Evaluating three Pre-trained Transfer Learning models for Pneumonia detection

Albin Thomsson, Viktor Smedberg and Maximilian Hilger

***Abstract*— Chest X-Rays are a medical imaging method used to detect pneumonia. The x-ray images have to be evaluated by a trained radiologist to determine if the patient is infected. To support the decision, medical computer vision techniques can be used. In modern computer vision, deep learning is the most common method. For applications with little training data, transfer learning can be used to apply a pre-trained model on different data. In this paper, we compare three different pre-trained models: GoogleNet, ResNet-50 and DenseNet. These are used on a dataset containing chest X-Rays of children. During training, the weights of these networks are fine-tuned to classify if the images contains pneumonia. We use t-SNE to visualize the features in different layers and GradCam to visualize on which image region the network bases its decision on. With the chosen approach, we are able to reach 93 percent accuracy for the ResNet-50 and comparable accuracies for the other networks.**

I. INTRODUCTION

Pneumonia is an inflammation in the lungs that is typically the result of a bacterial infection, but other example types are(but are not limited to): viral, hospital-acquired or fungal Pneumonia. The infection causes inflammation of the air sacs.

To diagnose whether or not a person is infected, one can use X-ray imaging, this is usually done by doctors for patients with more than mild symptoms. The data in these X-ray images can help a doctor to diagnose. The difference between a patient with Pneumonia and one without can be seen in these X-ray images but the difference is subtle, see Figure 1 for an example.

A common approach in many image classification tasks is to use convolutional neural networks (CNNs). Although prominent in the space, they do however require a large amount of data, which is something that is a common problem in the field of machine learning. A large set of labelled data by doctors and diagnosticians is rare, because it is expensive and tedious work is required. To overcome this issue since the dataset in this experiment is relatively small dataset, Transfer Learning and fine tuning was used. This allows the use of a model that has been pre-trained on a large amount of data.

II. RELATED WORK

Pneumonia detection with machine learning has been a known problem/research topic for many years, and there exists many papers on the topic with many different approaches.

Kundu et al. [4] created an ensemble of three convolutional neural network models: GoogleNet, ResNet-18, and DenseNet-121. Then by using a weighted average where the weights were set by some novel method. This method was tested and evaluated by four different metrics on data from two different publically available data sets and achieved an accuracy of 98.81% on Kermany et al [3] and 86.85% on RSNA [9]. They also performed statistical analysis on the ensemble method, which yielded positive results and showed viability for this approach. This approach is also domain-independent and can be applied to a large variety of image recognition tasks, not just pneumonia detection.

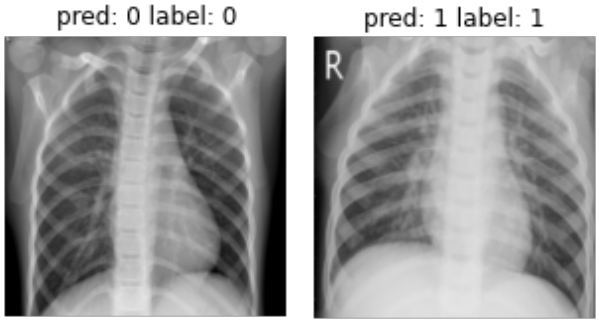


Fig. 1: Two example images from different classes showing the difficulty of correctly detecting pneumonia from two images. In this image, the left image is one of a normal lung X-Ray, while the one on the right is one with Pneumonia. The R in the image, means that this is the right side of the lung and it exists in most images to help doctors align the X-ray images accordingly.

Manickam et al. [5] evaluated different pre-trained models such as ResNet, InceptionNet,and InceptionResNet. In addition to this they also evaluated different optimizers: Adam and SGD and used visualization to help aid in fine tuning hyper parameters. To evaluate the results of the models with different hyper parameters (like learning rate and batch size) they used four different performance metrics(Accuracy, Precision, Recall, F1-Score,and ROC-AUC). The model with the best result, ResNet-50, reached an accuracy of 93.06%. This was compared to a multitude of other proposed and referenced models. Showing great promise for their approach.

III. METHODS

A. Data analysis

The dataset used consists of 5856 X-Ray images divided into two classes, "Pneumonia" and "Normal". The images are of children between the ages of 1-5 years old, and have

been diagnosed by two expert physicians and then evaluated by a third expert, Kermany et al [3].

Analysis of the dataset used in this experiment showed an initial train/test/val-split of 89/10.7/0.3, this was changed for every experiment in this paper into a 80/11/9-split by increasing the validation set by 10% of the train-set with stratified sampling to retain the distribution of the classes.

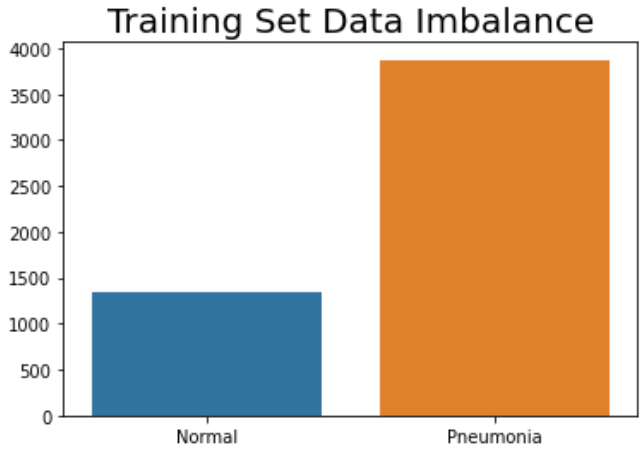


Fig. 2: Graph showing the train set imbalance.

Further, from the analysis an issue of data imbalance arose, see Figure 2. The dataset is biased towards data points of "Pneumonia". From figure 2 it is clearly visible that the training data set is imbalanced. The distribution of classes is 76% "Pneumonia" and 24% "Normal" in the train set, 50% "Pneumonia" and 50% "Normal" in the validation set and 63% "Pneumonia" and 37% "Normal" in the test set. In total there are $4273\ (73\%)$ "Pneumonia" X-ray images and $1583\ (27\%)$ "Normal" labeled X-ray images.

B. Handling data imbalance

There are many different ways to handle data imbalance, the method used was to add weight to the loss functions to the samples. This adds a weight to the calculated loss to counteract the imbalance. Another common way to handle this problem is to up-sample the minority class. Additionally to upsampling, the number of samples of a certain class can be enhanced by using data augmentation, for example by flipping images horizontally. For the given dataset, the augmentation should be used carefully. Each image has the letter R on the right side of the chest that would end up on the left side of the chest when flipping. In the worst case, the network would learn a detector for the flipped R on the other side to detect non-pneumonia cases. The approach with the least adjustments in the code to be made is adding a higher weight for the class with less samples. In this work, adding this weight is compared to using a naive approach that does not account for the imbalance.

C. Transfer Learning

Transfer learning is a technique used to train deep neural networks with pre-trained weights to shrink training time and allow training on less data. For CNNs, most networks are trained initially on ImageNet. ImageNet contains over 14 million images of different categories. Training the network on this dataset allows to learn abstract features. The key assumption of transfer learning is that these features are abstract enough to generalize also to data that is not used during training. To apply transfer learning to a new task, two approaches are commonly used: feature extraction and fine tuning. In feature extraction, the pre-trained networks weights are fixed such that the network just outputs a feature vector. This vector is used in a classifier network that is trained for the new task. In fine tuning, both the new classifier network and the pre-trained network are jointly trained. With this, the features can be adjusted to the new task. In this case, it is important to keep the learning low enough to not forget the previously learned features. For this work, fine tuning was chosen. As we saw in Kundu et al. [4], fine tuning yielded better results on the dataset compared to feature extraction. By this, the detected features are more suited towards the given task. As a drawback, it requires more computational resources because the gradient has to be back-propagated through the entire network.

D. Network architectures and hyperparameters

Three model architectures were used:

1) GoogleNet: GoogleNet is a CNN architecture published in 2014 by Szegedy et al. [7]. Its main feature is the usage of multiple identical inception modules. An inception module consists of parallel convolutions with different filter sizes and pooling. With this it can learn differently sized features. With the help of 1x1 convolutions, the outputs are resized such that the output size matches the input size.

2) ResNet: ResNet was introduced in 2015 by He et al [1]. ResNets incorporate Residual blocks. These blocks have skip connections that skip a layer. The skipped layer then learns the residual what leads to the name of this architecture. The skip connections allow the gradient to flow better through the network. With this, deeper architectures are possible. There exist several ResNet sizes. In this work, ResNet-50 is used.

3) DenseNet: DenseNet was published in 2016 by Huang et al. [2]. In DenseNets, each layer has directly access to every previous layer's activations. In contrast to ResNets, where the activations of the skip connection and the previous layers output are added element-wise, DenseNets use concatenations. With this, the number of feature maps grows with every layer. In this work, DenseNet-121 is used.

The networks were available pre-trained with the machine learning framework PyTorch. As classifier, a single fully connected layer is added after the last layer of the pre-trained network. Cross-Entropy loss was chosen as loss function. To cope with the data imbalance, it is possible to assign different weights to different classes. All models were tested both with and without higher weight for the samples without pneumonia. Additional, in the models with the weighted cost function, weight decay is used. All hyperparameters for the different experiments are given in Table I.

hyperparameter	GoogleNet	ResNet-50	DenseNet
batch size	32	16	32
learning rate	1e-5	1e-5	1e-5
weight decay	0 / 0.01	0 / 0.01	0 / 0.01
class weights	[1,1]/[100,1]	[1,1]/[100,1]	[1,1]/[100,1]

TABLE I: hyperparameters for models, without/with weighted classes in loss function

E. Evaluation Metrics

The most commonly used evaluation metrics are loss and accuracy. While these work well for balanced datasets, they have some drawbacks when applied to imbalanced data. If applied on the training data, classifying everything would already yield an training accuracy of 74.3%. A better score for imbalanced data is the F1-score. It is calculated using precision and recall. The precision gives a ratio how much of the positively tested samples are true positive. The recall calculates how many of the positive samples where detected correctly. Combined via the harmonic mean, they give a performance metrics that penalizes extreme values more than accuracy does. An even more intuitive metric is given by the confusion matrix. From it, it can directly observed how many false positives and false negative are classified by the network. The confusion matrix is used at test time to finally assess which of the network candidates performs the best.

F. t-SNE

T-Stochastic Neighbourhood Embedding (t-SNE) is a method to project high-dimensional data into a low-dimensional space introduced by van der Maaten and Hinton [8]. It can be used to project the activations of a layer into a two-dimensional space. The aim of t-SNE is to keep small distances in the high-dimensional space small in the low-dimensional space. It works by calculating a normal distribution over the data points in the high dimensional space. In the low-dimensional space, a second probabilty distribution is constructed using the Student-t distribution. The final lower-dimensional distribution is calculated via minimization of the Kullback-Leibler divergence between the both distribution. This can be done via gradient descent. Thus t-SNE can form distinct clusters that contain similar data points.

G. Grad-CAM

Gradient-weighted Class Activation Mapping (Grad-CAM) is a technique used to highlight the region in an input image that has a high influence in the output of the network and was developed by Selvaraju et al. [6] . It works as follows: Initially, a network layer has to be chosen. The visualization will show the gradients of the activations in this layer. Then, the input image is passed through the network. On the backward pass, the gradient is only back-propagated to the selected layer. There, each feature map of the layer is weighted according to the sum of its gradients. The final heat map is constructed by weighted summation of the activations in the channels and application of ReLU to only keep positive gradients. To overlay the heatmap with the input image, upsampling is used. Usually, the last convolutional layer of the network is chosen to apply Grad-CAM because it learns high-level features that are easier to understand.

IV. RESULTS

A. Model comparison

For the candidates with weighted loss functions, ResNet-50 trains faster with a better loss value. DenseNet initializes worse, but can reach almost the same loss as ResNet-50. GoogleNet initializes well, but does not improve much during training.

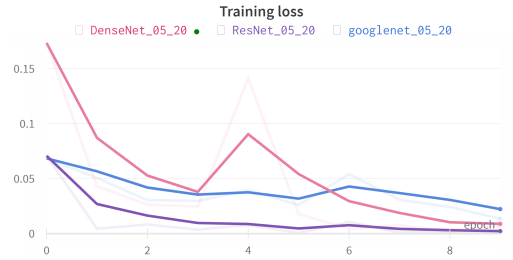


Fig. 3: training loss with weighted loss function

Comparing without weighted loss function the characteristics for the different models behave similar. Both ResNet-50 and DenseNet trains faster, but seems to reach a plateau at around epoch 8 and forward. DenseNet and GoogleNet is still decreasing but ResNet-50 starts increasing slightly.

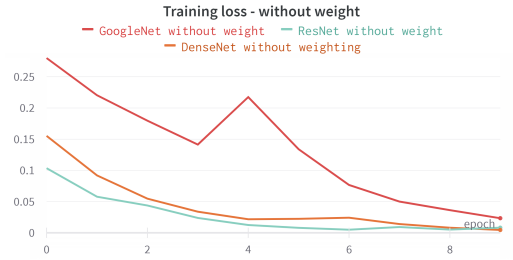


Fig. 4: training loss without weighted loss function

In figure 5 the validation accuracy over training time is shown for the loss function without weighting.

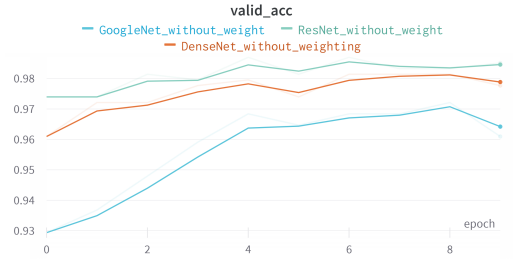


Fig. 5: validation accuracy without weighted loss function

The resulting values for the test accuracy for both weighted and non-weighted loss functions are shown in figure 6 and 7. While the weighting increases the accuracy for the ResNet-50, it harms the performance of the GoogleNet and DenseNet-121.

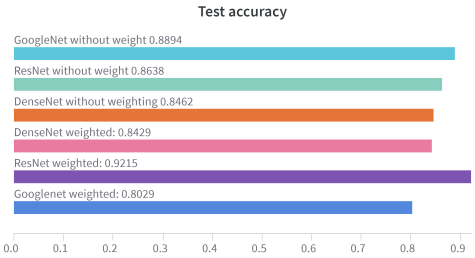


Fig. 6: test accuracy without weighted loss function

Model name	Epoch	F1-score	Accuracy	AUROC
GoogleNet – without weighting	10	0.875450849533081	0.8894230723381042	0.9678939580917358
ResNet - without weighting	10	0.8408197164535522	0.8637820482254028	0.964162826538086
DenseNet – without weighting	10	0.8178434371948242	0.8461538553237915	0.9443731904029846
DenseNet - weighted	10	0.8123434782028198	0.8429487347602844	0.938001275062561
ResNet - weighted	10	0.913726270198822	0.9214743375778198	0.9737398624420166
GoogleNet - weighted	10	0.801854133605957	0.8028846383094788	0.9536927342414856

Fig. 7: Performance summary for the different models

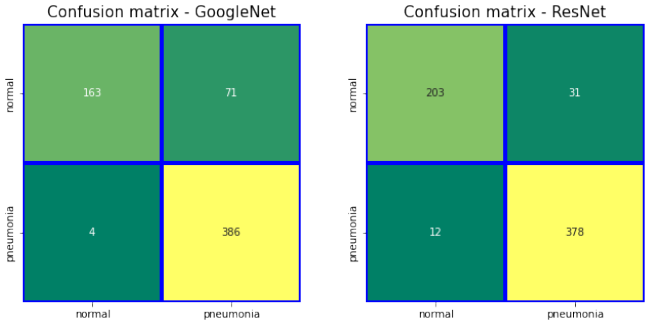
When looking at figure 5, it is obvious that the accuracy is not strictly increasing. Therefore, early stopping was examined to look for possible better results. For this purpose, ResNet-50 was chosen as it performed the best even without early stopping. With early stopping, the model performance could be improved a little bit more, giving an 93.1% accuracy for the test data.

To compare the effect of the weighted loss function on false positives and false negatives, the confusion matrices of the best-scoring non-weighted model and the best-scoring weighted model are shown in figure 8. As you can see in the figure, weighting reduces the number of false positives drastically, but this leads also to more false negatives. Overall, the accuracy is improved by this.

B. GradCam

An example result of the Grad-CAM is shown in figure 9. It is organized in an confusion matrix-like style. High activations are highlighted with red color, low activations with blue.

It can be observed that in the incorrectly classified images the networks tends to look into regions close to the edges of the edges. In the false positive case, it is looking at the throat and in the false negative case it is looking at the bottom of the image. In both cases, the activation at the region where the lung lies is low. In the correctly classified images, the largest activations contain the regions where the lung is located.



(a) Confusion Matrix GoogleNet without weighting (b) Confusion Matrix ResNet-50 with weighting

Fig. 8: Confusion Matrices

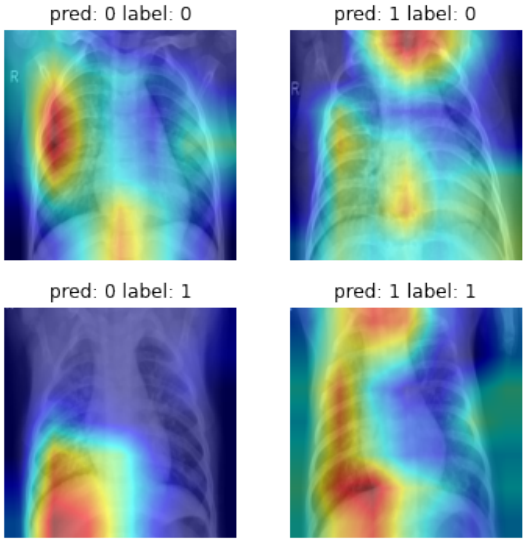


Fig. 9: Grad-CAM results

C. t-SNE

In the trained network, the features enter the classifier after flattening. To visualize the grouping of the features, t-SNE can be applied on the activations of the flattening layer. A t-SNE visualization of the features of the trained ResNet-50 is given in figure 10.

In the plot, it is visible that both classes build groups. The groups do not form distinct clusters, but are overlapping. Especially in this overlapping area misclassifications occur frequently. Some samples end up entirely in the other group. This is most likely due to insufficient feature extraction. The classifier seems to discriminate well between the clusters.

V. DISCUSSION

The results show that transfer learning is a suitable approach for medical image processing. In this study, all models performed well for the given task. Comparing the different models is difficult, as they all have slightly different hyperparameters and are not fully optimized. This is due

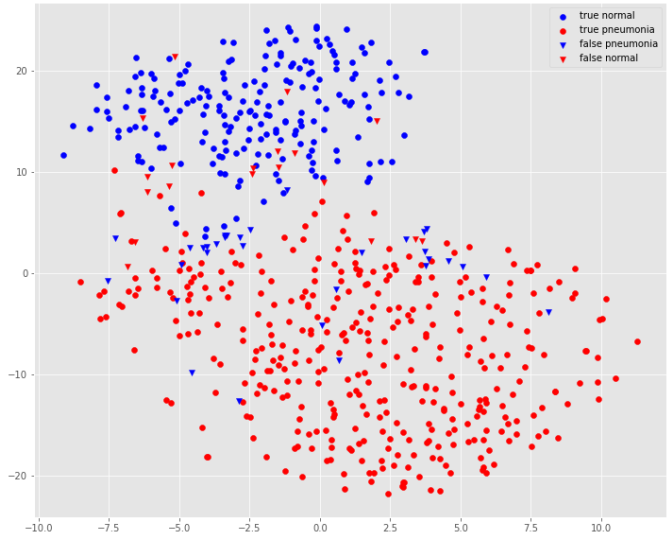


Fig. 10: t-SNE of ResNet-50. Correctly classified shown as dots, misclassified as triangles

to time- and resource limitations. Also, due to randomness and on the fly augmentation from the machine learning framework PyTorch the models employ different degrees of augmentation. Every single model might perform better if the hyperparameters were tuned better, by running more experiments on a cluster instead of a local machine or Googles Colaboratory environment. In this work, weighted ResNet-50 performs slightly better than the other architectures, something that is also seen in related work, see Section II. This is also based on the additional tuning effort put into this network. The other networks also showed promising results. From our limited data, and especially test data this model shows some promise, but we expect that the accuracy will decrease further for new data which means that the model might not generalize well.

Another insight is that using weights in the loss function can help with imbalanced data, but has to be tuned properly. While it increased test accuracy for the ResNet-50, it decreased it for GoogleNet and DenseNet.

VI. CONCLUSIONS

In this paper we have evaluated and compared different fine-tuned pre-trained models using two main performance metrics, F1-score and accuracy. The results were that the highest performing model, weighted ResNet-50, obtained an accuracy of 93.1% and an F1-score of 92.51%. We then examined this model in more detail using Grad-CAM and t-SNE visualization, to get an insight into the model. We also could have made use of cross-validation to merge the training and the validation sets to get our hyperparameters from this method instead of experimentally, this would however take more time and but could be done in future work.

REFERENCES

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[2] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

[3] Daniel S. Kermany, Michael Goldbaum, Wenjia Cai, Carolina C.S. Valentim, Huiying Liang, Sally L. Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, Justin Dong, Made K. Prasadha, Jacqueline Pei, Magdalene Y.L. Ting, Jie Zhu, Christina Li, Sierra Hewett, Jason Dong, Ian Ziyar, Alexander Shi, Runze Zhang, Lianghong Zheng, Rui Hou, William Shi, Xin Fu, Yaou Duan, Viet A.N. Huu, Cindy Wen, Edward D. Zhang, Charlotte L. Zhang, Oulan Li, Xiaobo Wang, Michael A. Singer, Xiaodong Sun, Jie Xu, Ali Tafreshi, M. Anthony Lewis, Huimin Xia, and Kang Zhang. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5):1122–1131.e9, 2018.

[4] Rohit Kundu, Ritacheta Das, Zong Woo Geem, Gi-Tae Han, and Ram Sarkar. Pneumonia detection in chest x-ray images using an ensemble of deep learning models. *PLOS ONE*, 16(9):1–29, 09 2021.

[5] Adhiyaman Manickam, Jianmin Jiang, Yu Zhou, Abhinav Sagar, Rajkumar Soundrapandiyan, and R. Dinesh Jackson Samuel. Automated pneumonia detection on chest x-ray images: A deep learning approach with different optimizers and transfer learning architectures. *Measurement*, 184:109953, 2021.

[6] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[7] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[8] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[9] X Wang, Y Peng, L Lu, Z Lu, M Bagheri, and R Summers. Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *IEEE CVPR*, volume 7, 2017.