Implementing Transfer Learning on the CheXNet Convolutional Neural Network to Automatically Identify the Presence of COVID-19 in Chest X-ray Images

Jhorman Perez (950214-3275) Ravi Bir (980313-1953)

May 20, 2020

Abstract

Coronavirus (COVID-19) is a newly discovered infectious disease that targets the respiratory system, with over 5million global cases and 300,000 deaths. Currently the best technique to control the spread of this disease is to quickly diagnose patients and quarantine them. There are some problems with the current method of diagnosis, so new methods are being researched. In this paper transfer learning was investigated as an approach to solve this problem. It was applied to the pre-trained CheXNet convolutional neural network, with the goal of automatically identifying the presence of COVID-19 in X-ray chest images. The results are promising, showing that applying transfer learning to a convolutional neural network can be used in the automatic detection and correct diagnosis of COVID-19 in chest x-rays.

1 Introduction

Coronavirus (COVID-19) is a newly discovered infectious disease that targets the respiratory system [1]. The majority of COVID-19 patients experience mild to moderate symptoms and can recover without requiring any particular treatment. However, for a small group of people, contracting the disease is very dangerous. It results in the development of severe symptoms requiring extreme medical intervention, and in some cases can lead to death. There are currently over 5 million global cases of COVID-19, with 300,000 confirmed deaths.

One of the key methods implemented to contain the spread of COVID-19 is to quickly and accurately identify and test people who many have contracted it, and quarantine them if they have. Diagnosing this disease is currently done by taking a swab of the nose and the back of the throat, with results being returned to the patient after just 48-72 hours. However, there have been some problems identified with the current testing procedure. There are not enough testing kits available, so only the most vulnerable and at-risk members of a population are currently being tested. Financial problems have also arisen from testing. Governments are struggling to provide funding for test kits, and in countries with private healthcare, some people can not afford to get tested. The test is also not 100% accurate. It is clear that a new testing technique to diagnose COVID-19 is needed. It could be used in conjunction with the current testing kits, and should make the testing procedure more accurate, affordable and comfortable.

In this paper transfer learning will be applied to the pre-trained CheXNet [2] convolutional neural network, with the goal of automatically identifying the presence of COVID-19 in X-ray chest images. Convolutional neural networks usually require a large dataset to accurately train them from scratch. However, there is only a limited supply of COVID-19 chest x-ray images freely available. Transfer learning is the best approach to use when this is the case. When a pre-trained neural network is used to perform a certain task, using transfer learning it is possible to utilise the knowledge learned from this task and use it to perform a different task. Different hyperparameters and methods will be investigated to find the most optimum results.

Even with the limitation of a small amount of COVID-19 data available, the results show that transfer learning can be applied to a convolutional neural network to automatically detect and diagnose COVID-19 in chest x-rays.

2 Related Work

Transfer learning will be applied to the CheXNet [2] model. CheXNet is a 121 layer convolutional neural network. It has been trained on the ChestX-ray14 dataset [3], which contains more chest x-rays than any another available dataset, spanning over 100,000 images of 14 different diseases that can be diagnosed from x-rays of the chest. It is trained to automatically classify and diagnose a chest x-ray of a patient into the correct disease class. It has been shown to have an improved performance when compared to that of radiologists who would manually classify the chest x-rays images. CheXNet was able to automatically diagnose each of the 14 diseases more accurately than the radiologists. These state of the art results mean that it is the perfect model to apply transfer learning to, in order to complete our required task of classifying COVID-19 chest x-rays.

Over the last month other researchers have performed similar investigations relating to transfer learning and its suitability in automatically detecting COVID-19 in x-rays. Rehman et al., [4] and Khalifa et al., [5] using different models and datasets, applied transfer learning in order to automatically distinguish COVID-19 x-ray scans from healthy scans. Both achieved promising results, so it is assumed that similar result can be achieved using the CheXNet model.

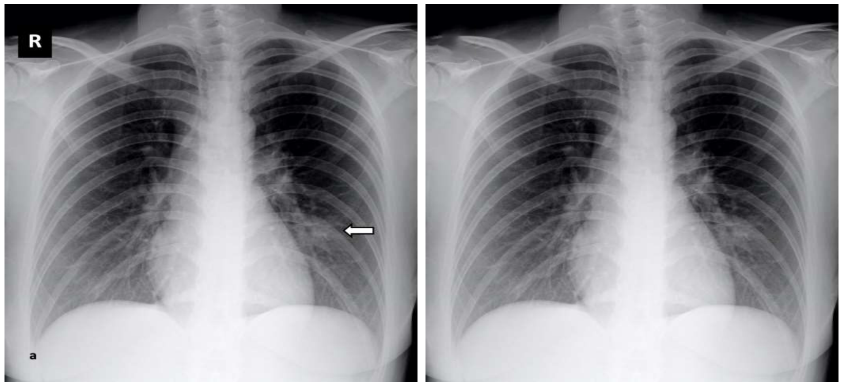
3 Data

The COVID-19 chest x-ray images that will be used for transfer learning have all been obtained from one public open dataset [6]. It contains chest x-ray and CT images of patients who have been diagnosed with COVID-19 or other viral and bacterial pneumonias (MERS, SARS, and ARDS). The images were collected from public sources (such as published papers) and hospitals throughout the world. Since we are only interested in the scans that show confirmed cases of COVID-19 and not other diseases, each image was cross-referenced to its row in the corresponding metadata table on the repository, and any scan that did not have a confirmed diagnosis of COVID-19 was removed from the dataset.

The dataset contains a small portion of scans from the sides of the chest. These were all deleted, as we wanted to be consistent and only train the model with scans from the front of the chest. The dataset also contains a small portion of CT scans, which produced very different images when compared to those produced by x-rays. Therefore, these were also removed from the dataset to maintain consistency in the images.

The main issue with this dataset is the watermarks and symbols found on each x-ray image. The scans have come from a wide range of sources, with each one adding its own features to the images they have provided. Some images contain arrows pointing to various parts of the scan, and some contain markers indicating the left and right sides of the scan. Some images contain various details regarding the origin of the scan, for example the name of the patient or the hospital it came from. There was no automated technique to remove these details from the images. Therefore each image was loaded one at a time into Adobe Photoshop, all the unwanted watermarks and symbols were selected, and they were removed using the 'Content Aware Fill' feature. An example of this can be found in the figure below.

The non-COVID-19 chest x-ray images that will be used for transfer learning have been collected from two different datasets. The first set of images comprises of 300 front chest x-ray scans from patients with healthy lungs [7]. The second set of images comprises of300 front chest x-ray scans from patients with a variety of different lung diseases, including pneumonia, fibrosis, and emphysema. They come from a subset of images from the ChestX-ray14 dataset [3] that were not used to train the CheXNet model. The scans from both datasets contained watermarks and symbols, which were removed using the same procedure as the COVID-19 scans.



(a) X-ray scan before removing symbols and watermarks

(b) X-ray scan after removing symbols and watermarks

4 Methods

4.1 Transfer Learning

The main concept behind transfer learning is taking knowledge that a neural network has learned from one task and applying that knowledge to a separate task [8]. It works because neural networks are layer-wise self contained. This means that the neural network will still work if all the layers are removed after a certain layer, and a new layer is connected with a different number of neurons and different weights [9]. During pretraining with the large dataset, the first layers of the network learn how to extract the features. Then when the new smaller dataset is given, the pretrained network does not have to learn how to extract the features again, because it already learned how to do this. Appropriate parameter values for these layers have already been set. The features in the new dataset have been successfully extracted, but can not be classified, so the last layers of the model are retrained with the new smaller dataset and learn how to classify the extracted features. This is possible because these layers are simple and have much less parameters compared to the feature extraction layers [8].

Transfer learning is most useful when not much data is available, which is the case with the limited COVID-19 data available. A model trained using only this small dataset is much more prone to overfitting. Using what the well-trained, well-constructed ChexNet network has learned over the large ChestX-ray14 dataset, is is applied to boost the performance of the COVID-19 classification task, which has a smaller available data set [10]. Other benefits of transfer learning include a higher starting accuracy, faster convergence and a higher accuracy the training will converge to [9]. In the neural network, the output layer and the weights leading up to the output layer are removed. A new output layer is added with a different number of nodes relating to the new problem (in our case 2 output nodes are added for the binary classification on x-ray image into classes COVID-19 and non-COVID-19). Weights leading up to the output layer are randomly initialised [11].

The neural network is then retrained on the new dataset (COVID-19 and non-COVID-19 scans), and the weights leading up to the new output layer are updated. The whole neural network can now be used for the new task. Another possible method for retraining involves pre-training and finetuning the parameters. During retraining using the new dataset, parameters in multiple layers of the neural network are updated, not just the final layer. But this is only feasible when we have a large dataset to retrain with [11].

4.2 Image Pre-Processing

70% of both the COVID-19 and non-COVID-19 were put into the training dataset, and the remaining 30% were put into the validation dataset. There was not enough images available to make a separate test set. Multiple methods were implemented to normalise, centre and standardise the images. As mentioned above, the watermarks and symbols on all images were removed. In Pytorch [12] both the training data and validation data were normalised based on mean and standard deviation values that are commonly use on ImageNet. The training data images were randomly resized and given a random aspect ratio of the original, then all made to the size 224 x 224. Some of the images were also randomly flipped horizontally. The validation data was all resized to 224 x 224 and cropped at the centre.

4.3 Implementation

The implementation of the transfer learning model for image-classification on the COVID-19 dataset was built in PyTorch [12], an open source deep learning framework highly supported by Facebook. A previous model [13] that had already been trained by replicating the CheXNet architecture was used as the reference neural network to train our one. The learnt parameters from this pre-trained network were loaded and used to initialise our network. Then a custom classifier was added to better suit the model's new task with two outputs to be trained from scratch, replacing the original one from CheXNet.

We completed our own training implementation, which meant a lot of components had changed from the original model's implementation. By taking the model's architecture and parameters, we were able to complete our own custom training and validation. Unlike the original implementation, we did not have a file containing all the labeled images, but instead the datasets and classes were organised by folders. PyTorch understood this format when it read and loaded the data (dataloaders). The data was either given the label 'covid' or 'no-covid'. We replaced the SGD optimizer that was used to train the model with the Adam optimizer, using recommended deafult values of $\beta_1 = 0.9$, $\beta_2 = 0.999$

and weight decay of 1e-4. Also, instead of Binary Cross Entropy Loss, we used a Cross Entropy Loss function which internally includes a Softmax activation. This gave us the probabilities for each class, allowing predictions to be made. The implementation was executed on the Google Colab GPU [14].

5 Experiments

We will be investigating the effect of adjusting the model and its hyperparameter values on the performance of the network in classifying COVID-19 images. The purpose of this is to achieve the most optimum results. First we will investigate the effect of changing the compositions of the training and validation datasets. The dataset compositions and the values they returned can be found in Table 1. Each composition was run 5 times for 20 epochs, and the average across the 5 runs is given. The datasets that return the highest accuracy are the ones where the number of images in the two classes COVID-19 and non-COVID-19 are equal. From now on we will be using these datasets for training and validation, as they returned the best result. Something to note is that the accuracies for the other 2 experiments are very similar to the proportion of non-COVID compared to COVID images in the datasets. This means overfitting has occurred. The non-COVID class is the only one being learned, so every image just gets classified as belonging to this class. This does not happen with the equal datasets.

Training Dataset	Validation Dataset	Max validation Accuracy	Min validation Loss
COVID-165 Healthy-210 Diseased-210	COVID-70 Healthy-90 Diseased-90	0.728	0.513
COVID-165 Healthy-165 Diseased-165	COVID-70 Healthy-70 Diseased-70	0.676	0.568
COVID-164 Healthy-82 Diseased-82	COVID-70 Healthy-35 Diseased-35	0.827	0.406

Table 1: Effect of different datasets on Accuracy and Loss

Next we investigated the affects of the number of epochs has on the performance, to see if running for more epochs is necessary. Early stopping has also been implemented, so if there has been no improvement in the accuracy for more than 10 epochs in a row, training will stop. The results are below in Table 2. Each epoch experiment was run 5 times, and the average values across the 5 runs is given. We can see that early stopping occurred for 20 and 50 epochs. The validation loss decreases as the number of epochs increases, and the maximum validation accuracy seems to always occur between epoch 10-20, so to ensure the maximum accuracy is captured, all experiments will run for 20 epochs.

No. of epochs	No. of epochs ran for	Max validation Accuracy	Min validation Loss
5	5	0.800	0.507
10	10	0.814	0.450
20	19	0.835	0.394
50	20	0.832	0.403

Table 2: Effect of number of epochs on Accuracy and Loss

Next we investigated the effect of learning rate on the performance. Each epoch experiment was run 5 times, and the average values across the 5 runs is given. Results can be found in Table 3. A learning rate of 0.001 returns the best result, so will be used from now on. A too small learning rate resulted in overfitting on one of the classes.

Learning rate	Max validation Accuracy	Min validation Loss
0.01	0.807	0.528
0.001	0.821	0.434
0.0001	0.621	0.638

Table 3: Effect of learning rate on Accuracy and Loss

Next we investigated the effect of fine-tuning. Instead of only updating the parameters leading up to the output layer, we investigated how freezing only some layers and re-training some other layers would affect the performance. These layers were not trained from scratch using random initialisations. Instead, the parameter values they had already learned from the CheXNet model were being updated.

The model is based on the architecture of DenseNet-121 that was used to run experiments over the ImageNet dataset, with a particular four-dense-block structure. We froze all the dense block layers except the classification layer (the final fully connected one) and the one preceding it (Dense Block 4 - called denseblock4 in PyTorch). This was done because only the last layers of a CNN have a big impact on the final predictions, given that they learn more task-specific patterns (they are close to the classification layer). On the contrary, it is acceptable to freeze earlier layers, since their parameters are ready to generically recognize simpler patterns (non-complex shapes, lines, among others). Unfreezing more layers implies more processing and memory resources, and it may not represent an actual improvement in the accuracy. In our case, since we have a limited dataset for COVID-19, it is not convenient to unfreeze other layers of the model. It could lead to inaccuracy and make the network unlearn deliberately

This experiment was run five times. obtaining an average maximum accuracy of 0.938 and an average minimum loss of 0.151. However, we achieved our very good result from one of these runs, with an accuracy of 0.957 and a loss of 0.134. By unfreezing the final layers and allowing them to be retrained, the model has better learned how to classify the extracted features, so returns a higher accuracy.

6 Conclusion

In this investigation the best results for classifying images into COVID-19 and non-COVID-19 were achieved when using the Adam optimiser, a learning rate of 0.001, 20 epochs with early stopping, equal number of images in each class in the training and validation datasets, and fine tuning the last block of layers. It is therefore concluded that applying transfer learning to a convolutional neural network can be successfully used in the automatic detection and diagnosis of COVID-19 in chest x-rays. Using transfer learning, the model can distinguish between COVID-19 scans and non COVID-19 scans made up of healthy and diseased patients. This model could be used as a cheap, fast and automatic COVID-19 diagnostic test to support other medical testing, without unnecessarily exposing hospital staff to infection. The testing procedure would become more accurate, safe and comfortable. However, access to x-ray equipment would be needed. To further support this result, more COVID-19 x-rays would be needed to train and test the model.

Code used in this investigation can be found at https://github.com/jhorapb/covid19-pytorch

References

[1] (2020) World health organization- coronavirus. [Online]. Available: https://www.who.int/health-topics/coronavirus

[2] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *arXiv preprint arXiv:1711.05225*, 2017.

[3] R. Summers, "National institutes of health clinical center- chestx-ray14," https://nihcc.app.box.com/v/ChestXray-NIHCC, 2017.

[4] A. Rehman, S. Naz, A. Khan, A. Zaib, and Imran, "Improving coronavirus (covid-19) diagnosis using deep transfer learning," 04 2020.

[5] M. Loey, F. Smarandache, and N. E. M. Khalifa, "Within the lack of chest covid-19 x-ray dataset: A novel detection model based on gan and deep transfer learning," *Symmetry*, vol. 12, p. 651, Apr 2020.

[6] J. P. Cohen, P. Morrison, and L. Dao, "Covid-19 image data collection," *arXiv 2003.11597*, 2020. [Online]. Available: https://github.com/ieee8023/covid-chestxray-dataset

[7] P. Mooney, "Chest x-ray images (pneumonia)," https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia, 2017.

[8] D. Sarkar, "A comprehensive hands-on guide to transfer learning with real-world applications in deep learning," *Towards Data Science*, 2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b

[9] C. von Csefalvay, "Transfer learning: the dos and don'ts," *StarsChema*, 2019. [Online]. Available: https://medium.com/starschema-blog/transfer-learning-the-dos-and-donts-165729d66625

[10] J. Brownlee, "A gentle introduction to transfer learning for deep learning," *Machine Learning Mastery*, 2017. [Online]. Available: https://machinelearningmastery.com/transfer-learning-for-deep-learning/

[11] Deeplearning.ai, "Transfer learning (c3w2l07)," 2017. [Online]. Available: https://www.youtube.com/watch?v=yofjFQddwHE

[12] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[13] J. Zech, "Reproduce chexnet," https://github.com/jrzech/reproduce-chexnet, 2018.

[14] Google, "Google colab," colab.research.google.com, 2017.