CS230

Transfer Learning from CheXNet to COVID-19

Ethan Lo
Department of Symbolic Systems
Stanford University
ethanlo@stanford.edu

Hashem Elezabi
Department of Computer Science
Stanford University
hashem@stanford.edu

Abstract

Within a span of a few months, novel coronavirus disease 2019, or COVID-19, initiated a global pandemic and put immense stress upon healthcare systems throughout the world. Naturally, there is a relative lack of chest x-ray (CXR) data for COVID-19, and tests are not yet widely available. However, since there already exists a significant amount of literature regarding the use of convolutional neural networks (CNNs) for detecting older types of thoracic pathologies, we evaluate the efficacy of applying a model pre-trained on non-COVID thoracic pathologies (CheXNet) to the task of identifying COVID-19. We find that various versions of our model do not perform well on this task in particular. Therefore, our findings support the idea that direct transfer learning from a DenseNet model trained on more general CXR data is not adequate for identifying COVID-19, and possibly hints that using CXR imaging might not be effective at all for detecting COVID-19.

1 Introduction

With almost 7 million confirmed cases and 400,000 deaths worldwide, COVID-19 is wreaking havoc upon our global society (Dong et al., 2020). At the same time, chest X-rays are the most common imaging examination tool used in practice, critical for screening, detection, and management of diseases including pneumonia (Rajpurkar & Irvin et al., 2017). However, due to the novelty of COVID-19, there is significantly less data available for chest X-rays of COVID-19 as there are for chest X-rays with pneumonia and other pathologies. Thus, we aim to leverage the relative abundance of thoracic pathology CXR data available for the novel task of identifying COVID-19 in patients. We hope that this will assist technologists and healthcare workers in detecting patients with COVID-19, despite the relative lack of data within this specific problem space. The input to our model is a single chest x-ray image, and the output is one of three conditions: normal, pneumonia, or COVID-19.

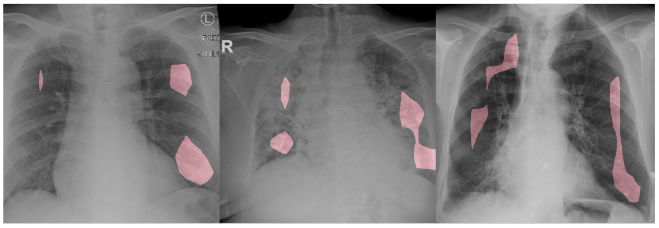


Figure 1: Example chest radiography images of COVID-19 cases from 2 different patients and their associated critical factors (highlighted in red)

CS230: Deep Learning, Spring 2020, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

2 Related Literature

According to the World Health Organization, two-thirds of the world's population does not have access to radiology diagnostics (Mollura et al., 2010). However, every continent, with the exception of Antarctica, has counted hundreds of thousands of cases of COVID-19 (Dong et al., 2020). This is why we think it is important to quickly create an automated system to assist in detecting COVID-19.

There currently exists a non peer-reviewed paper which builds a unique COVID-Net for detecting patients with COVID-19 (Wang & Wong, 2020). The creators of COVID-Net have compiled a dataset of COVID-19 cases from multiple sources and instutions to train their model. This dataset of 13,975 images includes augmented images by translating, rotating, horizontal flipping, and intensity shifting the original chest x-ray images. However, the bulk of their COVID-Net model is machine-generated by a private company called Darwin AI. Therefore, although the model reaches 92.6% accuracy, their process has two main drawbacks which we hope to avoid: uninterpretable/proprietary neural network architecture and a lack of training data on COVID-19 relative to data for other CNN's built for other pathologies.

Additionally, we reviewed the CheXNet paper, even though it is not suited directly for our task ((Rajpurkar & Irvin et al., 2017). This paper seems promising to build off of because of its prominence within the field and successful experimental results. For instance, CheXNet outperforms the best published results on all 14 pathologies in the dataset it uses. In detecting Mass, Nodule, Pneumonia, and Emphysema, CheXNet has a margin of >0.05 AUROC over previous state of the art results. Moreover, as determined by F1 score (the harmonic mean of the precision and recall), the CheXNet model outperforms three of four practicing radiologists and the average of all four radiologists. This is why we deemed it promising to use this as the foundation for our model.

Finally, it is a common technique to leverage transfer learning from a convolutional neural network trained upon ImageNet when building a CNN for a more specialized task (Huh, Agrawal, & Efros, 2016). One reason this has been viewed as an effective technique is because the lower-level visual structures required of certain image analysis tasks can be shared across even the most different end-tasks. Instead of following suit with what many other papers do by building directly from a model pre-trained merely on ImageNet, however, we transfer from a more specialized pre-trained model already related to our initial task of identifying COVID-19 via CXR.

3 Dataset and Features

Our model is built from two datasets: COVIDx (Wang & Wong, 2020) and ChestX-ray14 (Rajpurkar, Irvin, et al., 2017). We will explain both, with a larger focus on COVIDx since this is the dataset we actually use to train our model.

COVIDx is a conglomeration of three seperate datasets:

1. Dataset 1 - Consists entirely of scans from patients with COVID-19 (Cohen, Morrison, & Dao, 2020).
2. Dataset 2 - Consists of chest scans from patients with COVID-19 as well (Wang & Wong, 2020).
3. Dataset 3 - Provides COVIDx with all of the non-COVID scans. Originally, these scans were purposed for training a pneumonia detection model. We integrate them into our COVIDx dataset by not altering the examples labeled as normal, but bucketing all of the examples labeled as non-normal into a single class called 'pneumonia'. For instance, the condition 'Streptococcus' would be labeled as 'pneumonia' for our dataset's purposes.

The distribution for the train and test set are as follows:

- Chest radiography images distribution
  - Train: 7966 normal, 5451 pneumonia, 152 COVID-19, 13569 total
  - Test: 100 normal, 100 pneumonia, 31 COVID-19, 231 total
- Patients distribution
  - Train: 7966 normal, 5440 pneumonia, 107 COVID-19, 13513 total (unique)

- Test: 100 normal, 98 pneumonia, 14 COVID-19, 212 total (unique)

Because COVIDx has many more normal and pneumonia images than it does COVID-19 images, our test/train sets would be unbalanced if left alone. However, we do a rebalancing of each test/train batch where it up-samples for the COVID-19 cases in the overall test dataset and then randomly distributes those COVID samples throughout the test batches according to a hyperparameter we set (default is 30%). This allows us to use accuracy as a metric when evaluating and tuning our model. Finally, these images have been augmented via translating, rotating, horizontal flipping, and intensity shifting.

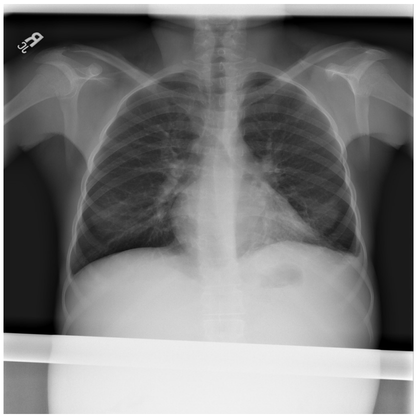


Figure 2: A training example from COVIDx.

The ChestX-ray14 contributes to our model as the dataset which our pre-trained layers were trained upon for CheXnet. The dataset was released by Wang et. al (2017) and contains 112,120 frontal-view X-ray images of 30,805 unique patients. Wang et al. (2017) annotate each image with up to 14 different thoracic pathology labels using automatic extraction methods on radiology reports. Rajpurkar et al. (2017) label images that have pneumonia as one of the annotated pathologies as positive examples and label all other images as negative examples. For the pneumonia detection task, Rajpurkar et al. randomly split the dataset into training (28744 patients, 98637 images), validation (1672 patients, 6351 images), and test (389 patients, 420 images). There is no patient overlap between the sets.

4 Methods

Our learning algorithm is founded upon the DenseNet-121 architecture and pre-trained on CheXPert data. DenseNet-121 is an architecture provided by Huang et al. (2016). The main benefit of using DenseNet-121 is derived from the fact that it contains shorter connections between layers close to the input and those close to the output. This means each layer is connected to every other layer in a feed-forward fashion. Our network therefore has $L(L+1)/2$ direct connections, where there are L layers total. This structure allows each layer to use the feature-maps of all earlier layers as inputs, meaning that there is no need to learn redundant feature maps. Moreover, additional practical advantages of a DenseNet include alleviating the vanishing-gradient problem and strengthening feature propagation. We then modify the later layers of the DenseNet to fit our desired classifications of 'normal', 'pneumonia', or 'COVID-19'.

We use Categorical Cross-entropy loss to optimize our model, because our output has 3 possible classes which are mutually exclusive. The equation is as follows:

$$-\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}1_{y_i\in C_c}log(p_{model}[y_i\in C_c])$$

Where i iterates over N observations, c iterates over C classes, $1_{y_i\in C_c}$ is the indicator function, and $p_{model}[y_i\in C_c]$ is the predicted probability of observation i belonging to class c.

Prior to deciding upon a DenseNet, we attempted to use a MobileNet as our pre-trained model. We thought that this might allow for faster training and therefore greater ability to iterate as more COVID-19 data came out. Additionally, Sandler et al. (2018) prove that the MobileNet architecture requires minimal storage. This means a MobileNet could be easily embedded on-device at low-resource hospitals. Initially, we used Tensorflow's pre-trained MobileNetV2. It is trained and tested on ImageNet data. However, we learned that we did not have the computational resources (more GPU's) nor time to train an entire MobileNet strictly on COVID-19 detection from scratch. Thus, we decided to not use MobileNet as our pre-trained model because relying on the generalized ImageNet data is what we aimed to avoid in the first place. In terms of findings, the pre-trained MobileNetV2 was achieving 85.5% training accuracy and 45.6% validation accuracy.

5 Experiments/Results

As far as hyperparameter tuning goes, we have 4 major options: mini-batch size, learning rate, covid-percent, number of trainable layers, and number of epochs.

First, it is important to note that covid-percent is the variable we use to guide our upsampling of COVID cases when creating our test and train batches. For example, if we set covid-percent = 0.3, this means that each test or train batch will consist of at least 30% of CXR images with COVID-19. These are then randomly distributed throughout the mini-batches of each overall test/train batch. This balancing of each batch is what ultimately allows us to use accuracy as an evaluation metric for our models, instead of AUCROC. We decided to keep covid-percent as 0.3, since this is what Wang et al. (2020) use in their paper. Additionally, we found through trial and error that a learning rate of 0.001 and a mini-batch size of 32 allowed for the fastest training while not harming training accuracy at each epoch.

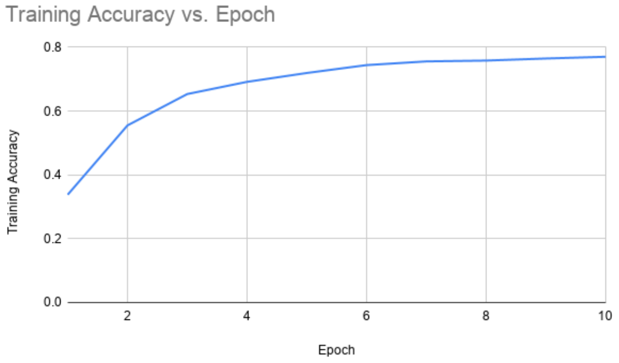


Figure 3: Training accuracy experiences diminishing returns with each increase in number of epochs.

As shown in $Figure\,3$, we found that an epoch of 10 brings us to the point of significant diminishing returns when it comes to achieving maximal training accuracy possible for a given model while ensuring reasonable total training time just under 2 hours.

The most significant gain in training accuracy came from increasing the number of trainable layers of the pre-trained CheXNet model. When we allow roughly 18% of the total layers in the DenseNet to be trainable, we see the improvement in training accuracy shown in $Figure\,4$.

As shown in $Figure\,5$, although training accuracy seems to have increased significantly because of the increased number of traininable layers, what matters most is the fact that test accuracy did not see a significant improvement despite improvements in training accuracy.

These results lead us to believe that our transfer model is indeed learning something while it trains, however, it is not learning anything medically relevant to detecting COVID-19. Therefore, our experimental evidence refutes our hypothesis that transfer learning from CheXNet would be effective when directly applied to detecting COVID-19.

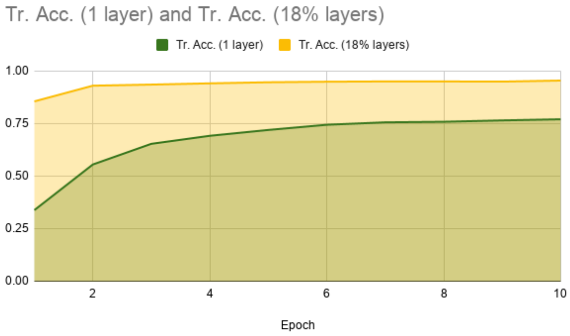
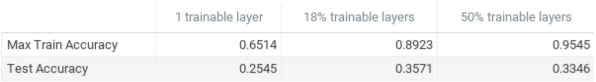


Figure 4: Training accuracy with 18% of the DenseNet layers as trainable is significantly higher at all epochs than training accuracy with only one trainable layer.



	1 trainable layer	18% trainable layers	50% trainable layers
Max Train Accuracy	0.6514	0.8923	0.9545
Test Accuracy	0.2545	0.3571	0.3346

Figure 5: Despite the training accuracy significantly increasing, the test accuracy does not perform well even as we increase the trainable layers to 18% and 50% of the total layers in our model.

6 Discussion/Conclusion

In theory, given the deep learning principles we have learned throughout this course, we had hypothesized that using transfer learning from a pre-trained CheXNet model would be effective for detecting patients with COVID-19. Our results, however, prove otherwise. We postulate that this unexpected result is likely due to factors within the field of medicine and due to the inherent nature of the COVIDx data. In fact, the American College of Radiology states that "the Centers for Disease Control (CDC) does not currently recommend CXR or CT to detect COVID-19" (ACR, 2020).

Additionally, we know that the COVIDx dataset draws all of its non-COVID datapoints from one dataset and its COVID datapoints from two other datasets. We believed that the data augmentation our scripts did would prevent our model from merely learning the institutional differences of where the data was collected from, but it is possible that this still ended up being what our model merely learned to identify from training. In the same vein, we hypothesize, as radiologist Luke Oakden-Rayner argues, that any COVID-19 CXR dataset is likely to suffer from selectivity bias because those with mild/no symptoms are not as likely to request a CXR scan (Oakden-Rayner, 2020), and most COVID-19 cases are mild anyways (CDC, 2020).

However, our results may hint at a more fundamental underlying reason for the inability to detect COVID-19 from x-ray data alone. From a medical perspective, in a non peer-reviewed article published by the Centre for Evidence-Based Medicine at Oxford, there are limited cues for differentiating bacterial and viral pneumonia, and COVID-19 is often associated with the development of multiple thoracic pathologies (Heneghan et al., 2020). Additionally, clinical decisions are rarely made on medical imaging data alone and often incorporate lab values from patients. These two ideas hint at the fundamental incompatibility of detecting COVID-19, let alone diagnosing it, from CXR data alone.

Finally, if we had more time, we would like to test the overlap of COVID-19 indicators with indicators of other infections to better understand how feasible it is for a human or machine truly differentiate COVID-19 from visual information alone. Moreover, we believe there is a way to use heatmaps or other neural network visualization techniques to build a deep learning system which can assist, rather than guide, radiologists in detecting COVID-19.

7 Contributions

1. Ethan (50%) - Brainstormed project ideas, generated COVID dataset, implemented baseline model, implemented mid-quarter model, implemented and debugged final model, wrote final report, recorded/designed final video
2. Hashem (50%) - Brainstormed project ideas, generated COVID dataset, implemented baseline model, implemented mid-quarter model, implemented and debugged final model, wrote final report, designed final video

References

[1] Joseph Paul Cohen, Paul Morrison, and Lan Dao. Covid-19 image data collection. *arXiv 2003.11597*, 2020.

[2] Du H. Gardner L Dong, E. An interactive web-based dashboard to track covid-19 in real time. *The Lancet*, 2020. https://coronavirus.jhu.edu/map.html.

[3] U.S. Centers for Disease Control (CDC). Severe outcomes among patients with coronavirus disease 2019 (covid-19), 2020. https://www.cdc.gov/mmwr/volumes/69/wr/mm6912e2.htm.

[4] Carl Heneghan, Annette Pluddemann, and Kamal R. Mahtani. Differentiating viral from bacterial pneumonia, 2020.

[5] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2016.

[6] Minyoung Huh, Pulkit Agrawal, and Alexei A. Efros. What makes imagenet good for transfer learning?, 2016.

[7] Daniel J. Mollura, Ezana M. Azene, Anna Starikovsky, Aduke Thelwell, Sarah Iosifescu, Cary Kimble, Ann Polin, Brian S. Garra, Kristen K. Destigter, Brad Short, Benjamin Johnson, Christian Welch, Ivy Walker, David M. White, Mehrbod S. Javadi, Matthew P. Lungren, Atif Zaheer, Barry B. Goldberg, and Jonathan S. Lewin. White paper report of the rad-aid conference on international radiology for developing countries: Identifying challenges, opportunities, and strategies for imaging services in the developing world. *Journal of the American College of Radiology*, 7(7):495–500, July 2010.

[8] Luke Oakden-Rayner. Ct scanning is just awful for diagnosing covid-19, 2020. https://lukeoakdenrayner.wordpress.com/2020/03/23/ct-scanning-is-just-awful-for-diagnosing-covid-19/.

[9] American College of Radiology (ACR). Acr recommendations for the use of chest radiography and computed tomography (ct) for suspected covid-19 infection, 2020. https://www.acr.org/Advocacy-and-Economics/ACR-Position-Statements/Recommendations-for-Chest-Radiography-and-CT-for-Suspected-COVID19-Infection.

[10] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, Matthew P. Lungren, and Andrew Y. Ng. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning, 2017.

[11] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks, 2018.

[12] Linda Wang and Alexander Wong. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images, 2020.

Code Libraries/Repos/Frameworks: Tensorflow, Keras, Open-CV, CheXNet-Keras, NumPy, h5py, PANDAS, Scikit-learn, Scipy, Scikit-image, imgaug