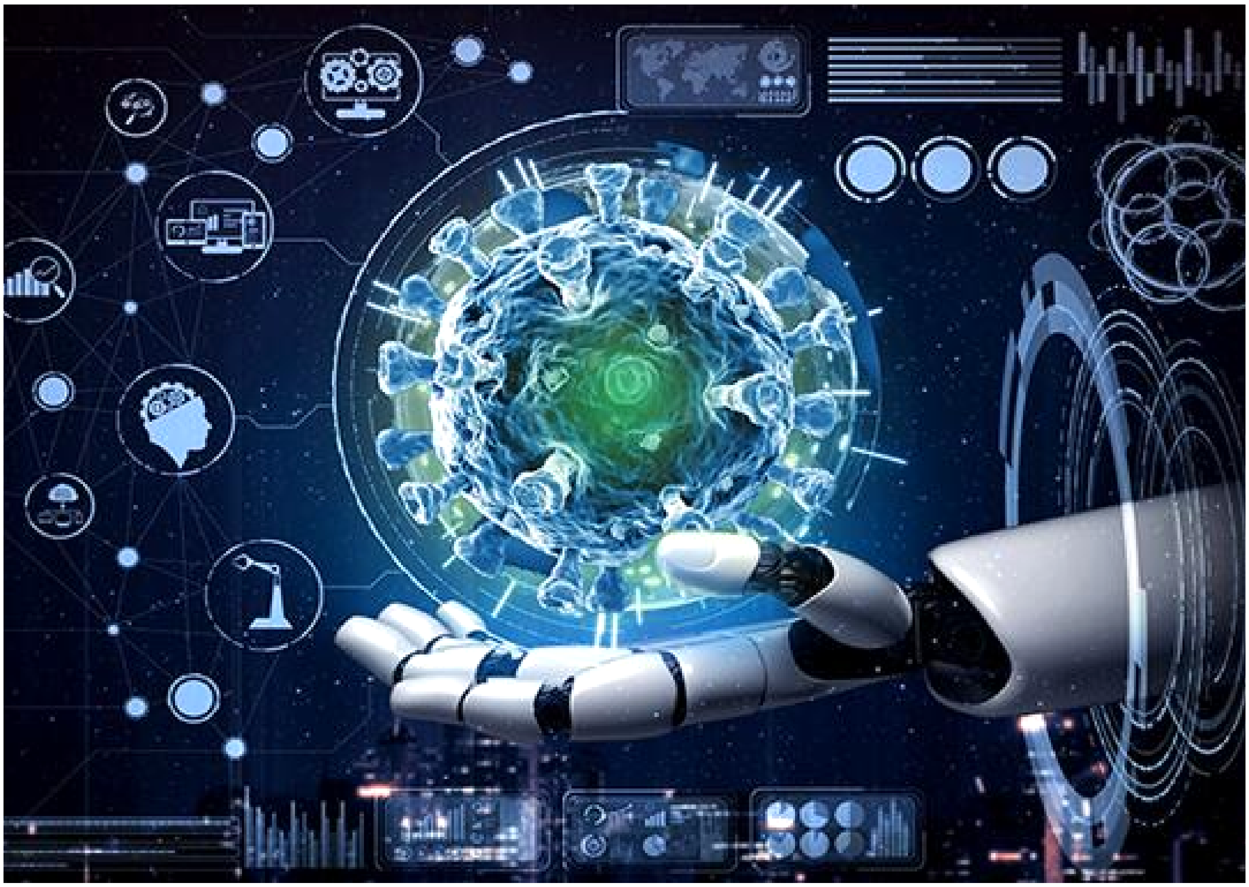
Smart Pneumonia Classification

An enhanced doctor's friend!



Cullan Oberle
Bryan Moore
Marsilinou Zaky

Dr. Wenlu Zhang
CECS 456 Spring 2021

GitHub Repo:

https://github.com/pcoberle/Machine-Learning-semester-project

INTRODUCTION

What is Pneumonia? Pneumonia is an infection that inflames the air sacs in one or both lungs. This dangerous disease can cause many problems, ranging from mild to life threatening. Some of the problems that it can cause is difficulty breathing for a patient; also, their air sacs in the lungs may get filled with fluid or pus. This disease has a mortality rate of 13.4 per 100k people, but is relatively easy to treat if identified correctly.

In this project we worked to create classifiers (CNN) that aim to detect and diagnose people with pneumonia, based on their x-ray image with more than ~94% accuracy rate. We created two different models to do so.

DATA SET

Link: Chest X-ray Images

The data set contains a total of 5856 chest X-ray images of children, including training and test data samples. We broke down the data further to have a separate validation set with the ratio 7-2-1 (training-validation-test).

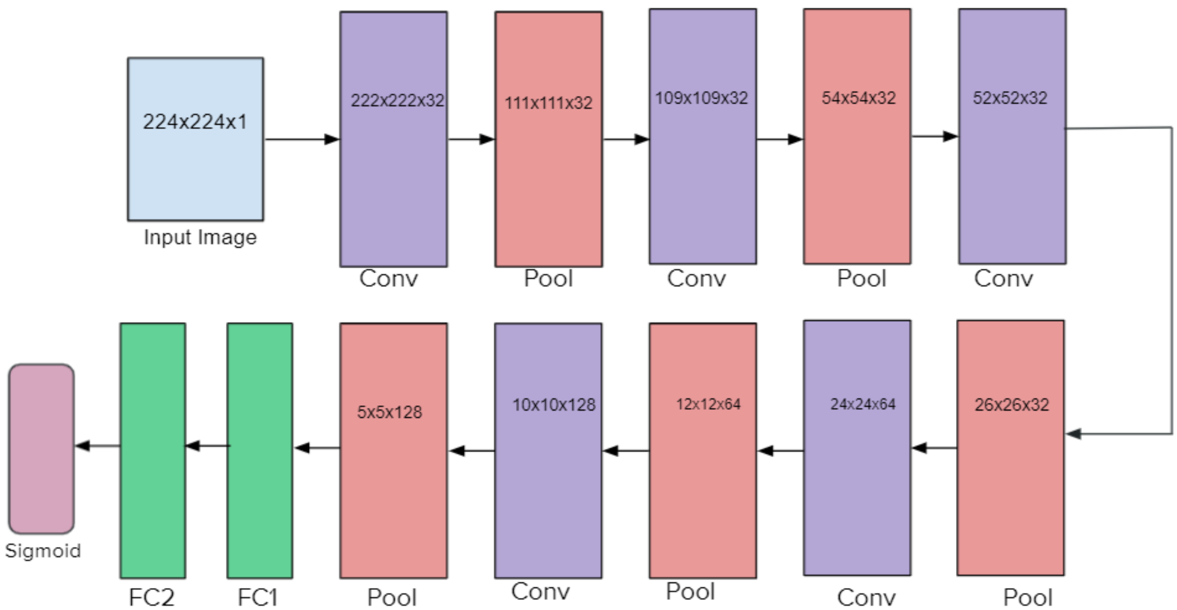
	Train	Validation	Test
Normal	733	557	293
Pneumonia	3366	614	293

PROCEDURE

Firstly, we used data augmentation on the images. The Keras ImageDataGeneratorclass applies multiple random transformations to each training image as it's passed along to the model. Basically, it allows for the expansion of the training data set, which is especially helpful given that our group does not have a big chest x-ray data set. Secondly, for both models we used a learning rate of .00005. We used binary cross entropy for the loss function since we are using binary classification to detect whether an x-ray shows pneumonia or not. Thirdly, we also used a sigmoid activation function on both models because we are doing binary classification. For both models we had a batch size of 32. Finally, before training the models, we resized all images to 244x244, with the 3rd dimension being 1 because the x-rays are grayscale images.

We ran each model for 50 epochs, pocketing each model at its minimum validation loss. Researching online, we found that it was common to use validation loss as the metric to watch because it shows the certainty a model has for its predictions, rather than accuracy, which gives a percentage of correct guesses. It should be noted that this is still a fairly subjective issue currently; however, the minimum validation loss and maximum accuracy are usually found in the same epoch anyway.

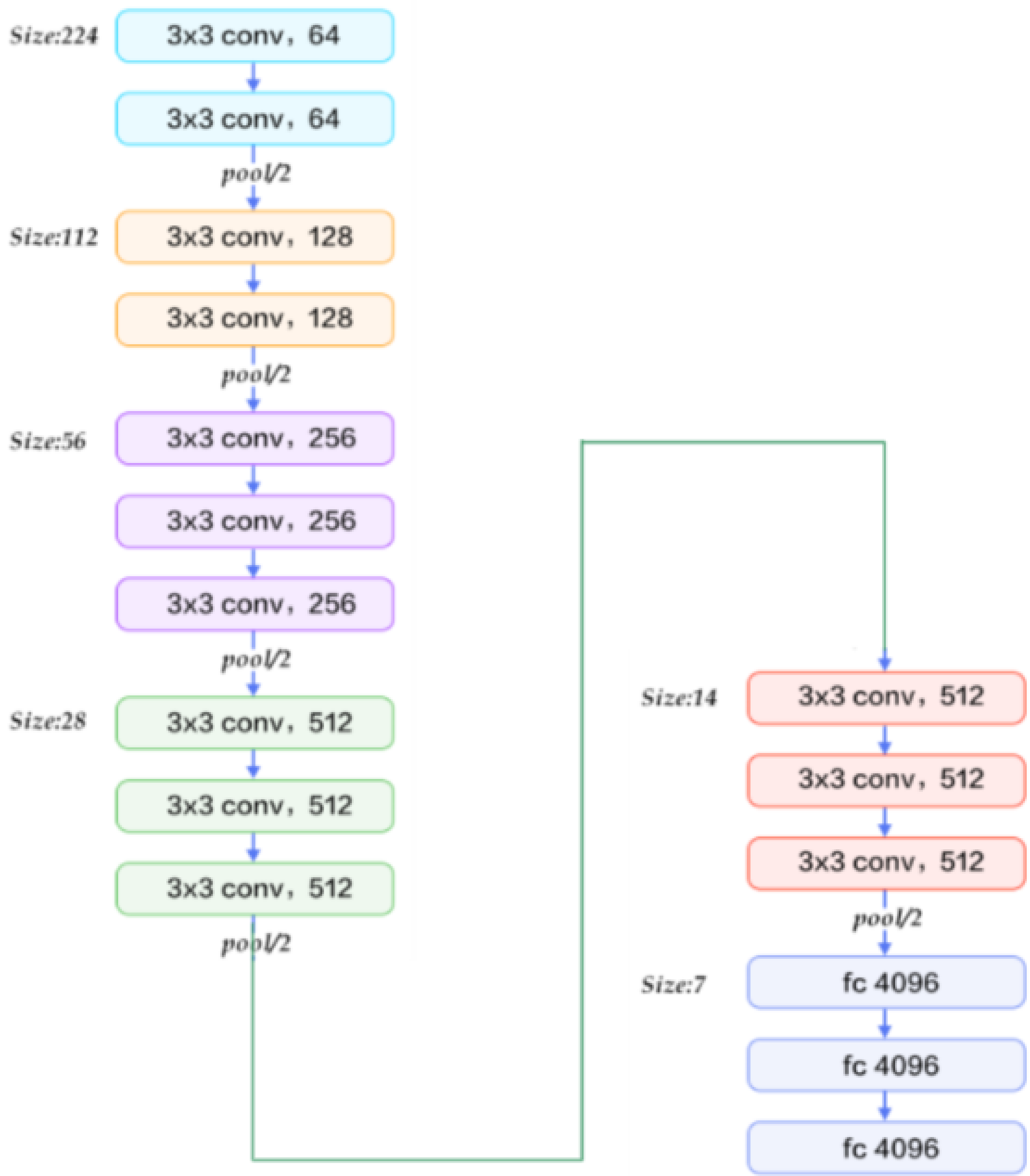
OUR MODEL



Total parameters: 521,025

For our custom model we added five different convolution layers alternating with max pool layers, followed by two fully connected layers. At the end we used a sigmoid activation layer. We used sigmoid activation because it was much better since we're dealing with binary classification (normal, abnormal). In this model we used a convolution window of 3X3 and stride 1X1, with 20 epochs. This model yielded 251,025 parameters. This model had accuracy of 92.2%.

VGG16

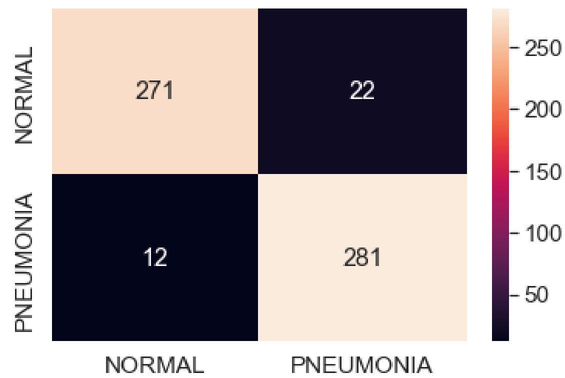


Total parameters: 134,263,489

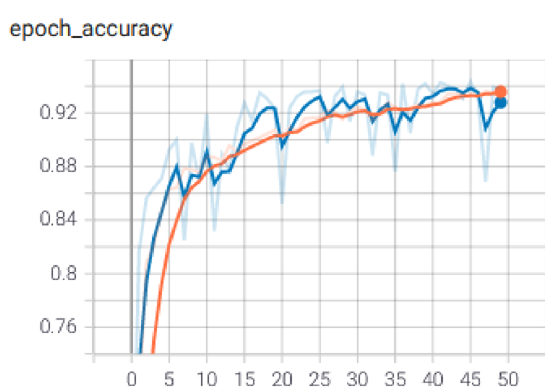
For the VGG16 model we have 13 convolution layers in total, and 5 max pooling layers. Convolution window was 3X3 for all layers, and the stride was 1X1 for all layers as well. We started with two convolution layers, followed by a max pooling layer. The result will be reduced by half, meaning the size will become 112. The same combination of convolution layers and max pooling layer is repeated again. After that there are three convolution layers followed by a max pooling layer. This combination is repeated three times. The size gets halved after every max pooling layer, so we end up with size 7. At the end there are three fully connected layers. After the first and the second layer we did a dropout of 0.5 with ReLU activation; in the last layer we used sigmoid activation. This model yielded 134 million parameters.

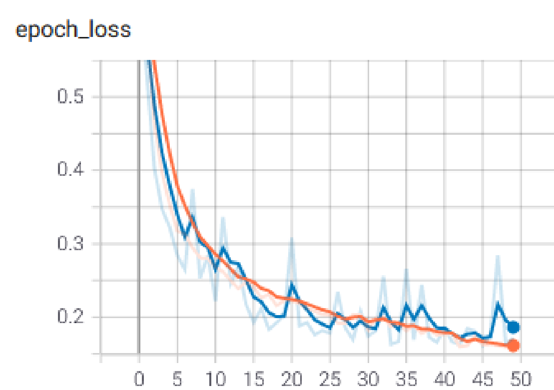
PERFORMANCE DATA

Our simple model's performance data:



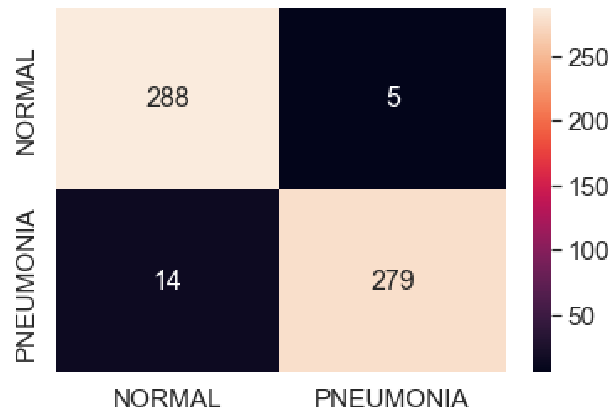
	precision	recall	f1-score	support
NORMAL	0.96	0.92	0.94	293
PNEUMONIA	0.93	0.96	0.94	293
accuracy			0.94	586
macro avg	0.94	0.94	0.94	586
weighted avg	0.94	0.94	0.94	586



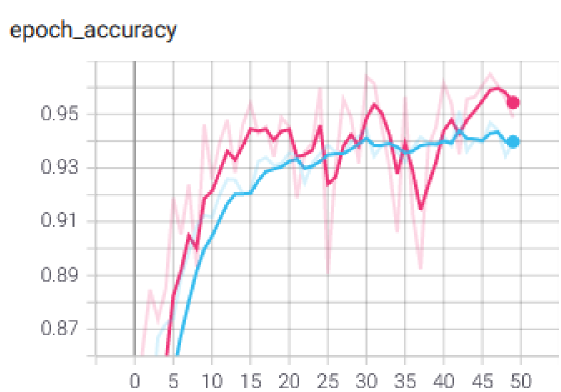


Blue is training data Orange is validation data

VGG16’s performance data:



	precision	recall	f1-score	support
NORMAL	0.95	0.98	0.97	293
PNEUMONIA	0.98	0.95	0.97	293
accuracy			0.97	586
macro avg	0.97	0.97	0.97	586
weighted avg	0.97	0.97	0.97	586



Blue is training data Red is validation data

RESULTS

From this data, we see that although the deeper VGG16 model had higher fluctuations in loss and accuracy over 50 epochs than the simpler model, the point at which the model did best (minimum validation loss) it had a higher f1-score (TP/(TP+0.5*(FP+FN))) at 0.97 vs the simple model's 0.94 giving VGG16 a 3% advantage.

CONCLUSION

Overall, with the help of data augmentation, we were able to artificially increase the data set which resulted in a better result for both models. The VGG16 model gave us better accuracy when it came to predicting whether an x-ray had pneumonia compared to our model. However, due to the VGG16 being deeper as a result of more parameters, it took much longer to go through 50 epochs. In the future, as we get more data and models improve, machine learning will have a big impact on our healthcare system due to its ability to diagnose medical conditions with such high accuracy.

REFERENCES

1. Kaggle: Chest X-ray Images by Tolga Dincer
2. Pneumonia - Symptoms and causes - Mayo Clinic