Diagnosing COVID-19 from Chest X-rays: A Simple Neural Network Perspective

Daniel Pham

Gurbir Bhangu

Student ID: 861196372

University of California Riverside

Data Science Specialization

1. **Abstract**

The use of AI, and more specifically Convolutional Neural networks (CNN), in medical scenarios to help screen and diagnose disease has been a heavily researched topic. With the rapid spread of coronavirus disease (COVID-19) across the globe in the past 3 years, we must consider what tools and techniques are necessary to reduce the burden on the current medical systems, and also prepare ourselves for future pandemics. Our paper aims to establish a methodology and model to help develop a lightweight CNN that can accurately diagnose COVID-19. Our proposed model was built from no previous architecture but was compared with an existing image classification model named VGG-16. Chest X-rays were classified into 4 classes: normal, lung mask, COVID-19, and viral pneumonia. All data was obtained from an openly accessible database on Kaggle. Our proposed model was able to achieve 93.83% accuracy, 97.33% sensitivity, and 98.44% specificity. This proposed model and methodology show promise as viable alternatives to help with the automatic screening of COVID-19. Overall, our project can help reduce the burden on medical professionals, while increasing the efficiency of medical services and systems.

1. **Introduction**

 The current healthcare industry will benefit from a disease diagnostic system that can detect certain diseases present in scans with speed and accuracy. However, due to current limitations on computing power, there remains a consideration between balancing performance and computational cost. We attempt to create a model with low computational cost while maintaining relatively high accuracy in detecting disease, and compare this with a complex model, with a larger associated computational cost. Our work will primarily attempt to detect the presence of coronavirus disease 2019 (COVID-19) and viral pneumonia from an X-ray scan of the chest mid-section. We first discuss the scope of the problem and the history of work done in the field.

* 1. **Coronavirus and the COVID-19 Pandemic**

Coronavirus disease 2019 (COVID-19) first emerged in December 2019, when clusters of patients with pneumonia with no known cause were identified in the city of Wuhan, China (Zhu et al., 2020). As of December 18th 2022, over 649,000,000 cases and 6,600,000 deaths from COVID-19 have been confirmed and reported globally (World Health Organization, 2020). With the introduction of effective vaccinations for the disease, varying degrees of control has been achieved over the pandemic (Christie et al., 2021).

The rapid spread of COVID-19 across the globe demonstrated a severe lack of preparedness in the country’s ability to respond to a pandemic of this scale. Additionally, ways to accurately diagnose COVID-19 during the early days of the pandemic primarily centered on PCR and antigen testing. PCR testing can detect the SARS-CoV-2 virus approximately 95% of the time, and antigen testing can detect the SARS-CoV-2 virus approximately 80% of the time (Food and Drug Administration, 2022). With this project, we hoped to introduce an alternative detection method for COVID-19, with potential generalizability to future pandemics, so that doctors and public health officials could respond with more urgency and precision in the future.

* 1. **Convolutional Neural Networks**

Convolutional neural networks (CNN) have shown incredible promise as a tool for solving pattern recognition problems, and have been the most general method to analyze visual imagery (Li et al., 2021). Neural networks are generally known as computing systems that draw inspiration from biological neural networks, such as those of the human brain (Cox & Dean, 2014).  Convolutional neural networks primarily operate with three different types of layers: convolution, pooling, and fully connected layers. Convolution and pooling layers are responsible for feature extraction, while the fully connected layers are responsible for mapping these features into classification outputs (Yamashita et al., 2018). As convolution and pooling layers are stacked upon each other, we can extract increasingly complex features which result in the high-performance levels seen in many CNN models.

Since our project is primarily a classification problem at heart (attempting to accurately classify lung scans with COVID-19 or viral pneumonia as such), the usage of a CNN model seems like the most natural choice. As previously mentioned, the number of layers increases the levels of complexity detectable by our CNN but also increases the computing cost of such a model. There arises a question about balancing computational costs and the performance of the overall model, which our project will attempt to glean insight into.

* 1. **Disease Detection using Artificial Intelligence models**

Disease detection and diagnosis by machine learning models is a rapidly growing field of research. The models involve inputting correct diagnostic patient records, the model then analyzing said records, and finally, a diagnosis is made and presented as the output (Fatima & Pasha, 2017). Based on this, doctors with access to such technology can diagnose patients quicker, diagnose patients more accurately, and more efficiently treat the patient. The introduction of highly reliable disease detection models across medical institutions can disrupt current medical systems. Doctors are increasingly feeling more burnt out across all stages of their careers than the general population (Dyrbye et al., 2014). It is feasible to introduce AI models to reduce the workload and burden on doctors, thus improving their mental health and reducing feelings of burnout (Rodriguez-Ruiz et al., 2019).

Our project will primarily focus on detecting COVID-19 or Viral Pneumonia cases from chest X-ray scans. By pre-screening chest x-rays through such a model, we hope that we can limit the number of scans that must be extensively reviewed by doctors to correctly diagnose patients. Additionally, if our model excels at correctly verifying scans that do not possess any disease, it can serve as a pre-screening tool that indicates a scan is likely clear. This paper will first cover the previous literature in the space of machine learning models to diagnose COVID-19 and associated viral pneumonia. Then, we will discuss the methodology employed to discover how to potentially create a lightweight CNN model, capable of diagnosing COVID-19 at comparable rates to a more complex CNN, VGG16. Our paper hopes to contribute to the wider topic of medical diagnosis via AI models.

1. **Literature Review**

The field of COVID-19 diagnosis by convolutional neural networks has been heavily explored since the onset of the pandemic. We will briefly review some of the efforts made in the field in prior years, what questions were possibly left unexplored, and how our work on this project contributes to a wider effort to integrate AI medical diagnosis at more significant levels into healthcare.

Goel et al. (2021) presented an “OptCoNet” model to diagnose COVID-19 from chest X-ray images. The Optimized Convolutional Neural network (OptCoNet) is primarily centered around optimizing feature extraction and classification components. The model achieved 97.78% accuracy with 97.75% sensitivity and 96.25% specificity.

Abbas et al. (2021) used a Decompose, Transfer, and Compose (DeTraC) model to address the same issue. The VGG-19 neural network was used in conjunction with the DeTraC method. The three-phase model investigated attempted to improve model outcomes with irregularities in the image dataset. An accuracy of 93.1% with a sensitivity of 100% was reported.

In a model using VGG-16 like our project, Sitaula et al. (2021) attempted to improve COVID-19 diagnosis accuracy by introducing an attention module that specifically helps to highlight the regions of interest where COVID-19 likely affects the lungs. The model was able to achieve an accuracy of 79.58% to 87.49% across multiple datasets that it was trained and tested.

Using a modified version of the existing CNN architecture provided by MobileNet and ResNet, Jia et al. (2021) classified COVID-19 infections, non-COVID-19 infections, and normal controls, and then further classified the images into COVID-19, Tuberculosis, viral pneumonia in non-COVID-19 cases, bacterial pneumonia, and normal controls. The modified MobileNet network addressed the former issue, and the modified ResNet network handled the latter issue. Test accuracies were 99.6% on chest X-ray data and 99.3% on CT images.

From this research, it is evident that the field of COVID-19 diagnosis using computer models is an intensely studied subject. However, the research primarily operates on using existing CNN architectures to base their model design. While this is an idea that logically makes sense, there seems to be a lack of exploration into how complex a model must be to achieve similar results to those seen above.

While all the above research has discovered incredible methods to handle and accurately diagnose COVID-19 based on chest X-ray images using CNNs, the question remains if it is all necessary. In our project, we approach the problem of building a model that can accurately detect COVID-19 present in chest X-rays from a novel perspective. Our proposed model and method are plausible alternatives to existing research methods that simply build from known, well-performing models.

1. **Methodology**

For our project, we implemented a basal-level CNN using TensorFlow and Keras API and then compared that to a sophisticated CNN known as VGG-16, a model that was the first runner-up in the ImageNet competition in 2014. The VGG-16 model is a CNN that is 16 layers deep and relies on the use of accuracy as its main measure. The overall architecture of the VGG-16 model is depicted in **Figure 1**.



Figure 1. VGG-16 architecture

There was an initiative led by a team of researchers from Qatar University to create a public database of chest X-rays for COVID-19 positive cases along with other cases, such as normal, lung opacity, and viral pneumonia scans (Chowdhury et al., 2020). They developed the database of COVID-19 x-ray images from the Italian Society of Medical and Interventional Radiology COVID-19 database and another dataset in GitHub that consists of images extracted from 43 different publications by Joseph Paul Cohen. By using the provided database of chest X-ray scans, we first created train, valid, and test folders to separate the images for training, validation, and testing. We chose to preprocess the images by using the very same preprocessing methods used for the original VGG-16 model. This is to minimize the variance between the VGG-16 and our model’s results so we can do a fair comparison. The preprocessing subtracts the mean RGB value of the image, computed on the training set, from each pixel color channel.

We designed our CNN, fine-tuning how many layers the network should have, and tested what optimizer would best suit the model. Then, we emphasized the accuracy of the model as our primary performance measure. Accuracy is a great indicator to see if the model is training properly, how it will generally perform on data it has not seen before, and how our lightweight model compares to the VGG-16 model. Since accuracy is acting as our main statistical metric, the goal is to get our model’s accuracy as high as we could without making the neural network too complex before we compare it to the VGG-16 model. Other statistical tests that were also used in this project include sensitivity and specificity as they were the most widely used measurements for COVID-19 diagnosis models from previous studies (Alyasseri et al., 2021).While keeping accuracy, sensitivity, and specificity in mind, we optimized for those metrics by tuning hyperparameters, addressing overfitting, and utilizing different kinds of data augmentation. The hyperparameters tuned include adjusting the number of epochs and the learning rate of the model. The methods used to alleviate overfitting issues include increasing sample size, implementing dropout, and applying early stopping. The use of data augmentation allows the end-user to artificially increase the training set by creating modified copies of existing data usually through minor geometric transformations. This generates new data points for the model to learn from, which enables the machine learning models to perform better and become more robust to unseen data.

In this project, various data augmentation techniques were carefully chosen and used. Certain augmentations are incompatible due to the nature of the data we are working with. The data augmentations techniques selected were in line with what we potentially see in real-world scans. The data augmentation techniques used in this project include random rotation, width shift, zoom, brightness, and horizontal flip. The random rotation augment can rotate images between 0 and 360 degrees. Sometimes when taking chest X-ray scans, people are not always going to be lying down perfectly straight. The random width shift augment can allow images to be shifted on the horizontal axis by a constant pixel value. In some cases when taking the chest X-rays, the patient may not always be in the center of the image. The zoom augmentation can randomly zoom in and out of the image. Generally, in the real world for X-ray scans, there will always be some varying degrees of zoom level, especially across clients in different parts of the world. The brightness augment can change the light intensity throughout the image. In general, X-ray machines are capable of utilizing varying grayscale contrast, usually controlled automatically or changed by doctors. The horizontal flip augment can be used to flip the images on the horizontal axis. This was used to increase the diversity of the dataset.

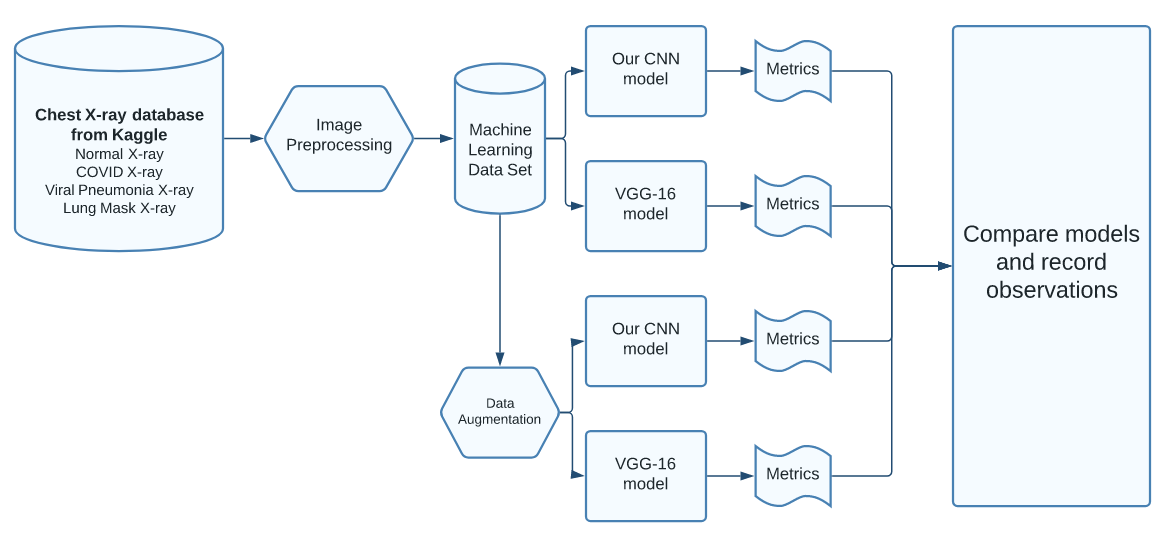


Figure 2. Model Building Process

Additionally, since we were using these statistical measures to test how our CNN performed, a visualization of the confusion matrix was implemented to help interpret results more easily. If any of the CNNs in this project can achieve a high value for all three statistical tests: accuracy, sensitivity, and specificity, then CNNs would be able to serve as a great technological tool in identifying COVID-19 cases quickly and as accurately as possible, especially when further optimized and implementing a more complex model. The overall approach in the model-building process is depicted in **Figure 2**.

1. **Results & Discussion**

We first implemented the VGG-16 model and ran the model through the chest X-ray scans dataset. Initially, the replicated VGG-16 model was having problems with the Adam optimizer. The Adam optimizer is an optimization algorithm that stands for adaptive moment estimation. This optimizer is an extension of stochastic gradient descent (SGD) because it uses estimations of the first and second moments of the gradient to adapt the learning rate for each weight in the neural network. The Adam optimizer and its variations are some of the most widely used optimizers in neural networks. However, no matter what we did with the replicated VGG-16 model, the model during training would only see a stagnant validation accuracy of 27.27% along with very minor improvements in the loss function and its training accuracy, which was roughly 29.94%. We started to doubt the use of the VGG-16 model in classifying COVID-19 chest X-ray scans due to the poor results. However, after swapping out the Adam optimizer for SGD, the replicated VGG-16 model started to produce results that were matching our expectations. The VGG-16 model with the use of the SGD optimizer had a training accuracy of 94.29% and a validation accuracy of 92.97%.

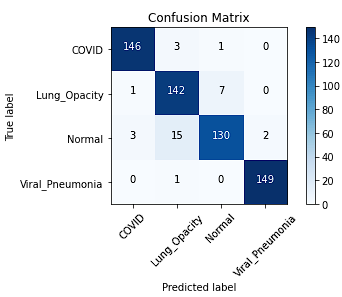


Figure 3. Confusion Matrix for VGG-16 using SGD

We then had the model make predictions on the test set and compiled the results into a confusion matrix, as depicted in **Figure 3**. The accuracy of the VGG-16 model on the test set is 94.5%. For this particular multi-class problem, we calculated the sensitivity and specificity for the COVID-19 class. The sensitivity is 97.33% and the specificity is 99.11%. These 3 values will serve as the standard values of what a complex model can achieve in this problem.

After establishing results with the VGG-16 model, we then moved on to developing the architecture of our model. We first started with a barebone CNN. It had 2 hidden layers, trained for 10 epochs, had a learning rate of 0.001, trained on a small sample size of 500 images consisting of 125 images from each class, and utilized the Adam optimizer for the model-building process. In the end, the model started to overfit after 5 epochs and we decided to mainly concern ourselves with how the CNN performed before it started to overfit going forward. Therefore, this model had a 100% training accuracy and an approximate 80% validation accuracy before overfitting. It was noted that this model with the aforementioned parameters was computationally expensive since the model had a total runtime of about 40 minutes, averaging 4 minutes per epoch. We first wanted to figure out a way to reduce this computational time since training future iterations would be infeasible.

We then tried adjusting the target size for images in the model. Originally, the target size was set to 224x224 pixels. This was the same target size used in the original VGG-16 model. We adjusted this to 128x128 pixels, which is about 3 times fewer inputs compared to before. Similarly, to the previous training, the model started to overfit after 5 epochs. Before overfitting, it had a training accuracy of 98.5% and a validation accuracy of 77.75% with each epoch having a decreased runtime averaging 80 seconds.

Next, we implemented dropout to the CNN. The dropout layer was only included in between the hidden layers with no dropout layer being added between the final hidden layer and the output layer. After training for 10 epochs, the training accuracy was 91.1% and the validation accuracy was 74.5%. Although it seemed like the model was performing worse, it was noted that both training and validation accuracy kept improving with every epoch and that overfitting did not seem prevalent. Later, we would realize that we did not provide this iteration of the model with enough epochs to reach its full potential since training and validation accuracy only kept improving.

Moving along, we doubled the sample size of each class, from 125 images to 250 images. This increased the total samples that the model trained on to 1,000 images. We observed that compared to the former, the training accuracy increased to 95.8% and the validation accuracy decreased to 69.5% with each epoch having an increased runtime of about 105 seconds. Initially, it was hard to gauge here what kind of effect the increased sample size had here for training and validation accuracy since both had minor positives and negatives. If either the number of epochs or the sample size were increased, we would have seen more definite effects. But the one thing we were able to see was that each epoch on average took 25 seconds longer than the previous iteration. Since the full actual dataset is much bigger, if we truly wanted to utilize most of this dataset, we still needed to figure out a more efficient way of decreasing the computational time.

The next thing we tried to do was adjust the learning rate of the model during training both ways. When we lower the learning rate from 0.001 to 0.0001, the model achieved a training accuracy of 99.95% but the validation accuracy was 74.5%. It seems that in this case, both the training and validation accuracy had slight improvements. But when we increased the learning rate to 0.01, the model had a training accuracy of 24.1% and a validation accuracy of 25%. The model was not able to converge to a global optimum properly due to the high learning rate and this was our worst iteration for the model. We chose to maintain the model’s learning rate of 0.001 going forward since it will let our model converge faster than a learning rate of 0.0001.

Moving along, we implemented another hidden layer to the model with dropout and increased the number of epochs to 100 for once to see what the results would look like. First, we looked at how the model performed after 10 epochs to understand the baseline of adding another hidden layer, compared to the previous iterations. After 10 epochs, the model had a training accuracy of 79.6% and a validation accuracy of 76.5%. After 100 epochs, the model had a training accuracy of 99.95% and a validation accuracy of 83%. On average, the runtime for each epoch was increased to 110 seconds and the total runtime for this iteration of the model was 210 minutes. When we compare the results from this iteration after 10 epochs with the previous iteration when the learning rate was still 0.001, the training accuracy decreased from 95.8% but the validation accuracy increased from 69.5%. As validation accuracy is a better indicator of the model’s performance to unseen data, adding another hidden layer to the model allows the model to achieve better results. This is further proven when looking at the results after 100 epochs as both training accuracy and validation accuracy still improved. Although the results of adding another hidden layer were promising, the accuracy can still be further improved, but the computational time was also steadily increasing.

The implementation of the model was all computed on the CPU. To address the increasing computational time, we added GPU integration to our model and saw significant improvements to the runtime. We re-ran the previous iteration of the model and saw that the runtime for each epoch was averaging about 18 seconds, which was an over 600% improvement in computational time. Since the computational time was significantly reduced, we adjusted the total sample size to 10,000 images for the model to train on. After 10 epochs, the training accuracy of the model was 88.57% and the validation accuracy was 91.09%, with each epoch averaging 20 seconds. We then reverted the target image size of 128x128 pixels to the original VGG-16 target image size of 224x224. After 10 epochs with this change, the model had a training accuracy of 87.43% and a validation accuracy of 87.27%.

Since we now have a good understanding of how each hyperparameter affects the model and a computationally inexpensive model setup, we looked to fine-tune the model’s accuracy without overfitting. We increased the number of epochs to 100 and implemented early stopping to the model, which allows the model to continue to train on the training dataset but stops training once the performance of the validation dataset starts to degrade. Essentially, early stopping interrupts the model once it starts to overfit. The trigger to interrupt the model was specified by patience, which is the number of epochs when the monitored metric has shown no improvement. Initially, the patience in our early stopping implementation was set to 2 but it was then adjusted to 3 since “results indicate that a slower criteria, which stop later than others, on average lead to improved generalization compared to faster criterias” (Prechelt, 1998). Lastly, we added a fourth hidden layer with dropout to the model. This is the last change we did to our model’s architecture.

To summarize, our finalized model has 4 hidden layers with dropout in between each of the hidden layers, trained for 100 epochs with early stopping interrupting the model once it starts to overfit, has a learning rate of 0.001, and is trained on a sample size of 10,000 images. This model and the VGG-16 model are then tested not only by using the Adam and stochastic gradient descent optimizers but also with and without data augmentation.

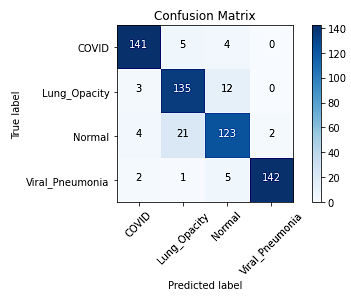


Figure 4. Confusion Matrix for Our Model using Adam

The results of our model utilizing the Adam optimizer and without any data augmentation are depicted in **Figure 4**. The accuracy of this model on the test set is 90.17%, the sensitivity is 94%, and the specificity is 98%.

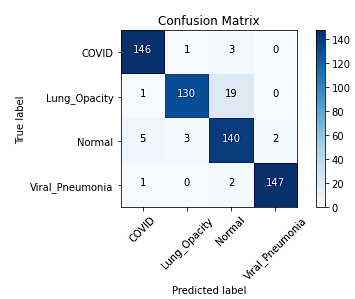


Figure 5. Confusion Matrix for Our Model using Stochastic Gradient Descent

The results of our model utilizing the stochastic gradient descent optimizer and without any data augmentation are depicted in **Figure 5**. The accuracy of this model on the test set is 93.83%, the sensitivity is 97.33%, and the specificity is 98.44%.

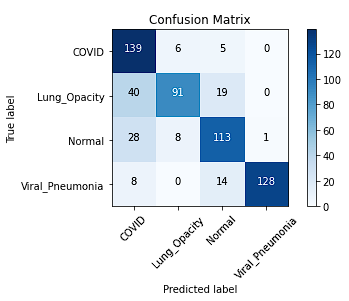


Figure 6. Confusion Matrix for Our Model using Stochastic Gradient Descent and Data Aug

The results of our model utilizing the stochastic gradient descent optimizer and data augmentation are depicted in **Figure 6**. The accuracy of this model on the test set is 78.5%, the sensitivity is 92.67%, and the specificity is 83.11%.

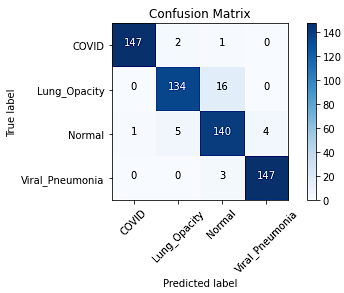


Figure 7. Confusion Matrix for VGG-16 using Stochastic Gradient Descent and Data Aug

The results of the VGG-16 model utilizing the stochastic gradient descent optimizer and data augmentation are depicted in **Figure 7**. The accuracy of this model on the test set is 94.66%, the sensitivity is 98%, and the specificity is 99.78%.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Optimizer | Data Aug | Accuracy | Sensitivity | Specificity |
| VGG-16 | SGD | û | 94.50% | 97.33% | 99.11% |
| Our | Adam | û | 90.17% | 94% | 98% |
| Our | SGD | û | 93.83% | 97.33% | 98.44% |
| Our | SGD | ü | 78.50% | 92.67% | 83.11% |
| VGG-16 | SGD | ü | 94.66% | 98% | 99.78% |

Figure 8. Summary of Results

The summary of the results is depicted in **Figure 8** for all the ran models that were in this project. The Adam optimizer was not successfully utilized as much as SGD because of the issues we encountered with it. In some cases, the model returned a stagnant validation accuracy for every epoch. In other cases, the model returned a NaN value for the loss function. According to the results, the model that used the Adam optimizer performed the worst when comparing models that did not utilize data augmentation. If we compared the base VGG-16 model to our best lightweight model, both of them had very similar results in the performance measures. Out of the three performance measures, we were most concerned about the sensitivity of the model since the sensitivity tells us out of all the cases that have the disease, how many of those cases were returned positive by the model? According to the results, both the base VGG-16 model and our lightweight model had the same sensitivity. However, the VGG-16 model did perform slightly better than our best lightweight model in accuracy and specificity. But recall that the VGG-16 model is much more complex than our lightweight model.

Interestingly, contrary to our expectations, our lightweight model that utilized data augmentation was our worst-performing model. Data augmentation is known as a technique that helps machine learning models to perform better and become more robust by generating new data points for the model to learn from. For one reason or another, our model performed worse with data augmentation than without. But, when we take a look at the results from the VGG-16 model utilizing data augmentation, it had the best performance metrics out of all of the models in this project. We speculated that the reason why this may be the case was that a model needed a certain degree of complexity to fully utilize data augmentation. If a model was not complex enough, it will perform worse with data augmentation than without it. However, once a model was complex enough, the model will then benefit from the usage of data augmentation. And according to Elgendi et al. (2021), their findings indicate that the usage of geometrical data augmentation in X-ray images may not be an effective strategy for detecting COVID-19 since their overall results when comparing the performance of several deep learning algorithms without augmentation were consistently higher than with augmentation.

1. **Clinical Perception of Artificial Intelligence**

Another issue to discuss is consumer receptivity to machine learning and artificial intelligence in medicine. Currently, in the healthcare industry, both doctors and patients are often reluctant to implement machine learning into their workflows. Some of the fears when utilizing machine learning and artificial intelligence in healthcare may include a lack of human oversight and the potential machine errors that may lead to mismanagement of the patient’s health. Uniqueness neglect, as described by Longoni et al. (2019), is a concern that AI providers are less able than human providers to account for consumers’ unique characteristics and circumstances, which drives consumer resistance to medical AI. Most people do not understand how AI works. Medical imagery is utilized by doctors to help diagnose the patient and provide treatment decisions. Errors do occur in the interpretation of medical images by both doctors and computers, impacting the patients' lives. But it is not like the patients are limited to just picking one or the other. Both doctors and artificial intelligence can work together to provide better patient outcomes and treatment options. Artificial intelligence is revolutionizing healthcare and is evolving. It is not here to replace doctors but rather to support them. With a better understanding of how artificial intelligence works for the general population, the sentiment for artificial intelligence in medicine will increase and improve patient outreach. In the case of our designed CNN, it was able to achieve high-performance measures despite it being a relatively simple model. With a more complex model, designed for the specific problem at hand, it can have an even better performance than what we made.

1. **Limitations and Recommendations for Future Work**

There were several limitations and biases in this project. The dataset used to train the and test the models were primarily comprised of chest X-ray scans of Europeans. This may introduce biases in the model such that our model may perform worse on predicting on chest X-rays of non-Europeans. Possible future spaces for our project would be to expand our data set to be more inclusive of different backgrounds. Similarly, our model could potentially be segregated into a male focused model and a female focused model, by creating male chest X-ray and female chest X-ray datasets. This could potentially help improve diagnostic outcomes due to key differences between sexes. Another possible future improvement would be to preprocess images with higher precision. By increasing signal in areas where COVID-19 is likely to manifest in the lungs, we can improve our models ability to diagnose COVID-19, and potentially reduce false positive outcomes.

1. **Conclusion**

Our project aimed to create a lightweight Convolutional Neural network that could produce highly accurate COVID-19 and viral pneumonia diagnoses. The results indicated that our model was able to positively diagnose with ~93% accuracy with the original dataset. In the presence of data augmentation, our model struggled compared to the VGG-16 model, and was only ~78% accurate. Although the comparison with the VGG-16 model might not be the most representative of modern diagnostic AI, our project demonstrates how lightweight models can be high-performance tools in diagnosing COVID-19 from chest X-rays. Much work remains to be done in the field of AI diagnostics, but our paper illustrates that stacking complexity does not necessarily increase model accuracy, but may positively contribute to model robustness.

1. **References**

Abbas, A., Abdelsamea, M. M., & Gaber, M. M. (2020). Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network. *Applied Intelligence*, *51*. https://doi.org/10.1007/s10489-020-01829-7

Alyasseri, Z. A. A., Al‐Betar, M. A., Doush, I. A., Awadallah, M. A., Abasi, A. K., Makhadmeh, S. N., Alomari, O. A., Abdulkareem, K. H., Adam, A., Damasevicius, R., Mohammed, M. A., & Zitar, R. A. (2021). Review on COVID ‐19 diagnosis models based on machine learning and deep learning approaches. *Expert Systems*. https://doi.org/10.1111/exsy.12759

Christie, A., Brooks, J. T., Hicks, L. A., Sauber-Schatz, E. K., Yoder, J. S., & Honein, M. A. (2021). Guidance for Implementing COVID-19 Prevention Strategies in the Context of Varying Community Transmission Levels and Vaccination Coverage. *MMWR. Morbidity and Mortality Weekly Report*, *70*(30). https://doi.org/10.15585/mmwr.mm7030e2

Chowdhury, M. E. H., Rahman, T., Khandakar, A., Mazhar, R., Kadir, M. A., Mahbub, Z. B., Islam, K. R., Khan, M. S., Iqbal, A., Emadi, N. A., Reaz, M. B. I., & Islam, M. T. (2020). Can AI Help in Screening Viral and COVID-19 Pneumonia? *IEEE Access*, *8*, 132665–132676. https://doi.org/10.1109/access.2020.3010287

Cox, D., & Dean, T. (2014). Neural Networks and Neuroscience-Inspired Computer Vision. *Current Biology*, *24*(18), R921–R929. https://doi.org/10.1016/j.cub.2014.08.026

Dyrbye, L. N., West, C. P., Satele, D., Boone, S., Tan, L., Sloan, J., & Shanafelt, T. D. (2014). Burnout Among U.S. Medical Students, Residents, and Early Career Physicians Relative to the General U.S. Population. *Academic Medicine*, *89*(3), 443–451. https://doi.org/10.1097/acm.0000000000000134

Elgendi, M., Nasir, M. U., Tang, Q., Smith, D., Grenier, J.-P., Batte, C., Spieler, B., Leslie, W. D., Menon, C., Fletcher, R. R., Howard, N., Ward, R., Parker, W., & Nicolaou, S. (2021). The Effectiveness of Image Augmentation in Deep Learning Networks for Detecting COVID-19: A Geometric Transformation Perspective. *Frontiers in Medicine*, *8*. https://doi.org/10.3389/fmed.2021.629134

Fatima, M., & Pasha, M. (2017). Survey of Machine Learning Algorithms for Disease Diagnostic. *Journal of Intelligent Learning Systems and Applications*, *09*(01), 1–16. https://doi.org/10.4236/jilsa.2017.91001

Food and Drug Administration. (2022). At-Home COVID-19 Antigen Tests-Take Steps to Reduce Your Risk of False Negative Results: FDA Safety Communication. *FDA*. https://www.fda.gov/medical-devices/safety-communications/home-covid-19-antigen-tests-take-steps-reduce-your-risk-false-negative-results-fda-safety

Goel, T., Murugan, R., Mirjalili, S., & Chakrabartty, D. K. (2020). OptCoNet: an optimized convolutional neural network for an automatic diagnosis of COVID-19. *Applied Intelligence*, *51*. https://doi.org/10.1007/s10489-020-01904-z

Li, H., Yue, X., Wang, Z., Wang, W., Tomiyama, H., & Meng, L. (2021). A survey of Convolutional Neural Networks —From software to hardware and the applications in measurement. *Measurement: Sensors*, *18*, 100080. https://doi.org/10.1016/j.measen.2021.100080

Longoni, C., Bonezzi, A., & Morewedge, C. K. (2019). Resistance to Medical Artificial Intelligence. *Journal of Consumer Research*, *46*(4), 629–650. https://doi.org/10.1093/jcr/ucz013

Jia, G., Lam, H.-K., & Xu, Y. (2021). Classification of COVID-19 chest X-Ray and CT images using a type of dynamic CNN modification method. *Computers in Biology and Medicine*, *134*(134), 104425. https://doi.org/10.1016/j.compbiomed.2021.104425

Narin, A., Kaya, C., & Pamuk, Z. (2021). Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks. *Pattern Analysis and Applications*, *24*. https://doi.org/10.1007/s10044-021-00984-y

Prechelt, L. (1998). Early Stopping - But When? *Lecture Notes in Computer Science*, 55–69. https://doi.org/10.1007/3-540-49430-8\_3

Rodriguez-Ruiz, A., Lång, K., Gubern-Merida, A., Teuwen, J., Broeders, M., Gennaro, G., Clauser, P., Helbich, T. H., Chevalier, M., Mertelmeier, T., Wallis, M. G., Andersson, I., Zackrisson, S., Sechopoulos, I., & Mann, R. M. (2019). Can we reduce the workload of mammographic screening by automatic identification of normal exams with artificial intelligence? A feasibility study. *European Radiology*, *29*(9), 4825–4832. https://doi.org/10.1007/s00330-019-06186-9

Sitaula, C., & Hossain, M. B. (2020). Attention-based VGG-16 model for COVID-19 chest X-ray image classification. *Applied Intelligence*, *51*(5). https://doi.org/10.1007/s10489-020-02055-x

World Health Organization. (2020). *Novel Coronavirus (2019-nCoV) situation reports*. Www.who.int. https://www.who.int/emergencies/diseases/novel-coronavirus-2019/situation-reports

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, *9*(4), 611–629. https://doi.org/10.1007/s13244-018-0639-9

Zhu, N., Zhang, D., Wang, W., Li, X., Yang, B., Song, J., Zhao, X., Huang, B., Shi, W., Lu, R., Niu, P., Zhan, F., Ma, X., Wang, D., Xu, W., Wu, G., Gao, G. F., & Tan, W. (2020). A Novel Coronavirus from Patients with Pneumonia in China, 2019. *New England Journal of Medicine*, *382*(8). https://doi.org/10.1056/nejmoa2001017