CS4347 Final Project Report Determining the Presence of Pneumonia in Chest X-Rays

Angela Phillips & David Elias
amp395 die9
Project code: GitHub

ACM Reference Format:
Angela Phillips & David Elias, *amp395 die9*, Project code: GitHub . 2020. CS4347 Final Project Report Determining the Presence of Pneumonia in Chest X-Rays. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

For our project, we decided to use computer vision machine learning to examine chest x-rays to determine whether or not the subject of the x-ray has pneumonia. There are two types of pneumonia that will be examined: bacterial and viral. The program will analyze x-rays of chests that have either bacterial or viral pneumonia, or x-rays that are clear, and will determine if the lungs have pneumonia present. It's important to note that the program doesn't distinguish the two types of pneumonia that we'll be viewing from each other; it only determines whether or not pneumonia is present.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1.1 Summary of the Problem

Identification of pneumonia is difficult for a number of reasons. For one, diagnosing based on a blood test can yield incorrect results, as the white blood cell count might be a marker for infection, or might be a marker for something else. Diagnosing based on symptoms can also lead to error, as pneumonia shares symptoms with the common cold and the flu. We want to develop a diagnostic tool that uses machine learning to help recognize and identify the presence and type of pneumonia in chest x-rays with a high accuracy rate.

1.2 Previous Works

Using machine learning for the purposes of medical diagnosis has been done before. In the study "Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning", Daniel Kermany and his team apply the transfer learning framework they created to diagnosing pediatric pneumonia. In addition to diagnosing whether the patient has pneumonia present in their lungs, they also attempt to diagnose the type of pneumonia the patient has.

Our work differs from previous works because we're setting baseline hyperparameters and changing our models (VGG-16, VGG-19, ResNet18, ResNet50, ResNet152, VGG-16 with batch normalization, and VGG-19 with batch normalization) to see how it affects the accuracy of our program, rather than using just one model.

1.2.1 Methods. Kermany and his team used a Convolutional Neural Network with their own model. They used a total of 5,232 chest X-ray images (the same images that we used in our dataset) and ran for 100 epochs, where the training "was stopped due to the absence of further improvement in both loss and accuracy."1

1.2.2 Results. Kermany's team achieved an accuracy of 92.8%, and their accuracy for distinguishing between bacterial and viral pneumonia had an accuracy of 90.7%.

2 PROBLEM DESCRIPTION

We want to apply a machine learning algorithm to the identification of certain features in chest X-rays, to determine whether or not pneumonia is present in the patient.

2.1 Application

We used a Convolutional Neural Network, or CNN, with different models. We kept the hyperparameters of the program the same for all runs, but would switch out our model each time. The models we used were VGG-16, VGG-19, ResNet18, ResNet50, and ResNet152, as well as VGG-16 with batch normalization and VGG-19 with batch normalization. We used these models because they're popular and effective for our use and comparison, and they're pre-trained, which is more efficient for us as well.

The benefit to using pretrained models is that we can now compare multiple against each other, rather than spending time building our own, which would be very difficult and time-consuming to do. In the time it takes to build one model of our own, we can run our program with seven models for 10 epochs each and compare our results and reach our conclusion.

2.2 Features

While there are many different types of pneumonia, our dataset only contained photos of viral and bacterial pneumonia. While our program doesn't distinguish whether an X-ray displays viral or bacterial pneumonia, it still needs to identify features of both to determine whether either type of pneumonia is present in the patient.

2.2.1 Viral. Hyun Jung Koo and his team's study of Radiographic and CT Features of Viral Pneumonia was essential for us to understand how viral pneumonia is identified and diagonsed. In this study, Koo's teamm identifies areas of inflammation and Ground-Glass Opacification (GGO) in the X-ray. In layman's terms, GGO is spotting or fogging found in the X-ray of the lung. Spotting and fogging are usually the most apparent features that can be seen in a pneumonia patient's X-rays.

In the image below, we can observe long veins of white coloration along the black arrows, as well as the typical spotting and fogging. This X-ray displays only a light to mild case of pneumonia, due to the overall low level of GGO.

2.2.2 Bacterial. Bacterial pneumonia is very similar to viral pneumonia; they can both be diagnosed using X-ray imaging. However, the features that identify bacterial pneumonia in an X-ray are different from viral pneumonia. Typically, cases of bacterial pneumonia exhibit Focal Lobar Consolidation (FLC). FLCs are essentially unsegmented, homogenous consolidations of fogging and spotting.

2.3 Challenges

The first and most impactful challenge we faced during this project was our dataset. While the dataset is extremely extensive and detailed, there are some issues with it.

Firstly, the number of X-rays that exhibit pneumonia greatly outnumber the number of normal X-rays, creating an imbalanced dataset. This imbalance can greatly influence the program during the testing phase.

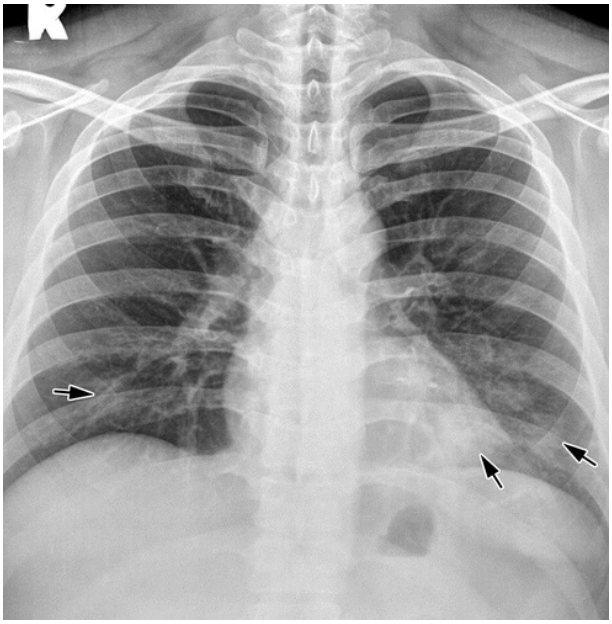
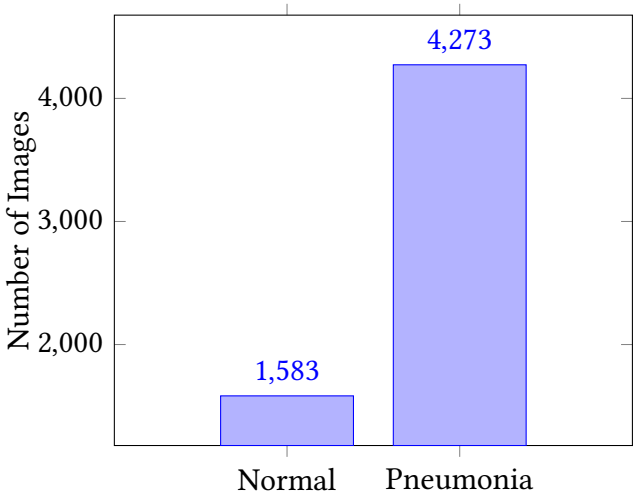


Figure 1: "Pneumonia due to MERS coronavirus in a 27-year-old man who presented with a cough and sputum. (a) Initial chest radiograph shows increased areas of ill-defined nodular opacity (arrows) in both lower lung zones, especially in the left retrocardiac area. (b-d) Axial CT images (3-mm section thickness) obtained on the same day at the level of the right inferior pulmonary vein (b) and the junction of the right atrium and inferior vena cava (c) and a coronal reconstruction image at the vertebral body level (d) show multifocal patchy and nodular consolidation with GGO (arrows) in both lower lobes."2



Another issue we faced with our dataset was the exceedingly small validation set. Of the nearly 6,000

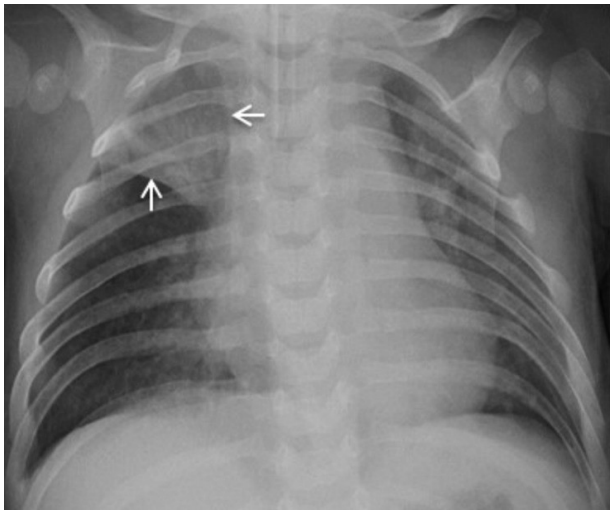


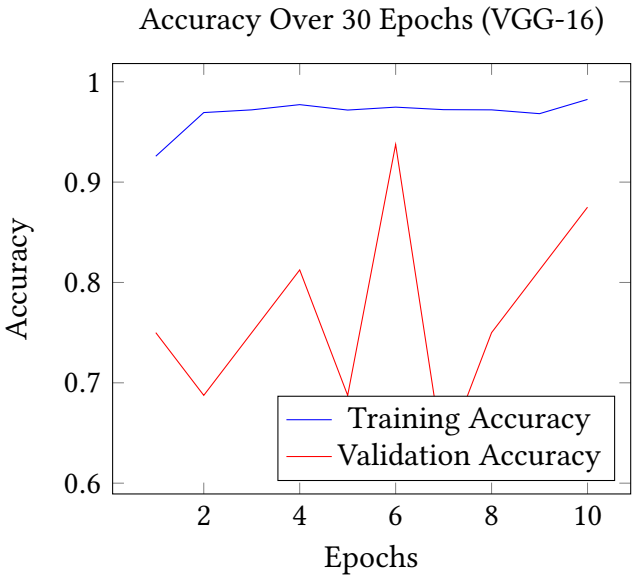
Figure 2: "Bacterial pneumonia typically exhibits a Focal Lobar Consolidation," which is located in the upper right lobe of this figure, where the white arrows indicate.1

images used in this dataset, only 16 were dedicated for validation. This small set limited the models' learning and created exceptionally skewed validation data. We suspect that given an overall more balanced dataset with a sufficiently large validation set, the models would perform at a higher level.

The last notable challenge we faced during our training and testing was the hardware available to us. All models were run on consumer grade hardware: specifically an RTX 2060 GPU and a Ryzen 7 2700X CPU. Originally, we had planned to run these models over 100 epochs, or until the training and validation accuracy were no longer exhibiting upward trends. However, processing nearly 6,000 images over just 40 epochs took nearly half a day. Realizing that we wouldn't be able to test all of our models in a timely manner, we adjusted and settled for a relatively low epoch of 10. This would allow us to test all our models with enough training to fairly compare their accuracies, although it might make reading where the data was going more difficult.

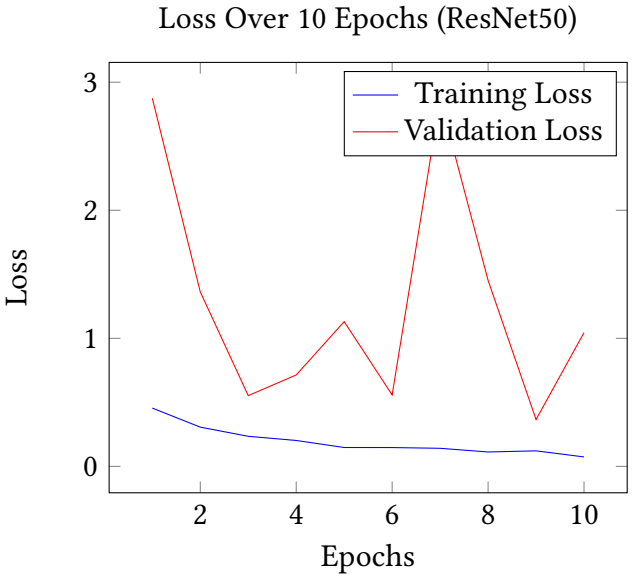
Despite these challenges, we were able to successfully run our program and reach results.

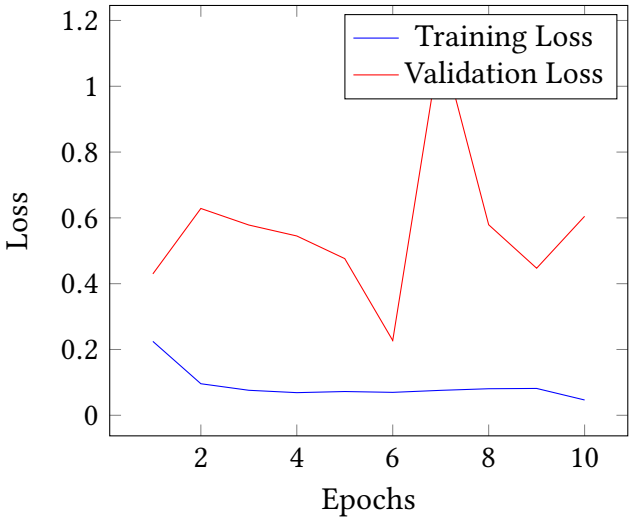
Type of X-rays in Dataset	Normal	Pneumonia
Test	234	390
Train	1341	3875
Validation	8	8

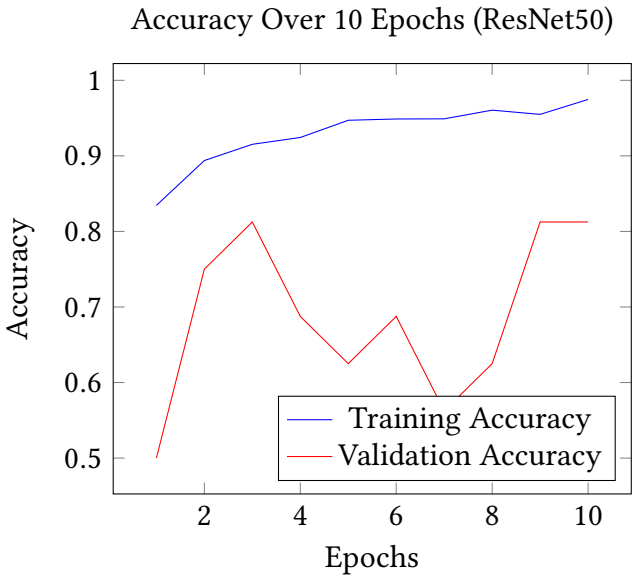


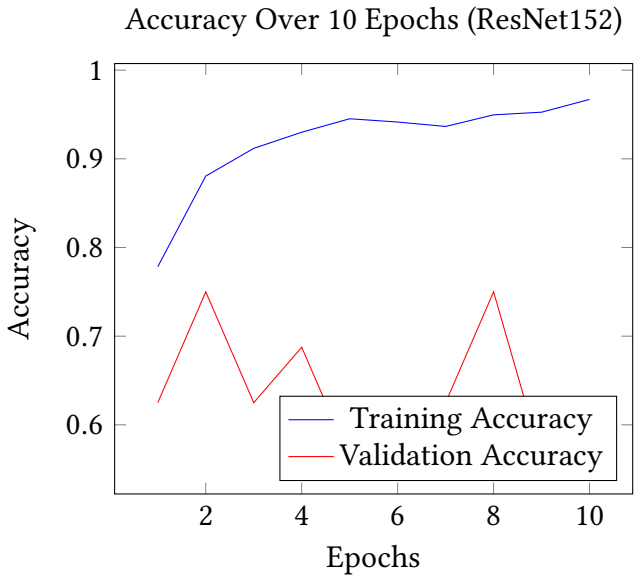
3 RESULTS

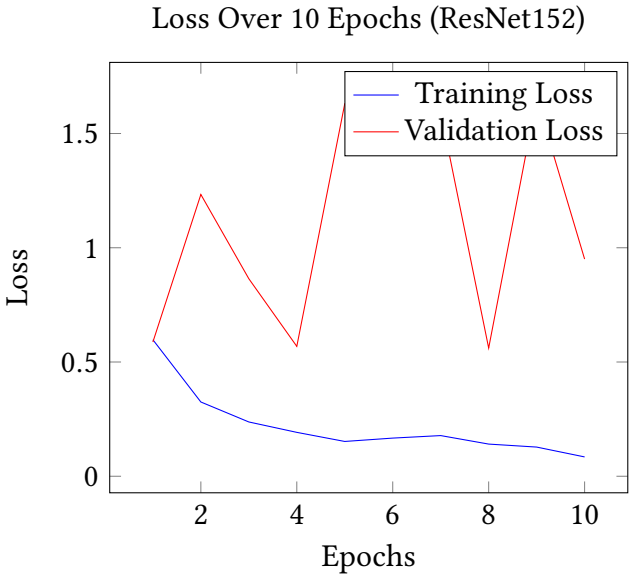
Below are the results of testing the dataset using various popular pretrained CNN models. Each test was run using the same hyperparameters, except for VGG-19 with batch normalization and VGG-16 with batch normalization. Due to the high memory cost necessary for these two models, the batch size was decreased from 16 to 4. All tests were run over 10 epochs using cross entropy loss, an SGD optimizer, and a learning rate of 0.001.

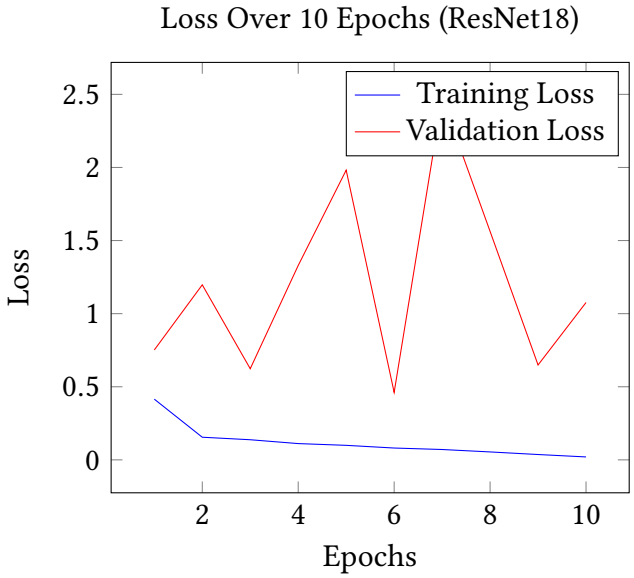


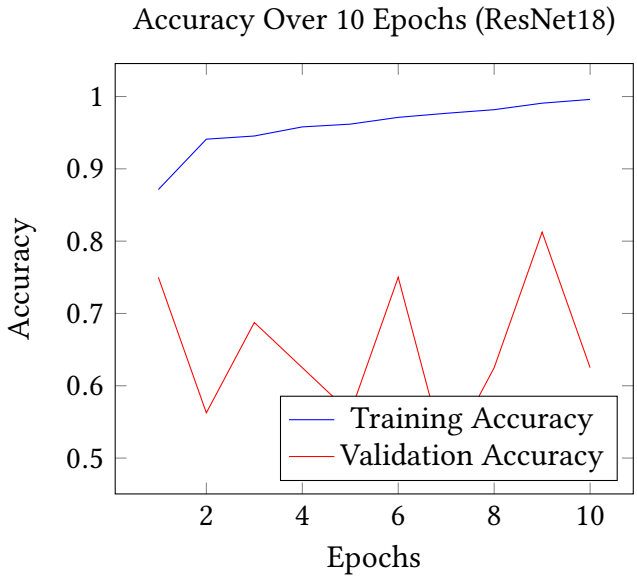


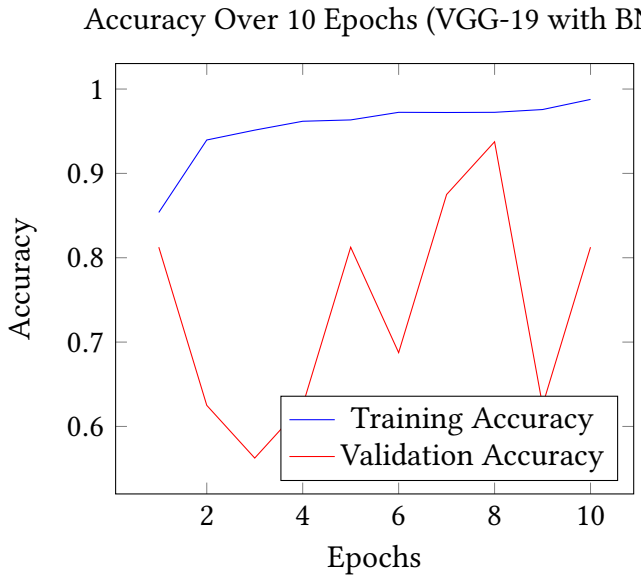

Accuracy Over 10 Epochs (ResNet50)
Accuracy
Epochs
Training Accuracy
Validation Accuracy

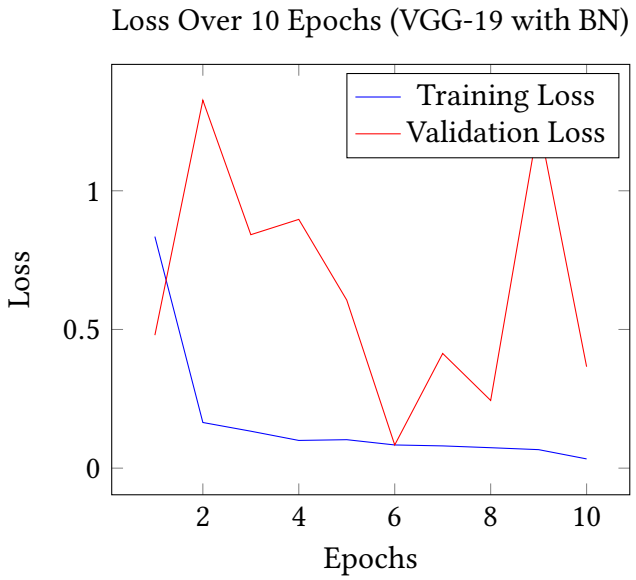

Accuracy Over 10 Epochs (ResNet152)
Accuracy
Epochs
Training Accuracy
Validation Accuracy

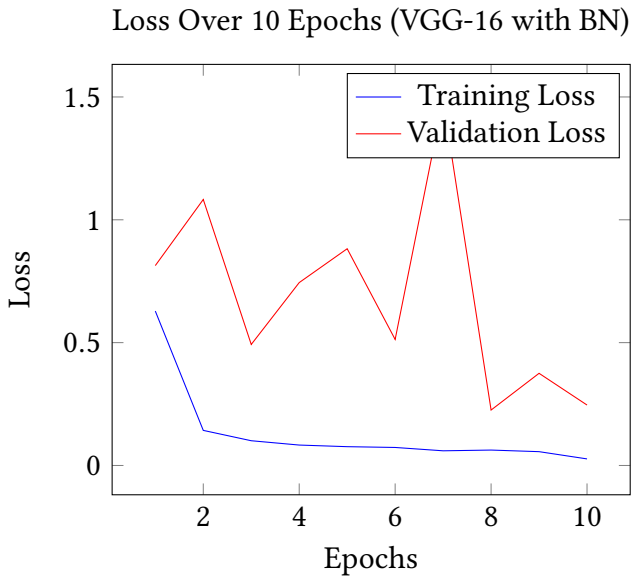

Loss Over 10 Epochs (ResNet152)
Loss
Epochs
Training Loss
Validation Loss

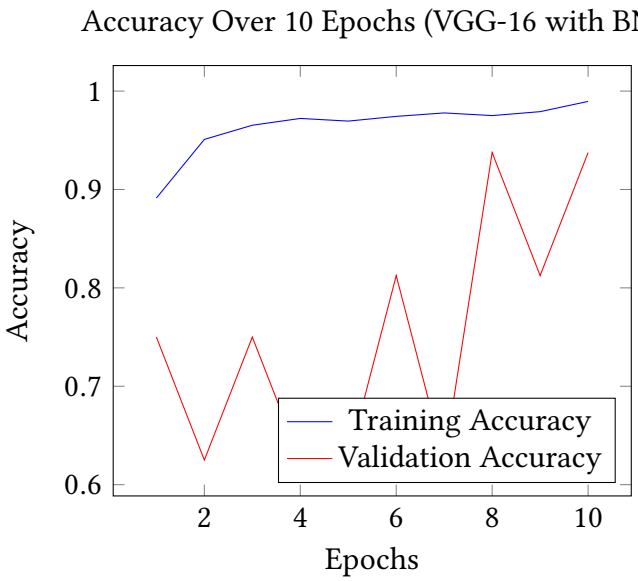

Loss Over 10 Epochs (ResNet18)
Loss
Epochs
Training Loss
Validation Loss

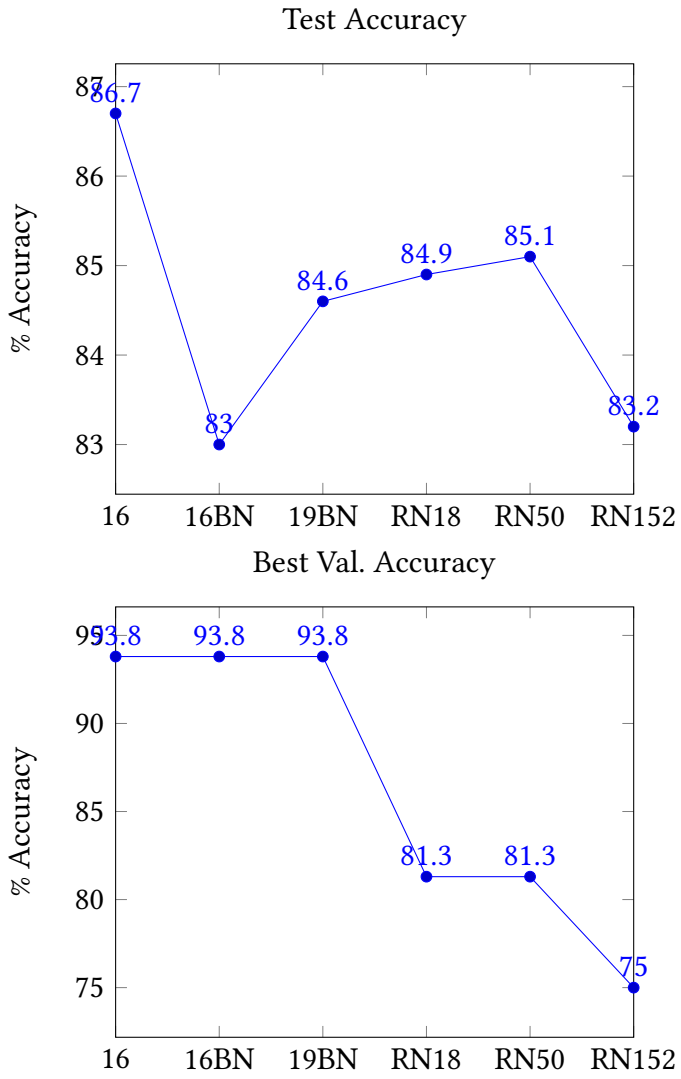

Accuracy Over 10 Epochs (ResNet18)
Accuracy
Epochs
Training Accuracy
Validation Accuracy


Accuracy Over 10 Epochs (VGG-19 with BN)
Accuracy
Epochs
Training Accuracy
Validation Accuracy


Loss Over 10 Epochs (VGG-19 with BN)
Loss
Epochs
Training Loss
Validation Loss


Loss Over 10 Epochs (VGG-16 with BN)
Loss
Epochs
Training Loss
Validation Loss





4 CONCLUSION

Within the scope of our program and throughout all of our runs, we found that the VGG-16 model had the highest testing accuracy over our set span of 10 epochs.

However, if we were able to run the program for as many epochs as we wanted, it's very likely that we would be getting very different results. Because we were limited by our epochs, our information from the results were also somewhat limited.

In addition to this, there was an imbalance in our dataset that very likely affected our results. Because our validation set was so small (only 16

images total), our training efficiency was essentially bottlenecked. The effect that the validation set size had on our data can be observed in our turbulent validation graphs in our result section.

The conclusion that our team reached was actually somewhat inconclusive, because we didn't have the necessary tools to reach something more conclusive. Although we can report that the VGG-16 model had the highest testing accuracy under the conditions we ran, running under better and more accurate conditions would very likely affect this outcome and give us a more precise result, in terms of which model worked better.

5 FUTURE WORK

There are many ways that we can improve and potentially branch off of this project in the future.

To better the results of our current work, we could run our program on more professional hardware so we can increase the number of epochs that we can use to a higher number. Even 30 epochs would make our data more accurate, although if we had the time and hardware to achieve 100 epochs for each run, that would be optimal.

Another way to optimize our current work would be to fix the imbalance in our dataset. We could do so by adding more images to our validation set, and bring our total of 16 current images in our validation set to 100 or more, which would fix the skew in our results.

We could also further optimize for accuracy, if we had more time. If we had the time, we could adjust our number of epochs, our learning rater, or use a different optimizer to find the best combination for use on each model and algorithm that we run.

For future work that branches off of what we currently have, we could use more than just one algorithm. Instead of just a CNN, we could also test algorithms like Logistic Regression and a Feed Forward Neural Network, among others. By doing so, we could also then determine which algorithm, not just model, is the best for use with our dataset, and which has the highest accuracy.

We could also potentially create our own model to use, rather than just depending on pretrained models. By doing so, we could develop a model for our dataset, and create one that will increase our accuracy throughout the run, since the model would be fitted and created for this specific dataset.

There are many potential directions that we can take this project in, but to do so, we'd need more time and better hardware.

6 CONTRIBUTIONS

Both members of our team worked to locate the dataset for use on Kaggle.com and came to a consensus about the implementation, as well as an agreement on using the dataset in question. David Elias managed the coding and decided on using and comparing different pretrained models. He also ran the code and recorded the results of each run under each model used.

Angela Phillips handled the documentation and compiled and graphed the final results into a LeTeX report form, although both members of our team brought the information they had to create the final report.

Decisions made for this project were done diplomatically, and the team worked in tandem on any issues not explicitly mentioned in the contributions section. The workload was shared and the group acted equally.

7 REFERENCES

1. Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning: https://www.cell.com 8674(18)30154-5

2. Radiographic and CT Features of Viral Pneumonia: https://pubs.rsna.org/doi/10.1148/rg.2018170048