Covid-Net: A PyTorch Implementation and Extension of POCOVID-Net for Automated Diagnosis of COVID-19 from Ultrasound Images

Ambuj Arora, Nathan Marshak, and Jordan Saethre

December 11, 2020

Introduction

The world is facing an unprecedented crisis due to the **COVID-19** pandemic. Being able to test patients for COVID rapidly is crucial to all containment efforts. However, not all countries and municipalities have the resources to rapidly test all patients, especially when the healthcare system is overwhelmed with patients, e.g., during a significant spike in coronavirus cases.

Therefore, imaging has been proposed as a way to complement existing methods for COVID testing, especially in "low-resource" situations [3].

Ultrasound, in particular, is appealing because ultrasound scanners are cheap, portable, readily available, and do not expose the patient to radiation the way that X-ray or CT imaging does [3]. Therefore, there is a demand for automated classification of ultrasound images for the purpose of diagnosing COVID-19.

The aim of this project is to accurately classify ultrasound lungs images into three categories, namely, COVID-19 infected, pneumonia infected, and healthy lungs.

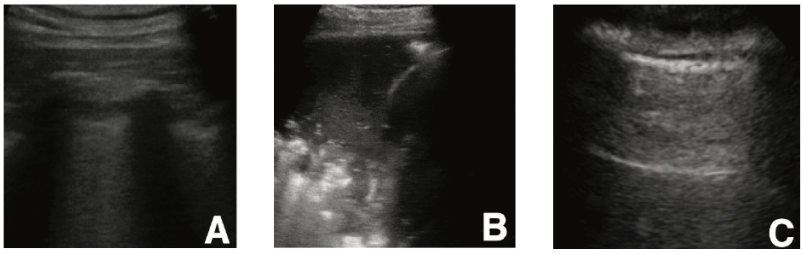

Figure 1: "Example lung ultrasound images. (A): A typical COVID-19 infected lung, showing small subpleural consolidation and pleural irregularities. (B): A pneumonia infected lung, with dynamic air bronchograms surrounded by alveolar consolidation. (C) Healthy lung. The lung is normally aerated with horizontal A-lines." (Image and caption from Born et. al. 2020 [1])

Why would deep learning solve this?

Deep learning has emerged in recent years as one of the best ways to classify images, as long as there is enough training data. Fortunately (or unfortunately, depending on your perspective), the number of training images for COVID is currently growing as the pandemic continues to spread. Therefore, a deep learning-based approach not only has the potential to be effective but would also become more promising in terms of classification performance as time goes on since more training data will become available.

Source of the Dataset

We will be using the dataset provided by the authors of the original POCOVID-Net paper. This dataset is freely available on GitHub [3].

Description of the Dataset

- The dataset consists of images (obtained from videos) which are labeled into the following three categories
  - COVID-19 infected
  - Pneumonia infected
  - Healthy
- The images and videos are recorded by an ultrasound transducer, also called a probe, which is a device that produces sound waves that bounce off body tissues and make echoes.
- The linear probe is a higher frequency probe yielding more superficial images.
- Depending on the type of probe, there are two types of data (*paraphrased from the authors* [3])
  1. **Convex Probe**
     - 162 videos (46x COVID, 49x bacterial pneumonia, 64x healthy, 3x viral pneumonia)
     - 20 videos from the Butterfly dataset (18 COVID, 2 healthy, see below how to use the provided scripts to process the data)
     - 53 images (18x COVID, 20x bacterial pneumonia, 15x healthy)
  2. **Linear Probe**
     - 20 videos (6x COVID, 2x bacterial pneumonia, 9x healthy, 3x viral pneumonia)
     - 6 images (4x COVID, 2x bacterial pneumonia)
     - 45 videos of possible COVID-19 patients collected in Piacenza at the peak of the crisis in Italy; there were not enough PCR tests available, so the label is not clear. For more information and comments by medical experts, see the metadata sheet or metadata csv.

Lung anatomy, ultrasound imaging of the lung, and lung pathology

Ultrasound images of healthy lungs, COVID-infected lungs, and lungs infected by bacterial pneumonia all tend to have distinguishing characteristics that can help clinicians make a diagnosis. One important feature to look for is the *pleural line* - a line in the ultrasound image that corresponds to the two-walled membrane that forms the boundary of the lung.

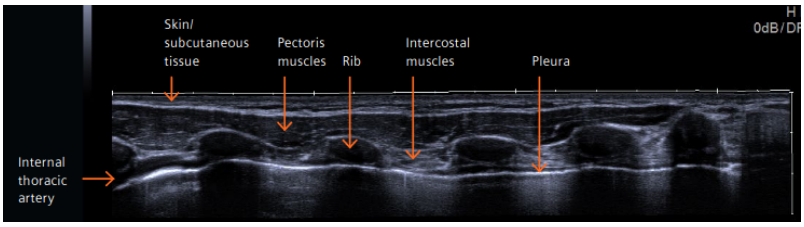

Figure 2: Sample ultrasound image showing lung anatomy [9]. Note the "Pleura", which appears to be important to look at when diagnosing lung disease.

A healthily aerated lung tends to have a regular pleural line, and also A-lines, which are echoes of the pleural line that appear in the ultrasound image [9].

COVID cases, by contrast, are often characterized by an *irregular* pleural line, and s*ubpleural consolidation* (consolidation is when air in the lung being replaced by something else), and this can appear as dark areas on the ultrasound [1]. Pneumonia cases, at least in this dataset, often feature bright white "spots" on the ultrasound, surrounded by darker areas. This is because there is consolidation in the lung that surrounds aerated areas, and these aerated areas are called *bronchograms* [1,9]. Both pneumonia and COVID cases can feature *B-lines*, which are roughly vertical lines that are indicative of fluid buildup. See some of the sample images below for reference.

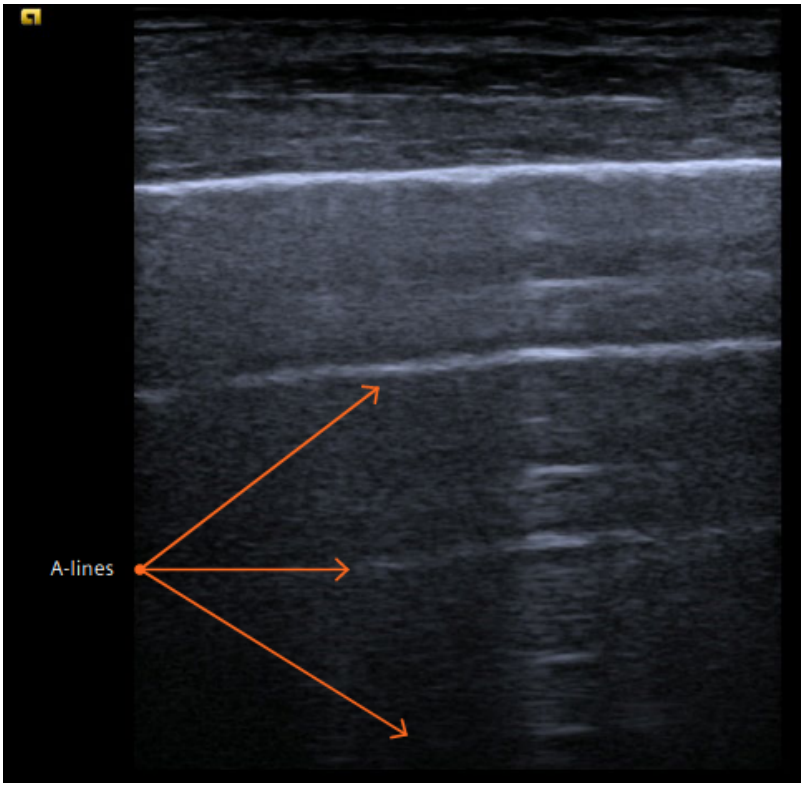

Figure 3: Example of "A-lines", which can be thought of as echoes of the pleura, and are often indicative of a healthy lung. (Image from Clevert's white paper [9].)

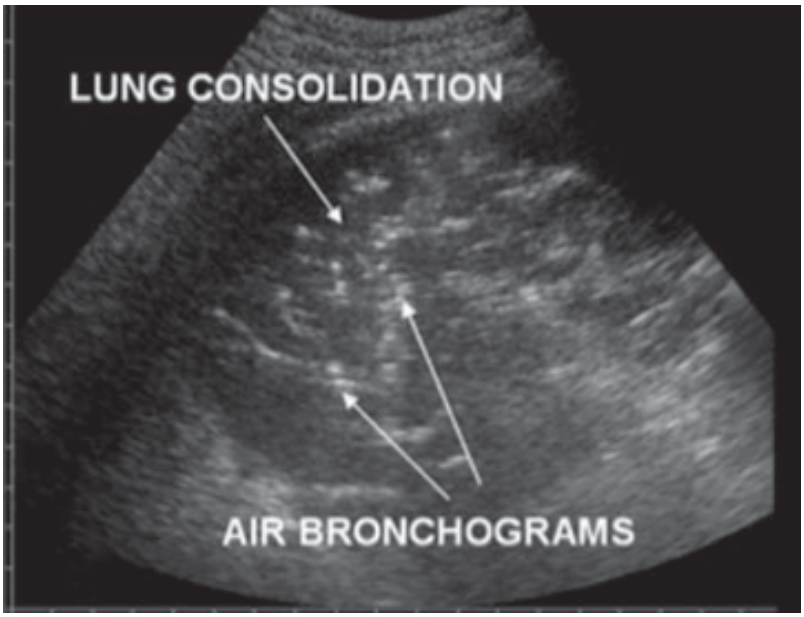

Figure 4: "Ultrasound imaging of [a] patient's lung. It shows an area of consolidation (defined by an area of hypoechoic hepatizated tissue) involving the whole left lower lobe, with minimal pleural effusion, with an estimated volume , 100 ml, and no pneumothorax. Within the consolidation, hyperechoic punctiform areas can be seen and were interpreted as air bronchograms." (Image and caption from Charbit et. al. 2012 [10])

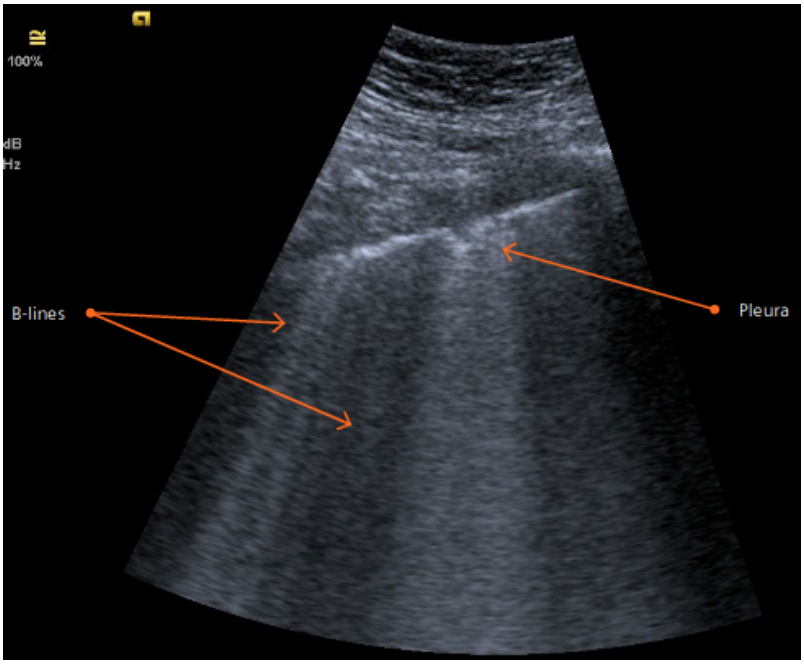

Figure 5: "COVID-19 pneumonia. Follow-up examination in the ICU using a curved transducer. Ultrasound findings including pleural thickening and irregularity. Additionally, consolidations of the lung with B-lines are detected." (Image and caption from Clevert's white paper [9].)

Related Work

This project is a reimplementation and a (small, class-project sized) extension of the work presented in the paper, "POCOVID-Net: automatic detection of COVID-19 from a new lung ultrasound imaging dataset (POCUS)", by Born et al. 2020 [1], which uses VGG-16 as a base architecture. The authors also do some follow-up work in a second paper [2], which explores network visualization, and also the use of variants of VGG as a base architecture, as well as a mobile neural network architecture. The original work was implemented in TensorFlow, while we reimplement this work in PyTorch. Furthermore, while we re-used the data and some data processing scripts from the original authors, we still had to implement our data loading and data augmentation in PyTorch. Finally, the original POCOVID-Net architecture uses VGG-16 as a base architecture. In addition to using VGG-16 as a base architecture, we also tried using ResNet-18 as a base architecture, which is something that the authors had not previously considered. We compare results from both base architectures.

For visualization, the original authors used Class Activations Maps (CAMs) to visualize the activity of the neural network [2]. We also perform visualization, but with different techniques. We use saliency maps (adapted from HW3) and occlusion sensitivity maps (our own implementation), which can be seen as approaches that complement CAMs for understanding which pixels in any given input image are most important for a classification decision made by the neural network. In addition, we create class visualizations - in other words, we use gradient ascent to find images that maximize score for each class. (Our approach had to be modified from the "off the shelf" algorithm from HW3 in order to avoid spurious colors in the results.) Finally, we use t-SNE to visualize the latent space in the final layer of our neural network, and these results seem to give us some insight into how well the neural network performs at "warping" the data into a space that allows classes to be separated with a linear decision rule / the relative difficulty of classifying at least some of the images in the dataset.

System architecture

VGG-16

The original model was built in TensorFlow and uses the convolutional part of VGG-16 with an additional hidden layer of 64 neurons with ReLU activation, dropout of 0.5 and batch normalization. The output layer used softmax activation. Only the weights of the last three layers were optimized during the training. We implemented this exact

architecture in PyTorch and were able to match the number of trainable and non-trainable parameters with the exception of 128 parameters which upon researching turned out to be a difference between Keras and PyTorch in the running means of batchnorm.

Our resulting architecture consisted of 2,392,963 trainable and 12,355,008 non-trainable parameters.

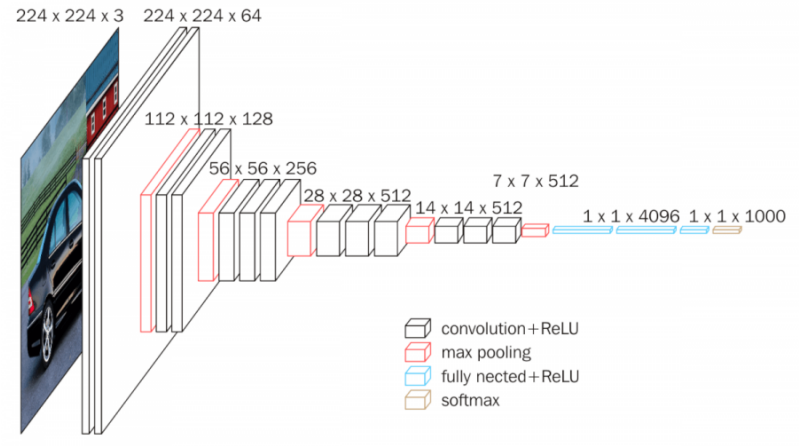

Figure 6: VGG-16 architecture diagram. (Image from [12]).

The VGG-16 model is a well established model that has shown high performance on many image datasets which is the reason for the original authors of [1] in choosing this architecture as a base model. However deep residual models have been shown to perform better than VGG based models in some respects on these same dataset. This led us to wonder if using a deep residual network as a base model for classifying ultrasound lung images might also perform better, [8].

ResNet-18

Our second model was based on the ResNet-18 with the all weights frozen except for the last convolutional layer. An additional fully connected layer of 64 neurons was added with ReLU activation resulting in 2,394,371 trainable and 11,209,539 non-trainable parameters. One thing to note while training this alternate model was that the GPU utilization was significantly lower than that of the VGG based model. This could be due either the ResNet model running more efficiently or perhaps the ResNet Model is less parallelizable. Further investigation is required to determine the implications of this observation.

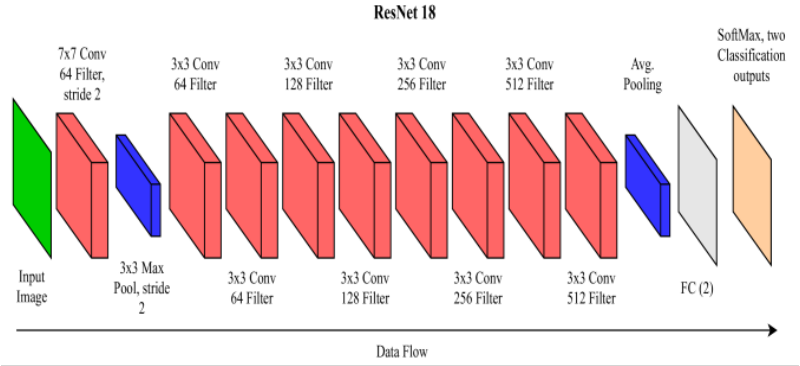

Figure 7: ResNet-18 architecture diagram. (Image from [11].)

Results

Classification results for VGG-16

Shown below are some examples of the classification results of a sample of images in the test set.

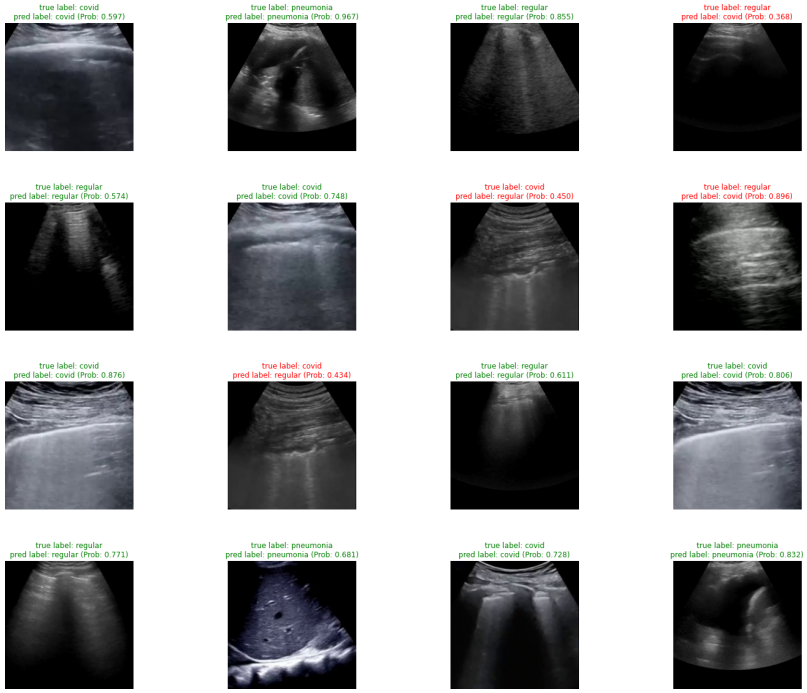

Figure 8: Example classification results of the VGG model

VGG Classification Report	Precision	Recall	F1-Score	Support
Covid	0.83	0.91	0.87	225
Pneumonia	0.94	0.85	0.90	96
Regular	0.88	0.83	0.86	204
			Accuracy	0.87

Classification results for ResNet-18

Shown below are some examples of the classification results of a sample of images in the test set.

ResNet Classification Report	Precision	Recall	F1-Score	Support
Covid	0.83	0.91	0.87	225
Pneumonia	0.94	0.85	0.90	96
Regular	0.88	0.83	0.86	204
			Accuracy	0.87

Comments about the ROC Curves

The Receiver Operating Characteristic (ROC) curve show the ratio of true positives to false positives for a classifier and is typically used for only binary classifiers. To extend to the multi-class situation each class is considered separately as a binary classification problem. The area under each class ROC curve is given and is interpreted to be a better classifier the closer this area is to 1. The micro-averaged ROC curve combines the information from each of the class ROC curves by "considering each element of the label indicator matrix as a binary prediction", [5]. The macro-average ROC curve, combines the information from each class curve by giving equal weight to each class [5].

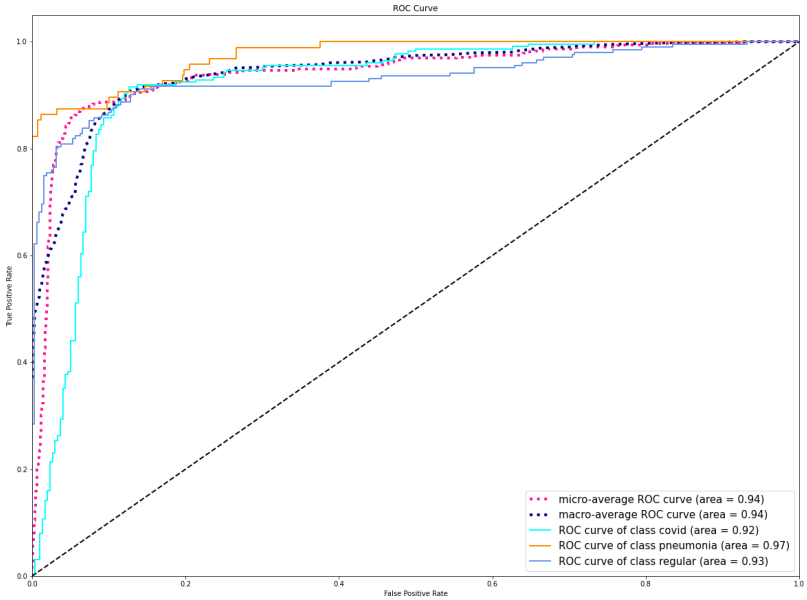

Figure 9: ROC Curves for the VGG model

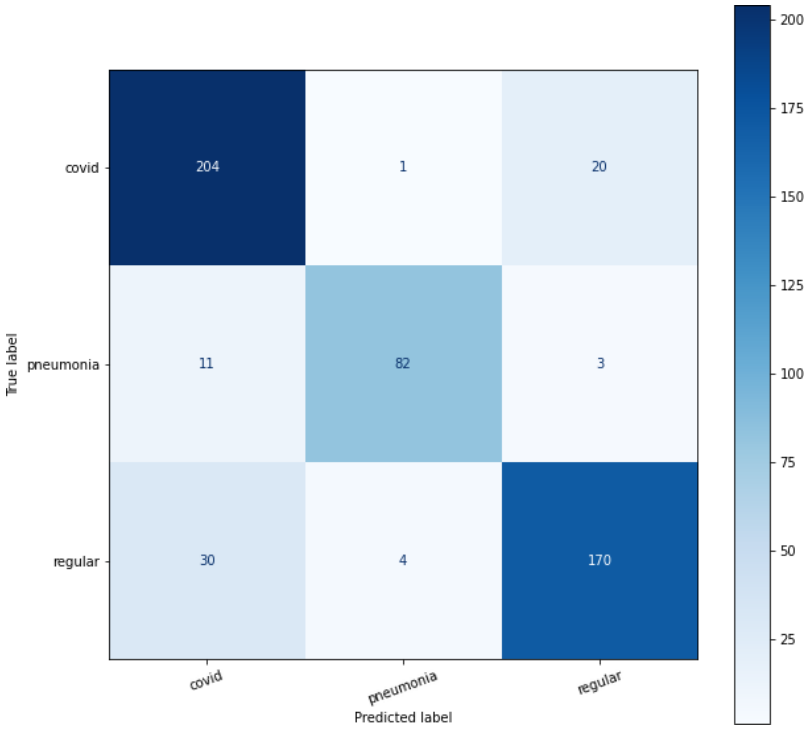

Figure 10: Confusion matrix for the VGG-16 based model

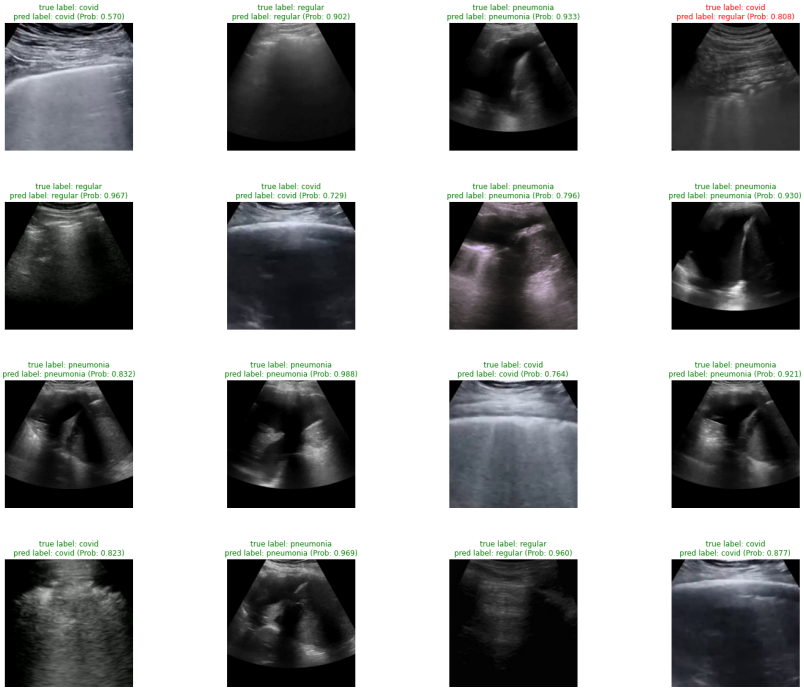

Figure 11: Examples of classification results for the ResNet Model.

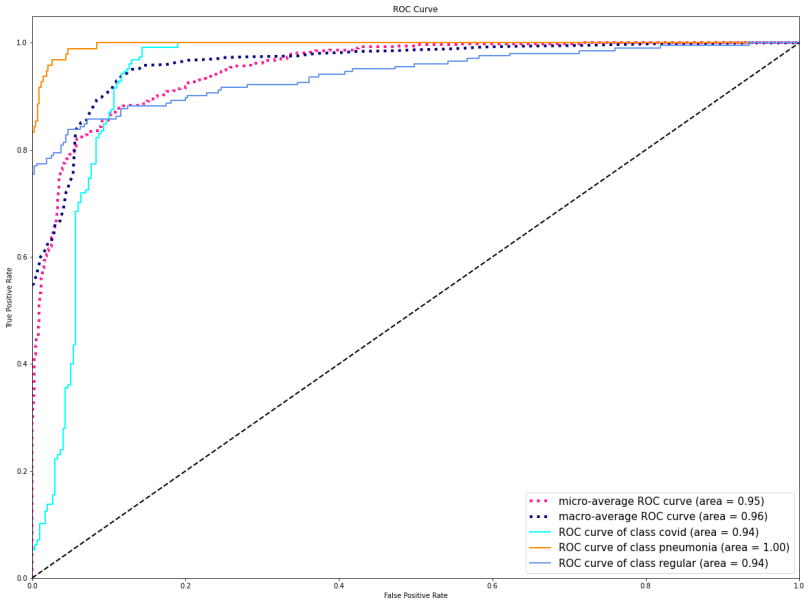

Figure 12: ROC Curves for the ResNet-18 Model

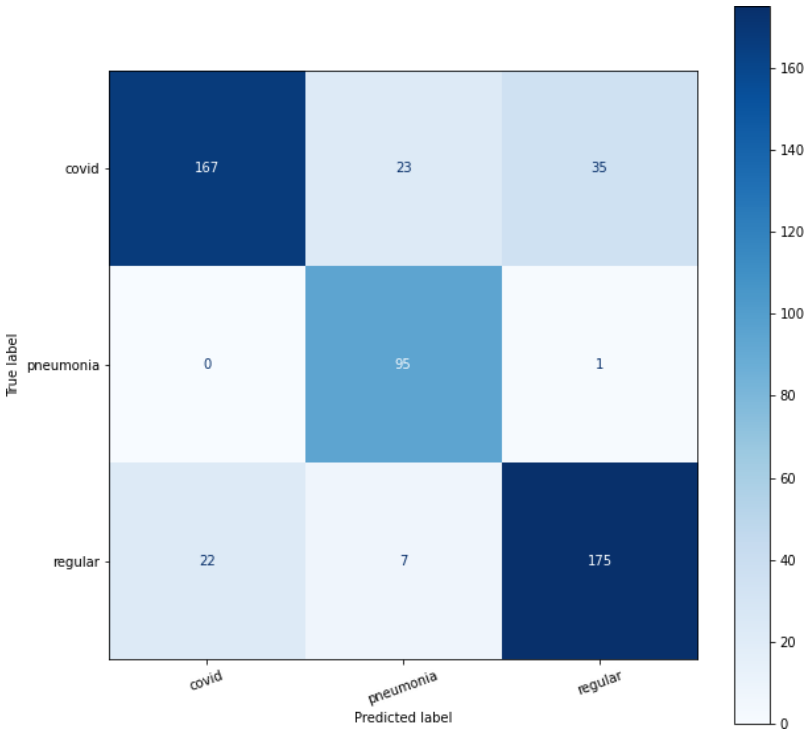

Figure 13: Confusion Matrix for the ResNet-18 Model

Visualization results

t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a nonlinear dimensionality reduction method which can be used to visualize the features a neural network among other things. The layers in a neural network can be thought of as points in space of dimension equal to the number neurons each layer. A probability distribution is constructed on the pairwise distances of these points such that points that are "close" in the high dimensional space are weighted with high probability and points that are "far" are weighted with low probability. The t-SNE method iteratively replicates this probability distribution on successively lower dimensions while optimizing the Kullback-Leibler divergence which is a measure of differences in probability distributions. The result is a 2 (or 3) dimensional representation of the features [4]. The more distinct each class cluster is the better the classifier.

For both the ResNet18 and VGG-16 models the second to last layer, i.e. the one before final classification, was fed into the t-SNE algorithm. For both models the second to last layer was of dimensions (N, 64), where N is the number of test samples. These points were reduced to (N, 2) so that the images can be easily represented in a two dimensional scatter plot.

Comments on t-SNE visualization

For both VGG16 and ResNet, what t-SNE tells us (assuming that t-SNE did its job correctly, and that points in latent space are close together in the 2D plot) is that our neural network does a fairly good job of clustering points from all three of our classes, but there are a nontrivial number of outliers - i.e. points in one class that are quite close to points in another class.

One reason for this may be, for example, that because COVID and pneumonia are both lung diseases, inevitably some pneumonia cases may "look" similar to pneumonia cases. This phenomenon is particularly salient in the t-SNE visualization for ResNet-18 - quite a few pneumonia cases look "similar" to COVID cases. It would be interesting to evaluate the accuracy of human clinicians for these outliers - perhaps it would be difficult for a human to classify these cases, not just a neural network. It would also be interesting, as an academic exercise, to attempt to train binary classifiers (as opposed to the current 3-way classifier) to try to distinguish between healthy and diseased lungs, or to distinguish between COVID and pneumonia. It's possible, for example, that the decision boundary

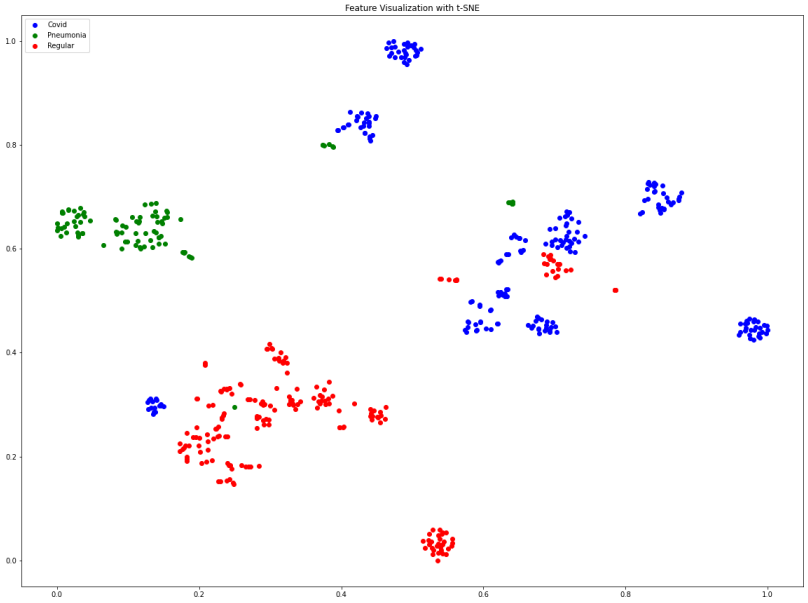

Figure 14: t-SNE plot for VGG-16, showing Covid, Pneumonia, and Regular class labels as colors (see legend in top left).

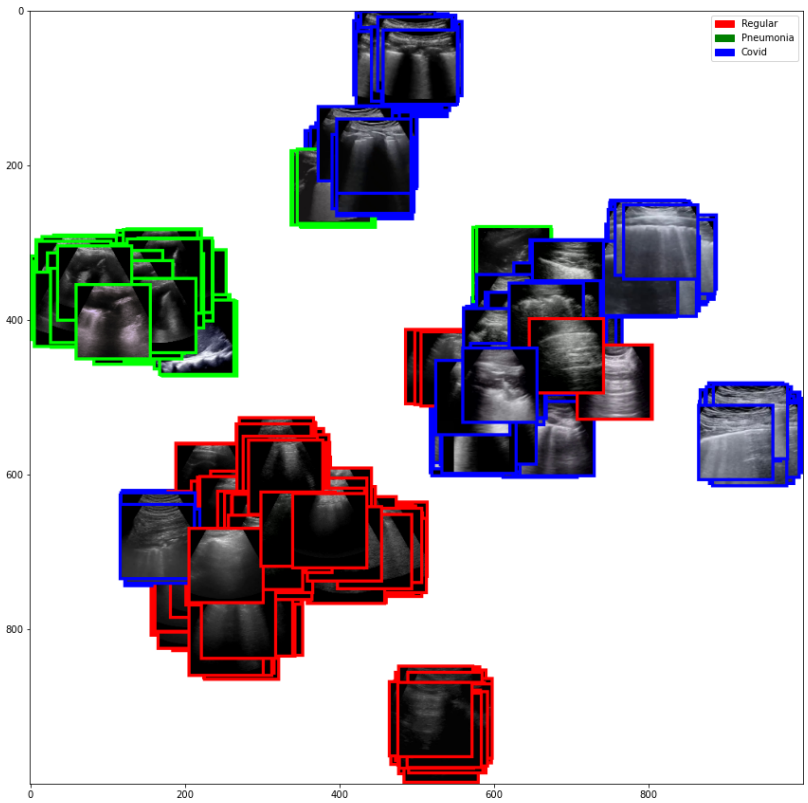

Figure 15: Alternate t-SNE plot for VGG-16, showing classified images from the dataset. The color of the border of each image denotes the class label (see legend in top right).

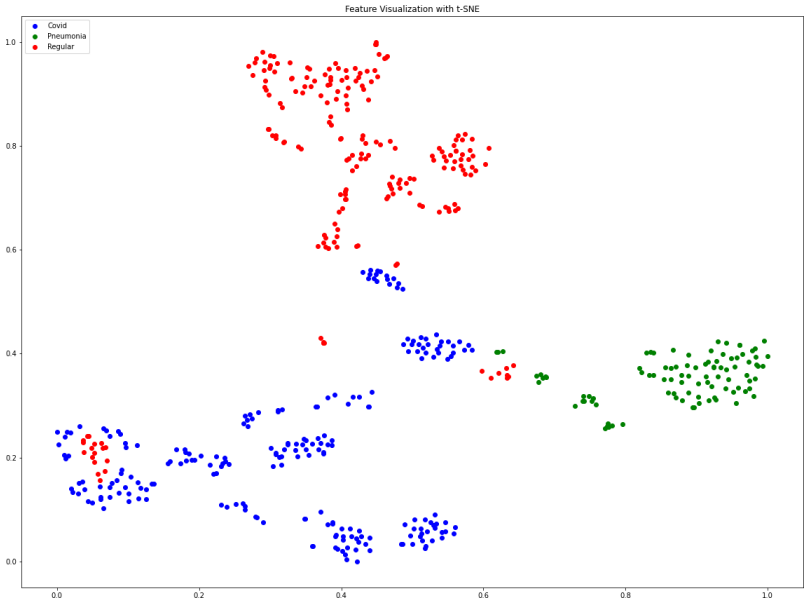

Figure 16: t-SNE plot for ResNet-18, showing Covid, Pneumonia, and Regular class labels as colors (see legend in top left).

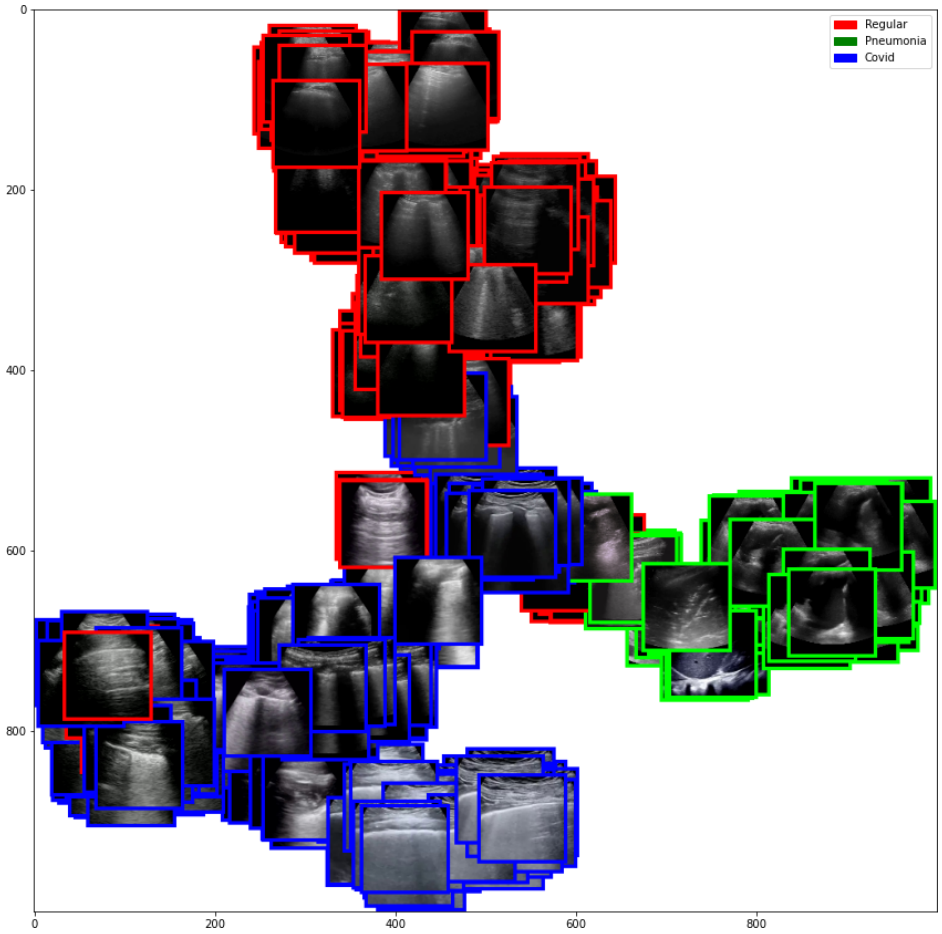

Figure 17: Alternate t-SNE plot for ResNet-18, showing classified images from the dataset. The color of the border of each image denotes the class label (see legend in top right).

between COVID and pneumonia is inherently more complex.

Another interesting phenomenon is that there appear to be at least some clusters within a class. E.g. looking at the t-SNE plots for VGG-16, there appear to be distinct clusters within the Covid class (clusters of "blue" points) and within the pneumonia class (clusters of "red" points). It may be interesting to investigate whether these clusters have any semantic meaning, but we would need assistance from a clinician / medical expert to be able to do so.

Saliency Maps and Occlusion Sensitivity Maps

In the follow-up to the original POCOVID-Net paper [2], the authors use Class Activation Maps (CAMs), to understand which areas of any given input image the network responds to. In order to complement their work, we created saliency maps and occlusion sensitivity maps. The saliency maps, based on HW3 and Simonyan et. al. [6], help us understand which pixels had the highest derivative with respect to the correct class. The occlusion sensitivity maps, based on Zeiler and Fergus [7], tell us how much the probability of the correct class changes when a square neighborhood centered at a particular pixel is occluded (i.e. we cover that part of the image with a square of uniform color). Sample results for a healthy lung, COVID-infected lung, and bacterial pneumonia infected lung are shown below - see the captions for comments.

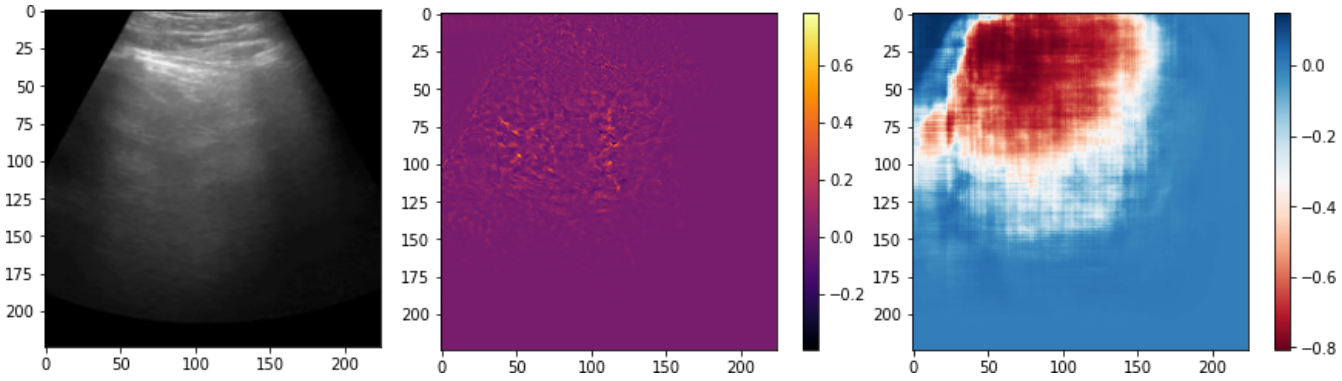

Figure 18: Healthy lung image (left), saliency map (center), and occlusion sensitivity map (right). Both maps are for VGG-16. The saliency map shows that the neural network appears to be responding to the upper left region in general, and there may be "hotspots" corresponding to A-lines, which are echoes of the pleural line, and are indicative of a normally aerated lung. By contrast, the occlusion sensitivity map seems to indicate that the upper left region of the ultrasound image contains important information - probably because that is where the pleura can be seen clearly.

General observations on saliency and occlusion sensitivity maps

1. The occlusion sensitivity maps and the saliency maps produce substantially different results. This likely shows that no single visualization technique gives us all the information that we need to know - instead, multiple, complementary visualization techniques are needed in order to paint a more complete picture of the neural network's behavior.
2. Most of the higher gradient values occur in the upper left of the image, in the lower right the gradient is usually zero. This is despite the fact that the data itself isn't really biased towards the upper right. This is interesting and perhaps warrants further investigation. This phenomenon was not clearly visible in the CAM visualizations that were shown in the authors' original work [2].
3. There are a few examples that seem to show the network picking up on information outside the boundary of images that come from a convex probe. Above, this is visible in the upper left of the COVID and pneumonia images. This may be because the network is trained on images from both linear and convex probes, and images from the former do not feature the black "out of bounds" regions that appear in images from the latter. Furthermore, images from different convex probes may have slightly different "out of bounds" regions. Perhaps this issue could be mitigated if the images are warped to some sort of common coordinate system. Alternatively, linear probe data could be dropped, since there seems to be an abundance of images from convex probes.

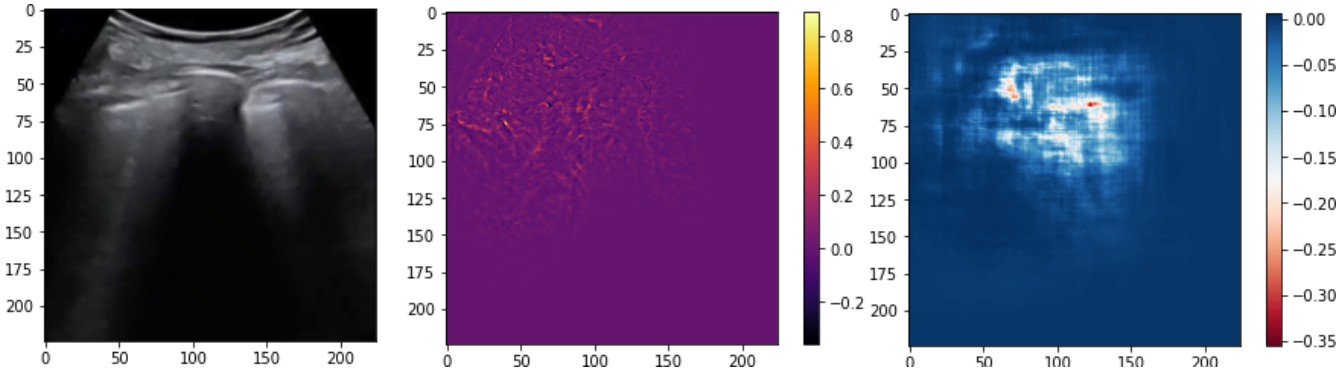

Figure 19: COVID lung image (left), saliency map (center), and occlusion sensitivity map (right). Both maps are from VGG-16. The saliency map indicates that the neural network responds to the upper left region of the image, and there are "hotspots" that are perhaps located in regions where there is pleural irregularity. There is some response in the saliency map outside the boundary of the ultrasound image, in particular in the upper left and top of the image. This indicates that the network may respond to some areas with irrelevant information. The occlusion sensitivity map reveals different areas of importance - in particular, regions in the upper center of the image seem important, with peaks that appear to map to pleural irregularities that are different from those highlighted by the saliency map.

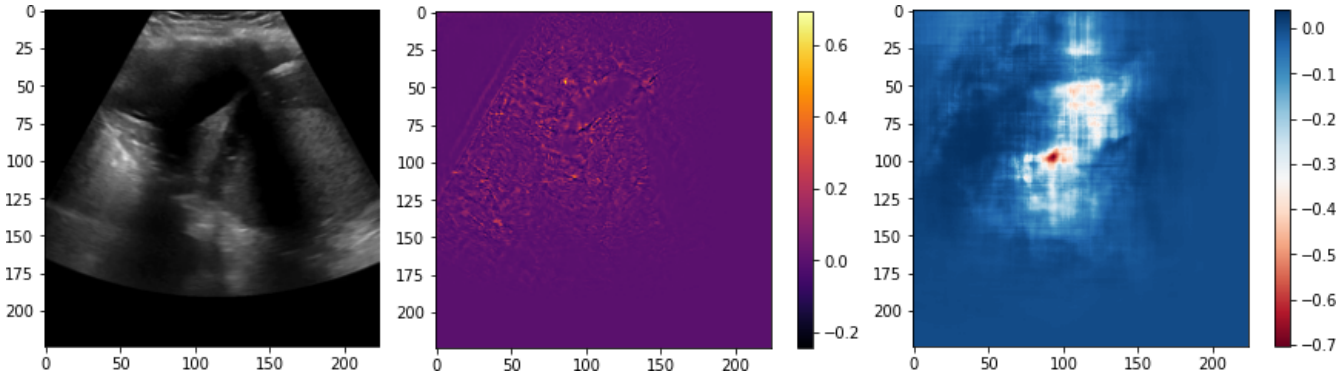

Figure 20: Bacterial pneumonia infected lung (left), saliency map (center), and occlusion sensitivity map (right). Both maps are from VGG-16. The saliency map appears to respond to edges near dark regions - this could be a response to "dark" regions in the lung, which may indicate consolidation of the lung caused by fluid. Alternatively, as can be seen on the left, it could be a response to the edge of the ultrasound image, which means the network may be picking up on data that is not really relevant. There also appear to be peaks in the saliency map, e.g. at around (x=75,y=50) but it is less clear what they are. The occlusion sensitivity map appears to indicate that the region most sensitive to occlusion is somewhere around (x=100,y=90) - with another important region in the neighborhood of (x=110,y=60). Both are dark regions, so these may be indicative of lung consolidation. Also note that in the occlusion sensitivity map, there is some response to areas outside of the actual ultrasound image, particularly in the upper left.

Class visualization

Shown in the figure below, we use gradient ascent to find an image that maximizes activation for a particular class, based on the HW3 code. However, we introduce an additional regularization that **was not present in HW3**, which is to make sure that the starting noise image is *grayscale*, and then to make sure that update applied during **gradient ascent is also grayscale**. (The latter is done by taking the gradient, finding its average across all three channels, then setting all three channels of the "gradient" to be the same as this average. Then we update by the learning rate times this monochrome "gradient".) The reason why we do this is because all (or almost all) of the training data appears to be grayscale, despite the neural network *assuming a 3-channel image* since it is based on an off-the-shelf pretrained architecture for color images. Therefore, color in any "maximal activation image" is meaningless! If we do not use the aforementioned regularization, spurious colors will appear.

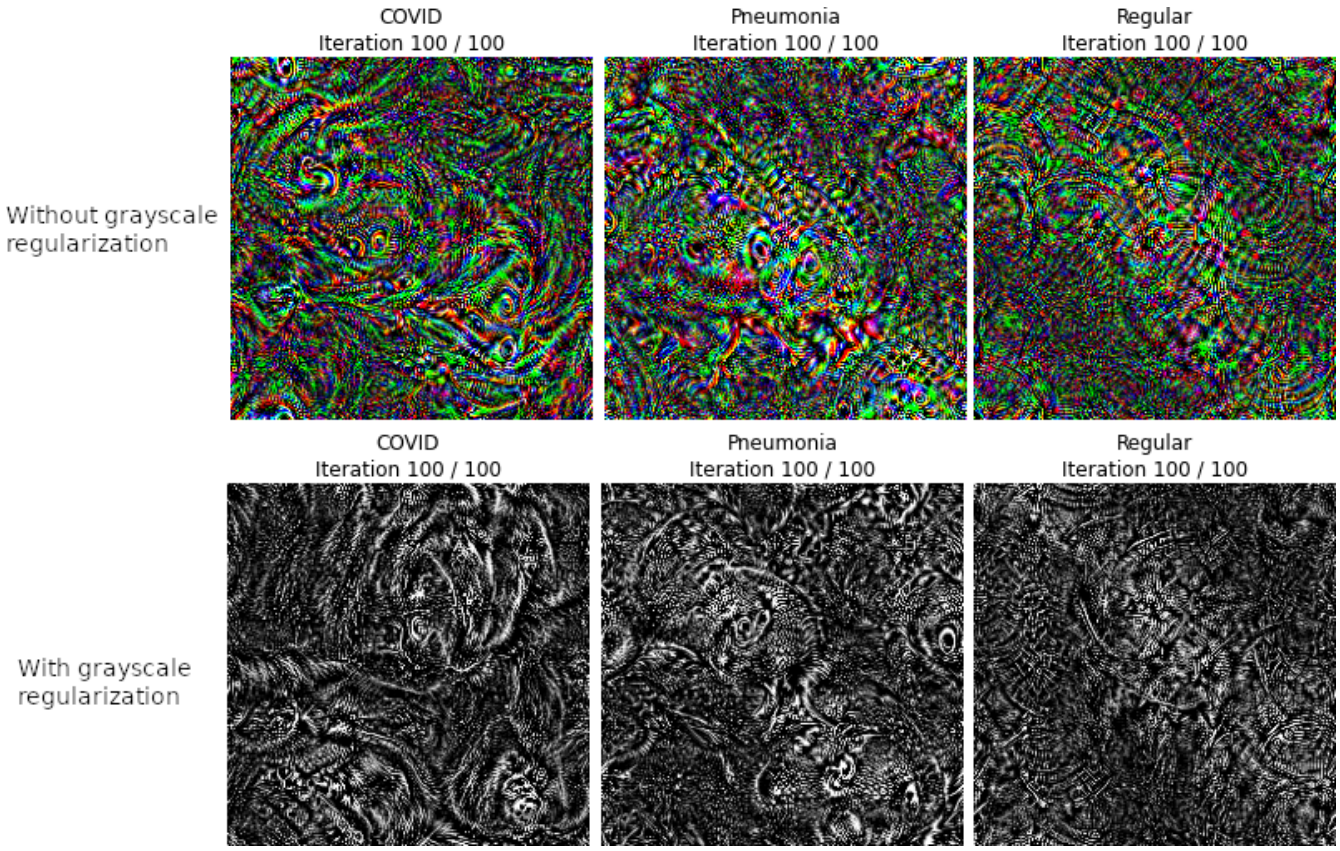

Figure 21: Class images (a.k.a. "maximal activation" images) from VGG-16. Top row: Without regularization used to remove spurious colors. Bottom row: With regularization. These images are easier to interpret than those on the top row. Bottom row observations: The "Regular" image, which denotes a healthy lung, thin, slightly curved lines that occur more often than in the other images. This thin lines likely denote a regular pleural line and its echoes (A-lines). Note that the "COVID" image features thicker lines, likely denoting irregular pleura and possibly B-lines, which are both indicators of COVID infection. Finally, note that the "Pneumonia" image features large, round, white "spots"/"blobs" that are not as prevalent as in the other images. These spots likely correspond to the "bronchograms" that are a tell-tale sign of pneumonia-infection, but appear less prevalent, at least in this dataset, in COVID cases and in healthy lungs.

Further Improvements

Currently the training script for both models does not include a function that automatically cross validates across folds. Each time the training script is run the fold to be used for validation must be specified by the user. An improvement would be to implement a function that would automatically train each model on each fold while keeping track of the labels given to each test sample. This added functionality would then give a final classification based on a majority vote on the label of each class. This would likely improve the overall results of each model.

Another improvement would be to adjust the training script to allow for further training based on new samples as they come available. As samples are collected additional layers could be added to the end of the existing architectures so that only the most recent layer is optimized.

More visualization techniques could be explored. One thing that could be particularly interesting is feature inversion and related methods for exploring the latent space at the end of both the complete neural network, and the end of the pretrained model used as a base. This could perhaps allow us to understand the differences between different base models, and point to a way to find a better base model.

As mentioned before, we also realized that most of the training images appear to be grayscale, yet the neural network assumes 3-channel color input, because it uses a pretrained model (either VGG or ResNet) as a base. (This is why our class visualization technique had to be modified from the off-the-shelf approach.) Perhaps a base model that has been trained on grayscale images specifically could be more effective here, because it means that we can do away with unnecessary model parameters.

Conclusion

- We were able to port the original POCOVID-Net architecture (written in TensorFlow) to PyTorch, and we achieved comparable results using both VGG-16 (the base model used by the authors) and Resnet-18.
- We experimented with our own choice of base model, ResNet-18, and found hyperparameters that could give us a result comparable to what the original authors achieved with VGG-16.
- We were able to complement the authors' CAM visualization with our own visualizations based on saliency maps, occlusion sensitivity maps, a customized class visualization technique, and t-SNE.
- Our visulizations gave us insight into which parts of the image contribute to class score (and this changed very significantly depending on technique), what features the neural network is "looking for" for each class, and how both our VGG-16 and ResNet-18 based architectures "cluster" the data prior to the final linear decision boundaries.

References

1. Born, J., Brändle, G., Cossio, M., Disdier, M., Goulet, J., Roulin, J. and Wiedemann, N., 2020. POCOVID-Net: automatic detection of COVID-19 from a new lung ultrasound imaging dataset (POCUS). arXiv preprint arXiv:2004.12084. https://arxiv.org/abs/2004.12084
2. Born, J., Wiedemann, N., Brändle, G., Buhre, C., Rieck, B. and Borgwardt, K., 2020. Accelerating COVID-19 Differential Diagnosis with Explainable Ultrasound Image Analysis. arXiv preprint arXiv:2009.06116. https://arxiv.org/abs/2009.06116
3. https://github.com/jannisborn/covid19_pocus_ultrasound
4. Vidiyala, Ramya. "What, Why and How of t-SNE." *Medium*, Towards Data Science, 19 May 2020, towardsdatascience.com/what-why-and-how-of-t-sne-1f78d13e224d.
5. Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.
6. Simonyan, K., Vedaldi, A. and Zisserman, A., 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*. https://arxiv.org/abs/1312.6034
7. Zeiler, M.D. and Fergus, R., 2014, September. Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818-833). Springer, Cham. https://cs.nyu.edu/~fergus/papers/zeilerECCV2014.pdf
8. Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. (2015). Deep Residual Learning for Image Recognition.
9. Clevert, Dirk-André. White Paper: Lung Ultrasound in Patients with Coronavirus COVID-19 Disease. *Siemens Healthineers white paper*. https://www.siemens-healthineers.com/en-us/ultrasound/lung-ultrasound-covid-19

10. Charbit, B., Funck-Brentano, C., Benhamou, D. and Weissenburger, J., 2012. Effects of oxytocin on Purkinje fibres. *British journal of anaesthesia*, *108*(6), pp.1039-1041. https://academic.oup.com/bja/article/108/6/1039/311354
11. Ghorakavi, R.S., 2019. TBNet: Pulmonary Tuberculosis Diagnosing System using Deep Neural Networks. arXiv preprint arXiv:1902.08897.
12. https://neurohive.io/en/popular-networks/vgg16/