Detecting COVID-19 In Chest X-Rays Using Deep Learning

Mina Vukovic & Prabesh Pokharel
COGS 181

***Abstract*—A defining part of 2020 is the COVID-19 and how it has rapidly spread across the whole world. Due to its infectious nature, it is important to be able to detect its presence in someone as quickly as possible to prevent spreading it further. Here we leveraged pretrained models such as AlexNet [9], VGG-19 and ResNet-50 [6] to speed up the training process and train our own convolutional neural networks (CNN) on Chest X-Ray images of patients with COVID-19, pneumonia and no disease. We explore the effectiveness of using deep learning to classify patients as whether or not they have coronavirus as a faster aid to help professionals detect the virus in patients. The overall results showed AlexNet to have the highest accuracy at 88%, followed by our convolutional network in second, and then VGG-19, then lastly ResNet. The results of this study are a step forward into exploring alternative methods for classifying this prominent virus.**

1. Introduction

One of the most important ways to help contain a viral disease is through limiting its ability to spread. In doing so, it is vital to be able to quickly classify if someone is sick or not with high precision and recall so that it is not passed on further. The current method of testing is through a swab that is inserted through the nose, and is then held there for a few seconds in order to collect the secretions [3]. Naturally, this test is rather invasive and can be uncomfortable for the patients being tested, in addition to the results not being instantaneous and needing to be analysed in a lab. They can also have up to a 30% false negative rate [11], although the most common test distributed by Roche achieves around a 95% accuracy [12]. Therefore it is useful to explore other means of detection such as training convolutional neural networks on chest X-ray imagery which can increase the speed and potentially accuracy of COVID-19 detection.

We opted to use Convolution Neural Networks to classify the chest X-ray images into COVID-19 or Pneumonia (a type of lung inflammation caused by bacterial or viral infection) and normal lungs categories. Traditionally we train an entire convolutional neural network from scratch by randomizing the initial parameters or use Xavier/He initialization. However, this process takes a lot of data and time. Due to the limitation of covid chest X-ray data and deep learning architecture being such a data hungry model, we used transfer learning techniques to increase the performance of models as well as substantially decrease the training time. Transfer learning is the process where we transfer the knowledge gained from one task to another task. It usually works best when we take a model trained on a large dataset and transfer its knowledge to a smaller dataset [2].

(a) (b) (c)

Figure 1. (a) COVID-19 (b) Pneumonia (c) Normal

1.1. Related Works

Covid-Net [13] is also a Deep Convolutional Neural Network which is tailored towards detection of COVID-19 cases from chest Xray images. They proposed lightweight PEPX design patterns similar to ResNet's bottleneck design patterns. Each PEPX module consists of two 1×1 convulsions and 3×3 convolutions followed by another two 1×1 convulsions. Covid-Net is able to classify images with 93.3% accuracy.

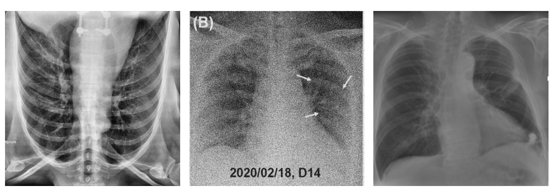
2. Method

2.1. Data

Due to the novelty of the disease, there is not as much data widely available and research is still relatively in its infancy. Thus any dataset that currently exists for public use is limited. The data that we used originates from 5 different sources, which was then combined using this script: https://github.com/lindawangg/COVID-Net/blob/master/docs/COVIDx.md. The data in this script contains different chest X-Ray datasets that are COVID-19, pneumonia and normal X-Rays. The script combines all of these different images into one large test and train dataset. In total, there were 470 COVID-19 images, 6050 pneumonia images and 8851 normal images, for a total of 15,371 images. This is an unbalanced dataset because there were so few COVID-19 images compared to the others, which would lead to unoptimised classification performance for the neural networks.

First we split the data into a shuffled 80:20 train and test split for each class. Then we chose to solve this problem of limited training covid data compared to the rest by augmenting the current training COVID-19 data. For each COVID-19 image, we applied a rotation (45 degrees, 30 degrees and 72 degrees), shift (by 25x25 pixels and 30x30 pixels), horizontal and vertical flip, added noise (sigma=0.1, 0.2 and 0.3) and blurred the image (by factor of 2.5 and 3.5). An example can be seen in Figure 2. This allowed for some variety on the limited images we had to train the models on different orientations or appearances of the images.

Next, since there were still more normal images in comparison to the number of training COVID-19 and pneumonia images we now had, we undersampled this class to be 5000 samples. This finally resulted in the number of images for training being 4888 for COVID-19, 4841 for pneumonia and 5000 for normal. The test data set contains 94 COVID-19 chest X-rays, 1211 pneumonia chest X-rays, and 1771 normal patient's chest X-rays.



(a) (b) (c)

Figure 2. Some data augmentations: (a) Vertical flip (b) Noise (c) Blur

TABLE 1. NUMBER OF IMAGES IN EACH CLASS FOR THE TRAIN AND TEST SPLIT.

CLASS	TRAIN	TEST
COVID-19	4,888	94
PNEUMONIA	4,841	1,211
NORMAL	5,000	1,771

2.2. Neural Networks

We used four models to classify the images and compare the performance: AlexNet, ResNet-50, VGG-19 and our own 4 layer convolutional network.

AlexNet: This is an 8-layer network that comprises 5 convolutional and 3 fully-connected layers [6]. It uses the ReLU function to add nonlinearity instead of the standard tanh function, which leads to a faster training time [8]. Additionally, the authors introduced the concept of dropout, where certain neurons are turned off with a predetermined probability [14]. The network also uses overlapping pooling instead of using the traditional method of pooling neighbouring groups of neurons without overlapping, which decreases the changes of the model to overfit the data [8].

ResNet-50: We used this residual network with 50 layers, although it has many different variants of the architecture using a different number of layers. ResNet uses a global average pooling function, and contains mostly 3x3 filters [10].

VGG-19: This network is 19 layers deep and contains 16 convolutional and three fully connected layers. It uses 3x3 filters like ResNet-50 with stride size of 1, max pooling using 2x2 windows, and also uses the ReLU function for non-linearity [1]. We used VGG with batch normalization which normalizes consequent activation layers to improve training [9].

Our Convolution Neural Network: Our network had 4 convolutional and 2 fully connected layers. Each convolutional layer consists of a 3x3 convolution and Batch Normalization to prevent overfitting and followed by ReLU as our choice of nonlinear function and 2x2 Average Pool at the end as shown in Figure 3. After that, we have two fully connected layers, where we use dropout with the probability of 0.5 to minimize overfitting followed by two linear layers with ReLU in between and softMax for the final classification. An drawing of the network can be seen in Figure 3.

3. Experiment

Our data was gathered from multiple sources so the images were in multiple dimensions. Knowing that CNN does not accept images in the different dimensions, we decided to crop all the images into 224x224. All of our data was transformed into tensor and normalized before being fed into our models. We had an excess of 16 GB of RAM and one Nvidia 1080 Ti GPU which were used to train our models on. For our convolution neural network we used the batch size of four and two workers to simultaneously put data into RAM. We also used Xavier initialization instead of randomizing the weights and biases of each neuron to decrease the vanishing gradient problem. Our choice of loss function is cross entropy loss whose output is a probability value between 0 and 1 and we used Adam optimization algorithm instead of stochastic gradient descent. We trained our models for 20 epochs.

Besides training our small convolution neural network we also used a transfer learning method in our experiment. There are many transfer learning strategies but we used off-the-shelf pre-trained models as Feature Extractors [6]. There are also many pretrained models to choose from, but for our task we chose AlexNet, VGG-19 with batch normalization, and ResNet-50 pretrained models. Our pretrained AlexNet, VGG-19 and ResNet-50 models were trained on ImageNet, with millions of images and thousands of classes [7]. We took these pre-trained models and removed the fully connected layers and created our own fully connected neural network tailored towards our classification task. We treated the rest of the convolution layer of both architecture as a fixed feature extractor.

Just like our small convolution neural network, these pre-trained models were trained on one Nvidia 1080 Ti GPU with 16 GB of RAM. We used a batch size of four and

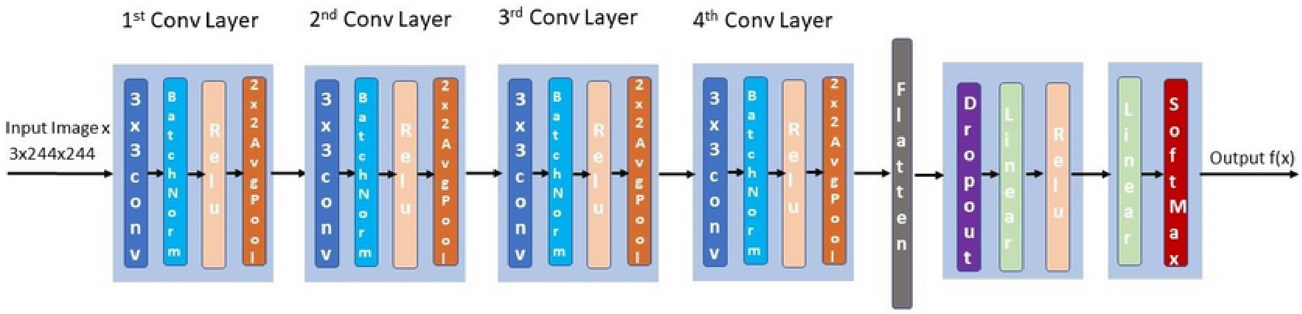


Figure 3. Our convolutional network.

two workers to simultaneously put data into RAM as well. For the fully connected layer of each pretrained model we have four linear layers followed by ReLU in between and then the SoftMax function for the final classification. Additionally, we used the cross entropy loss function and Adam optimization algorithm. We trained our each pre-trained models for 20 epochs as well.

4. Results & Analysis

The results of the experiment showed that overall, AlexNet performed the best with 88% accuracy, followed by our convolutional network, VGG-19-bn and lastly ResNet-50 , as seen in Table 2. These results are relatively surprising, given that AlexNet, VGG-19 and ResNet-50 are high performing models which we did not really see in our results. ResNet-50 performed the worst in our experiment with only 67% accuracy. This might be because ResNet-50 is a large, deep network so it loses the local details of the chest X-rays and there are too many degrees of freedom. Our model performed the second best because we used the best practice of using Batch Normalization and Dropout to reduce overfitting and Xavier initialization to reduce the vanishing gradient problem. We can see the training loss curve in Figure 4.

One likely factor in this overall lower performance is a result of the lack of data we had for COVID-19 chest X-rays. Although we conducted data augmentation to increase the number of images in the training set, we may have done this too aggressively and created too many augmentations of the same image. Thus there is not enough variety in the training set, so our models are unable to truly learn the individual features of the images (especially for COVID-19) to be able to classify them accurately.

Another possible reason for low performance could be the number of epochs we had. Although increasing the number of epochs to be too high can lead to overfitting, perhaps our number of 20 epochs was too low and did not have a chance to reach optimal accuracy.

TABLE 2. ACCURACY OF ALL NETWORKS PER CLASS AND OVERALL.

	COVID-19	Normal	Pneumonia	Overall
Our ConvNet	74.16%	90.78%	84.46%	87.00%
AlexNet	76.04%	92.00%	83.82%	88.00%
VGG-19-bn	43.52%	92..83%	80.73%	84.00%
ResNet-50	20.15%	86.45%	78.28%	74.00%

5. Conclusion

As the effect of COVID-19 on the world continues to grow, it is becoming increasingly important to be able to detect new cases as quickly and accurately as possible. We propose alternative methods of detecting this virus using deep learning classification. Although we do not have high accuracy, it is a step towards exploring this alternative method that given time might improve and become more reliable. As more chest X-ray data becomes available, this experiment can be re-examined so that we do not have to rely on data augmentation techniques and can use more accurate original data. Additionally, we could redo the experiment using higher computing power so that we could increase parameters and the training process which might lead to better performance. Overall, we are not medical experts and these results do not yet compete against the swab method of detecting COVID-19, but there is room for improvement which might lead to more promising results.

References

[1] Aakash Kaushik. "Understanding the VGG19 Architecture." OpenGenus IQ: Learn Computer Science, OpenGenus IQ: Learn Computer Science, 26 Feb. 2020, iq.opengenus.org/vgg19-architecture/. Accessed 13 June 2020.

[2] CS231n Convolutional Neural Networks for Visual Recognition. Github.Io, 2013, cs231n.github.io/transfer-learning/. Accessed 13 June 2020.

[3] Daley, Jim. Here's How Coronavirus Tests Work-and Who Offers Them. 27 Mar. 2020, www.scientificamerican.com/article/heres-how-coronavirus-tests-work-and-who-offers-them/.

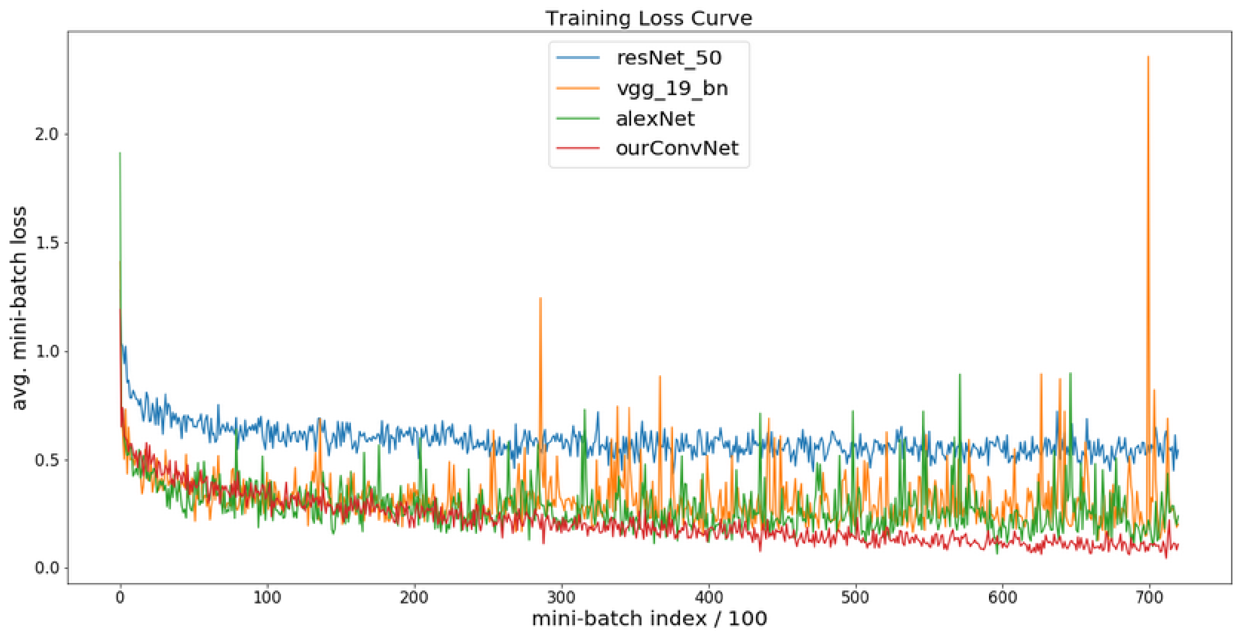


Figure 4. Training loss curve of all the neural networks.

[4] F D. “Batch Normalization in Neural Networks - Towards Data Science.” Medium, Towards Data Science, 20 Oct. 2017, towardsdatascience.com/batch-normalization-in-neural-networks-1ac91516821c. Accessed 13 June 2020.

[5] Hao Gao. “A Walk-through of AlexNet - Hao Gao - Medium.” Medium, Medium, 7 Aug. 2017, medium.com/@smallfishbigsea/a-walk-through-of-alexnet-6cbd137a5637. Accessed 13 June 2020.

[6] He, Kaiming, et al. Deep Residual Learning for Image Recognition.

[7] ImageNet. Image-Net.Org, 2017, image-net.org/. Accessed 13 June 2020.

[8] Krizhevsky, Alex, et al. ImageNet Classification with Deep Convolutional Neural Networks.

[9] Krizhevsky, Alex. One Weird Trick for Parallelizing Convolutional Neural Networks. 2014.

[10] ResNet, AlexNet, VGGNet, Inception: Understanding Various Architectures of Convolutional Networks - CV-Tricks.Com. CV-Tricks.Com, 9 Aug. 2017, cv-tricks.com/cnn/understand-resnet-alexnet-vgg-inception/. Accessed 13 June 2020.

[11] W. Wang, Y. Xu, R. Gao, R. Lu, K. Han, G. Wu, and W. Tan, “Detection of SARSCoV-2 in Different Types of Clinical Specimens,” JAMA, 03 2020.

Walsh, James D. “How Reliable Are COVID-19 Tests?” Intelligencer, Intelligencer, 10 June 2020, nymag.com/intelligencer/2020/06/how-accurate-are-covid-19-tests.html. Accessed 13 June 2020.

[12] Wang, Linda, et al. COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images.

[13] Wei, Jerry. “AlexNet: The Architecture That Challenged CNNs - Towards Data Science.” Medium, Towards Data Science, 3 July 2019, towardsdatascience.com/alexnet-the-architecture-that-challenged-cnns-e406d5297951. Accessed 13 June 2020.