Transfer learning from non-medical datasets to medical datasets

Bachelor Endproject (BEP) of	:	Floris J. Fok (0909649)
Daily supervisor	:	Dr. ir. Veronika Cheplygina
Supervising staff member	:	Dr. Dragan Bosnacki
Head of IMAG/e	:	Prof.dr. J.P.W. Pluim
Period BEP	:	10/18 – 01/19
Report number	:	B19/01

IMAG/e Medical Image
Analysis Group
Eindhoven

Transfer learning from non-medical source data to medical datasets

Floris Fok
IMAG/e group
TU/e - Technical university of Eindhoven

***Abstract*—Normally, deep neural networks take a large amount of data to be significant. Medical data sets, which are frequently smaller data sets, are therefore more difficult to use. Transfer learning is a method that enables knowledge learned from the source data to help classify the target data. This method shows to work effectively with small medical datasets. There is still, however, some uncertainty if transfer learning from non-medical data is effective for medical data. In this paper, we compare the performance of transfer learning from non-medical and medical source data to medical target data. To generalize the experiment, we experiment with two transfer learning methods, feature extraction and fine-tuning and comparing them with conventional training. Measuring the area under the ROC (receiver operating characteristic) curve to indicate the performance of the classified target data. By visualizing the filters or the layers of the weights of convolutional neural network (CNN) and note the similarity or dissimilarity between them. It turns out, transfer learning with both medical as well as non-medial data on average improves results. Fine tuning outperforms the other transfer learning method in this paper. Finally, the filters from a pre-trained CNN correlate with the performance of transfer learning. We obtained a better understanding of why certain models achieve better performance when using transfer learning and proposing different data characteristics to be examined to extend this understanding further.**

***Index Terms*—Transfer, learning, non-medical, filters, fine-tuning, Feature, weights**

I. Introduction

Data analysis strategies, such as deep learning are growing substantially, and a few commonly observed pitfall are repeatedly being made[2]. One of these pitfalls is when the number of samples is realistically too low to attain significance or to train a model. Insufficient training data could diminish the learning capabilities, and insufficient validation data hinders the true evaluation of the hypotheses. Learning neural network models from a small number of training examples is an important challenge in computer vision. People, however, can learn new categories from just a few examples. To achieve similar performance with machines learning it is likely that visual features learned, must be transferred for other tasks.

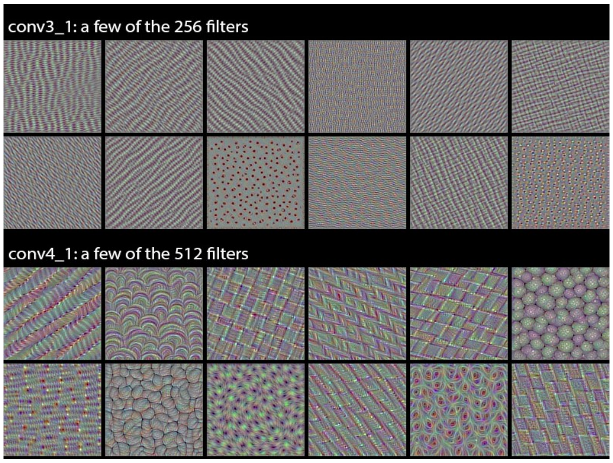

conv3_1: a few of the 256 filters
conv4_1: a few of the 512 filters

Figure 1: 12 filters each from both the VGG16 conv3.1 and conv4.1 layer visualized. The model is trained on ImageNet and the method used to visualize these images can be found in this blog post [20].

Transfer learning, sometimes also referred to as Knowledge transfer, is a term in machine learning that focuses on transferring knowledge gained solving one problem and applying it to a different but related problem [1]. To understand the mechanics of transfer learning, we first start with traditional learning. Usually, a new deep neural network (DNN) is trained from randomly initiated weights and to make useful predictions with such a DNN, it takes a large and diverse dataset to covert all the weights into useful classifiers and not cause any over-fitting. To avoid these pitfalls, one can import weights from a pre-trained model, trained on the so-called source data. These weights contain knowledge that can extract features of an image. The data we want to classify, or target dataset, can benefit from a pre-trained set of features to improve its classification [3]. As mentioned before, smaller datasets are not able to attain significance. It has, therefore, more benefits for smaller then larger datasets [4].

Transfer learning is a becoming a more popular method in medical image analysis as described in a review concerning medical image research [9]. Being able to use non-medical data has a positive impact on the performance of medical models, as those datasets usually are much larger and more accessible than medical data. So far, there is a contradiction between certain papers whether non-medical data or medical data is necessary during the use of transfer learning on medical data sets [3] [5]. Papers frequently use a single target data, on multiple source data. Additionally, it is difficult to compare papers, since they all use different methods and performance indicators. Therefore, in this paper, we apply a systematic approach to compare different data sets to see if medical source data is necessary in case there is a medical target data set.

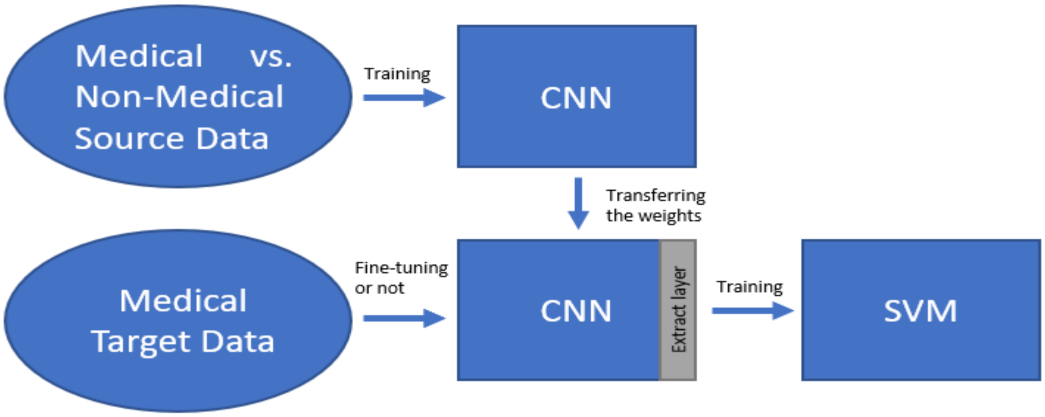


Figure 2: Schematic of transfer learning methods used in this paper. The grey area is the part of the model that is cut off to extract the feature vector to train the support vector machine (SVM). The target data will always be medical, but the source data can be medical or non-medical

Medical data is data that can be clinical analyzed, thus has not has to do anything with the characteristics of the image. However, most cases, medical images differ from standard images as being grayscaled or artificially stained to enhance certain features. Non-medical data is everything else, but in this paper, we focus on the object-oriented natural images. Natural images are images which have a vibrant local covariance structure. Source data is regarding transfer learning the data you train your model with from scratch, and target data is the data that uses the pre-trained model, which we want to classify.

The following deep learning image techniques are currently in use [9]: Segmentation, detection, classification, and registration. Each having subcategories, such as object, region or texture. In this paper, we focus on the object-oriented classification. CNN is the most popular network within the image classification research field [9]. This paper focuses on medical image classification and therefore only focus on CNN and CNNs proof to capture features from images that most images recognition tasks need [15]. Popular CNN architectures are ResNet [16], AlexNet [17], GoogleNet [18] and VGG16 [19]. The frequent use of image classification does not exclude other networks or task from using transfer learning. It can also be found in gaming, translation and speech recognition [8] [6] [7].

New development enables us to look into neural networks and see the filters that are responsible for the success of the CNN [26]. The filter, a 2D representation of weights, is a layer of the weights in a model. Usually, training a model from scratch gives the model randomly initialized weights, which requires large amounts of data to convert the model to a good performing model. In other words, it takes a large amount data to form the filters that extract useful features. Transfer learning gives the model a head start, by supplying the network with filters from the beginning. In this paper, we visualize and examine the filters of the pre-trained model to see if any patterns derive from it. In figure 1 we show such an example of this technique, extracting some filters of the several layers in the weights of two model layers.

In this paper we examine the influence of the source data on the target data, to conclude whether it requires medical source data for transfer learning with medical target data. Additionally, finding a reason for the contradictions, but more importantly getting a better understanding of why specific a model is performing better than others. To conclude which model is performing better, we examine both the performance and filters of the pre-trained network's weights. With this new knowledge, we want to propose different strategies or directions to examine the success of transfer learning.

II. Prior Work

Transfer learning for medical target data is a method that is repeatably used in medical image analysis. Unfortunately, the amount of target and source datasets are over-represented, as noticed in some reviews concerning transfer learning. Litjens et al. states that papers were too small for general conclusions and finds contradictions on whether fine-tuning or feature extraction is the better option [9]. Cheplygina et al. point out the popularity of transfer learning and the contradiction between whether medical source data is necessary when transfer learning with medical target data [3] [5]. In short, they both call out for a systematic and a more generalized approach on transfer learning with medical target data.

Before creating a more generalized approach, we must first know what the most common methods are. The method we found the most under transfer learning paper is the feature extraction [10] [11] [12] [13]. Huynh et al. compare custom made features to feature extraction to classify its target data. The two score equally but combined increases the score. Fine tuning is less popular, but a promising method. Antony et al. show that is can outperform feature extraction [**?**]. Therefore it is interesting to take both these methods and use them for both medical and non-medical source data.

Both Alexnet and VGG16 are popular with feature extraction related transfer learning methods, because of their sequential architecture and Dense layers with a vector with a size of 4,096 as output. These vectors simplify future extraction. Additionally, both architectures have a fully sequential model, which is easier to manipulate [20]. Alexnet is less expensive to train compared to the other architectures mentioned [25]. VGG16 is versatile in medical classification, due to small kernel size and deeper layers [21] [22] [23]. The models generalize well to a wide range of tasks and datasets, matching or outperforming other complex models [**?**]. The goal is to , and with better GPUs available, therefore we choose VGG16 as our go-to model.

Zhou et al. compare transfer learning in non-medical image classification with different tasks [30]. Comparing the training of Natural Images and Images of scenes between object-centric images as occur in natural images, and scene-centric images as occur in images of the scenes. The results show that CNN trained on a similar data set, scene to scene and object to object, results in higher accuracy of the model. We are examining to see if this relation of scene to scene and object to object also occurs with medical and non-medical.

Finally, Zintgraf et al. are pointing out to the importance of knowing how models used for medical purposes make their decision [31]. We agree on his point of view and want to understand more why some models perform better. However, the methods they use is highly computationally intensive and differs for each image. Therefore we visualize the weights of the model instead of decision activation. The weights are a more generalized visualization of the model.

III. Methods and Data

There are four methods used, traditional training of the VGG16 for reference, visualization of the weights for better understanding and two transfer learning methods for comparison. Describing these methods in more detail in the subsections below. The transfer learning methods are common methods in the field of medical imaging, Feature extraction, and Fine-tuning [32] [33]. In this paper, both methods use a feature vector extracted from a layer of the CNN to classify the test set with a second learning algorithm. The difference between the methods is that the fine-tuning re-trains the weights of the pre-trained model. We convert all the results to a single digit score to compare the different methods.

A. Data

The data for the experiments contains free to the public data sets only, contains both medical and non-medical data sets. The data is imported into Python and resized to 224 by 224 in RGB colors. Data sets with unbalanced classes use class weights during training. This gives an equal impact of the classes on the weights. For example; Chest got approx. 4,500 normal and 1,500 not normal cases, the weights translate to one and three. The images are shuffled using a random seed. The validation and test sets are a subset of the shuffled data. The validation set is 1/10th of the data, and the test set 1/5th of the data. Some images classification models take other data into account, such as age and gender to determine the outcome [34], this is not the case in this paper. The source data consist of the first five datasets. The target data consist of the last two and Chest x-ray dataset.

Algorithm 1 Pseudo script of the transfer learning experiments, Running target data for transfer learning. This represents the code in python for most part, oversimplified.

```
procedure TARGET DATA($file, parameters$)
    Import data
    Randomize split data
    Load VGG model
    if $No - FT$ then
        Load pre-trained weights
        Extract features from model
        Preform SVM
    else if $FT$ then
        Load pre-trained weights
        Train model weights
        Extract features from model
        Preform SVM
    Calculate AUC
    Save results
```

1) Imagenet: This data is from the Large Scale Visual Recognition Challenge (ILSVRC) 2012, the challenge contains an enormous amount of images ranging from airplanes to dog breeds. From this datasets, we did not use the actual 1.2 million images containing 1,000 classes, but the weights. These weights are free to the public and are very common in transfer learn papers [4]. The classes have a uniform distribution.

2) Cat Dog: This data is from a 5-year-old Kaggle challenge and contains approximately 25,000 pictures of cats and dogs with a uniform distribution over the two classes.

3) Natural images: The Natural Image dataset is a subset of other larger set, including flowers of Imagenet and dogs from the dog and cat set. There are in total 8 classes and almost 7,000 images, which show a uniform distribution over the classes.

4) Chest x-ray Pneumonia: The Chest dataset is a medical dataset. Bad CT scans were removed from the data set by experts. The not-pneumonia class is over-represented. The images are black and white. The dataset contains 6,000 images.

5) Diabetic Retinopathy Detection: Also referred to as KaggleDR, consists of high-resolution retina images taken under varying conditions and is a medical dataset. The classes represent the scale of Diabetic Retinopathy from zero to four. The image is for a large part black and is not as bright as other pictures. The more than 35,000 images do not have a uniform distribution, the healthy state, class zero, is the majority.

6) ISIC melonoma: The ISBI Challenge 2017 / ISIC Skin Lesion Analysis towards Melanoma Detection data set. It contains 2,000 pictures and 3 classes. The classes do not have a uniform distribution, and because of the size dataset, we use data augmentation to enhance the minority to make equal amounts of classes. Data augmentation is helpful to improve the performance of smaller datasets [32].

7) Blood cell images: This dataset contains 12,500 augmented images of blood cells with accompanying cell type classes. There are approximately 3,000 images for each of 4 different cell types grouped according to cell type. The cell types are Eosinophil, Lymphocyte, Monocyte, and Neutrophil.

B. Pre-training

Before we apply any transfers learning, we train the source data on the untrained, random initiated VGG16 model. Except for Imagenet weights, which we download from the Keras module. The data is re-sized to 2242243 and shuffled each epoch. All models trained 50 epochs, and because of the larger datasets, we used no data argumentation. For loss function, we use Categorical cross entropy and for the optimizer Stochastic gradient descent(SDG) with steps of 0.001 and Nesterov momentum of 0.01. Saving the weights for transfer learning after each training of a model. The pre-trained models scores are above 90% accuracy, except for the Retina model which scores little over 70%.

C. SVM

The SVM in this paper is the One vs. the rest SVM linear classifier from the sklearn multiclass package. This strategy requires the base classifiers to produce a real-valued confidence score for its decision, rather than just a class label; discrete class labels alone can lead to ambiguities, where multiple classes predicted for a single sample [35]. The SVM trains on the train data and tests on the test data. Using the vector from the second dense layer, with a length of 4,096 as feature vector [36]. Huynh, B. Q. et al. suggests layers closer to the input give better performance [37], but the size of the dataset and selection of network gives doubt to the applicability in this paper.

D. Fine-tuning

Fine tuning is training the pre-trained model trained by source data again with the target data. This enables the model to adapt its weights to the new dataset to enhance performance. The fine-tuning happens with the SGD optimizer. We apply the same SGD optimizer and loss function settings as mentioned within the section pre-training. These settings are proven to work with small data sets and VGG16 [29], which is important when working with VGG networks as they can easily suffer from vanishing gradients [38]. Besides, cross entropy is proven to be best for classification [39]. Nesterov momentum can correct the jump afterward or also refer to as steering the momentum [40]. This anticipatory update prevents us from going too fast and results in increased responsiveness, which has significantly increased the performance of Neural networks [41]. All layers are trained, since training all layers of the network during fine-tuning is proven to be the most effective approach [42]. An Early stop function prevents overtraining. Which means, if the validation loss did not change or descents during the last 10 epochs, it stops the training of the model. If this does not occur, it continues to 50 epochs.

Table I: Summary of VGG16, describing each layer of the model. The table separates the different so-called blocks of the network. In this paper, the input size is 2242243, but using other input sizes changes the size of the model. Param stands for the number of parameters. Max Pool takes the maximum value of the kernel and passes it through the next layer, and this reduces the size of the model. Conv2d layers are applying a kernel over the image, to create the next layer. Dense layers are fully connected

Layer	Output Shape	Param
Input Layer	(None, 224, 224, 3)	0
Conv2D	(None, 224, 224, 64)	1792
Conv2D	(None, 224, 224, 64)	36928
MaxPool2D	(None, 112, 112, 64)	0
Conv2D	(None, 112, 112, 128)	73856
Conv2D	(None, 112, 112, 128)	147584
MaxPool2D	(None, 56, 56, 128)	0
Conv2D	(None, 56, 56, 256)	295168
Conv2D	(None, 56, 56, 256)	590080
Conv2D	(None, 56, 56, 256)	590080
MaxPool2D	(None, 28, 28, 256)	0
Conv2D	(None, 28, 28, 512)	1180160
Conv2D	(None, 28, 28, 512)	2359808
Conv2D	(None, 28, 28, 512)	2359808
MaxPool2D	(None, 14, 14, 512)	0
Conv2D	(None, 14, 14, 512)	2359808
Conv2D	(None, 14, 14, 512)	2359808
Conv2D	(None, 14, 14, 512)	2359808
MaxPool2D	(None, 7, 7, 512)	0
Flatten	(None, 25088)	0
Dense	(None, 4096)	102764544
Dense	(None, 4096)	16781312
Dense	(None, 1000)	4097000
Total params: 138,357,544 Trainable params: 138,357,544 Non-trainable params: 0		

E. Weights Visualization

The VGG16 model counts roughly 14 million weights, it would be a mess to visualize them all. The method of our choice is the visualisation of the filters, these filters are a two-dimensional representation of the weights. They are called filter because they filter certain features from the image, and pass that information to the next layer. Using Keras backend to visualize inputs that maximize the activation of the weights in different layers of the VGG16 architecture, we can reveal the filter. A randomly generated image passes through a function that optimizes loss of a particular filter, this is the inverse of what happens during regular training of the model. This process iterates for 20 times or till the loss gets below zero while multiplying the weight changes with the randomly generated image instead of the weights during regular training. We are examining the remaining image for comparison with the other models.

Table II: Results of the experiments in AUC score. No transfer means the reference experiments. All numbers are the average of three experiments with the deviation displayed as the maximum deviation. Differences are calculated by subtracting feature from fine tuning. Mean, is the mean of the three sub sets of ISCI.

	Source data	**Retina**		**ImageNet**		**CatDog**		**Chest**		**Natural**	
Target data	No transfer	FE	FT	FE	FT	FE	FT	FE	FT	FE	FT
Malignant vs. Benign	0.778 ± 0.066	0.750 ± 0.012	0.810 ± 0.010	0.845 ± 0.021	**0.889** ± 0.027	0.770 ± 0.018	0.808 ± 0.019	0.758 ± 0.003	0.756 ± 0.012	0.675 ± 0.090	0.775 ± 0.009
Melanoma vs. Benign	0.798 ± 0.020	0.779 ± 0.016	0.828 ± 0.008	0.899 ± 0.026	**0.911** ± 0.022	0.772 ± 0.023	0.832 ± 0.004	0.790 ± 0.009	0.790 ± 0.045	0,671 ± 0.034	0,789 ± 0.025
Melanoma vs. Carcinoma vs. Benign	0.825 ± 0.063	0.807 ± 0.005	0.842 ± 0.032	0.912 ± 0.005	**0.920** ± 0.024	0.793 ± 0.022	0.825 ± 0.028	0.782 ± 0.023	0.795 ± 0.023	0,776 ± 0.034	0,818 ± 0.020
Mean ISIC	0.800	0.779	0.827	0.885	0.907	0.778	0.822	0.777	0.780	0,707	0,794
Differences			0.048		0.021		0.043		0.004		0,087
Blood	0.501 ± 0.001	0.676 ± 0.011	0.841 ± 0.035	0.865 ± 0.003	**0.889** ± 0.089	0.778 ± 0.007	0.809 ± 0.037	0.726 ± 0.005	0.863 ± 0.021	0,802 ± 0.005	0,805 ± 0.007
Differences			0.165		0.024		0.031		0.137		0,003
Chest	0.987 ± 0.001	0.958 ± 0.009	0.984 ± 0.007	0.980 ± 0.005	**0.991** ± 0.010	0.971 ± 0.013	0.984 ± 0.020	N.A	N.A.	0,980 ± 0.008	0,986 ± 0.017
Differences			0.026		0.011		0.013				0,006

F. AUC

The performance of a model, or result of an experiment is equal to the AUC score. Better performance means a higher score. The AUC normally ranges from 0.5, random guesses, to a perfect 1.0. AUC stands for the area under the ROC curve. The ROC curve, and hence the AUC, gives an indication of the true versus the false positives. AUC is a fundamental evaluation tool in medical data [43]. Besides, the AUC is especially beneficial when single digit results are necessary [44].

G. Experiments

The first experiment consists of training the model from scratch with each target dataset, as described aboveIII-B. This experiment used as reference or comparison to measure improvement of performance. From the test data, the AUC is calculated to measure its performance. The experiment is repeated three times to add consistency on every target data and represent the first column of the result table. Secondly, each source data trains in the same manner as the first experiment, and we use these models for both transfer learning and visualization later in the experiments. After that, we begin with the transfer learning experiments. As for each target data set and each pre-trained model, we perform transfer learning with and without fine-tuning. To compare the transfer learning with the reference experiments and to compare the two methods with each other. So for each source data, we get two columns; one with transfer learning without fine-tuning, using the SVM as classifier and AUC as a measurement of performance; and one with transfer learning with fine-tuning, again using the SVM as classifier and AUC as a measurement of performance, counting a total of eleven columns. Finally, we visualize the filters of every pre-trained model using the method describe aboveIII-E. This enables us to compare the performance of the pre-trained models during transfer learning and the filters of the pre-trained models.

H. Materials

The experiments are conducted on a desktop with a GTX 1070ti and running Python 3.6 (64-bit) with Keras and Tensorflow-gpu 1.5 as backend. The full code is on Github VIII. A full explanation of the code given in the read me of the respiratory.

IV. Results

Table II is showing the results of the experiments including the maximum deviation. The reference experiments are in the column labeled no transfer, the experiments without fine-tuning are in the column labeled FE and the experiments with fine-tuning are in the column labeled FT.

A. Score

Fine-tuning the model pre-trained with ImageNet outperforms all other methods. Fine-tuning increases the AUC score when using the method practiced in this paper. The results with larges deviations occur more in the fine-tuning results than they occur in the no fine-tuning results. The Blood data set benefits the most from the transfer learning, with Natural data set as an exception. The ISIC data subsets, all follow the same pattern as the mean ISIC score. Except for ImageNet, other source data does not have a consistent performance on the target datasets. For example, the retina model performs second best with fine tuning on ISIC, but not on the other two target datasets.

Figure 2 shows the average performance of the pre-trained models so that we can compare the performance of transfer learning between the source data. The figure is again showing the effects of fine-tuning performing better than no fine-tuning. Comparing the performance of transfer learning with the performance of no transfer learning shows, an average increase in AUC for every model. Were the Natural Image model reads only a mere 0.003 difference. Non-medical has a 0.025 advantage over medical. If we exclude ImageNet, summing the two methods, give an average difference of mere 0.005 in favor of the medical data sets.

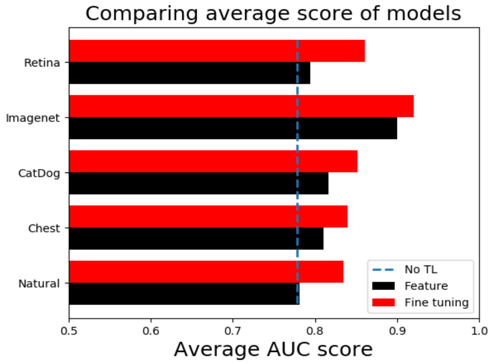
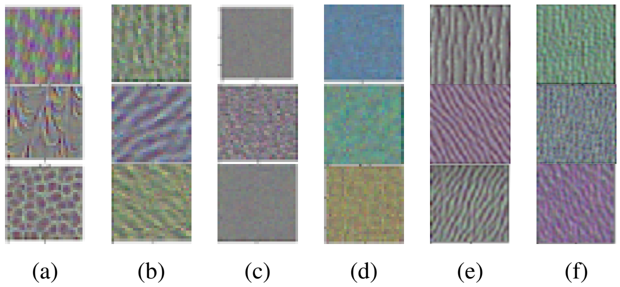


Figure 3: The sum of model performance for each source dataset. Feature means only preforming the SVM and fine tuning means both training of the model and SVM is preformed. No TL is the average of no transfer learning.

B. Weights

There are roughly 500 filters in the first three layers. Therefore we categorize the weights in certain filter types. Figure 3 shows three examples of each category, complex (structural), directions (structural), color (non-structural) and noise (non-structural) and three examples of both deep and shallow filters. The percentages of the structural filters are highest at Imagenet, 79% followed by the Retina model with 40%. The CatDog gets 30%, Natural 14% and Chest 6%. The percentage of structural filters follows the line of success in fine-tuning, except for Chest model. The Natural model has the most shallow filters, which made it hard to count the structural filters.

A remarkable observation is that the retina model showed all it's complex filters to be green. A similar observation was a large number of purple filters with the Nat model, while other models have more various and greyish filters. Both Retina and Natural perform poorly at no fine-tuning.



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

Figure 4: Filters from the model downsized to 30x30 pixel. Three pictures in a column each represents a group a) Complex structures filter b) Direction filter c) Noise filter unstructured d) Color filters e) Deep filters f) shallow filters.

V. Discussion

In this paper, we compared several medical and non-medical source data and their performance in transfer learning on medical target data. Besides, we visualized the weight to see if any patterns would arise from comparing it with the transfer learning performance of that particular pre-trained model. We found that non-medical source data can improve performance when using transfer learning on medical target data if fine-tuning is applied. This is best visible in figure 3. Besides, this figure tells us that there is no distinct difference between medical and non-medical, which derives from the average of the non-medical and medical datasets, which had either an advantage for non-medical or performed equally on average.

In agreement to Menegola et al. [29], transfer learning with fine-tuning shows an advantage over not using fine-tuning. It shows that different layers can influence the effects of feature extraction and thus, the results of this experiment**??**. Choosing a layer closer to the input layer could reduce, increase or reverse the effects of fine-tuning.

The visualization of the filters shows a correlation with the performance of transfer learning, especially when using fine-tuning. The chosen layer could cause the reason why the pattern does not apply for the no fine-tuning method. We expect that when a part of the decision layer is in the transfer, the extracted vector contains more features vital to the source data, but when fine-tuned contains more features vital to the target data. So the performance on the fine-tuning correlates with the ability to extract features as a whole, and the no fine-tuning correlates with a similarity of essential features in target and source data. To conclude this correlation, this phenomena must be examined in future research.

The performance of a model is highly likely due to the size of data seen by the model, especially with the number of images per class. The models' performance in figure 3, match the average image count per class. The consistency could be that models which see more data per class, have the change to develop deeper and more structural filters. A different experiment must be conducted before we can conclude whether data size is the cause, or that other characteristics are responsible. For example, by comparing filters and transfer learning performance of small and large data sets in combination with contrast and blurred images. The main advice is to step away from the classification of medical and non-medical and find more general characteristics since these terms are too broad to be used as classification.

The contradictions in the literature are more likely due to the frequent use of only one target dataset, which has either an advantage or disadvantages compared to the medical source data. It seems that the classification of a data set as medical or non-medical doesn't directly influence the effectiveness of transfer learning. Each medical target data set response differently to the giving pre-trained models. This behavior is visible in table II if the results of both the CatDog and Chest model, which behave contradictory to whether non-medical data is suitable as source data for transfer learning on medical target data. An explanation for this could be the class uniformity when the source and the target data have uniform classes the transfer learning is more successful. This should be tested by comparing medical datasets with uniform classes and non-medical datasets without uniform classes.

The Chest target data results performed too well, which made it hard to use as any proof. This result could be related to using the deep VGG16 instead of a more shallower model in combination with the image selection of the experts, that made it easier to classify. The Chest target data only tells us that the Retina model performs worse, which was unexpected as the images both show the most similarity and are medical. However, after visualizing the weights of the Retina model, it shows the model only has green complex filters and no other colors. For example, the ImageNet filters contain random colors, which to the eye look evenly distributed and most of the time grey. In other words, ImageNet can tell differences concerning shape instead of color, which is more favorable for the originally grey scaled Chest dataset. In future research, it could be interesting to see if data containing a greyscale, uniform RGB colors or single RGB color influence the success of transfer learning.

The fine-tuning result could show a more stable score if we use the best scoring models from each experiment by only saving the model with the lowest validation loss or highest training accuracy of all the epochs. Currently, the models performed with quite a large deviation due to sudden drops of accuracy or loss. In general, the settings for the fine-tuning and training will influence the score a lot and can favor specific datasets, so conducting these experiments with different setting could lead to a different conclusion. For future research, it is recommended to apply the optimal optimizer, and loss function for each dataset in combination with the save best only mode, so all data set to perform at its best.

The influence of layer choice is something that could be affecting the performance of fine-tuning. If less of the model takes part in the extraction of the feature vector, the influence of the fine-tuning and pre-trained model is smaller [31]. Using other layers could differ the results and should be taking into account when using the results of the paper.

Medical data sets, free to the public, are still highly outnumbered by non-medical images. Therefore data sets such as ImageNet are hard to compete with, and medical dataset with a class counts of 1,000 classes and over a million images do not exist. The significant difference between class and image count makes it impossible to compare with non-medical data sets on a large scale. It would be interesting in the future, to put a few data sets of approximately the same image size (and class size) together. Future research would benefit from this large dataset and could enable more testing with class and image counts.

VI. Conclusion

Non-medical data is suitable for transfer learning with medical target data, especially when fine-tuning is applied. When comparing different Transfer leaning methods, fine-tuning improves the score more than no fine-tuning. The success of transfer learning is more likely to be correlated with the amount of data seen by the model and the development of structural filters; then similarity in data sets. Future research should focus on more general characteristics of the dataset that can improve the performance of transfer learning.

References

[1] West, J., Ventura, D., Warnick, S. (2007). Spring research presentation: A theoretical foundation for inductive transfer. Brigham Young University, College of Physical and Mathematical Sciences, 1.

[2] Ravishankar, H., Sudhakar, P., Venkataramani, R., Thiruvenkadam, S., Annangi, P., Babu, N., Vaidya, V. (2016). Understanding the mechanisms of deep transfer learning for medical images. In Deep Learning and Data Labeling for Medical Applications (pp. 188-196). Springer, Cham.

[3] Cheplygina, V. (2018). Cats or CAT scans: transfer learning from natural or medical image source datasets?.

[4] Kornblith, S., Shlens, J., Le, Q. V. (2018). Do Better ImageNet Models Transfer Better?. arXiv preprint arXiv:1805.08974.

[5] Cheplygina, V., de Bruijne, M., Pluim, J. P. (2018). Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis.

[6] Zoph, B., Yuret, D., May, J., Knight, K. (2016). Transfer learning for low-resource neural machine translation. arXiv preprint arXiv:1604.02201.

[7] D. Wang and T. Fang Zheng. 2015. Transfer learning for speech and language processing. arXiv preprint arXiv:1511.06066.

[8] Banerjee, B., Stone, P. (2007, January). General Game Learning Using Knowledge Transfer. In IJCAI (pp. 672-677).

[9] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. Medical image analysis, 42, 60-88.

[10] Van Opbroek, A., Ikram, M. A., Vernooij, M. W., De Bruijne, M. (2015). Transfer learning improves supervised image segmentation across imaging protocols. IEEE transactions on medical imaging, 34(5), 1018-1030.

[11] Li, Z., Hoiem, D. (2018). Learning without forgetting. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(12), 2935-2947.

[12] Huynh, B. Q., Li, H., Giger, M. L. (2016). Digital mammographic tumor classification using transfer learning from deep convolutional neural networks. Journal of Medical Imaging, 3(3), 034501.

[13] Shie, C. K., Chuang, C. H., Chou, C. N., Wu, M. H., Chang, E. Y. (2015, August). Transfer representation learning for medical image analysis. In Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE (pp. 711-714). IEEE.

[14] Antony, J., McGuinness, K., O'Connor, N. E., Moran, K. (2016, December). Quantifying radiographic knee osteoarthritis severity using deep convolutional neural networks. In Pattern Recognition (ICPR), 2016 23rd International Conference on (pp. 1195-1200). IEEE.

[15] Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S. (2014). CNN features off-the-shelf: an astounding baseline for recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 806-813).

[16] F. Cero, A. Oliveira, and G. Botelho, Deep learning and convolutional neural networks in the aid of the classification of melanoma, in SIBGRAPI, 2016.

[17] J. Kawahara, A. BenTaieb, and G. Hamarneh, Deep features to classify skin lesions, in ISBI, 2016, pp. 1397280931400.

[18] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542(7639), 115.

[19] X. Sun, J Yang, M. Sun, and K. Wang, A benchmark forvisual classification of clinical skin disease images. ECCV, 2016, pp.

[20] www.medium.com/@sidereal/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5

[21] Qi, J., Le, M., Li, C., Zhou, P. (2017). Global and Local Information Based Deep Network for Skin Lesion Segmentation.

[22] Yan, K., Wang, X., Lu, L., Summers, R. M. (2017). Deeplesion: Automated deep mining, categorization and detection of significant radiology image findings using large-scale clinical lesion annotations.

[23] Kieffer, B., Babaie, M., Kalra, S., Tizhoosh, H. R. (2017, November). Convolutional neural networks for histopathology image classification: Training vs. using pre-trained networks. In 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA) (pp. 1-6). IEEE.

[24] Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[25] You, Y., Zhang, Z., Hsieh, C. J., Demmel, J., Keutzer, K. (2017). 100-epoch imagenet training with alexnet in 24 minutes. ArXiv e-prints.

[26] Simonyan, K., Vedaldi, A., Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps.

[27] https://www.kaggle.com/competitions

[28] Christodoulidis, S., Anthimopoulos, M., Ebner, L., Christe, A., Mougiakakou, S. (2016). Multi-source transfer learning with convolutional neural networks for lung pattern analysis.

[29] Menegola, A., Fornaciali, M., Pires, R., Bittencourt, F. V., Avila, S., Valle, E. (2017, April). Knowledge transfer for melanoma screening with deep learning. In Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium on (pp. 297-300). IEEE.

[30] Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A. (2014). Learning deep features for scene recognition using places database. In Advances in neural information processing systems (pp. 487-495).

[31] Zintgraf, L. M., Cohen, T. S., Adel, T., Welling, M. (2017). Visualizing deep neural network decisions: Prediction difference analysis.

[32] Xue, D. X., Zhang, R., Feng, H., Wang, Y. L. (2016). CNN-SVM for microvascular morphological type recognition with data augmentation. Journal of medical and biological engineering, 36(6), 755-764.

[33] Ravishankar, H., Sudhakar, P., Venkataramani, R., Thiruvenkadam, S., Annangi, P., Babu, N., Vaidya, V., 2016. Understanding the mechanisms of deep transfer learning for medical images

[34] Kooi, T., Litjens, G., van Ginneken, B., Gubern-Mérida, A., Sánchez, C. I., Mann, R., ... Karssemeijer, N. (2017). Large scale deep learning for computer aided detection of mammographic lesions. Medical image analysis, 35, 303-312.

[35] Nasrabadi, N. M. (2007). Pattern recognition and machine learning. Journal of electronic imaging, 16(4), 049901.

[36] Ravishankar, H., Sudhakar, P., Venkataramani, R., Thiruvenkadam, S., Annangi, P., Babu, N., Vaidya, V. (2016). Understanding the mechanisms of deep transfer learning for medical images. In Deep Learning and Data Labeling for Medical Applications (pp. 188-196). Springer, Cham.

[37] Huynh, B. Q., Li, H., Giger, M. L. (2016). Digital mammographic tumor classification using transfer learning from deep convolutional neural networks. Journal of Medical Imaging, 3(3), 034501.

[38] Hoo-Chang, S., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., ... Summers, R. M. (2016). Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE transactions on medical imaging, 35(5), 1285.

[39] https://stackoverflow.com/questions/36515202

[40] stats.stackexchange.com/questions/179915/whats-the-difference-between-momentum-based-gradient-descent-and-nesterovs-acc

[41] Ruder, S. (2016). An overview of gradient descent optimization algorithms.

[42] Tajbakhsh, N., Shin, J. Y., Gurudu, S. R., Hurst, R. T., Kendall, C. B., Gotway, M. B., Liang, J. (2016). Convolutional neural networks for medical image analysis: Full training or fine tuning?. IEEE transactions on medical imaging, 35(5), 1299-1312.

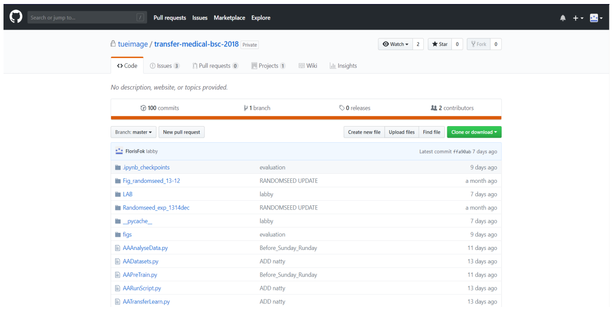
[43] Zweig, M. H., Campbell, G. (1993). Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine. Clinical chemistry, 39(4), 561-577.

[44] Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern recognition, 30(7), 1145-1159.

VII. Appendix

VIII. GitHub

https://github.com/tueimage/transfer-medical-bsc-2018



tueimage / transfer-medical-bsc-2018 Private

No description, website, or topics provided.

FlorisFok labby — Latest commit ffa90ab 7 days ago

.ipynb_checkpoints	evaluation	9 days ago
Fig_randomseed_13-12	RANDOMSEED UPDATE	a month ago
LAB	labby	7 days ago
Randomseed_exp_1314dec	RANDOMSEED UPDATE	a month ago
__pycache__	labby	7 days ago
figs	evaluation	9 days ago
AAAnalyseData.py	Before_Sunday_Runday	11 days ago
AADatasets.py	ADD natty	13 days ago
AAPreTrain.py	Before_Sunday_Runday	11 days ago
AARunScript.py	ADD natty	13 days ago
AATransferLearn.py	ADD natty	13 days ago

Figure 5: Snap shot of GitHub respiratory