with a driver license from the united states. We generalized some categories, such as credit cards, where we generalized all credit, paying, debit and balance transfer cards and mapped them all to the category credit cards. However, we have split big categories that cover a lot of different physical forms into smaller ones that are more visually distinct, such as money where we split on coins and cash.

To counter the possibility of having a biased training set, we opted for an equal representation of every category in the data set. This means that there should be a similar number of samples per class. When we finished the image scraping we had a total amount of 2425 unique images, with roughly 150 images per category, whereas the maximal amount of images was 167 and the minimal was 111.

3.4 Data augmentation and preprocessing

3.4.1 Augmentation

To evaluate whether different data setups would influence the overall classification results of our models, we created multiple datasets exposed different selections of augmentation techniques. Image augmentation is

the process of taking images that are already in the data set and manipulating them to create many altered versions of the same image. This both provides more images to train on, but can also help expose our network to a wider variety of lighting and coloring situations so as to make it more robust.

The selected augmentation techniques consist for the most part out of techniques that preserve the features representative for eDiscovery themes. Most features in these images are text-based or symbolic. For example text on an id card should not be altered. So augmentation that alters these features too much will lower the overall accuracy. Focusing on traditional augmentation techniques helped us to preserve most of these features. These techniques consisting of cropping, resizing, rotating and flipping the images and are known to be highly successful in training a neural network [PW17]. We find sharpening, contrast normalization and perspective transform also very applicable for preserving a lot of detail and are included in the selection.

(a) original news paper

(b) flipped left-right

Figure 3.6: Standard augmented image

Augmentation enables us to increase the number of training samples without actually using other data sets. Therefore it is a good alternative to combat the high expense of collecting thousands of training images, for example, applying one transformation on every image would double the data.

For curiosity, we wanted to research the effects of retraining a model with a data set made out of heavily augmented images. The symbolic and textual features are most likely lost in the vast part of the dataset, but this would expose our model to a more wide variety of lighting and coloring situations in comparison to models exposed to more traditional data sets. Some of the techniques used for heavy augmentation are:

- Techniques to transform perspective
- Color saturation techniques
- Pixel drop out techniques
- Standard augmentation techniques
- Contrast augmentation techniques

- Blurr techniques such as gaussian blur or simplex noise
- Light augmentation techniques

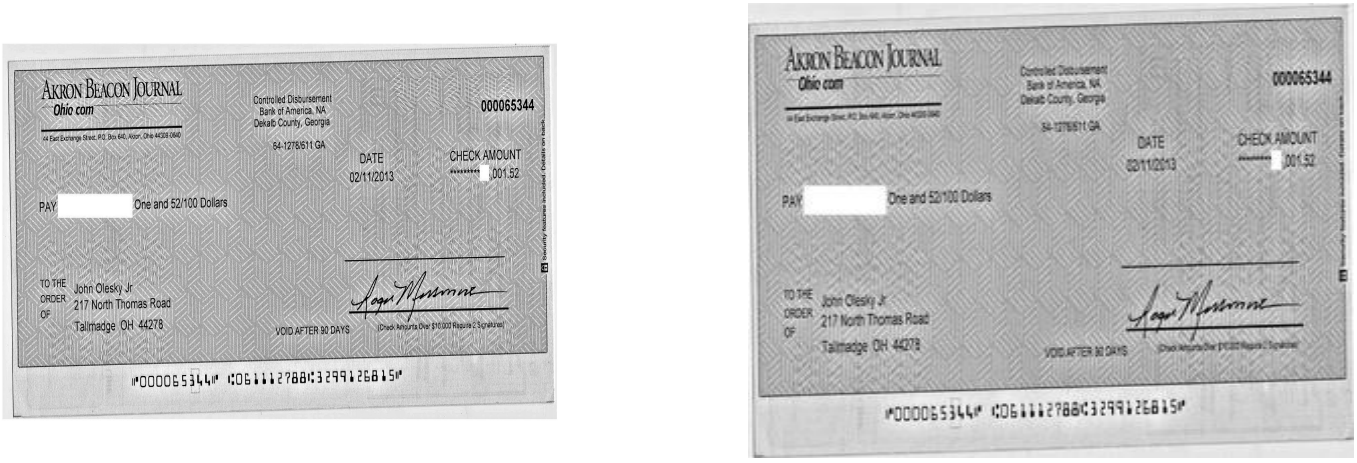
Figure 3.7: Heavy accumulated augmentation on a ID card

3.4.2 Preprocessing

We normalized all the images to a standard and consistent format. Without normalization applied the images would have different sizes and resolutions, so in order to be usable for transfer learning, we normalized the size across the whole data set. Inception V3 needs images of the dimension 300 $\times$ 300 pixels, so we first resized every image and re-scaled them to fit the desired ratio. Large images with high resolutions will lose a lot of detail with resizing, but the dimension necessary by the model still allowed us to preserve a lot of the features. Especially with eDiscovery images, where a lot of features are text-based or symbolic, higher resolution will greatly improve the accuracy. Before we applied to resize, we first removed all the white space of the images. Removing white space will help us in preserving more detail, with regards to the redundant white space when applying to resize.

3.5 Retraining

The retraining is done with 4000 training steps. With each step randomly selecting 10 images from the data set, then for each image, a bottleneck value is calculated. Bottlenecks, synonymous for classification layer, will be used to form a new penultimate layer. This layer will produce a set of values used to distinguish the new classes, keeping in mind that it can create a meaningful and compact summary based on the recognition ability obtained from the training on our own data set. The reason the retraining of the final layer works is the fact that the model general recognition ability is good enough to distinguish general shapes to form a



(a) original

(b) whitespace removed and resized

Figure 3.8: preprocessing: whitespace removal and resizing

foundation for visual classification of new objects. Throughout the process of training and validation, accuracy and cross entropy will be monitored, and the training will be finished with a final test, done with a dataset containing randomly selected data samples separated from the training and validation sets.

3.6 Evaluation

In order to get a general overview of the classification accuracy of each model and compare them against each other, we have selected a number of metrics that will help us to determine how well the model will perform in a general use case. For the overall accuracy, we used a top-1 accuracy test and the top-k error rate. The top-k error rate will be evaluated on the top 1, 3 and 5. To evaluate the performance of the ranking, we will use the mean reciprocal rank and mean average precision. These metrics are in line with previous studies ([DDS+09] & [GBJ17]), this also allows us to compare the results to other studies, which will give us a good indication of the overall performance for visual classification tasks. More precise evaluation will be done with confusion matrices and precision-recall curves. The dataset for evaluation consists entirely out of unique images not used in the training process. This data set is fixed and used for every model, with most of the images being self-made.

3.6.1 Top-k error rate

The top-k error rate is a well-known metric to evaluate the classification quality of neural networks and is widely used in the image net classification challenges. This metric is the fraction of images for which the correct label is not among the k labels most probably by the classification of the model. Because the inception V3 model makes a prediction for each label in a multinomial distribution, the summation of prediction results to 100%. This means that a high percentage score corresponds to a greater confidence in the made prediction. In the case of a top-1 error rate, we check if the class with the highest probability is the same as the ground

truth. In the case of a top-3 error rate, we check if the target label is within the top 3 predictions. The resulting error rate is the score computed as the times a predicted label matched the target label divided by the number of data-points evaluated. Here means a higher probability for the error rate a worse overall classification accuracy.

Top-1 error rate

- All first predictions are mapped directly to the set of **top-1 predictions A**
- All correct first predictions are mapped directly to the set of **correct labels B**
- Generate $\mathbf{C} = \forall x \epsilon \mathbf{A}, y \epsilon \mathbf{B} : x \neq y$
- error rate $= \frac{|\mathbf{C}|}{|\mathbf{B}|} \times 100$

3.6.2 Precision, recall, average precision

Precision (P) can be seen as the exactness that defines the obtained predictions. A high precision means in our case that the model returns substantially more relevant classes (true positives) than irrelevant ones (false positives). For example, classes that are difficult to distinguish such as id cards and driver license could show low overall precision. Precision(P) is defined as the number of true positives $(\mathbf{T_p})$ over the number of true positives $(\mathbf{T_p}) +$ the number of false positives $(\mathbf{F_p})$, $\mathbf{P} = \frac{\mathbf{T_p}}{\mathbf{T_p}+\mathbf{F_p}}$.

Recall(R) is the measure of completeness of the classification. Lack of correct classifications result in a low recall. High recall means that a model returned most or all of the relevant results. This metric is defined as the number of true positives $(\mathbf{T_p})$ over the number of true positives $(\mathbf{T_p}) +$ the number of false negatives $(\mathbf{F_n})$, $\mathbf{R} = \frac{\mathbf{T_p}}{\mathbf{T_p}+\mathbf{F_n}}$

Typically the metrics of precision and recall are inversely related, for example as precision increases, recalls falls and vice-versa. In order to obtain the best model, a balance needs to be achieved between the two. To evaluate the performance of a model related to these metrics, we will make use of a precision-recall curve where we will plot the relationships between them for every class.

Average precision (AP) is very similar to mean reciprocal rank, it summarizes a precision-recall curve as the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the weight.

$\mathbf{AP} = \sum_n (\mathbf{R_n} - \mathbf{R_{n-1}})\mathbf{P_n}$

This score based on the ranking of classifications. This means that the ranking of predictions influences the AP score, and a correct classification ranked highly, say a top-3 rank, will lead to a higher score than a correct classification on rank 4. This metric is useful because it characterizes the relationship between precision and recall. The idea of AP is conceptually viewed as finding the area under the curve of the precision-recall graph.

3.6.3 Mean Reciprocal rank (MRR)

The mean reciprocal rank is a metric for evaluating a process that produces a ranked list of possible responses ordered on probability of correctness. The MRR considers the ranking of the labels by calculating the reciprocal rank (RR) for a query of response. The RR is 1 for the first place $1 \div 2$ for the second place, $1 \div 3$ for third place and so on. The MRR is then the mean of all the RR's resulted from the queries.
MRR = $\frac{1}{|\mathbf{Q}|} \sum_{i=1}^{|\mathbf{Q}|} \frac{1}{rank_{\mathbf{i}}}$

In our case:

Correct label	Classification results	Rank	Reciprocal rank
ID card	Credit card, Driver license, ID card	3	$1 \div 3$
News paper	Office memo, News paper, email	2	$1 \div 2$
Email	Email, News paper, spreadsheet	1	1

This results in a total MRR of $(1/3 + 1/2 + 1)/3 = 11/18$ or about 0.61. if the ranking was done poorly, the score would be close to 0, a score closer to 1 means a better overall ranking quality.

3.7 Experiments

We set up 3 experiments. In the first experiment, we wanted to analyze if a retrained inception model on a data set containing only unique images, with unique meaning an original image with no augmentation, themed around eDiscovery could obtain satisfactory classifying results. Next, we wanted to evaluate the influence of image augmentation techniques on the classification performance and compare them to a model trained on a data set made out of only unique images. Lastly, we wanted to evaluate whether or not we could obtain a better performing model then one trained on only unique images with the use of the same data set and augmenting it with different augmentation configurations.

3.7.1 First experiment

The main goal of this experiment is to evaluate if transfer learning is applicable to eDiscovery themed images.

We create a dataset with K categories and for every category N training images. Retrain 2 models distinguishable by the use of preprocessing techniques and evaluate whether the obtained results are satisfactory for classifying the chosen categories. Our data set was formed with the eDiscovery categories outlined earlier consisting out of images obtained from ZyLAB and the image scraper. We used resizing and whitespace removal as a preprocessing technique. Also, we wanted to analyze the influence of removing whitespace from images to the classification results.

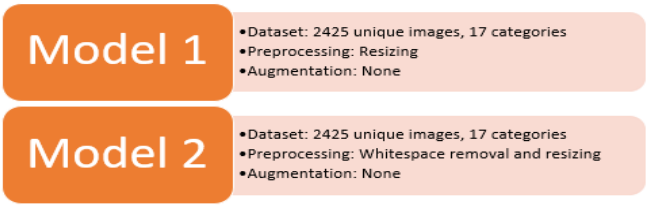


Figure 3.9: Overview of models of expirement 1

3.7.2 Second experiment

The main goal of this experiment is to evaluate the influence of augmentation on transfer learning.

Create a dataset with K categories and for every category N training images. Form 3 models with the following setup.

- Model A: dataset with K categories and N images, size is $K \times N$
- Model B: dataset with K categories and $p\% \times N$ images, size is $K \times (p\% \times N)$
- Model C: dataset is the same as Model B, supplemented with augmentation techniques to the size of the data set of A, resulting in a data set with size $K \times N$.

To evaluate the results you can use the following steps:

- The performance of A is most likely the upper limit.
- The performance of B is most likely the lower limit, because of overfitting and will show worse results than A.
- The performance of C shows the influence of image augmentation, and to what extent it can reach the performance level of A.

In our case, we used the 17 eDiscovery categories en 100 images per category resulting in a data set of 1700 unique images for A. We chose p to be a percentage of 35%, which led to a data set consisting out of 17

categories with every 35 images. C was trained with the data set of B but enhanced with image augmentation to the size of A.

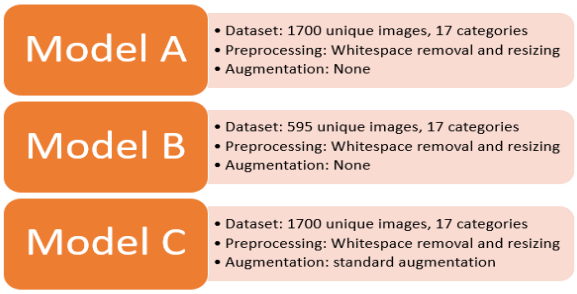


Figure 3.10: Overview of models of expirement 2

3.7.3 Third experiment

The main goal of this experiment is to create the best performing model, with the knowledge obtained from the previous experiments.

Use a base data set to create N data sets differentiated on the combinations of augmentation techniques. Train N models and evaluate them against each other and the standard model trained on the base data set.

We mainly wanted to monitor the effects of standard augmentation techniques, discussed in the data section, and the difference between single and multiple augmentations for the performance of a model. We made a distinction between individual and accumulated augmentation.

- Individual augmentations means that at most one augmentation technique can be applied per augmented image. This means with a selection of K techniques, at most K + 1 images can be formed from one plain image, including the original.
- Accumulated augmentation means that for every image multiple augmentation techniques can be applied to a single image. This means with a selection of K techniques, at most 2 images can be formed, the augmented and the original image.

Also, we created a data set consisting of heavily augmented images.

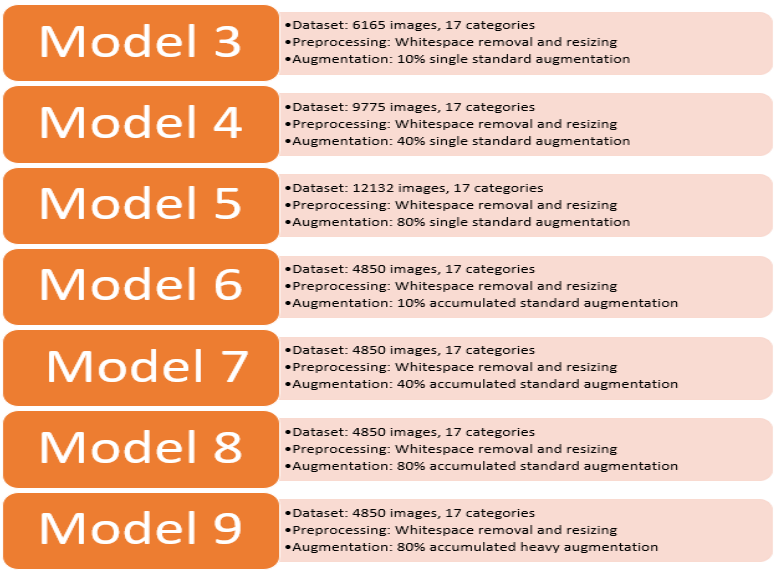


Figure 3.11: Overview of models of expirement 3

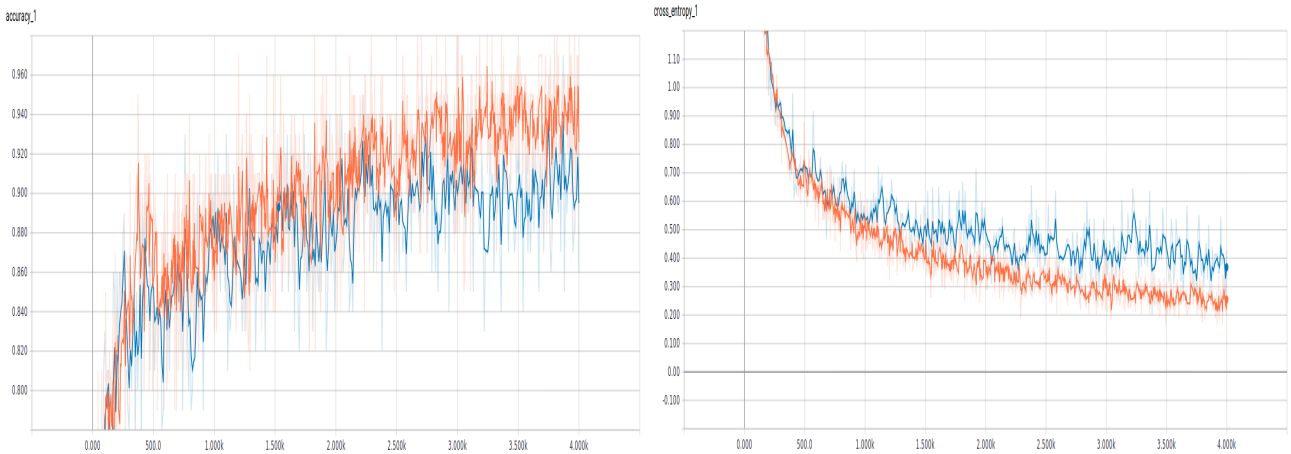
Chapter 4

Results

4.1 Experiment one

4.1.1 Training process model one

The training process showed that the training accuracy stagnated around 90%. The corresponding validation accuracy showed a similar curve. This was a good sign that could indicate that our model was not overfitting due a small dataset. The final random test had a score of 89.6%. Also, the cross-entropy curve demonstrated that it was performing well on the training data, the slope indicates even that there is room for improvement by feeding more training samples.



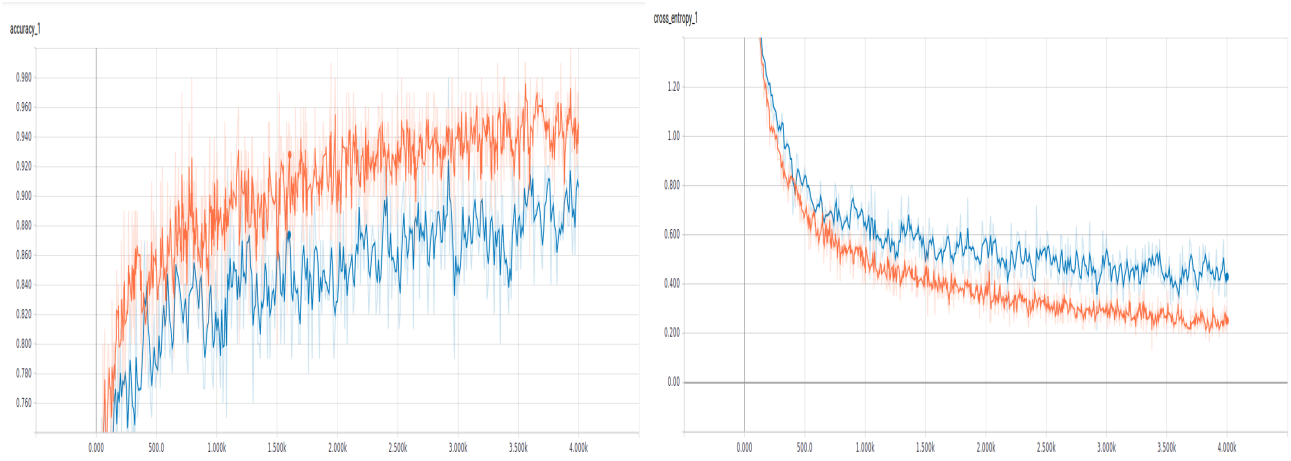
(a) Training and validation accuracy

(b) Cross entropy

Figure 4.1: Training statistics, orange is training and blue is validation statistics

4.1.2 Training process model two

The training process showed that the training accuracy stagnated around 94%, with the corresponding validation accuracy being similar to model 1 around 90%. The final random test resulted in 91.2% classification accuracy. From these training statistics 2 showed slightly better results in comparison to 1. The two statics from the models are very similar, and both demonstrate room for some minor improvements, keeping in mind the cross-entropy curves are declining, but the training accuracy is stagnating, by providing a larger dataset.



(a) Training and validation accuracy

(b) Cross entropy

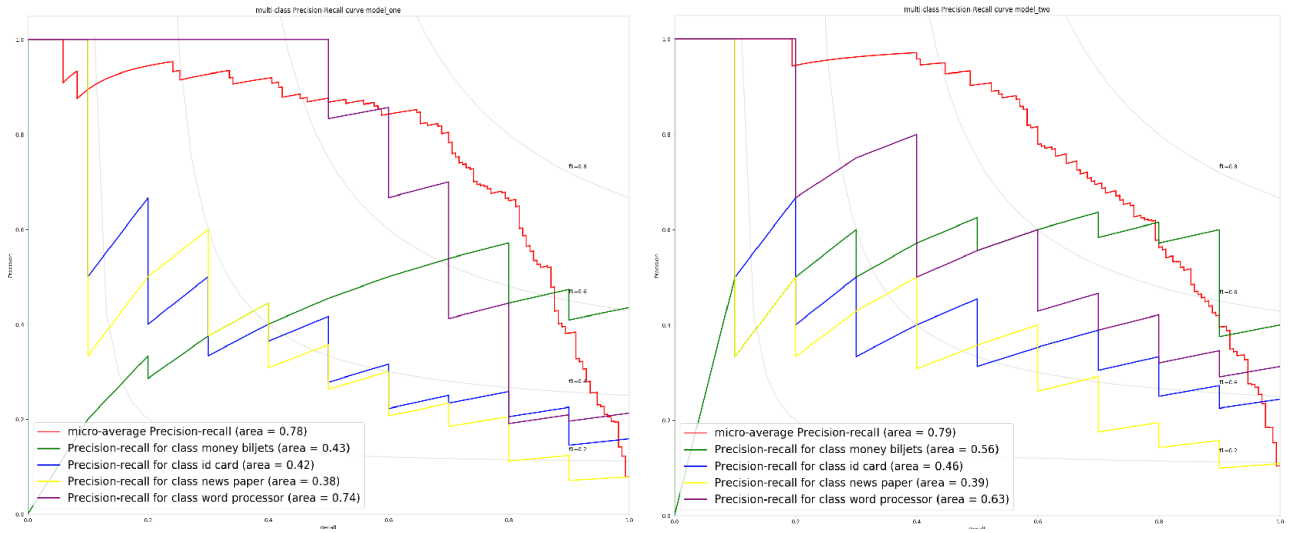
Figure 4.2: Training statistics, orange is training and blue is validation

4.1.3 Evaluation

Model	Top-1 accuracy	Top-1 error	Top-3 error	Top-5 error	MRR	Mean average precision
Model 1	**72.3%**	**27.6%**	**10.0%**	**4.1%**	**0.82**	0.78
Model 2	70.5%	29.4%	**10.0%**	**4.1%**	0.81	**0.79**

The results indicate, basing on the top-1 accuracy scores combined with the ranking statistics, that the performance is good enough for ZyLAB to be used for classification of eDiscovery images. Especially the ranking metrics indicate that an implementation where the top 3 or top 5 predictions are evaluated will perform well at visual classification tasks. The one thing that is standing out, is the fact that 1 performs slightly better than 2. This demonstrates that the preprocessing didn't improve the classification accuracy, one a side note, not every image needed to be preprocessed, there where only a small amount of images that needed removal of white space, thus we are evaluating almost identical models.

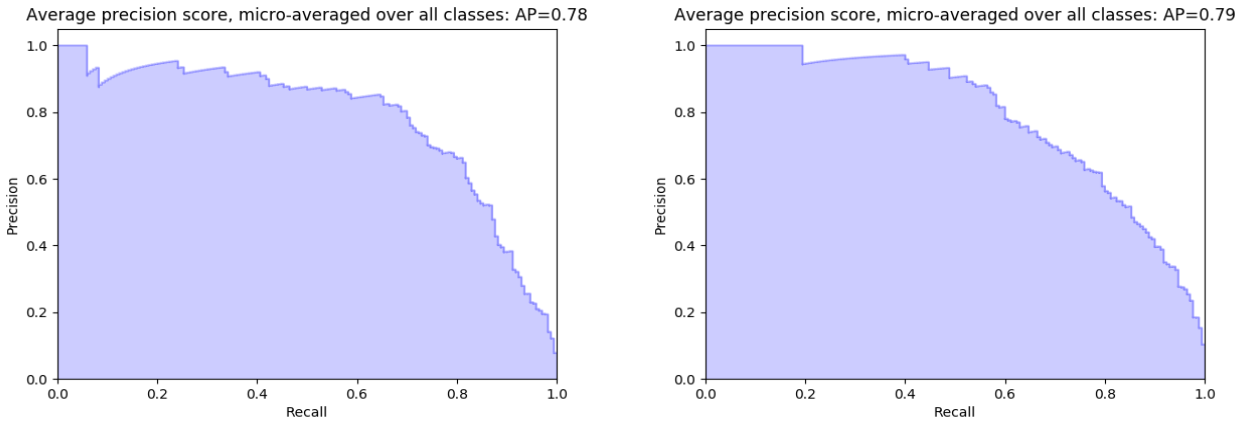
The fact the top-1 accuracy score is far worse than that of the training process could indicate that our models have overfitted on the training data, or it could indicate that the images in the custom evaluation set were difficult to classify.



(a) Model 1 (b) Model 2

Figure 4.3: Precision and recall curves for model 1 and 2

Looking at the individual precision and recall curves of the two models and in particular the worst performing categories, we see that both models are similar in the categories ID card and newspaper. Money billets show also difficulties to be classified, but 2 is slightly better than 1. Furthermore, there can be seen that 1 is better in classifying word processor, this may be caused by altering the images of word processors by whitespace removal. This may have led to the removal of features necessary for classification of this category. This could be prevented by a better implementation of the whitespace removal algorithm.



(a) Model 1 (b) Model 2

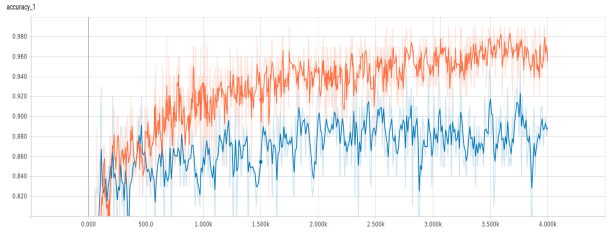
Figure 4.4: Average precision curves for model 1 and 2

Looking at the curves of the average precision curves, we can see that model 2 has better precision on lower recall, indicating the other categories show higher precision in comparison with 1. This is contrary to the inspection of the worst performing categories, where we get the impression that 1 is performing better than 2. If we look further along the curve we can see that both models show almost the same results, indicating that there is no significant performance difference.

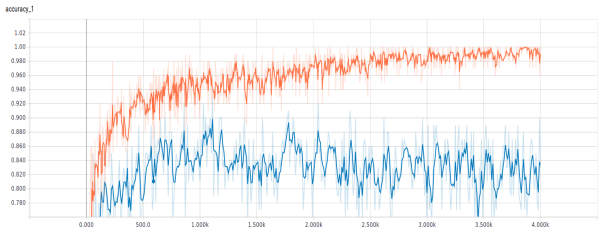
4.2 Experiment two

4.2.1 Training

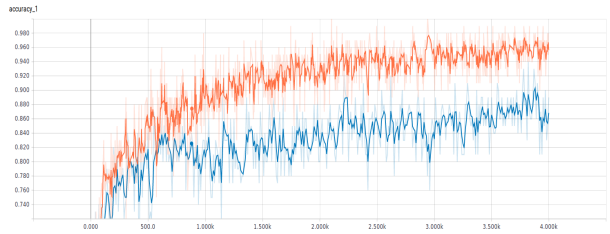
If we look at the training process we can conclude that model B is overfitting on it small dataset, just as we predicted. Model A and C show good overall training and validation accuracy with A being slightly better. The reason B shows the best score on the random test is probably it is being tested on a small number of images.



(a) Model A



(b) Model B



(c) Model C

Model	final random test
Model A	90.1%
Model B	93.5%
Model C	92.4%

(d) Random valuation test

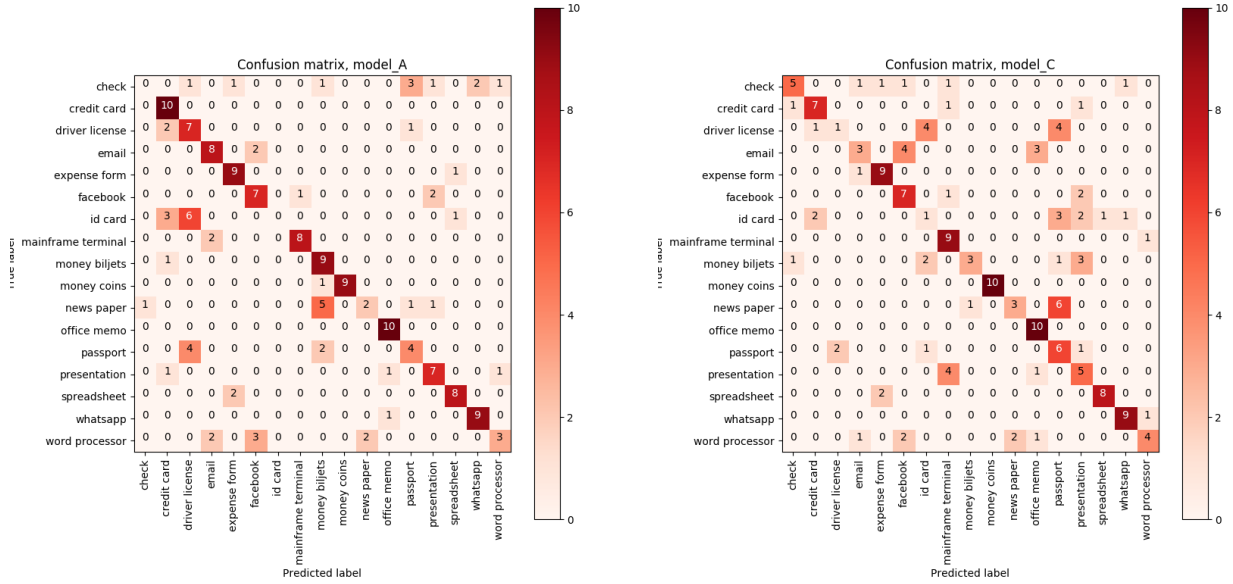
Figure 4.5: Training statistics for models A, B and C

4.2.2 Evaluation

Model	Top-1 accuracy	Top-1 error	Top-3 error	Top-5 error	MRR	Mean average precision
Model A	**64.7%**	**35.3%**	**14.1%**	8.8%	**0.79**	**0.72**
Model B	55.1%	44.7%	20.0%	11.2%	0.70	0.60
Model C	58.9%	41.2%	**14.1%**	**5.9%**	0.73	0.68

As we can see Model A has a better top-1 accuracy, which is self-explanatory due it being trained on only unique images. But when we compare A to C, the results are for most metrics similar, with the top 5 error rate even being better. Comparing C to B we can clearly see that there is a significant difference between applying augmentation to enlarge the data set and using only unique images. Only the top-1 accuracy and top-1 error are similar.

When comparing the confusion we can conclude A is far better on the top 1 accuracy. This demonstrates especially in categories that have a lot of text-based features such as ID-cards, and driver license. This is probably due to the fact that these categories have characteristic features related to text and symbols. Augmentation alters primarily shape and colors, so the use of selected augmentation techniques on these categories is not applicable. Noteworthy is the fact that C is significantly better at classifying checks and word processors. This can indicate that important features of checks and word processors are more related to

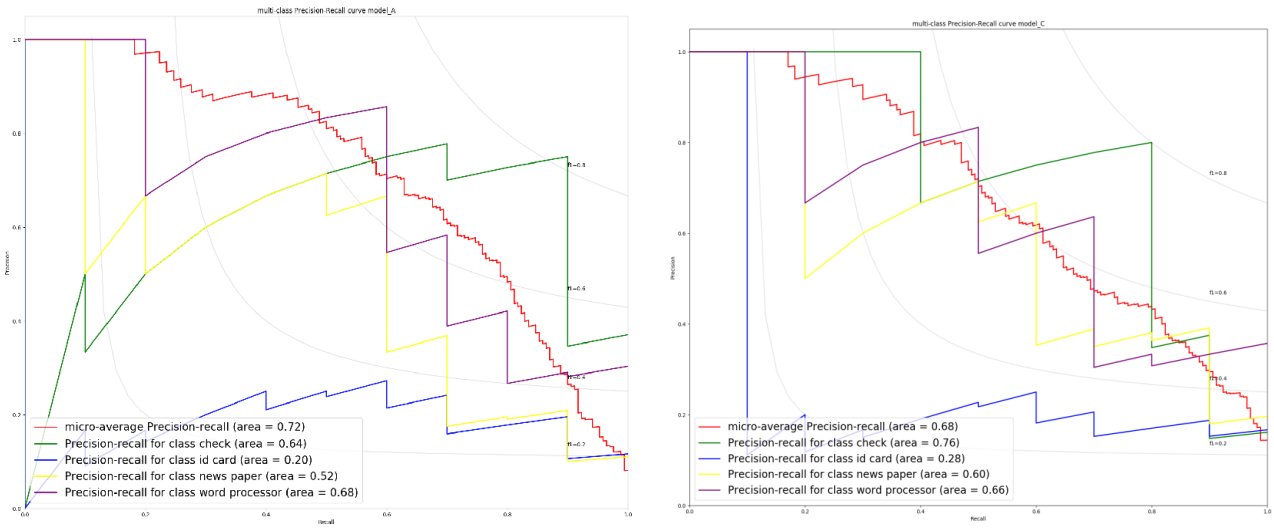


(a) Model A

(b) Model C

Figure 4.6: Confusion matrix model A and C

shape and color than symbols and text. A probably focused too much on these characteristics overlooking shape and color. This shows that augmentation can improve the focus on shape and color, thus improving the classification accuracy for some categories. Concluding from the predictions mapped in the confusion matrix we can clearly see A outperforms C, by the number of correct classifications.



(a) Model A

(b) Model C

Figure 4.7: Precision-recall curve model A and C

Concluding from the precision and recall curves of check, ID card, newspaper and word processor, A performs better on low recall than C, but higher values of recall show similar results for both models. From these statistics, keeping in mind A performs better on all the other categories, we can conclude that C performs almost on par with A, but lacks the precision coming from training with unique images. However one notable mention can be done about the curve of check, where it stands out that C has much more accuracy with the lower recall, but just as in the confusion matrix, this could be a result of A focusing on features not representative for the specific category.

4.3 Experiment Three

We already showed that image augmentation had a positive relationship on the overall classification quality of the models. Concluding from the previous experiments we didn't expect an overall better top-1 accuracy, but with the use of augmented images, we were primarily interested in the overall ranking quality and expected better results focused on ranking than models trained with only original images.

4.3.1 Evaluation

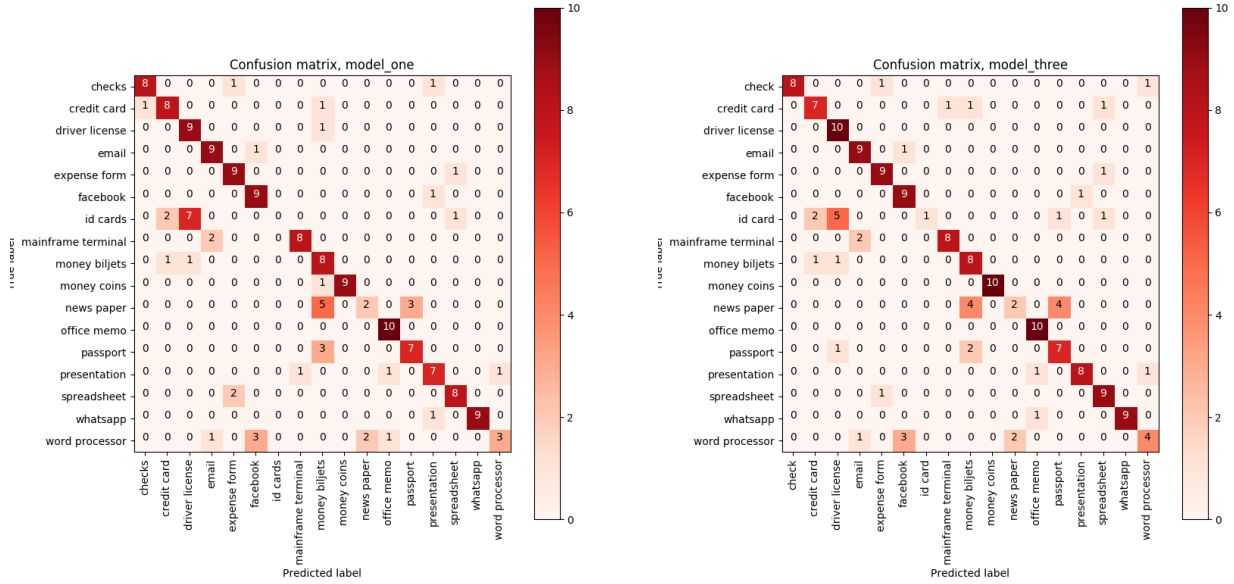
Model	Top-1 accuracy	Top-1 error	Top-3 error	Top-5 error	MRR	Mean average precision
Model 3	**75.3%**	**24.7%**	**7.6%**	**2.9%**	**0.84**	**0.82**
Model 4	73.5%	26.5%	10.0%	**2.9%**	0.83	**0.82**
Model 5	70.5%	29.4%	8.8%	**2.9%**	0.83	**0.82**
Model 6	72.9%	27.1%	9.4%	4.1%	0.82	0.81
Model 7	72.3%	27.6%	8.8%	3.5%	0.82	0.81
Model 8	69.4%	30.6%	10.6%	4.1%	0.81	0.78
Model 9	71.8%	28.2%	8.2%	5.3%	0.82	0.79

The results show that model 3 is on every metric better or on par with another model. Thus we can conclude that this is the overall best performing model. Comparing the results to experiment 1 we see that 4 models perform better or as good on the top-1 accuracy and almost all models perform better in the top-error rates, but the most prominent difference is the fact that these models perform almost all better in ranking the categories.

Comparing 3 to the best performing model of experiment 1 we see the following differences.

Model	Top-1 accuracy	Top-1 error	Top-3 error	Top-5 error	MRR	Mean average precision
Model 1	72.3%	27.6%	10.0%	4.1%	0.82	0.78
Model 3	**75.3%**	**24.7%**	**7.6%**	**2.9%**	**0.84**	**0.82**

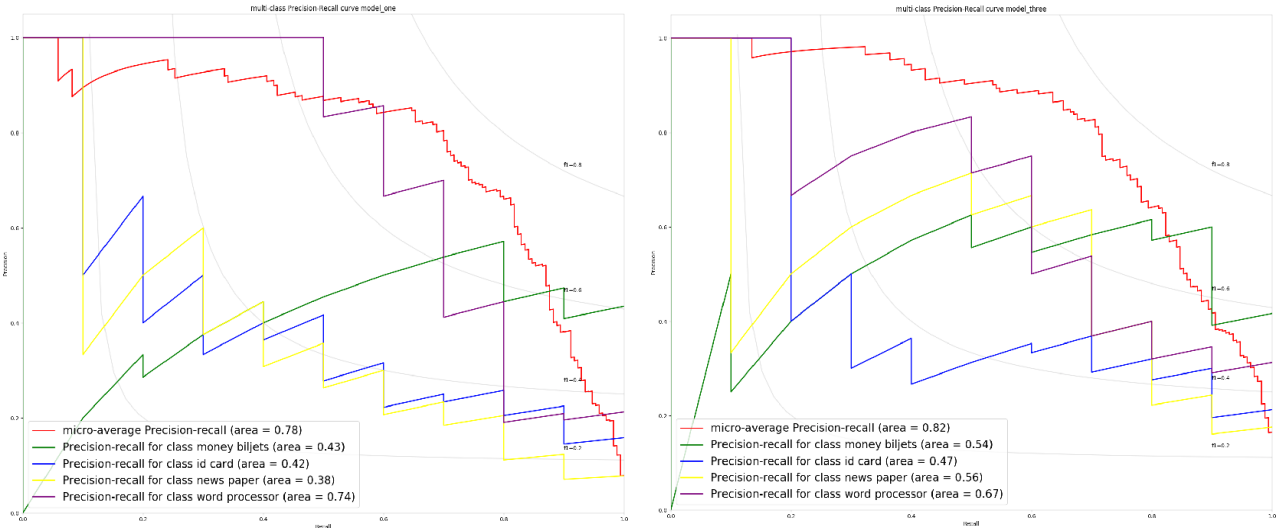
As we can see 3 performs better than 1 on every metric, but the most striking is the top-1 accuracy. In this experiment, a model trained with augmented training samples is better in correctly qualifying new images. Also as we predicted from previous experiments, we can see that the ranking quality improves with the use of augmentation. One noteworthy fact is the performance of model 9, which was trained on a heavy augmented data set. The results indicate that the performance is comparable to data sets that consist of lesser augmented images. It even outperforms the models trained on 80% standard and accumulated augmentation. This might indicate that eDiscovery images have significant dependencies on features related to overall shape whereas we first assumed it depends a lot on text and symbolic features.



(a) Model 1 (b) Model 3

Figure 4.8: Confusion matrix model 1 and 3

The confusion matrices of 1 and 3 show very comparable results, we see a small improvement in the classification of ID cards but for some classes, a deterioration can be observed. Basing on these metrics we can assume that the performance difference is mainly based on the overall ranking improvements. It looks like models ranking quality can be improved by training with a large amount of unique images combined with augmented images, keeping in mind that the task at hand is by nature a multinomial distribution classification.



(a) Model 1 (b) Model 3

Figure 4.9: Confusion matrix model 1 and 3

Comparing the precision-recall curves of the worst performing categories of model 1, we see that 3 shows higher precision with a lower recall for almost all categories, except word processor, is worse although average precision becomes comparable when recall increases. The results give us the idea that augmentation of the

images leads to better top-1 accuracy for categories that were previously underperforming.

Chapter 5

Implementation

5.1 System limitations

Zylab's back end software is primarily implemented with the programming language C#. Tensorflow currently supports python, c++, java, Go and swift, so our options where limited to a certain extend. Luckily there are third party projects that allow people to load Tensorflow graphs with other programming languages. One such module was Tensorflowsharp, a .NET binding library which allows user to load and use Tensorflow graphs. Another more general option is the creation of a python web API. This will allow users to make API calls and implement the results in other software. We made two implementations, one with Tensorflowsharp and another with a web api.

5.2 Implementation Tensorflowsharp

Tensorflow sharp is a .NET bindings to the TensorFlow library, and supports the entire low-level TensorFlow API, lacking some functionality coming with the high level python implementation. This restrict users in using the full library with C#, a common use case with this, is to train you're models with the TensorFlow python api or the high level keras implementation and load load them into C# with the use of Tensorflowsharp. Another downside with this framework is the fact that it will not help with the configuration of the model. So functions implemented in the used model need to be known beforehand. This means in order to use the model at least the input and output function names need to known and called. Also specific configuration such as input parameters and resizing of images is for the user to consider. The models need there inputs represented in tensors, which is the same way as in Tensorflow python. The framework has built in classes that model these matrices and are easy to use.

To give a specific insight into our implementation, we wanted to load multiple models into the application. The code used for loading and using the TensorFlow graph is the same for every model, the only variables

here, in our case, are the input/output functions and creation of tensors for the inputs. The loading of the graphs is done by the creation of a *"TFGraph"* object. This object forms a model from a graph.pb file just as in the python Tensorflow bindings. Then this graph needs to be feeded by an input tensor containing the image with the input function specific for the model. The results can be retrieved from the output function. The input tensor are model specific, therefore a custom implementation need to be made for preprocessing and normalizing the image. Also users need to consider the output results, because these are also model specific. To combat the variability, we made our implementation with use of a template pattern.

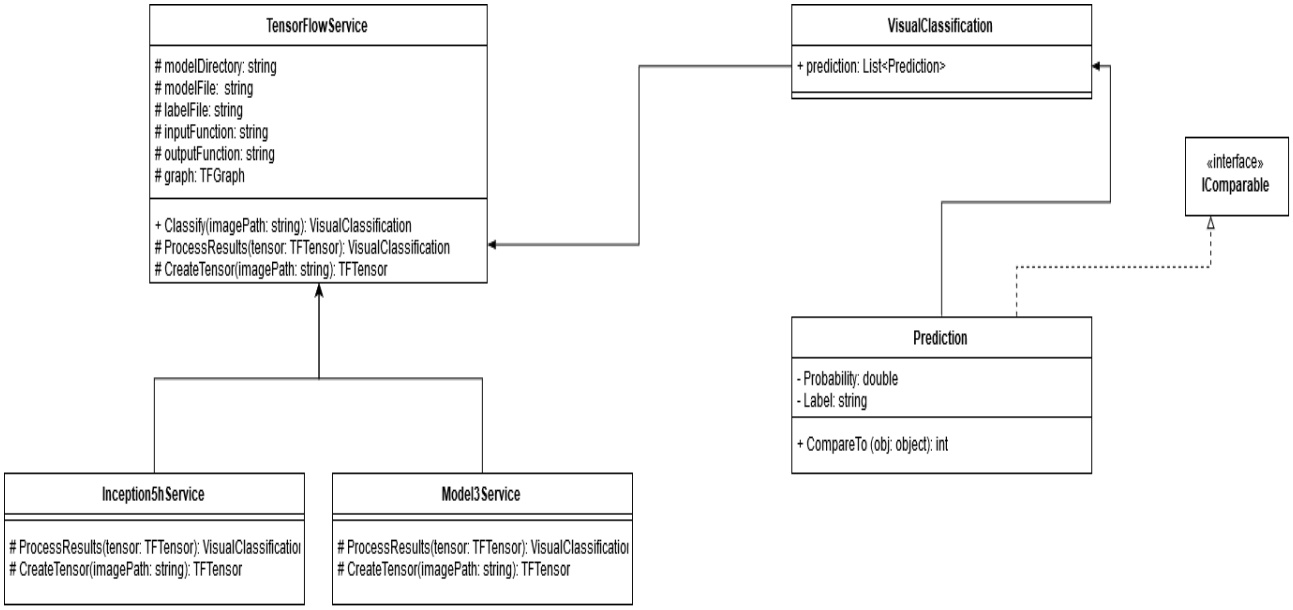


Figure 5.1: class diagram of TensorFlowSharp implementation

The Abstract class implements the template method classify, this method manages the feeding of the input tensor and processing of the results. The creation of the matrix is done by the primitive method *"Create tensor(String imagePath)"*. The processing of the results is done by *"ProcessResults(TFTensor tensor)*. Both these methods are called from the template method and the implementation is for the concrete classes extending the abstract class *TensorFlowService*. Also the used input and output functions can be initialized by these concrete implementations, for example with the constructor. This architecture allows us to use other models and make use of there own set of rules belonging to the specific model.

To give a concrete idea, we implemented an *Inception5H* model. This model had the input functions *input* and *output*. The input tensor needed to have a dimension of 224×224. In the *"Create tensor(String imagePath)"* function, we first resized the image and scaled it, before we created the tensor holding the data. The model returned a tensor of 1000 classifications, we mapped these to a list in the implementation *"ProcessResults(TFTensor tensor)* and saved the results.

5.3 Web API

The setup for this solutions is much less complicated in comparison with that of Tensorflowsharp. The core concept is very straightforward, where we create an API around the Tensorflow framework. We used FLASK, which was easy to implement. For every model we created a specific API endpoint. This gave the user the flexibility to chose from the different models. He or she only needed to make a REST request, where the data section contained the binary representation of an image, the API we made could resize and preprocess it, and feed it to the chosen model. We returned the results as a XML, which could be processed easily by most programming languages. This approach will give much more flexibility than other solutions. The only problem, is the fact that the program will make use of an external solution. Also in our implementation in order to use the API we had to deploy it in an always online environment, this can be a drawback for most implementations but a solution could be to deploy the api locally. In our example we loaded multiple models into the web API, and used the in C# the library restsharp to make requests to the API.

Chapter 6

Conclusion

In this research, we considered the overall recognition quality of eDiscovery categories with a retrained inception V3 model using Tensorflow. We also evaluated different training setups and analyzed the effects on overall classification quality. First, we looked at the retraining of the model for our own categories and concluded that a satisfactory result could be obtained with a relatively small dataset consisting of roughly 150 samples per category. Only considering the top-1 accuracy we could obtain a score of 72.3%, this shows that acceptable results could be obtained, keeping in mind that eDiscovery images are mostly text and symbolically based and thus hard to classify, with the general recognition quality of a pre-trained inception model. This could also lead to future research where we could consider the amount of classification quality could be obtained from different sets of categories.

Also, the fact that our model showed good results with training with a relatively small data set could indicate that satisfactory results could be achieved with an even smaller data sets. We evaluated the effects of this with our second experiment. Here we also considered the influence of image augmentation to enhance a small data set and compare the results to a model trained with a data set consisting out of entirely unique images. The results showed that our model trained on the smallest data set was overfitting on the training images basing on the training graph. This was also noticeable from the top-1 accuracy score with a value of 55%. The other two models performed significantly better with the one trained entirely on unique images showing the best top-1 accuracy results. However considering the fact that unique images will most of the time result to a better recognition quality, the model trained with augmented images demonstrated that the overall ranking quality was very comparable and even better on the top-5 error rate. We could conclude that a model trained with augmented images was more generalized than the one without and leads to more evenly distributed multinomial probabilities improving the overall ranking. Without augmentation, models are more certain of there first classification, causing lower probabilities in other classifications. Thus we can conclude that exposing a model to a wider variety of lighting and coloring situations with the same images will have beneficial results for the overall ranking and classification quality in visual classification tasks.

With the knowledge obtained from the previous two experiments, we wanted to analyze which training setup

would result in the best performing model. We constructed 6 data sets, by exposing the data set used in experiment 1 to augmentation techniques which were fairly standard, we split the 6 sets in two, with one subset being limited to at most one augmentation per image and the other having multiple augmentations per image. The models in these subsets where diversified by the number of augmented images. We also made a data set with augmentation techniques that were more drastic in comparison with the standard augmented sets. The results showed that for almost all models the top-1 accuracy was very comparable to that of experiment 1, but just as in experiment 2 the ranking performance improved significantly. Especially model 3 augmented with 10% light standard augmentation, which showed better ranking performance but also a better top-1 accuracy score, was a confirmation of the improvement obtained from applying augmentation techniques. The only negative effect which occurred was the fact that some categories showed worse precision with a low recall. This could indicate that augmentation could alter the features characteristic for categories too much, which results in lower probabilities on the top-1 accuracy. This was especially noticeable in the category word processor, most likely having features that lose there influence when being altered. Future implementations should evaluate whether certain categories will improve from augmentation or have too much dependency on certain feature characteristics.

6.1 Future work

Just as we concluded, it would be interesting to know which visual objects are too specific that transfer learning is not possible. Images related to eDiscovery where already very specific, but still satisfactory results are obtained. Also, the influence of augmentation could be further researched. We could notice that some categories performed better with augmentation, but some categories showed worse results. A research which determines the features that could benefit from augmentation would be interesting. This research could also extend to researching the best selection of augmentation techniques for specific visual features. Further, it is interesting to know how the results differ when the research is done with transfer learning or with an entirely new model.

6.2 Discussion

Comparing our results with other visual classification studies, we can conclude that the results are very comparable. For example, our top-5 error rate is better than the top 5 error rate of the Inception V3 model [SVI+15] on the image net classification challenge. Although this challenge is to classify 1000 categories, it still shows with the use of transfer learning a pre-trained model can be repurposed for classification of new categories with roughly the same error rate.

Bibliography

[ABB+15] A. Abadi, M.and Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Man, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Vigas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: Large scale machine learning on heterogeneous systems. Web, 2015. https://www.tensorflow.org/.

[Car98] Rich Caruana. *Multitask Learning*, pages 95–133. Springer US, Boston, MA, 1998.

[Cas] E. Casey. *Handbook of Digital Forensics and Investigation*. Academic Press.

[CC08] P. Cane and J. Conaghan. *The New Oxford Companion to Law*. Oxford University Press, London, 2008.

[Chu17] V. Chu. We need to go deeper: A practical guide to tensorflow and inception. Web, 2017. https://medium.com/initialized-capital/we-need-to-go-deeper-a-practical-guide-to-tensorflow-and-inception-50e66281804f.

[DDS+09] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009.

[Dea17] Lea C. Dearing. Top five tips for using forensics to win the case. *Daily Report*, 2017.

[Die15] S. Dieleman. Classifying plankton with deep neural networks. Web, 2015. https://benanne.github.io/2015/03/17/plankton.html.

[DJV+13] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. A deep convolutional activation feature for generic visual recognition. *DeCAF*, 2013.

[Eur16] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *Official Journal of the European Union*, L119:1–88, May 2016.

[GBJ17] Herve Goeau, Pierre Bonnet, and Alexis Joly. Plant identification based on noisy web data: the amazing performance of deep learning (LifeCLEF 2017). In *CLEF 2017 - Conference and Labs of the Evaluation Forum*, pages 1–13, Dublin, Ireland, September 2017.

[GC10] Maura R. Grossman and Gordon V. Cormack. Technology-assisted review in e-discovery can be more effective and more efficient than exhaustive manual review. *Richmond journal of law and technology*, 2010.

[IS15] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Proceedings of The 32nd International Conference on Machine Learning*, 2015.

[ITW93] S. Imade, S. Tatsuta, and T. Wada. Segmentation and classification for mixed text/image documents using neural network. In *Document analysis and recognition, 1993., proceedings of the second international conference on*, October 1993.

[Law98] David S. Lawyer. Text-terminal-howto. Web, 1998. https://linux.die.net/HOWTO/Text-Terminal-HOWTO-1.html.

[LBD+89] Y LeCun, B Boser, J. S Denker, D Henderson, R. E Howard, W Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation 1(4)*, pages 541–551, 1989.

[LRM+12] Q. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng. Building high-level features using large scale unsupervised learning. *ICML*, 2012.

[MDG+12] G. Mesnil, Y. Dauphin, X. Glorot, S. Rifai, Y. Bengio, I. Goodfellow, E. Lavoie, X. Muller, G. Desjardins, D. Warde-Farley, P. Vincent, A. Courville, and J. Berkgstra. Unsupervised and transfer learning challenge: a deep learning approach. *JMLR*, 2012.

[Mis07] Frederic S. Mishkin. *The Economics of Money, Banking, and Financial Markets*, page 8. Addison Wesley, Boston, MA, 2007.

[Pra93] L. Y. Pratt. Discriminability-based transfer between neural networks. *Morgan Kaufmann Publishers*, pages 204–211, 1993.

[PW17] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *CoRR*, abs/1712.04621, 2017.

[RBLP07] R. at Raina, A. Battle, H. Lee, and A. Packer, B.and Ng. Self-taught learning: Transfer learning from unlabeled data. *ICML*, 2007.

[Ron10] A. Ronacher. Flask is a microframework for python based on werkzeug, jinja 2 and good intentions. Web, 2010. http://flask.pocoo.org/.

[SLJ+15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[SSP] P. Simard, Y., D. Steinkraus, and P Platt, C. Best practices for convolutional neural networks applied to visual document analysis. *Microsoft Research*.

[SVI+15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *arXiv preprint arXiv:1512.00567*, 2015.

[Thr96] S. Thrun. Is learning the n-th thing any easier than learning the first? *NIPS*, 1996.

[WWCN12] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 3304–3308, November 2012.