Improving COVID-19 Detection by Using Chest X-Rays

Tiancheng Kuang, Songqiao Li, Yanqiao Li, Heng Yuan

Abstract

Due to the outbreak of COVID-19 pandemic, the infection testing is always fundamental. We aim to build a machine learning model based on the images of patients' lungs X-ray, in order to investigate how our model can successfully identify each image if the patients are having COVID or viral pneumonia. Thus, we think our study is meaningful, and could potentially make contributions to the pandemic. We will be comparing accuracy using two different models, CNN and VGG, later to see which model performs better.

Convolutional Neural Network, also known as CNN, is a class of neural networks that specializes in processing data that has a grid-like topology, such as an image. VGG model is a type of CNN architecture in which the model weights are freely available and can be loaded and used.

Finally, we came up with two models' ACC scores for comparison. As a result, CNN indicates approximately 0.92 ACC and VGG indicates 0.98 ACC, which performs better and gives more accurate classification.

1. Introduction

As of today (3/12/2022), there are 79,761,407 positive cases, with almost 1 million deaths just in the U.S. There are more than 455,234,964 positive cases, with over 6 million deaths worldwide since 2019. Covid-19 is an infectious bacteria caused by severe acute respiratory syndrome coronavirus. The most common symptoms include fever, continuous cough, and breathlessness. Covid-19 is usually diagnosed by an PCR test and often is complemented by chest X-ray. X-ray machines are widely used worldwide and could provide image results quicker than any other test. Moreover, chest X-ray could effectively show the status of infection. Therefore, chest X-ray method is commonly recommended when testing.

Even though Chest X-rays are a fast and inexpensive test that may potentially diagnose COVID-19. However, chest imaging is not a first-line test for COVID-19 due to low diagnostic accuracy and confounding with other viral pneumonias.

Why do we need Chest X-ray if we already have a PCR test which has high accuracy while the chest X-ray is not that precise? Indeed, the most reliable test for the diagnosis of COVID-19 is PCR testing, which involves a throat swab. However, according to a study, PCR provides very few false positives with the test, but some reports suggest a sensitivity of 60% - 70%, meaning that there may be a significant number of infected people who have a negative PCR test. Therefore, the secondary test is required if the condition of the patient gets worse. Combining with PCR and chest X-ray, a more accurate combined result will be better serving the community, especially with the initial lack of sufficient test kits and the rapid growing of positive COVID cases.

Our project goal is to contribute to improving COVID-19 detection using just Chest X-rays. Helping the medical and research community by encouraging them to contribute extensively.

1.1 Dataset

The dataset is relatively simple, splitted with training and testing images, contained with 3 categorical variables: Covid, Normal and Viral Pneumonia (Figure 1.) We will be building some machine learning models to detect if the image is indicating COVID or Viral Pneumonia or neither, by recognizing different chest X-rays images that were given in the dataset.

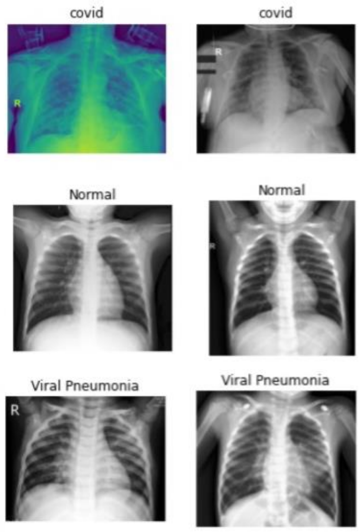


Figure 1. Chest X-ray with different conditions

1.2 Data Preprocessing

To preprocess for our CNN model, we used ImageDataGenerator to rescale, rotate, flip, and shear the images. The reason we did such is because CNN's in-variance property can classify objects even when images in different sizes, orientations, or different illumination. As we can see in the raw images data (Figure 1.), some of the images are not in the same scale with others, some are illuminated, and some are flipped. Hence, we decided to take the images and transform the objects to different sizes by zooming in or zooming out, flipping them vertically or horizontally, or changing the brightness. Therefore, we created a rich, diverse data of images with many variations that could also avoid overfitting problems. In addition, in the ImageDataGenerator, we came to use the parameter rescale = 1/255, such a parameter brings two benefits. (1) Firstly, the 255 is the maximum pixel value, while some images are high pixel range, and some are low. However, all images share the same model, weight and learning rate. Scaling each image will make images contribute more evenly to the total loss. (2) In addition, scaling also helps to reference learning rate from others' results.

2. CNN Model

2.1 Introducing Convolutional Neural Networks (CNN)

CNN belongs to the Neural Network family and allows users to extract higher representations for the image content. Compared to other classic image recognition techniques/methods, for which we have to preprocess the image features ourselves, CNN takes the images' raw pixel data points, trains the model, then extracts the features automatically for better classification.

Similar to the functionality of human brains, CNN classifies the images based on potential meaningful features and capture patterns in order to classify the image as a whole. This distinguishes CNN from other methods as it provides a faster processing speed while sufficient accuracy. In a more technical way of saying this is that, CNN has features parameter sharing and dimensionality reduction. The core concept of CNN is, it applies convolution of image and filters to generate invariant features which are then passed on to the next layer. In this way, the features in the next layer are convoluted with different filters to generate more invariant and abstract features and the process continues till one gets a final feature/output which is invariant to occlusions.

As one researcher points out, "convolutional layers exploit the fact that an interesting pattern can occur in any region of the image, and regions are contiguous blocks of pixels. But one of the reasons why researchers are excited about deep learning is the potential for the model to learn useful features from raw data. Now, convolutional neural networks can extract informative features from images, eliminating the need of traditional manual image processing methods."

2.2 Building and Training the base model

In a Conv Layer, the depth of every kernel is always equal to the number of channels in the input image. So every kernel has K^2 * C parameters, and there are N such kernels. In this case, we've selected 128 as the dimensionality of the output space, and 5 as the kernal_size, specifying the height and width of the 2D convolution window. Here, Rectified Linear Unit (ReLU) is applied, which mimics our neuron activations on a "big enough stimulus" to introduce nonlinearity for values x>0 and returns 0 if it does not meet the condition. This method has been effective to solve diminishing gradients. Weights that are very small will remain as 0 after the ReLU activation function. Moreover, MaxPool is introduced as it replaces output with a max summary to reduce data size and processing time. This allows you to determine features that produce the highest impact and reduces the risk of overfitting.

W_c = Number of weights of the Conv Layer.
B_c = Number of biases of the Conv Layer.
P_c = Number of parameters of the Conv Layer.
K = Size (width) of kernels used in the Conv Layer.
N = Number of kernels.
C = Number of channels of the input image.

$$W_c = K^2 \times C \times N$$
$$B_c = N$$
$$P_c = W_c + B_c$$

Figure 2. CNN parameter

Usually, multiple layers help NN models to learn more detailed and more abstraction relationships within the data and how the features interact with each other on a non-linear level. In this case, we added another layer with half of the dimensionality filter and lower kernel size that allows a hierarchical decomposition of the input. The filters that operate on the output of the first line layers may extract features that are combinations of lower-level features, such as features that comprise multiple lines to express shapes.This process continues until very deep layers are extracting images.

Moreover, a flatten layer is added as it converts the data into a 1-dimensional array for inputting it to the next layer to create a single long feature vector. Also, Softmax activation is added for multinomial probability distribution prediction in this multi-class classification case.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 146, 146, 128)	9728
max_pooling2d_2 (MaxPooling2	(None, 71, 71, 128)	0
conv2d_3 (Conv2D)	(None, 69, 69, 64)	73792
max_pooling2d_3 (MaxPooling2	(None, 34, 34, 64)	0
flatten_1 (Flatten)	(None, 73984)	0
dense_2 (Dense)	(None, 128)	9470080
dense_3 (Dense)	(None, 3)	387

Total params: 9,553,987
Trainable params: 9,553,987
Non-trainable params: 0

Figure 3. CNN parameter choice

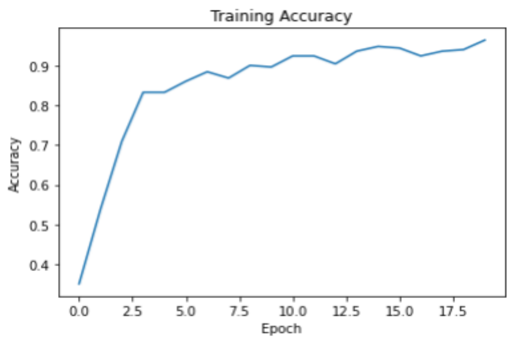
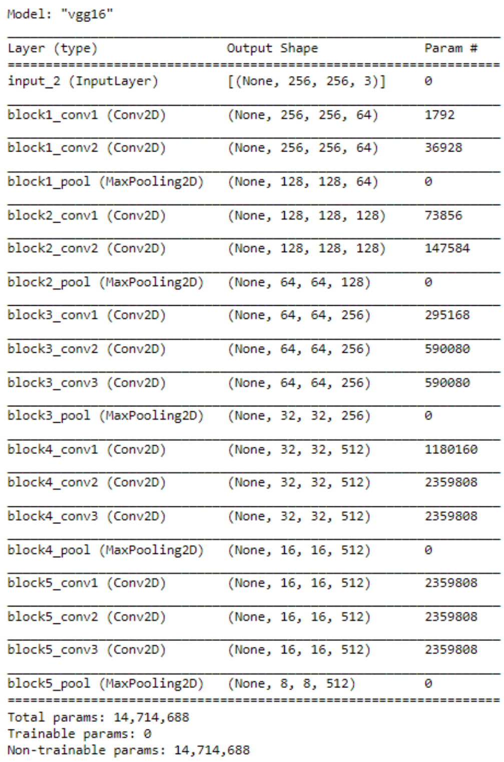


Figure 4. CNN Accuracy

3. VGG Model

3.1 Introducing VGG model

VGG stands for Visual Geometry Group, which is a standard Convolutional Neural Network architecture with multiple layers. The VGG architecture is the basis of ground-breaking object recognition models. Developed as a deep neural network, the VGGNet also surpasses baselines on many tasks and datasets beyond ImageNet. In this case, we applied VGG16, and it supports 16 layers. This means that VGG16 is a pretty extensive network and has a total of around 138 million parameters.



```
Model: "vgg16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 256, 256, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 256, 256, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 256, 256, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 128, 128, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 128, 128, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 128, 128, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 64, 64, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 64, 64, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 64, 64, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 64, 64, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 32, 32, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 32, 32, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 32, 32, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 32, 32, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 16, 16, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 16, 16, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 16, 16, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 16, 16, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 8, 8, 512)         0
=================================================================
Total params: 14,714,688
Trainable params: 0
Non-trainable params: 14,714,688
```

Figure 5. VGG model

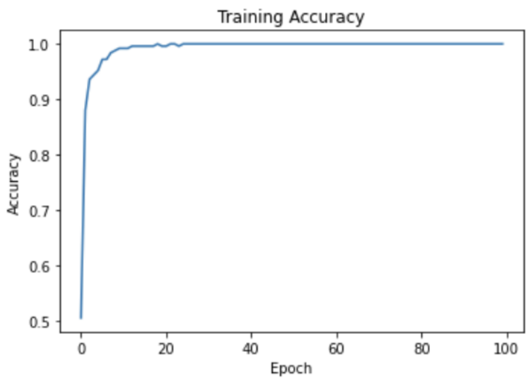


Figure 6. VGG Accuracy

4. Result

Our purpose of analyzing Chest X-ray is to build sufficient enough tool that can help deep learning and AI Enthusiasts to contribute on improving COVID-19. Then, we decided to use accuracy, precision, recall and F-1 score to evaluate the performance and effectiveness of our model. Before we move on to the performance metric, we need to need to emphasize that the dataset we have and training for the case is imbalanced, which means it is necessary to use all the Metrix to evaluate the model's performance.

In order to effective the metric score we used to evaluate the model efficiently, we have to briefly talk about the confusion Metrix since all the calculation are based on it. Confusion Metrix that often used to describe the performance of a classification model on a set of test data for which the true value are know. All the measures can be calculated by True positive(TP), True Negative(TN), False Positive(FP) and False Negative(FN).

True Positive(TP): These are the correctly predicted These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes.

True Negative(TN): These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no.

False Positives (FP) – When actual class is no and predicted class is yes.

False Negatives (FN) – When actual class is yes but predicted class in no.

Below is a Confusion Metrix.

		Predicted Class Label	
True Class Label		Yes	No
	Yes	True Positive (TP)	False Negative (FN)
	No	False Positive (FP)	True Negative (TN)

Figure 7. Confusion Metrix

Accuracy score is the major intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. We could think that, if we have high accuracy then our model is best. And for sure, accuracy is an excellent measure but only when you have symmetric datasets where values of false positive and false negatives are almost on the same level. Therefore, you have to look at other performance Metrix to evaluate the performance of your model.

Accuracy = TP+TN/TP+FP+FN+TN

Precision - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all images that labeled as covid, how many actually covid? High precision relates to the low false positive rate.

Precision = TP/TP+FP

Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class. The question recall answers is of all the images that are truly covid, how many did we label?

Recall = TP/TP+FN

F1 score - F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Although it is not as intuitive as accuracy, F1 is frequently more useful than accuracy, especially if the class distribution is unequal. Accuracy works best if false positives and false negatives have similar cost. It's best to look at both Precision and Recall if the cost of false positives and false negatives is considerably different.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

Below is a summary of these metrics for our first CNN model.

	Precision	Recall	F1-score	Support
Covid	1.00	0.92	0.96	26
Normal	0.80	1.00	0.89	20
Viral Pneumonia	1.00	0.85	0.92	20
Accuracy			0.92	66
Marco. Avg	0.93	0.92	0.92	66
Weighted. Avg	0.94	0.92	0.93	66

Table 1: First CNN Model's Performance 1

CNN Model. Table one show the Metrix of our first CNN model. The model did well on precision for both covid and Viral Pneumonia. However for the normal, instead of putting all of normal chest x-ray into correct category, it recognize it into covid and viral pneumonia, and this is not what we want. The precision of Normal and Recall of Viral Pneumonia could be better, and we believe that we can improve the model.

Below is a summary of these metrics for our second CNN model.(VGG as the basement)

	Precision	Recall	F1-score	Support
Covid	1.00	1.00	1.00	26
Normal	0.95	1.00	0.98	20
Viral Pneumonia	1.00	0.95	0.97	20
Accuracy			0.98	66
Marco. Avg	0.98	0.98	0.98	66
Weighted. Avg	0.99	0.98	0.98	66

Table 2: Second CNN Model's Performance

VGG Model. Table two show the Metrix of our VGG model. The model still did an excellent job on precision for both covid and Viral Pneumonia. And this time, the model has improved the performance for normal images, the precision for normal images can reach to 0.95 and the F1-score for normal images almost equal to 1. At this point, we can say that we got almost 100% accuracy and F-1 score among all categories, which is a distinguishable improvement.

5. Conclusion

In conclusion, our project aims to make contributions to the COVID-19 pandemic, by helping to develop an accurate model to classify chest X-ray images, to identify if the patients have COVID-19 or viral pneumonia. Further, we accomplished the model development by trying two different methods, CNN and VGG. By comparing the result, we successfully built a model which could classify lung X-ray images with an accuracy being 0.98.

We first introduced Convolutional Neural Network (CNN) as our base model. CNN provides excellent image processing capability that captures meaningful patterns in the image instead of doing pixel-wise analysis. Moreover, Visual Geometry Group (VGG16) is used as it provides more layers based on the standard Convolutional Neural Network, providing an extensive network that has a total of around 138 million parameters. Our final model (VGG 16) delivered almost 100% accuracy in all metrics among all categories.

There are few of the challenges we encounter during our Analysis. When considering kernel size, due to our limitation of background in the medical field, we were having trouble deciding kernel size since it is hard to tell which kernel size covers diseased area. Moreover, when deciding the number of layers and learning rate, we spent quite some time tuning to get the optimal performance.

After our analysis, we think there's more techniques we can apply on our model, such as ResNet50 and Xception. These could potentially boost our model performance if we are going to handle a larger similar structured dataset.

Beside our data analysis, we recommend X-ray test as additive testing for PCR. PCR provides very few false positives with the test, but some reports suggest a sensitivity of 60% - 70%, meaning that there may be a significant number of infected people who have a negative PCR test. Therefore, the secondary test is required if the condition of the patient gets worse. Combining with PCR and chest X-ray, a more accurate combined result will be better serving the community, especially with the initial lack of sufficient test kits and the rapid growing of positive COVID cases.

References:

Global COVID-19 Tracker & Interactive Charts | Real Time Updates & Digestable Information for Everyone | 1Point3Acres. (2022). 1point3acres. https://coronavirus.1point3acres.com/

Chest X-ray and CT Scan for COVID-19. (2022, February 25). Verywell Health. https://www.verywellhealth.com/medical-imaging-of-covid-19-4801178

Khandelwal, R. (2021, December 12). *Data Augmentation techniques in python - Towards Data Science*. Medium. https://towardsdatascience.com/data-augmentation-techniques-in-python-f216ef5eed69

Jahn, A. (2021, February 8). *Keras Image Preprocessing: scaling image pixels for training*. Linkedin. https://www.linkedin.com/pulse/keras-image-preprocessing-scaling-pixels-training-adwin-jahn

Covid-19 Image Dataset. (2020, April 29). Kaggle. https://www.kaggle.com/pranavraikokte/covid19-image-dataset

Bhatia, R. (2021, February 15). Why Convolutional Neural Networks Are The Go-To Models In DL. Analytics India Magazine. https://analyticsindiamag.com/why-convolutional-neural-networks-are-the-go-to-models-in-deep-learning/

Exsilio, S. (2016, September 9). Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures. Https://Blog.Exsilio.Com. Retrieved March 10, 2022, from https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/