COVID-19 Classification in Chest Radiographs with Deep Convolutional Neural Networks using Transfer Learning

Jim Cuijpers[1][2572680], Leyu Liu[1][2630429], and Liselore Borel Rinkes[2][11168307]

[1] VU University Amsterdam, De Boelelaan 1105, 1081 HV Amsterdam
[2] University of Amsterdam, Science Park 904, 1098 XH Amsterdam

1 Introduction

Being in the midst of the COVID-19 pandemic, insight into the spread and recovery of the virus is of significant importance. With nearly 5.1 million confirmed cases and a death toll of approximately 330 thousand globally at the time of writing, such insights could save the lives of many. Considering the fact that in reality these numbers are likely to be substantially higher, the immediate urgency becomes even more apparent. Due to the growth of publicly available data concerning various aspects of COVID-19, many initiatives in the field of data mining have started as well. For instance, research has been conducted using natural language processing techniques on full text sources to possibly reveal answers to key open scientific questions on COVID-19 [21]. Additionally, John Hopkins University has created a dashboard displaying the amounts of positive tested and recovered people alongside the death amount, per region [8]. This visualization of the spread and recovery is of utmost importance for tracking the virus' behavior across the world.

2 Challenges

As the corona crisis affects the entire world's public health as well as the economy, a collaborative approach for data collection would be to the benefit of all. Centrally stored hospital data would be of tremendous value in tackling COVID-19. Unfortunately, for privacy as well as geopolitical reasons, this is not realistic [5]. It needs to be kept in mind that fighting the virus comes with sharing data responsibly. While it is a struggle to find medical data at all, there are some publicly available sources which could be used in tackling the problem at hand. The University of San Diego published a data set containing COVID and non-COVID CT-scans online and have trained an AI model on it [27]. Although their result of 83% accuracy is decent, it is worth further exploring the capabilities of data mining techniques to acquire more reliable and better performing models, before such models can be applied to determine medical diagnoses.

To classify COVID-19 in chest radiographs we will explore some well known deep convolutional neural networks that have proven to be effective in the past.

However, to train such models we need to have a lot of data, which we already identified as a major challenge. Therefore, we will be using transfer training to make use of the pre-existing weights in the lower layers of a CNN. Lower layers contain information on more generic image features which are useful by a wide array of tasks, while we will be training the higher layers of the CNN that are more specific to the details of the domain classes that are contained in our dataset. In an attempt to acquire a model valuable to the COVID-19 crisis, instead of using CT-scans as was done in the previous work by the University of San Diego we exploit available chest radiographs, both types of scans have the ability to detect COVID-19[9], though CT-scans are ten times more expensive and thus chest radiographs are overall more accessible[22]. In this paper we focus on exploring well known deep convolutional neural network architectures, such as to acquire insights in in its capabilities of classifying COVID-19 in chest radiographs.

Section 3 discusses background information on research done in both COVID-19 and classification of COVID-19, additionally we discuss background information on the deep convolutional networks that we are going to use, transfer learning, data augmentation and data balancing. In section 4 we discuss the set up of the experiment. the results are presented in section 5, and the conclusion and discussion are covered in the final section.

3 Background

3.1 Related Work

While the vast majority gets moderate symptoms, in approximately 15% COVID-19 can cause severe inflammation in the lungs and possibly pneumonia [6]. With the pneumonia caused by COVID-19, fluid will fill the air sacs in both lungs, just as in regular pneumonia cases. This might lead to severe breathing problems, as the reduced volume limits the lungs' capacity in taking up oxygen. When the COVID-19 pneumonia progresses, more sacs will fill which eventually might even lead to the destructive Acute Respiratory Distress Syndrome (ARDS) which is a from of lung failure [23]. Considering this potential course of events, early detection is crucial. Speeding up the chest X-ray interpretation process hence would be greatly valuable.

There has already been done some research on COVID-19 X-ray classification [15]. However, while they achieved an accuracy of 0.97 for the InceptionV3 model, performance of Inception-ResNetV2 did not surpassed 0.87. Further, they omitted data pre-processing, augmentation and balancing. Another study did investigate various pre-processing techniques thoroughly [1]. However, they focused on CT-scans which, due to the aforementioned availability disadvantages, might not be preferable when in global crisis.

3.2 Convolutional Neural Networks

The task of classifying images is a rather complex one, leading to the need of complex models. Convolutional Neural Networks are a multi-layer neural network as shown in Figure 1 and without going in depth about the different kind of layers, they can either be an input layer which holds the raw pixel values of the images, a convolutional, relu or pooling layer which are associated with feature learning, or a fully connected layer which is associated with computing the class scores and classification in general.

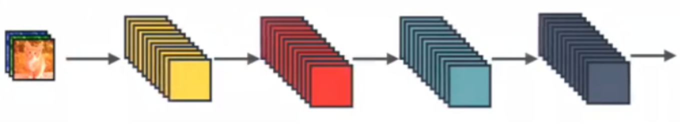

Fig. 1. Convolutional Neural Network Architecture [25]

Since the introduction of LeNet5 in 1998, a lot of researchers have been designing and developing CNNs to perform image classification [13]. A few of them are:

DenseNet attempts to solve a common problem of CNNs, namely that they become increasingly deep, with as a result that the vanishing gradient problem becomes more prominent [12]. The DenseNet architecture solves this problem by connecting all layers directly rather than solely connecting adjacent ones using so-called dense blocks.

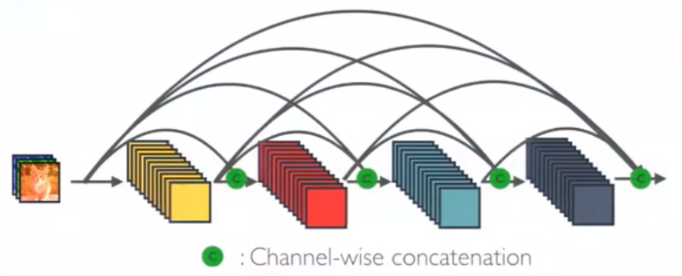

Fig. 2. Densely Connected Neural Network Architecture [25]

Note that this is not the same as fully connected networks, where each individual neuron in a certain layer is connected to all neurons in the following layer. Figure 1 and 2 display the difference between the networks. Using this solid structure ensures better supervision thus a more reliable loss function [24].

However, it also results in many parameters to be learned and need for high computational resources [19].

ResNet (Residual Networks) contrary to a DenseNet, utilize shortcuts to jump over certain layers, creating so-called Residual blocks of code. So, rather than linking all layers, some interlayer connections are skipped. This reduces the number of parameters, thus making ResNets less memory hungry than DenseNets [10].

InceptionV3 is the enhanced version of InceptionV1, also known as GoogleNet, which is assembled using a so-called Inception architecture [25]. InceptionV1 is based on the structure introduced by [14] of Networks within Networks to improve the discriminative power. InceptionV1 is not only deep in the direct sense but the depth is also apparent within the used Inception modules. Inception modules create sparsity by applying 1×1, 3×3, and 5×5 convolutions after dense components. Additionally, 1×1 convolution is used as non-linear dimension reduction before 3×3 and 5×5 convolutions to maintain sparsity and prevent an increase in computational requirements [19][25]. The Inception module architecture can be found in Figure 3. The network mainly consists of stacked Inception modules. The occasional introduction of Max-Pooling down-samples the features by halving the resolution. These modules allow the focus on higher features at higher network levels, enabling scaled feature processing [19].

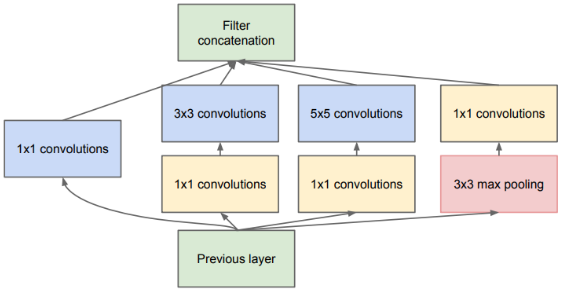

Fig. 3. Inception Module as used in InceptionV1 [17]

InceptionV3 has been widely adopted in the field of image recognition. In contrary to the module presented in Figure 3, InceptionV3 factorizes convolutions, i.e. 5×5 to two 3×3 filters and $n \times n$ to a size of $1 \times n$ and $n \times 1$, improving a model's speed [20]. Additionally, some modules' filters are expanded to prevent dimensionality reduction, hence information loss, at crucial stages. These two factorization methods next to filter expansion naturally results in three different Inception modules rather than just one. Further, factorized 7×7 convolutions are introduced. Lastly, label smoothing is added in an attempt to reduce overfitting by preventing the model to gain too much confidence on a certain class.

Inception-ResNetV2 is a more expensive hybrid Inception model which has shown performance improvement over the simpler InceptionV3 model [18]. It uses the same Inception modules but includes additional reduction blocks, which reduce the width and height of the grid. InceptionV3 already incorporates dimensionality reduction within the Inception moduels as explained, so in an implicit manner.

3.3 Transfer Learning

Transfer Learning is similar to fine-tuning, but instead of retraining the whole network, we leverage a pre-trained model and exploit the hierarchical representations of CNNs by transferring the lower layers of an existing CNN to a new domain, while we train the higher layers of the network which are more specific to the details of the domain classes. The benefits of transfer learning is that it greatly reduces the time to train a complete model and can result in a lower generalization error when you only have a small data set[26].

The three major Transfer Learning scenarios are:

1. **CNN as fixed feature extractor** by removing the last fully-connected layer and treat the rest of the CNN as a fixed feature extractor for the new dataset. We then replace the removed fully-connected layer with our own classifier and retrain the top of the CNN.
2. **Fine-tuning the CNN** by replacing and retraining the classifier on top of the CNN, but also backpropagate through the network to fine-tune the weights of the pre-trained network.
3. **Pretrained models** released by others who can use the networks for fine-tuning.

In this paper we will make use of the first scenario, where we use the CNN as a fixed feature extractor by replacing the classifier with our own to fine-tune it to our dataset, additionally we will add a few extra layers, such to prevent overfitting.

3.4 Data Augmentation and Balancing

Class imbalance is a common issue where the number of data in different classes varies significantly. It can cause the classification model favoring the class with more data. To solve this problem, various methods can be used such as oversampling and undersampling. In the study[2] performed by Buda, Maki and Mazurowski on the class imbalance problem it was concluded that in cases of extreme ratio of imbalance between classes, undersampling of the majority class performs as good as oversampling the majority class and that oversampling the minorty class does not cause overfitting of convolutional neural networks.

A common technique used for oversampling in deep learning is by replicating randomly selected samples from the minority class. To augment our dataset we use SMOTE (Synthetic Minority Oversampling Technique) as simply duplicating examples from our minority class doesn't provide any additional information to the model, meanwhile SMOTE synthesizes new examples from the minority class by interpolating neighboring data points [3].

Additionally, data augmentation is used to prevent overfitting by transforming the original training data to a more heterogeneous set, for example by adding rotation to the image or adding random jitters to increase the generalizability of the model.

4 Methodology

In this paper we classify COVID-19 in chest radiographs by using the CNN architectures of Inception-ResNetV2, DenseNet, InceptionV3, ResNet which have respectively a depth of 572, 201, 159 and 101 layers. We replace the fully-connected top layer with our own classifier and freeze the other layers while training, which is better known as transfer learning. The following sections will discuss the origin of the data, the pre-processing of the images, the hyperparameters used in training the model and our classifier.

4.1 Data

The data consists of 140 chest radiographs of cases diagnosed with COVID-19 which are collected by Cohen, Morrison and Dao[4] and 939 chest radiographs of healthy people collected from the Chest X-Ray Images dataset on Kaggle by Paul Mooney[16].

4.2 Pre-processing

As can be seen in several datasets [7][16], radiographs are often taken from different perspectives. Additionally, the presence of noise within radiographs, e.g. in the form of pointers, might further prevent models from making fair comparisons. Therefore, pre-processing images is a necessary step to prepare data for neural network models, which can eliminate noises and enhance the interpretability of images to improve the classification precision[11]. The following preprocessing techniques will be applied to the images:

1. **Resizing the image:** All the images will be resized to 224x224 pixels.

2. **Swapping color channels:** Swapping the color channel from BGR to RGB.

3. **Normalization of the pixel values:** Scaling the pixel intensities to the range [0, 1].

4. **Training data augmentation:** SMOTE has been used to generate new images of the minority class, while Keras' ImageDataGenerator is providing random jitter to the images.

4.3 Hyperparameters and Classifier model

As mentioned before we will use the pre-trained CNNs as a fixed feature extractor and remove the fully-connected top layer and construct a new one, to reduce overfitting we introduce the following classifier:

- A dropout layer with 50% ratio of dropped outputs for InceptionV3 and DenseNet and with a 20% ratio of dropped outputs for ResNet and Inception-ResNetV2.

- A global average pooling layer to reduce the spatial dimensions of the three-dimensional tensor.

- A dense layer with 128 units, ReLu as activation function and we add the activity regularizer with an L1 and L2 regularization factor of 0.001.

- A batch normalization layer for speeding up the convergence and improving the generalization of the model.

- A dense layer with 2 units and a softmax activation function. We add the activity regularizer with an L1 and L2 regularization factor of 0.001.

The hyperparameters used for training the CNNs are:

Table 1. Hyperparameters used in the training model

Batch size	Epochs	Learning rate	Optimizer	Loss function
8	100	0.0001	SGD(momentum=0.5)	Binary crossentropy

Although larger batch sizes allows for computational speedups when training the model, it also the case that a batch size that is too large will lead to poor generalization, while smaller batch sizes show a faster convergence to good solutions as they offer a regularizing effect and lower generalization error. However, when using a smaller batch size we are not necessarily guaranteed that we converge to the global optimum.

As we are using a small batch size we also use a lower learning rate as the updates are now more noisy. A lower learning rate may allow the model to learn a more optimal set of weights but may take significantly longer to train and therefore we train the model for at least 100 epochs. In the training graphs shown in the next section we can determine that 100 epochs is certainly enough to reach a point of convergence.

We initialize SGD as optimizer with a momentum parameter set to 0.5. Momentum is a method which helps accelerate gradients vectors in the right directions, thus leading to faster converging. Finally, as this is a 2-class problem we use a binary crossentropy loss.

5 Results

This section presents the training graphs acquired from training the four CNNs as indicated earlier, subsequently we present the classification report acquired from using the trained networks to classify COVID-19 in chest radiographs.

5.1 Training

Figure 4 to 7 illustrates the training and validation loss and accuracy for each model used in the training process. All the loss graphs follow the similar trend that both training loss and validation loss are decreasing as the number of epochs increases, though the loss might fluctuate within one epoch as the batch sizes are rather small. On the other hand, the training accuracy and validation accuracy are increasing with the number of epochs, which corresponds to the loss graphs indicating that a decrease in loss goes hand in hand with an increase in accuracy.

Based on the graphs we see that when it comes to training the networks, two of them stand out. Inception-ResNetV2 has a heavy fluctuating validation loss, which intuitively could mean that some portion of the examples in the test set are classified randomly, this is also visible in the the validation accuracy as this is fluctuating as well.

The second set of training graphs that stand out are the ones belonging to ResNet. After epoch 40 we see that the accuracy on the training set keeps improving while the accuracy on the validation set decreases. This is a sign of overfitting as the model potentially is continuing to learn patterns in the training data, but the same patterns don't not necessarily exist in the validation set, more misclassifications will occur.

Both the sets of training graphs from InceptionV3 and DenseNet form a rather smooth curve in both the validation accuracy and validation loss, indicating that no overfitting occurs on the training data and the classifier is doing a better job at modelling the relationship between the input data and the output targets at each epoch.

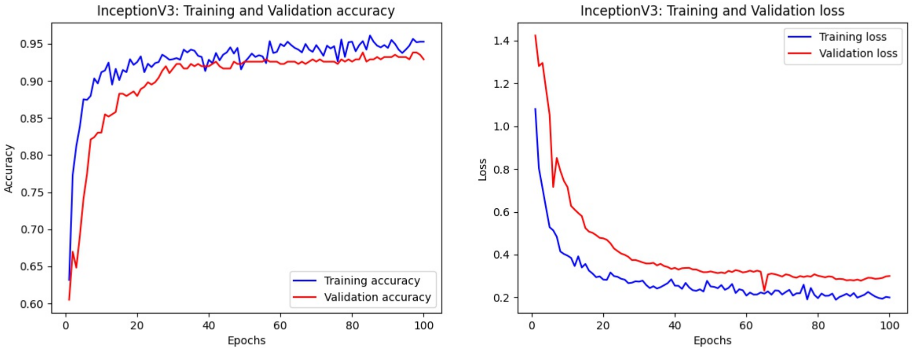

Fig. 4. InceptionV3: training and validation accuracy and loss per epoch

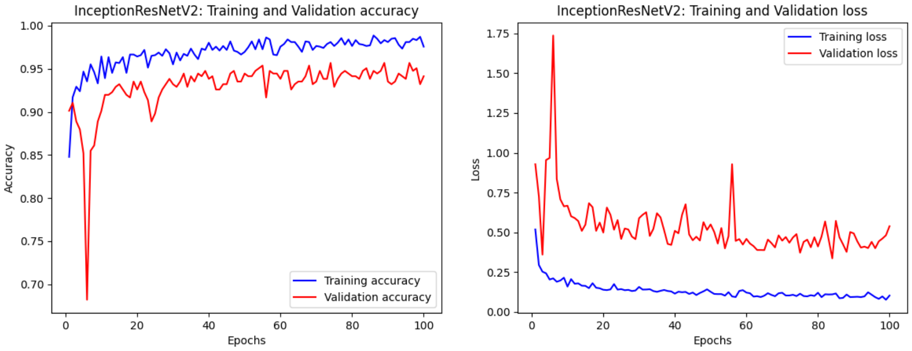

Fig. 5. Inception-ResNetV2: training and validation accuracy and loss per epoch

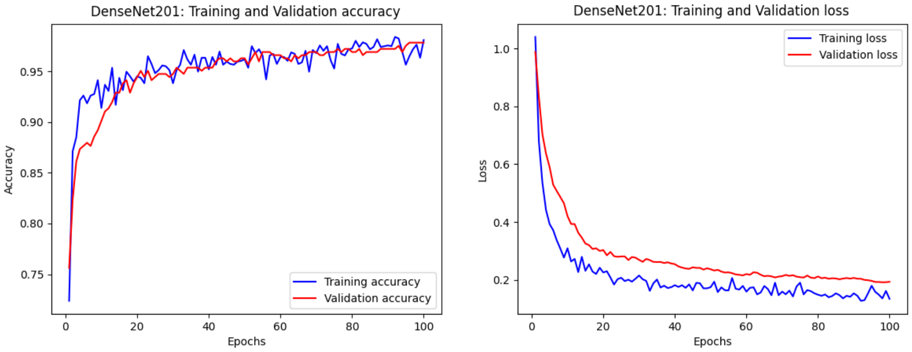

Fig. 6. DenseNet: training and validation accuracy and loss per epoch

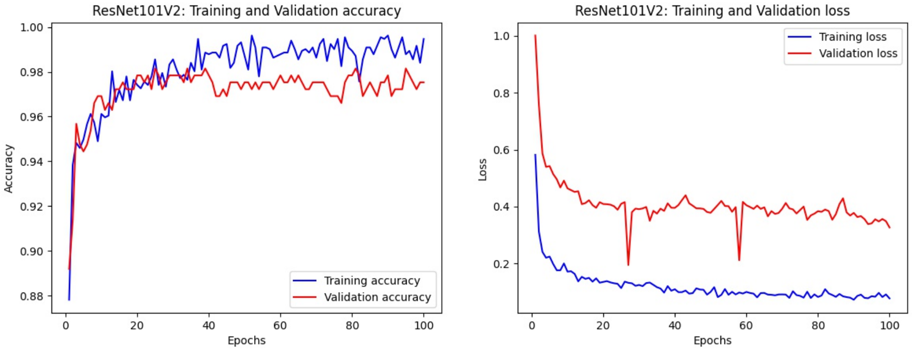

Fig. 7. ResNet: training and validation accuracy and loss per epoch

5.2 Evaluation

Rather than focusing solely on accuracy, several other metrics will also be taken into account when evaluating the models. The main target is to detect COVID-19 patients as accurate as possible making recall, precision and f1-score suitable evaluation metrics. From a public healthcare point of view we regard a missed classification of a COVID-19 patient as healthy catastrophic, as that person could potentially infect others, while a misclassification of someone who is healthy as having COVID-19 might have safety risks of getting hopital-accquired infection, which could for example be averted by implementing save procedures for a second opinion. Therefore, as it shows in table 2, we evaluate the training models using four different metrics as follows:

$$\text{precision} = TP/(TP + FP) \tag{1}$$

$$\text{recall} = TP/(TP + FN) \tag{2}$$

$$\text{f1-score} = 2 * ((precision * recall)/(precision + recall)) \tag{3}$$

$$\text{accuracy} = (TN + TP)/(TN + TP + FN + FP) \tag{4}$$

where TP, FP, TN, FN represent the number of true positives, false positives, true negatives and false negatives respectively. Note that the accuracy score is calculated for the whole model, however, precision, recall and f1-score are calculated per class. Based on the equations above, we can see that f1-score and accuracy consider the number of positives and negatives more thoroughly and appear to be better representations for the results, therefore, we focus on the comparison between these two during evaluation.

Table 2. Model evaluation results

Evaluation result of InceptionV3

	precision	recall	f1-score
COVID-19	0.93	0.60	0.72
NORMAL	0.94	0.99	0.97

TP	FP	FN	TN	accuracy
25	2	17	280	94%

Evaluation result of InceptionResNetV2

	precision	recall	f1-score
COVID-19	0.96	0.64	0.77
NORMAL	0.95	1.00	0.97

TP	FP	FN	TN	accuracy
27	1	15	281	95%

Evaluation result of ResNet101V2

	precision	recall	f1-score
COVID-19	0.97	0.88	0.93
NORMAL	0.98	1.00	0.99

TP	FP	FN	TN	accuracy
37	1	5	281	98%

Evaluation result of DenseNet201

	precision	recall	f1-score
COVID-19	0.93	0.90	0.92
NORMAL	0.99	0.99	0.99

TP	FP	FN	TN	accuracy
38	3	4	279	98%

Among all four models, DenseNet201 and ResNet101V2 achieve the best accuracy score of 98% and best f1-score of 0.93 and 0.92 on COVID-19 and an f1-score of 0.99 on NORMAL, while all the models have an accuracy well over 90% we need to take into account that there was a significant class imbalance and that if the models would disregard classifying anything as COVID-19 we still would achieve high accuracy percentages in an absolute sense.

Of all the radiographs that InceptionV3 identified as COVID-19 93% of them were actually true, however out of the whole dataset it was only able to correctly identify 60% of the actual COVID-19 cases. Inception-ResNetV2 gives us a slightly better performance when it comes to predicting COVID-19 cases, where 96% of the identified COVID-19 cases were actually true and out of the whole dataset it was able to correctly identify 64% of the actual COVID-19 caes. As indicated earlier misclassification of COVID-19 as healthy can be catastrophic to public health as this person would be able to infect others. Therefore, we consider the performance of these two models as rather poor, since only a bit more than half of the actual cases is predicted correctly.

However, ResNet provides us with promising results as 97% of the identified COVID-19 cases were actually true and it correctly predicted 88% of the COVID-19 cases over the whole dataset. The best performing model from a public healthcare point of view is DenseNet with a slightly lower precision of 93%, but also a slightly higher recall of 90%. Although it does tend to classify radiographs of healthy people as someone who has COVID-19 a little bit more than ResNet does, it also tend to misclassify less radiographs with people diagnosed with COVID-19 as healthy. If we evaluate from a quantitative point of view, then ResNet is the better performing model, since it has the least misclassifications, which is also visible in its f1-scores being the highest amongs the four models.

6 Conclusion and future work

In this paper we made use of transfer learning on pre-trained deep convolutional networks to further explore the capabilities of deep learning on classification of COVID-19 in chest radiographs. ResNet and DenseNet performed the best in terms of accuracy, where both models were able to reach an accuracy of 98%. ResNet has a slight edge over DenseNet in terms of the f1-scores on COVID-19 and NORMAL classification, with an f1-score of 0.93 and 0.92 for COVID-19 respectively and an f1-score of 0.99 for both models on NORMAL classification.

To further extend on the work done in this paper, one could explore the effects of fine-tuning the CNNs by not only replacing and retraining the classifier on top of the CNNs, but also backpropagating through the network to fine-tune the weights of what was already pre-trained. Secondly, one could further optimize the hyperparameters to find optimal hyperparameters to train the networks as this was not the main focus in the scope of this research.

References

1. Mucahid Barstugan, Umut Ozkaya, and Saban Ozturk. Coronavirus (covid-19) classification using ct images by machine learning methods. *arXiv preprint arXiv:2003.09424*, 2020.
2. Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. 2018.
3. Nitesh Chawla, Kevin Bowyer, Lawrence Hall, and W. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res. (JAIR)*, 16:321–357, 01 2002.
4. Joseph Paul Cohen, Paul Morrison, and Lan Dao. Covid-19 image data collection. *arXiv 2003.11597*, 2020.
5. Fighting corona starts with sharing data responsibly (Universiteit Leiden). https://www.universiteitleiden.nl/en/news/2020/03/fighting-corona-starts-with-sharing-data-responsibly.
6. Coronavirus and Pneumonia. https://www.webmd.com/lung/covid-and-pneumonia.
7. Kaggle COVID-19 Detection X-Ray Dataset. https://www.kaggle.com/darshan1504/covid19-detection-xray-dataset.
8. Kaggle Novel Corona Virus 2019 Dataset. https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset.
9. Soon Ho Yoon et al. Chest radiographic and ct findings of the 2019 novel coronavirus disease (covid-19): Analysis of nine patients treated in korea. *Korean Journal of Radiology v21 n4 (2020): 494-500*, 2020.
10. Vincent Fung. An overview of resnet and its variants. *Towards Data Science*, 2017.
11. Vijaya. G. Artificial intelligent techniques for the effective diagnosis of lung cancer. 2016.
12. Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.
13. Yann Lecun, Leon Bottou, Y. Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278 – 2324, 12 1998.
14. Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.
15. Ali Narin, Ceren Kaya, and Ziynet Pamuk. Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks. *arXiv preprint arXiv:2003.10849*, 2020.
16. Kaggle Chest X-Ray Images (Pneumonia). https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia.
17. Bharath Raj. A simple guide to the versions of the inception network. *Retrieved from Towards Data Science website: https://towardsdatascience. com/a-simpleguide-to-the-versions-of-the-inception-network-7fc52b863202*, 2018.
18. Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.
19. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

20. Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
21. Fighting the Covid-19: All the datasets and data efforts in one place. https://towardsdatascience.com/fighting-the-covid-19-all-the-datasets-and-data-efforts-in-one-place-4d6aeb0157ab.
22. CT scans across 3 states: 4 things to know The out-of-pocket costs of X-rays. https://www.beckershospitalreview.com/finance/the-out-of-pocket-costs-of-x-rays-ct-scans-across-3-states-4-things-to-know.html.
23. What Coronavirus Does to the Lungs. https://www.hopkinsmedicine.org/health/conditions-and-diseases/coronavirus/what-coronavirus-does-to-the-lungs.
24. Edna Chebet Too, Li Yujian, Sam Njuki, and Liu Yingchun. A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, 161:272–279, 2019.
25. Sik-Ho Tsang. Review: Googlenet (inception v1)—winner of ilsvrc 2014 (image classification), 2018.
26. Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3320–3328. Curran Associates, Inc., 2014.
27. Jinyu Zhao, Yichen Zhang, Xuehai He, and Pengtao Xie. Covid-ct-dataset: a ct scan dataset about covid-19. *arXiv preprint arXiv:2003.13865*, 2020.