Transfer Learning for COVID-19 Detection

Mikael Ljung (milju@kth.se) **Johanna Dyremark (dyremark@kth.se)**

Hannes Kindbom (hkindbom@kth.se) **Ershad Taherifard (ershadt@kth.se)**

Abstract

Artificial Intelligence has demonstrated great potential in areas such as radiology. During the pandemic of COVID-19, governments around the world are enacting policies that try to mitigate the spread of the virus. Latest technologies are applied to track, diagnose and treat COVID-19 patients. This paper applies the VGG-16 architecture to classify patients' chest X-Ray images as either *normal*, *pneumonia* or *COVID-19*. The concept of transfer learning is examined, to overcome the challenge of having a small dataset due to the limited number of COVID-19 X-ray scans so far. By using the pre-trained VGG-16 model, that was trained on ImageNet, a varying number of layers could be trained on COVIDxPLUS, a set of 15150 publicly available CXR images. The model performance was also evaluated on a smaller subset called COVIDxMINI. Four types of experiments, with varying degrees of transfer learning on the VGG-16 model, were conducted. The highest achieved test accuracy was 83.5% on COVIDxPLUS with a precision and recall of 100% and 22.6% respectively on the COVID-19 class. However, the highest recall of 90.3% was achieved on COVIDxMINI. The hope is that the insights gained in this project, could further push the development of deep learning models for detecting COVID-19 and thus facilitate better treatment for the patients.

1 Introduction

The world has been severely disrupted by a pandemic which has forced governments around the world to enact policies at an unprecedented scale to try to slow down the spread of this novel coronavirus. However, the measures in place have caused economic turmoil and rising unemployment. There are several know coronaviruses that causes respiratory infections and they vary in severity. The latest coronavirus, the SARS-CoV-2 virus results in what to today is know as COVID-19. By practicing social distancing and good personal hygiene, the number of infected people can be kept at manageable levels so that the health care system do not get overwhelmed. Also, by tracking and isolating infected cases, the spread of the virus can be further slowed down. The standard way of testing today is laboratory testing, more specifically reverse transcriptase-polymerase chain reaction (RT-PCR) testing. But it is time-consuming and the process is manual. A faster alternative is machine learning in medical image-based testing which uses computed tomography (CT) scans, MRI, X-rays, etc to determine whether a patient is infected or not. The models are trained to detect abnormalities in chest radiography images, that characterizes COVID-19 patients. When there is a need for professional radiologists, machine learning can play a vital role in assisting classification of medical images [5].

However, one issue with applying deep learning methods is the need for large high quality datasets and a large supply of CXR images is not yet the case for the novel coronavirus outbreak. Instead other methods must be applied. For instance, by using previous datasets of similar medical images together with a small set of COVID-19 images, the parameters of the model can be fine tuned and still produce good results despite this constraint.

Project report for DD2424

The purpose of this project is to compare how different levels of transfer learning can successfully be used to classify COVID-19 CXR images. Apart from this, the performance of transfer learning on a small compared to a larger dataset, will also be evaluated.

2 Background

2.1 Related Work

There have been many initiatives to battle this pandemic by leveraging technology. Despite the novelty of the virus, there have been efforts to try to streamline the treatment process and speed up medical diagnostics using machine learning. In [16] a tailored CNN, called COVID-Net was used to detect the disease. It was trained on 13,800 chest X-ray (CXR) images that was open source and available to the general public. The data set is referred to as COVIDx and it is comprised of a mix of normal, non-COVID-19 infections and COVID-19 infections. It achieved 92.6% test accuracy. The model sensitivity was further examined for each infection type and it was found that the recall for COVID-19 was 87.1% but it should be noted that the number of such patient cases is limited. This is the main paper that will be referred to and finally benchmarked against.

Another paper proposed a COVID-RENet which incorporated edge and localization based analysis on COVID-19 image data. The proposed approach was a concatenation of the COVID-REnet and a custom VGG model, which together achieved the best performance. 98.2% accuracy was reached however on a quite small dataset compared to other papers. The authors further discuss the disparities in accuracy among conduced studies and argue that the lack of a consolidated data repository results in many models being evaluated on different distributions of the dataset [12].

2.2 Dataset

To be able to benchmark the various prediction scores of the developed models, the same dataset as in the reference paper [16], was used. However, additional images were gathered to form a validation set, used during model building. This extended dataset, which is referred to as COVIDxPLUS, consists of 15150 CXR images. The images were originally gathered, using the provided generation script, from three different publicly available data sources: 1) COVID-19 Image Data Collection [6], 2) COVID-19 Chest X-ray Dataset Initiative [1] and 3) RSNA Pneumonia Detection Challenge dataset [10]. COVIDxPLUS is partitioned into three different classes: *normal*, *pneumonia* and *COVID-19* and the distribution of data over the train, validation and test sets is shown in table 1. The class imbalance is noticable and the lack of publicly available COVID-19 CXR images increases the difficulty of the problem. In order to examine the impact of such imbalance as well as the requirements on amount of data in transfer learning, an alternative, smaller dataset was extracted from the COVIDxPLUS dataset. The dataset COVIDxMINI, was sampled from the COVIDxPLUS to achieve a more balanced class distribution as shown in table 2. During the implementation, the normal, pneumonia and COVID-19 class have been labelled 0, 1 and 2 respectively.

Table 1: Class Distribution of COVIDxPLUS

	Normal (0)	Pneumonia (1)	COVID–19 (2)	Total
Train	7966	5451	152	13569
Validation	785	494	71	1350
Test	100	100	31	231

Table 2: Class Distribution of COVIDxMINI

	Normal (0)	Pneumonia (1)	COVID–19 (2)	Total
Train	200	200	152	552
Validation	100	100	71	271
Test	100	100	31	231

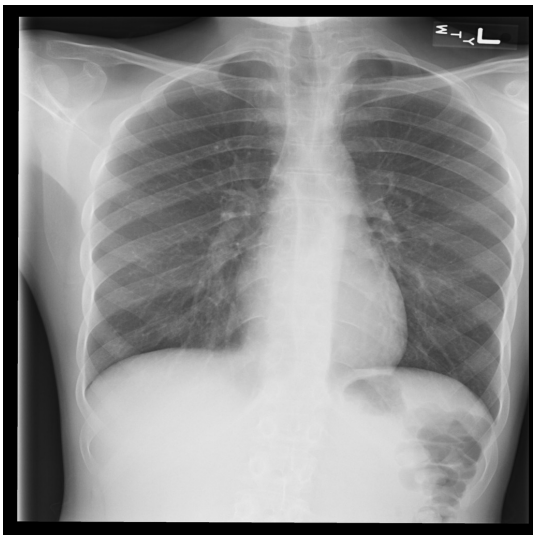


Figure 1: Example CXR image

2.3 Transfer Learning

There are many machine learning methods applied in the real-world, but often assumptions made in theory do not hold in reality. The notion of *data dependence* is one of the greatest hurdles facing the field of deep learning. Training data are in many circumstances difficult to obtain and deep learning is dependent on massive training sets. That is why the concept of transfer learning is applied, where data from different domains are use in a new setting. That means that the pre-trained network weights are used as initialisation and they are fine tuned with a smaller set you aim to learn [7]. The idea behind transfer learning is to relax the assumption that training data must be independent and identically distributed with the test data. That is, the model must not be trained on data from scratch in the target domain but instead pre-trained on some other similar dataset. The knowledge from the source domain is possible to transfer to the target domain by relaxing the independence assumption [13].

One example of a dataset, often used as source data, is the ImageNet which is a large dataset that is used to identify edges, textures, shapes, and object composition. It contains 10 million images and objects distributed over 1000 categories, which makes it a useful source dataset that can be migrated to a target data set [17].

2.3.1 Notation and Definitions

Let a domain be represented as $\mathcal{D} = \{\mathcal{X}, P(X)\}$, where $\mathcal{X}$ is the feature space, $P(X)$ is the marginal probability distribution and $X = \{x_1, ..., x_n\} \in \mathcal{X}$. A prediction can be represented as $\mathcal{T} = \{y, f(x)\}$where y is the label space and $f(x)$ is the prediction function.

Given a learning task $\mathcal{T}_t$ based on a target dataset $\mathcal{D}_t$, one can utilize a source dataset $\mathcal{D}_s$ for the learning task $\mathcal{T}_t$. The goal is to improve the performance of the predictive function $f_{\mathcal{T}}$ for a learning task $\mathcal{T}_t$ by discovering and transferring the latent knowledge from $\mathcal{D}_s$ and $\mathcal{T}_s$ where $\mathcal{D}_t \neq \mathcal{D}_s$ and/or $\mathcal{T}_t \neq \mathcal{T}_s$. Often $\mathcal{D}_s$ is much larger than $\mathcal{D}_t$ [13]. In this case, $\mathcal{D}_t$ is COVIDxPLUS or COVIDxMINI and $\mathcal{D}_s$ is the ImageNet.

2.3.2 Fine Tuning

The idea behind fine tuning is to take the weights of a trained model and use it to initialize a new model that is trained on a different dataset. This is suitable if the target data are limited or if one wants to speed up training. To use the pre-trained model, a number of layers are frozen depending on different factors such as datasize and data similarity. There are four steps to fine tuning. First a neural network, the source model, is pre-trained on $\mathcal{D}_s$. Then a new model, the target model, is created. The target is a replicate of the source model except for the output layer. The assumption is that the knowledge learned from the source dataset $\mathcal{D}_s$ is applicable on the target dataset $\mathcal{D}_t$. Thus

the learned parameters can be copied to the new model, as can be seen in figure 2. The size of the output layer is the number of classes in $\mathcal{D}_t$. Finally the target model is trained on $\mathcal{D}_t$. That is, the unfrozen layers are trained from scratch while the parameters of the previous layers are already set from the source model [17].

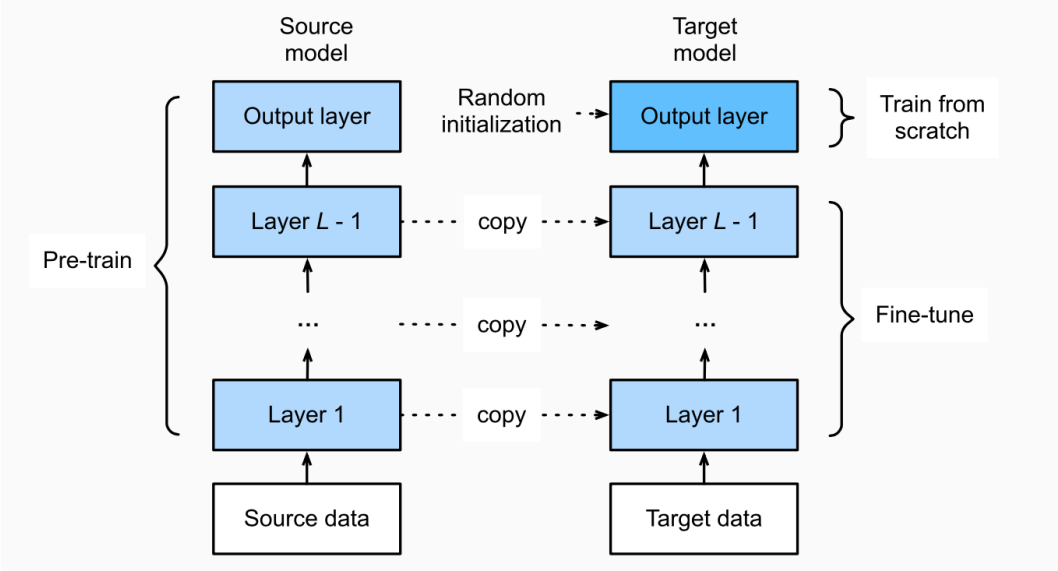


Figure 2: Graphical illustration of how fine tuning works

2.4 CNN and VGG

There are several types of Neural Networks and the architecture that has shown superior performance in Image Processing are Deep Convolutional Neural Networks (CNN). CNNs use feature extraction to automatically learn representations from the data and one vital factor is the special architecture of the model.

One of the most widely used CNNs is the VGG. There are two main configurations of this model; VGG-16 uses 16 layers whilst VGG-19 uses 19 layers. Both are specifically trained on the ImageNet dataset. The complexity of the model is regulated by using 1x1 convolutions in between the convolutional layers. After each convolutional layer, a max-pooling is placed. At the final block, there is a fully connected classifier, where the majority of all the calculated parameters are generated.

The VGG architecture is known for its simplicity, homogeneous topology and increased depth shown in figure 3. It manages to solve classification and localization problems, both important in medical image recognition. However, the greatest restriction is its use of 138 million parameters, which makes it computationally expensive [8].

3 Method

3.1 Packages

A pre-trained VGG-16 model was used and fine-tuned on the COVIDxPLUS and COVIDxMINI datasets. The model was created in Python by using Tensorflow's high-level API, Keras. The implementation was inspired by *Transfer Learning in Keras with Computer Vision Models* and *An attempt- Detection of COVID-19 presence from Chest X-ray scans using CNN Class Activation Maps 2*. [4]. Several alternative methods for transfer learning in COVID-19 detection where discussed, in particular building a CNN model from scratch without any pre-training. However, early results achieved with a pre-trained model and fully trainable layers (see experiment 2 and 4) indicated that the amount of data would not be sufficient enough to train such model. There are multiple network architectures for deep learning object recognition available in Python, but VGG-16 is considered to

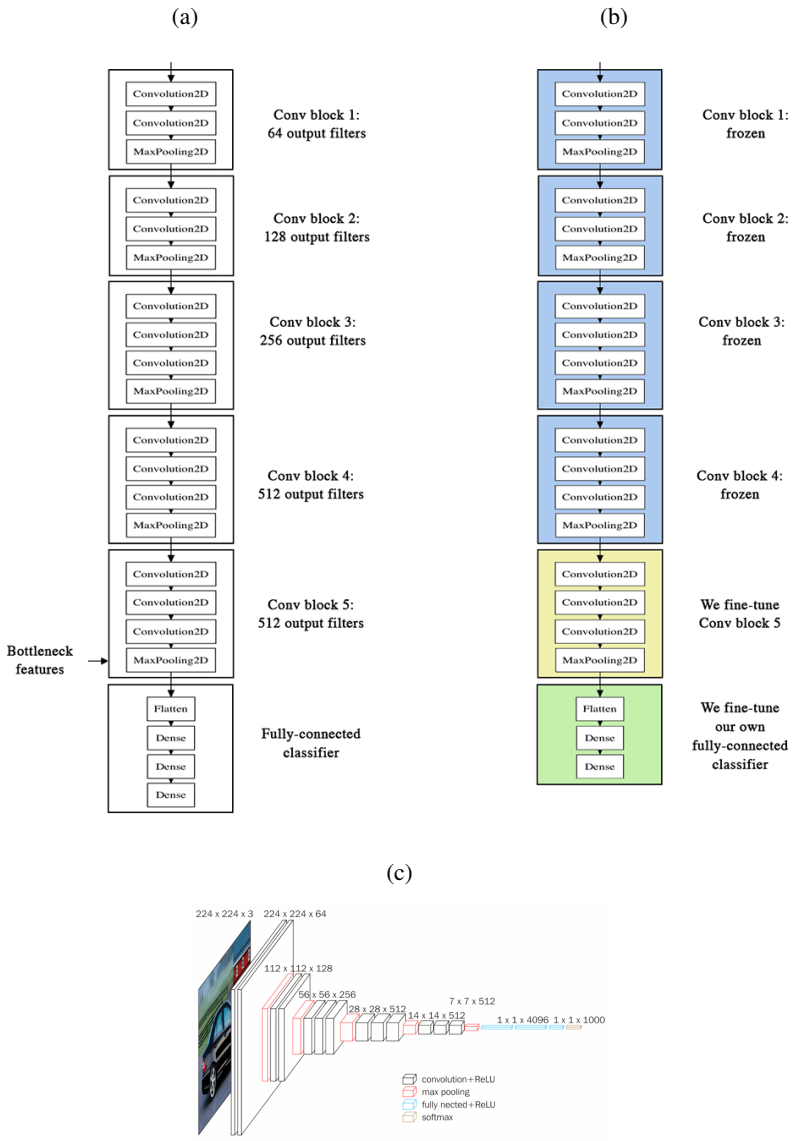


Figure 3: The architecture of VGG-16[15][4]

.

be one of the excellent modern vision architectures to date while still offering a rather straightforward implementation with relatively few hyperparameters. [14]

3.2 Optimizers

To fit the model, Adam optimizer was used. It is based on a type of stochastic gradient descent method. The method is computationally efficient, has little memory requirement and is good for problems that are large in terms of data/parameters [3].

3.3 Data Augmentation

It is widely accepted that bigger datasets result in better deep learning models. But there are often constraints on how much data is available. The manual effort of labeling and collecting data can be quite difficult to overcome. This is certainly true in medical image datasets due to the rarity of diseases, patient privacy and medical expert labeling. To mitigate this problem some data augmentation was

performed on the COVIDxPLUS dataset as well as on the more balanced but smaller COVIDxMINI dataset. By doing so, a larger set of samples was created to train the model on which improves generalization [9].

In this report, rotation, translation and scaling have been applied to transform the original X-ray images. Table 3 contains parameter specific information about this process, which has been replicated from the reference paper [16].

Figure 4: Example of an augmented image that has been translated and rotated

Table 3: Augmentation settings

	Value
Rotation	10 degrees
Translation width	0.1
Translation height	0.1
Scale	1/255

3.4 Class Weights

When handling imbalanced datasets where a miniority class is more important to the classification problem, a larger error weighting can be applied to a minority class. [2] The class weights used in experiments with the COVIDxPLUS dataset have thus been set to 1:1:12 for normal, pneumonia and COVID-19 samples respectively, identically to the class weights being implemented in the reference paper [16]. This applies a stronger penalty of wrongly classifying a COVID-19 sample than wrongly classifying a normal or pneumonia sample. As the COVIDxMINI dataset is significantly more balanced, experiments on this dataset applied 'balanced' class weights computed through Sci-Kit Learn, with each class weight being calculated as $\mathcal{N}/(\mathcal{K} * \mathcal{Y})$, where $\mathcal{N}$ is the number of training samples, $\mathcal{K}$ is the number of classes and $\mathcal{Y}$ occurrences of the specific class [11].

3.5 Parameters and Model Choice

3.5.1 Final model

All experiments were conducted using a target model consisting of a pre-trained VGG-16 model with initial weights 'ImageNet' as source model and four additional layers: a flattening layer, two fully-connected layers with 1024 ReLu units and a final layer with three softmax units. These four last layers have thus not been pre-trained and are trained from scratch in all experiments. The proposed

deep learning model, shown in figure 5, is publicly available at `https://github.com/hkindbom/Transfer-Learning-COVID-19-Detection`.

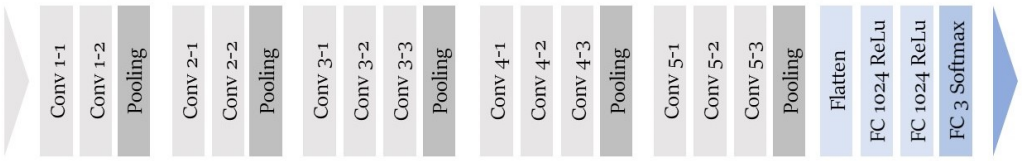


Figure 5: Final model architecture

3.5.2 Parameters

The experiments used initial learning rate=2e-5 and batch size=8. The Adam optimizer was applied with a learning rate policy where the learning rate is reduced by a 'factor' once learning stagnates for a 'patience' number of epochs, using parameters patience=5, factor=0.7, monitor='loss'. The parameter settings were inspired by the implementation in the reference paper [16].

3.6 Experiments

Four experiments (see table 4) were conducted using different combinations of datasets, class weights, number of epochs and number of trainable layers. Since experiments using the smaller COVIDxMINI dataset required less epochs to reach a converging accuracy, 12 epochs were used for the COVIDxMINI while 22 epochs were used in experiments with the COVIDxPLUS dataset. To investigate the impact of freezing layers, all layers except the last four, not pre-trained, layers were frozen in experiment 1 and 3. In experiment 2 and 4, no layers where frozen and all 22 layers thus trainable. The results were compared to the ones in the reference paper [16] to assess model performance.

Table 4: Experiment settings

	Dataset	Class Weights	Epochs	Trainable Layers
Experiment 1	COVIDxPLUS	1:1:12	22	19–22
Experiment 2	COVIDxPLUS	1:1:12	22	1–22
Experiment 3	COVIDxMINI	Balanced	12	19–22
Experiment 4	COVIDxMINI	Balanced	12	1–22

4 Results

4.1 Experiments

As seen in figure 6a, experiment 1 resulted in training accuracy reaching above 90% whilst the validation accuracy is oscillating at around 86%. In 6c the validation loss is decreasing significantly and stabilizing after 10 epochs and increases slightly towards the end. This suggests that the model might be somewhat overfitted. When examining the confusion matrix it shows that the model manages to differentiate normal medical images and pneumonia images. However, 19 test images that were COVID-19 positive, were classified as pneumonia. 5 positive COVID-19 cases were classified as normal. The final accuracy in experiment 1 is 83.5%. In experiment 2, all layers were trained and as shown in figure 7b, the model performs badly. All images were classified as normal and the model basically never learns. The accuracy for experiment 2 is 43.3% which is exactly the share of normal data in COVIDxPLUS.

In experiment 3 and 4, COVIDxMINI was analyzed instead and similar results were achieved. In experiment 3, shown in figure 8a, the training accuracy go beyond 80% whilst the validation accuracy stays at around 75%. The training loss does also stabilize. Unlike experiment 1, this model performs worse on normal data as it tends to predict normal cases as pneumonia cases. However, this model

is better at distinguishing pneumonia cases from COVID-19 cases. Only 3 COVID-19 cases were classified as pneumonia cases and 6 pneumonia cases were classified as COVID-19 cases. The accuracy for experiment 3 is 77.9%. In the final experiment 4, the results were poor as in experiment 2. In both these models, all layers were trainable. In this final experiment all images were classified as pneumonia, as show in figure 9b. The accuracy for experiment 4 is the share of pneumonia images, which is 43.3% as in experiment 2.

In conclusion, the produced scores were overall slightly lower than in the related work, presented earlier. Our best test accuracy was for instance 83.5%, compared to the other papers at 92.6% and 98.2%. However worth to point out is that the COVID-19 recall in experiment 3 is higher than that in the reference paper [16]. Finally, the number of trainable layers seems to have a great impact on the model performance. In experiment 2 and 4, where all layers where trainable, no useful learning has been done.

Experiment	Accuracy (%)
1	86.9
2	43.4
3	77.9
4	43.4

Table 5: Table describing the accuracy of all the experiments

Experiment	Precision (%)	Recall (%)
1	100.0	22.6
3	75.7	90.3

Table 6: Table describing precision and recall on the COVID–19 class from the top performing models

5 Conclusion

In this study, several aspects of transfer learning, based on the VGG-16 model, have been investigated. The effects of the amount of target data as well as the number of trainable layers were studied in four experiments. Comparing the results on the larger dataset COVIDxPLUS to the smaller COVIDxMINI, it seems that similar scores can be achieved independently of the amount of target data. Elaborating with data augmentation as well as different class weights and epochs was crucial in achieving high performance on both datasets. The highest test accuracy on COVIDxPLUS was higher (83.5%) than that on COVIDxMINI (77.9%), whereas the model trained on COVIDxMINI shows better recall (90.3%). It may be argued that recall is particularly important in this application, since it is important to detect and provide treatment for those infected with COVID-19. Regarding the level of transfer learning, it can be observed that training all 22 layers of VGG-16 yielded significantly worse performance than training only the last four, as can be observed in figures 7b and 9b. This is most likely a result of too little target data, but may be explored further. Lastly, is should be emphasized that this is by no means a production ready model, especially since the reference paper [16] achieved overall better performance on the same test set. Nonetheless, this study may serve as a foundation for further exploration of using small CXR datasets in transfer learning. It seems like high scores can be achieved with little data if the weighing and number of trainable layers are adjusted accordingly. Those insights may be useful to develop a model quickly next time a similar virus is encountered.

References

[1] Chung et al. *COVID-19 chest x-ray data initiative*. 2020. URL: https://github.com/agchung/Figure1-COVID-chestxraydataset.

[2] Jason Brownlee. *Imbalanced classification with Python*. Machine Learning Mastery, 2020, p. 229.

[3] François Chollet. *Adam Optimizer*. https://keras.io/api/optimizers/adam/s. 2015.

[4] François Chollet. *Building powerful image classification models using very little data*. https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.htm. 2016.

[5] *Diagnosing COVID-19 using AI-based medical image analyses*. 2020. URL: https://www.quantib.com/blog/diagnosing-covid-19-using-ai-based-medical-image-analyses.

[6] Paul Morrison Joseph Paul Cohen and Lan Dao. *COVID-19 image data collection*. 2020. URL: https://arxiv.org/abs/2003.11597.

[7] Taghi M. Khoshgoftaar DingDing Wang Karl Weiss. "A survey of transfer learning". In: *Journal of Big Data* 3.9 (2016). DOI: https://doi.org/10.1186/s40537-016-0043-6.

[8] Asifullah Khan et al. *A Survey of the Recent Architectures of Deep Convolutional Neural Networks*. 2020. URL: https://arxiv.org/pdf/1901.06032.pdf.

[9] Connor Shorten Taghi M. Khoshgoftaar. *A survey on Image Data Augmentation for Deep Learning*. 2019. URL: https://link.springer.com/article/10.1186/s40537-019-0197-0#Abs1.

[10] Radiological Society of North America. *RSNA pneumonia detection challenge*. 2019. URL: https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data.

[11] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[12] Anabia Sohail et al. *Coronavirus Disease Analysis using Chest X-ray Images and a Novel Deep Convolutional Neural Network*. Apr. 2020. URL: https://www.researchgate.net/publication/340574359_Coronavirus_Disease_Analysis_using_Chest_X-ray_Images_and_a_Novel_Deep_Convolutional_Neural_Network.

[13] Chuanqi Tan et al. *A Survey on Deep Transfer Learning*. 2018. URL: https://arxiv.org/pdf/1808.01974.pdf.

[14] Rohit Thakur. *Step by step VGG16 implementation in Keras for beginners*. 2019. URL: https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c.

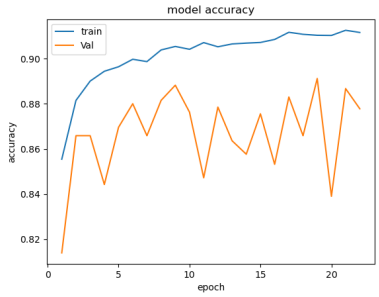
[15] *VGG16 – Convolutional Network for Classification and Detection*. 2020. URL: https://neurohive.io/en/popular-networks/vgg16/.

[16] Linda Wang, Zhong Qiu Lin, and Alexander Wong. *COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images*. 2020. URL: https://arxiv.org/pdf/2003.09871v3.pdf.

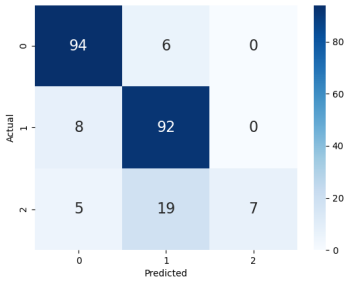
[17] Aston Zhang et al. *Dive into Deep Learning*. https://d2l.ai. 2020.

6 Appendix

(a) Accuracy plot



(b) Confusion matrix (test data)



(c) Loss plot

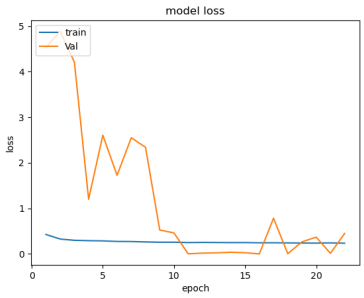
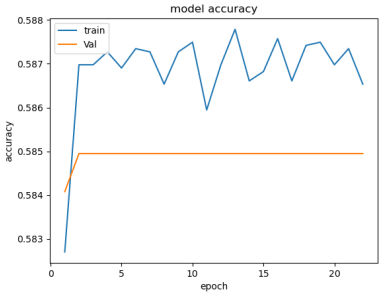
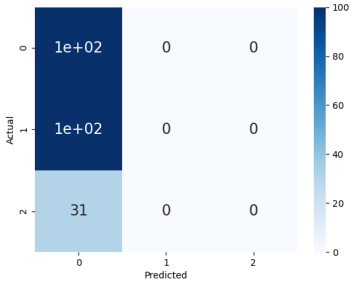


Figure 6: Experiment 1

(a) Accuracy plot



(b) Confusion matrix (test data)



(c) Loss plot

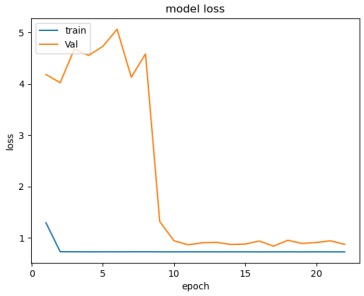
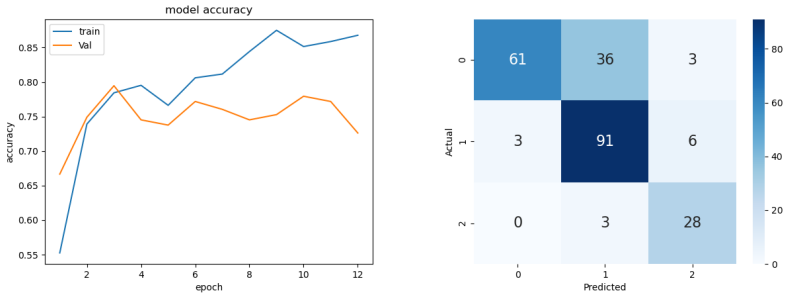


Figure 7: Experiment 2

(a) Accuracy plot (b) Confusion matrix (test data)



(c) Loss plot

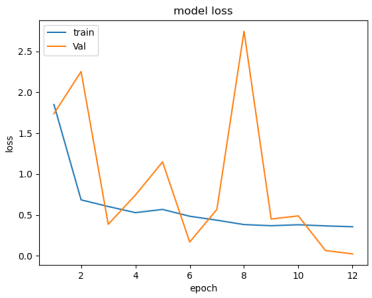
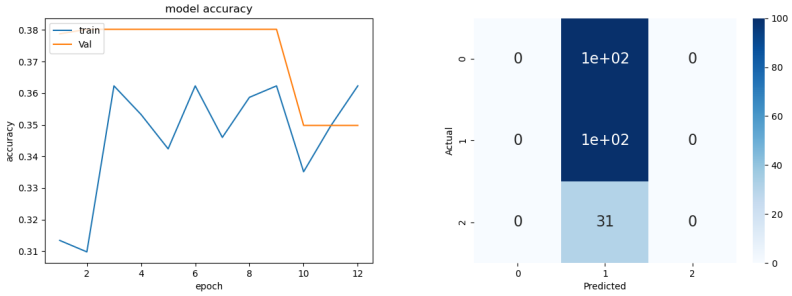


Figure 8: Experiment 3

(a) Accuracy plot (b) Confusion matrix (test data)



(c) Loss plot

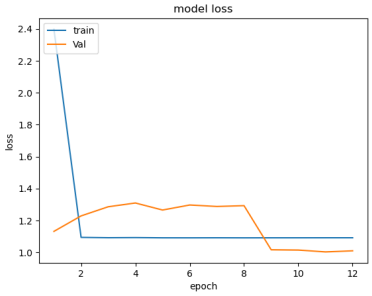


Figure 9: Experiment 4