CS598 Project Report

https://mediaspace.illinois.edu/media/t/1_zmxe0f0c

https://github.com/amtoney524/cs598-team1149-final-project

Vijay Selvaraj vs27@illinois.edu

Susie Kuretski skure2@illinois.edu

Ashley Toney atone3@illinois.edu

Regan Chan ttchan2@illinois.edu

ABSTRACT

We explored deep learning methods described in Deep-COVID [1], COVID-Net [2], and COVID-Net combined with a long short-term memory recurrent network [3] to classify COVID-19 and non-COVID-19 from chest X-ray images. These original studies were published approximately a year ago in early to mid-2020, and they all suggested further analysis was needed due to limited COVID-19 X-ray data available.

Therefore, we utilized a larger dataset of approximately 13,200 chest X-rays, including 3,616 COVID-19 chest X-rays, and evaluated transfer learning performance using ResNet18, ResNet50, SqueezeNet 1.1, DenseNet-121, and InceptionV3, similarly to the Deep-COVID paper. We also compared these pre-trained models to COVID-Net and a slightly modified version of COVID-Net that uses a long short-term memory (LSTM) network.

The code for this project can be found here: **https://github.com/amtoney524/cs598-team1149-final-project**

Keywords

COVID-19, deep neural network, deep learning, transfer learning, X-Ray, medical imaging

1. INTRODUCTION

In 2021, COVID-19 continues to be a global pandemic with 2.7 million deaths and 125 million reported cases [4] since its first public announcement by the Wuhan Municipal Health Commission on December 31, 2019 [5]. Even with worldwide vaccination efforts, COVID-19 detection remains imperative for patient outcomes and public health. Reverse transcriptase polymerase chain reaction (RT-PCR) remains the standard for COVID-19 testing, but results can take hours to days, and it can yield false negatives. In immediate and emergency care settings, it is vital for clinicians to paint a comprehensive clinical picture beyond one lab test. Radiography is one of the fastest, readily available diagnostics to gain insight on lung disease and its progression.

In previous studies for COVID-19 classification through chest X-rays with deep learning, a common limitation was lack of data [1,2,3,6]. Magulo and Nanni [6] used AlexNet in two different experiments involving four different datasets of chest X-rays with COVID-19 images. In both of those experiments, the training and test sets were a combination of all four datasets. From this, they concluded some protocols may be biased since the models learn to predict based upon features of the source dataset instead of the relevant medical information. Their future work suggested incorporating new COVID-19 datasets to create diversity and limit bias.

Some studies have used extensive data augmentation and various other techniques to address the gap [1,7]. Now in 2021, there is more data available, and we hope to evaluate the performance further by continuing the work of Deep-COVID, COVID-Net, and a modified version of COVID-Net containing a LSTM network.

2. LITERATURE REVIEW

For the purpose of this project, we reviewed literature pertaining to deep learning for classification with medical imaging or specifically regarding COVID-19 classification. The main areas of interest were convolutional neural networks, transfer learning techniques, and more novel approaches for the purposes of comparison.

2.1 Convolution Neural Networks

Convolutional neural networks have become the standard for computer vision and image classification thanks to their ability to extract features through convolution and other hidden layers. By convolving, or applying a kernel, CNNs can get a local summary of the neighborhood akin to humans extracting features in areas of an image instead of scrutinizing an image pixel by pixel. A basic CNN has one or many convolutional layers, an activation function (commonly ReLU), pooling layers e.g. maximum or average pooling, and one or many fully connected layers.

2.2 Transfer Learning and Deep-COVID

Transfer learning is the process of taking a pretrained model and using it for a similar task. It is useful when the data is sparse and/or the tasks are very similar. Questions could be raised with using CNN transfer learning for medical imaging since databases, like ImageNet, generally contain more natural images as compared to X-rays or computed tomography scans. However, Tajbakhsh et. al. [8] specifically tested this with medical images in four different classification scenarios. They found that while shallow tuning of pretrained models typically led to poorer results, deeper tuning of pretrained models actually met or could exceed CNNs trained from scratch. In addition, their findings also showed that pretrained models converged faster than CNNs from scratch.

The authors of Deep-COVID [1] used data augmentation techniques and transfer learning with ResNet18, ResNet50, SqueezeNet, and Dense-Net-121. Their pre-trained model was used more as a feature extractor rather than adapting the weights

to the new task of COVID-19 classification. This approach was taken because their number of COVID-19 images was limited. Their COVID-Xray-5k dataset contained 184 COVID-19 images compared to the 5,000 non-COVID images. The 184 COVID-19 images were split with 84 into the training set and 100 into the test set. After data augmentation, the training set grew to 420 images. Deep-COVID achieved 98% sensitivity rate and 90% specificity rate. However, the authors did conclude that further experiments were needed on a larger set of COVID-19 images.

2.3 COVID-Net

Published in May 2020, Wang et. al. shared an open source network designed for COVID-19 detection, called COVID-Net, and COVIDx, an open access dataset with 13,975 chest X-ray images [2]. COVID-Net utilized deep residual learning and created a projection-expansion-projection design. With COVID-Net, they also hoped to gain more transparency with GSInquire, an "explainability method that is a critical aspect of the generative synthesis strategy" since deep learning models can seem like black boxes [2].

With their dataset of 13,975 chest X-ray images, which contained 358 labelled as COVID-19, 8,066 labelled as normal, and 5,538 labelled as non-COVID pneumonia chest X-ray, they achieved an accuracy of 93.3%, sensitivity of 91.0%, and a positive predictive value of 98.9%.

2.4 Combined CNN-LSTM Network

In June 2020, Islam et. al. [3] published a paper which utilized a combined CNN with a long short-term memory network to detect COVID-19 through chest X-rays. The CNN focused upon feature extraction, while the LSTM focused on detection. In a traditional CNN, a general fully connected layer will be used for classification. However, the authors leveraged the RNN memory of a LSTM. Unlike traditional recurrent neural networks, LSTMs address the exploding or vanishing gradient problem by its structure of a cell state where it can retain useful information and forget unnecessary information. They used 4,575 chest X-rays, of which 1,525 were labelled as COVID-19. With only the CNN, they achieved an accuracy of 98.5%, sensitivity of 99.0%, specificity of 98.2%, and F1-score of 97.7%. With the CNN plus LSTM, they achieved slightly better results all around with an accuracy of 99.4%, sensitivity of 99.3%, specificity of 99.2%, AUC of 99.9%, and F1-score of 98.9% in COVID-19 detection.

3. METHOD

With our 13,200 chest X-ray dataset, we did some pre-processing and performed some data augmentation in regards to mirroring, rotating, and increasing the number of COVID-19 images in our training set to 2,880 from 1,440. We then trained multiple models: ResNet18, ResNet50, SqueezeNet 1.1, DenseNet-121, InceptionV3, COVID-Net, and COVID-Net+LSTM network and compared the results.

3.1 Data

Our dataset, named COVID-19 Radiography Database hosted on Kaggle, was compiled by Rahman et. al. [9,34,35]. It contains 10,192 normal, 6,012 non-COVID lung infections, 1,345 viral pneumonia, and 3,616 COVID-19 chest X-ray images. The COVID-19 images are a compilation of images from the Medical Imaging Databank of the Valencia Region (Padchest) [11], a German medical school [12], SIRM [13], GitHub [10,14,15,16], and Kaggle [17,18]. It should be noted that all of the original COVID-19 images from the Deep-COVID and COVID-Net are included in this dataset [10,19]. Normal images originated from RSNA and Kaggle [17]. Non-COVID lung infections and abnormalities were sourced from RSNA as well, and the viral pneumonia images were from a specific pneumonia database [18].

All images were 299x299 pixels in the Portable Network Graphics (PNG) format.

3.2 Preprocessing

For our purposes, we wanted to utilize approximately 13,200 images in total. We did a 60/40 split for train and test respectively, and then an 80/20 split for train and validation. Our data breakdown was as follows:

Split	COVID	Non-COVID
Train	1440 (2880) Augmented up to 2880	4800 - 2079 Normal (43.3%) - 642 Viral pneumonia (13.4%) - 2079 Other lung disease (43.3%)
Validation	580	1199 - 519 Normal (~43.2%) - 161 Viral pneumonia (~13.4%) - 519 Other lung disease (~43.2%)
Test	1200	3999 - 1732 Normal (43.3%) - 535 Viral pneumonia (13.3%) - 1732 Other lung disease (43.4%)

Table 1. Number of images per category

From the 21,165 images available from the dataset, we randomly chose the 13,218 X-rays from each respective category to the desired split destination. From here, we utilized the tool FiftyOne [26] for recognizing and removing duplicates. With multiple sources of data, this step was important in order to curate a diverse dataset. To do so, we used MobileNetV2 to create embeddings for each image. From these embeddings, we calculated the similarity between images using cosine similarity. The average similarity was approximately 0.935, which was understandable given all the images were chest X-rays. We then defined the threshold in which to remove images at 0.99. In total, 100 duplicate images were removed.

To augment our training COVID-19 images, we used Augmentor [31]. With this tool, we did slight rotation to images with a 50% probability. We also performed a flip over the Y-axis at a 40% probability.

Prior to training (except in the case for InceptionV3) images were downsampled to 256x256 pixels and cropped to 224x224 pixels. For InceptionV3, images stayed at 299x299 pixels. Then, they were all normalized within the range of 0-1 with an average mean of 0.449 and an average standard deviation of 0.226 in all three color channels.

3.3 Models

3.3.1 ResNet18

ResNet [20] is one of the most popular CNN architectures that provides easier gradient flow for more efficient training. The core idea of ResNet, or Residual Net, is introducing a so-called identity shortcut connection that enables the network to provide a direct path to the very early layers in the network, making the gradient updates for those layers much easier. Many ResNet architectures are available, one of which is ResNet18 that has 18 layers.

3.3.2 ResNet50

ResNet50, much like its sibling ResNet18, utilizes identity shortcut connections. It does, however, as the name suggests, has 50 layers. It consists of five stages where each stage has a convolution and identity block. The convolution block has three convolution layers, and the identity block also has three convolution layers. With the shortcut connections, the increased number of layers do not pose a significant gradient issue.

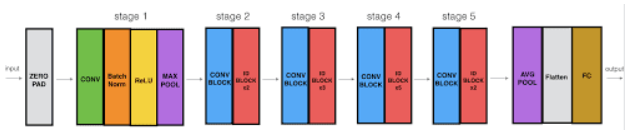

Figure 1. ResNet50 with 5 Stages [29]

3.3.3 SqueezeNet 1.1

In 2016, SqueezeNet was introduced as a competitor to AlexNet with similar accuracy, 50x less parameters, and <0.5MB model size [22]. It has one convolutional layer, eight fire modules, and a final convolutional layer. A fire module consists of a squeeze convolutional layer, usually 1x1, which leads into a layer with a mixture of 1x1 and 3x3 convolution filters.

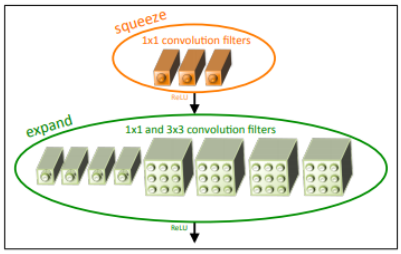

Figure 2. Fire module where $squeeze_{1x1} = 3$, $expand_{1x1} = 4$, $expand_{3x3} = 4$ **[22]**

3.3.4 DenseNet-121

In 2017, densely connected convolutional neural networks were introduced. While the neural network has 121 layers, it addresses the vanishing gradient problem, strengthens feature propagation, encourages feature reuse, and reduces the number of parameters by taking input from all previous layers [23]. Because of this, the network can be slimmer.

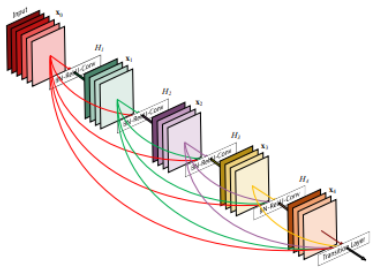

Figure 3. A 5-layer dense block with growth rate k =4 where each layer takes all preceding feature-maps as input. [23]

3.3.5 InceptionV3

InceptionV3 [33] is the third edition of Google's Inception CNN that mainly focuses on utilizing less computational power. In the Inception v3 model, several improvements such as factorized convolutions, regularization, dimension reduction, label smoothing, and parallelized computations are made for optimizing the network.

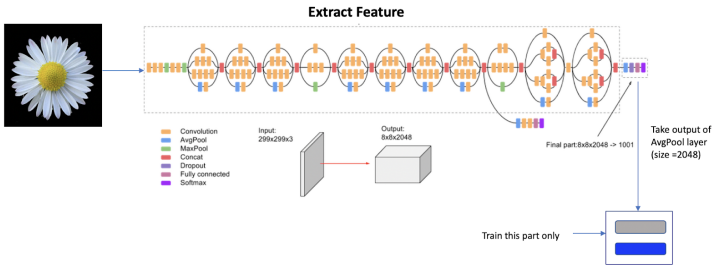

Figure 4. InceptionV3 architecture [25]

3.3.6 COVID-Net

We used the original COVID-Net in hopes of gaining a baseline for comparing pre-trained models with novel models. COVID-Net aims to maintain a lightweight pattern by utilizing a projection-expansion-projection (PEPX) design. This is described by Wang et. al. (2020):

- First-stage projection with 1x1 convolutions for projecting input features to a lower dimension
- Expansion with 1x1 convolutions for expanding features to higher dimension
- Depth-wise representation with 3x3 convolutions for learning spatial characteristics to minimize computational complexity while preserving representational capacity
- Second-stage projection with 1x1 convolutions for projecting features back to a lower dimension
- Extension with 1x1 convolutions that extend channel dimensionality to a higher dimension to produce the final features (6,7)

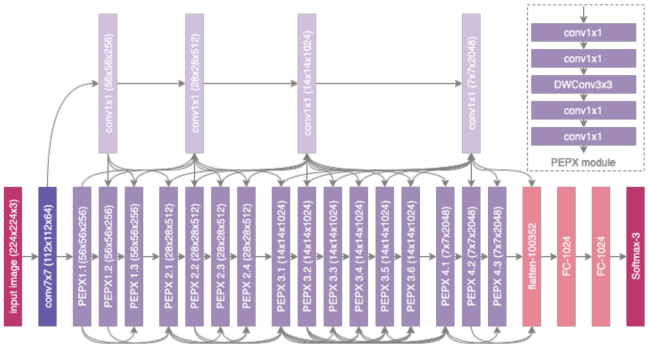

Figure 5. COVID-Net. PEPX design [2]

The overall architecture of this model includes 5 convolution layers, 15 PEPX layers, and finally fully connected layers as a classifier activated by a softmax function. The four intermittent densely connected convolutional layers within the PEPX layers act as a central meeting point to maintain long-range connectivity. They are sparsely intermixed to keep data representation without sacrificing computational complexity. Using frequent long-ranged connected layers would significantly limit performance in terms of memory and computation time [2].

3.3.7 COVID-Net + LSTM

With a baseline from COVID-Net and inspiration from Islam et. al. [3], we modified the original COVID-Net by including a LSTM layer in hopes of improving the classification layer. Therefore, we removed a fully connected layer and replaced it with the LSTM layer after the series of PEPX layers and right before the last fully connected layer in COVID-Net model. The output shape of the PEPX block would be of size (batch_size, 7, 7, 424).

Before passing the output of the PEPX block to the LSTM layer, the PEPX output is reshaped to size (batch_size, 49, 424) in order to make it compatible for the LSTM layer. The LSTM layer is used to extract time information from the feature map. After analyzing the time characteristics, the output of the LSTM layer is passed through the fully connected layer to predict the category of the image.

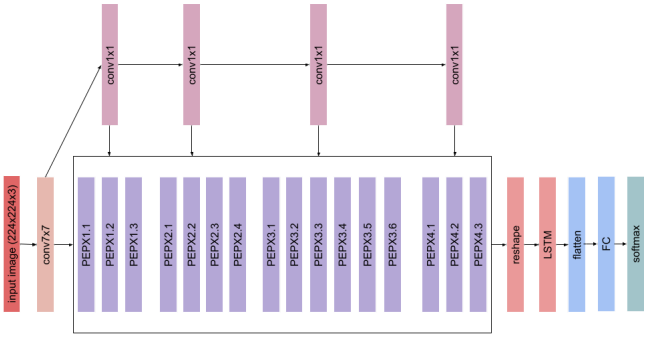

Figure 6. COVID-Net+LSTM

3.4 Implementation

For code collaboration, we used Google Colaboratory Pro and GitHub. Google Colaboratory Pro ranges in GPU and memory available depending on demand. GPU access can range from NVIDIA Tesla T4 to P100. We used PyTorch version 1.31, Python version 3.7.9, SciKit-Learn version 0.23.2, and TorchVision version 0.4.2.

We initially started with the original code and data of Deep-COVID, which only included a ResNet18 implementation, from GitHub to get a baseline [24]. We achieved similar results in accuracy, sensitivity, and specificity with their repository. From there, we adjusted the code to our needs by introducing additional models: ResNet50, SqueezeNet 1.1, DenseNet-121, InceptionV3, COVID-Net, and COVID-Net+LSTM. With the pre-trained models on ImageNet, we only updated features in the last fully connected layer, meaning doing only feature extraction. For COVID-Net and COVID-Net+LSTM, we used the original code from COVID-Net [30] and added a LSTM layer for the variation.

Our hyperparameters were generally a learning rate of 0.004, 50 epochs, momentum of 0.9, and batch size of 32.

Adam optimizer, an extension of stochastic gradient descent, was used in order to adapt to the learning rate which can help with overfitting. We also used cross-entropy loss defined as:

$$Loss_{CE} = -\sum_{i=1}^{N} p_i log q_i$$

where p_i and q_i denote the ground-truth, and predicted probabilities for each image, respectively.

To assess for overfitting or underfitting in all models during training, we kept track of training and validation loss at each epoch. For brevity, we will include one plot from ResNet18 training, but overall, we tuned our hyperparameters to get high accuracy while maintaining similar loss values in training and validation, as seen in the learning curve in Figure 7. It was also important to ensure a loss plateau at an assumed local optima.

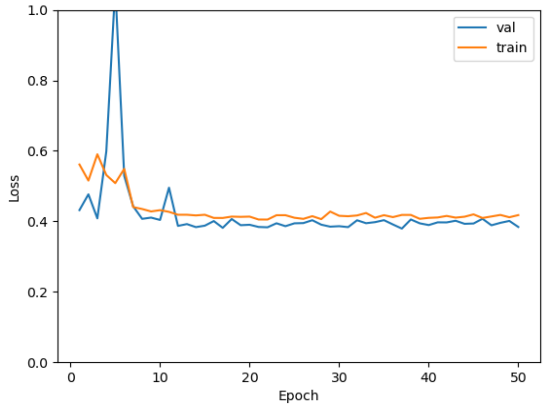

Figure 7. Training loss of validation vs train

For testing our models, we determined the best threshold for binary classification for our slightly imbalanced dataset based on the maximum geometric mean at each threshold using the true positive rate and false positive rate.

$$G_{mean} = \sqrt{TPR * (1 - FPR)}$$

This cutoff threshold averaged 0.36 for these models.

3.5 Metrics

In order to assess the performance and gain insight on these machine learning models, we wanted to use standard metrics. For our models, we used the 5 metrics [21] - sensitivity, specificity, precision, accuracy and F1-score. Specificity and sensitivity were added since our dataset is still imbalanced and would make the comparison to the other models easier. The metrics are defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$F1 - score = \frac{2 * Precision * Sensitivity}{Precision + Sensitivity}$$

We also reported ROC-AUC as well for a visual of true positive versus false positive rates for this classification problem.

4. RESULTS

The best possible outcome we could have hoped for was a model, or multiple models, beating their original metrics with our enlarged dataset. The next best possible outcome would be meeting their results in hopes of solidifying the techniques as a standard for future comparison.

4.2 Comparison of Models

Model	Sensitivity	Specificity
ResNet18	0.81	0.873
ResNet50	0.891	0.869
SqueezeNet 1.1	0.911	0.774
DenseNet-121	0.898	0.851
InceptionV3	0.862	0.88
COVID-Net	0.852	0.937
COVID-Net + LSTM	0.852	0.926

Table 2. Comparison of sensitivity and specificity

Model	Precision	Accuracy	F1 Score
ResNet18	0.657	0.858	0.725
ResNet50	0.671	0.874	0.766
SqueezeNet	0.548	0.806	0.684
DenseNet	0.644	0.862	0.75
InceptionV3	0.683	0.876	0.762
COVID-Net	0.803	0.918	0.827
COVID-Net + LSTM	0.775	0.909	0.812

Table 3. Comparison of precision, accuracy, and F1 score

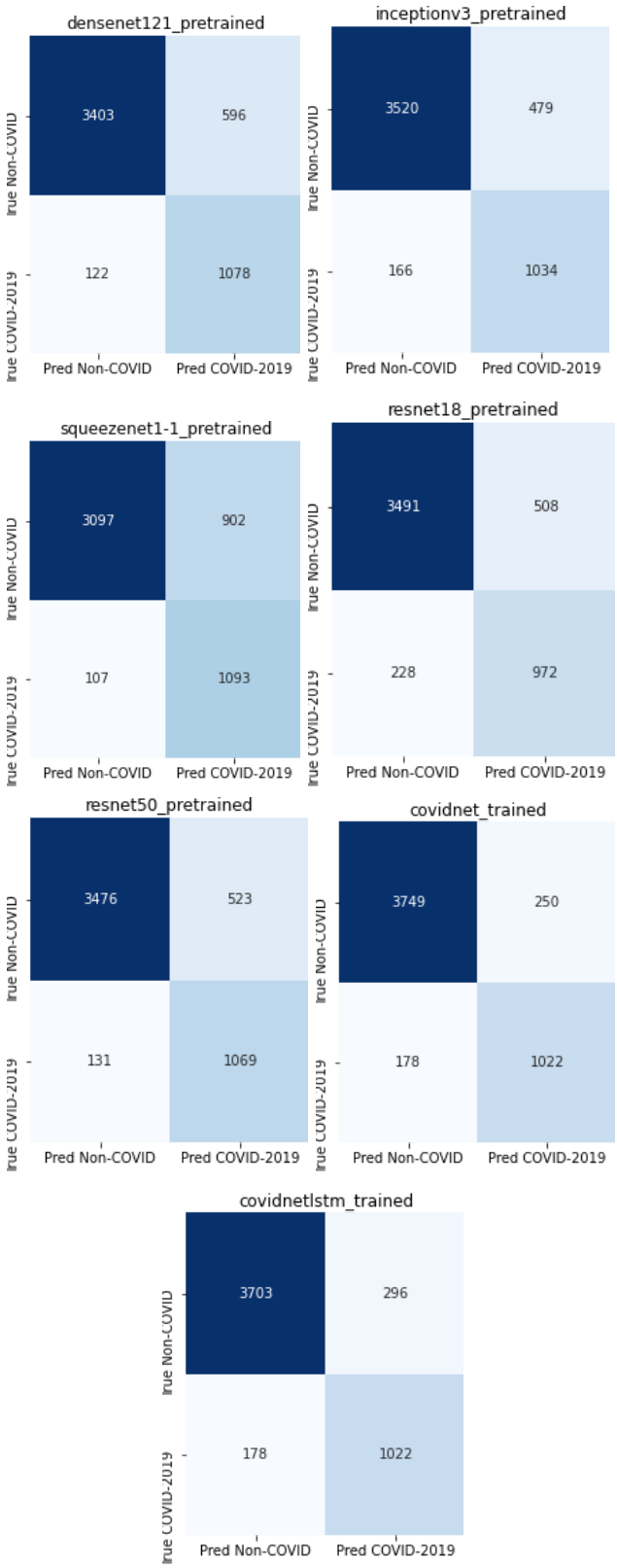

Figure 8. Confusion matrices of all models.

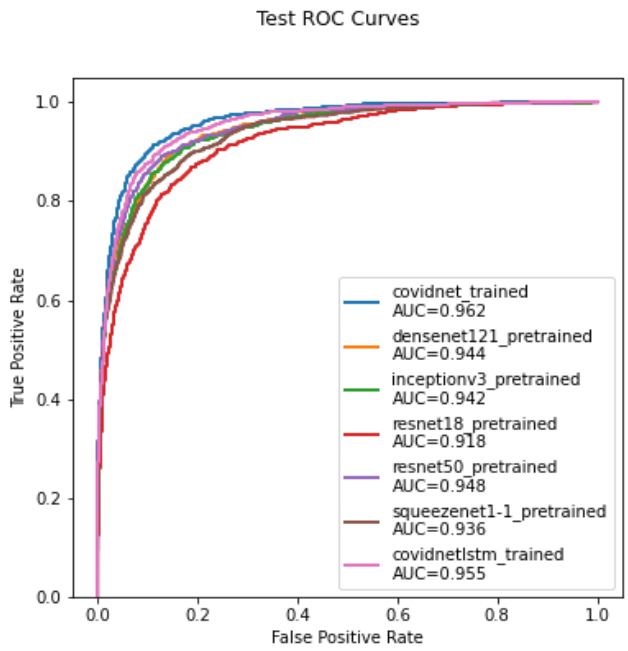

Figure 9. ROC-AUC of all models.

5. DISCUSSION

The evolution of our final results and implementation since the draft phase had a variety of focus and improvements. First, we took another pass at our dataset, more specifically the train vs test vs validations splits. We increased the dataset from about 13,000 X-rays to 13,200 X-rays. With our training set, we implemented some data augmentation techniques like rotation and flipping. We also increased the size of our validation set. In the draft run, the validation set was far too small and did not even reach 20% of the training set. Second, we tuned our hyperparameters by comparing training loss in the training and validation sets. By observing the loss, we were able to adjust learning rate and its relation to batch size and epochs. Third, we explored more novel approaches to COVID-19 classification to compare with our pre-trained models. We hoped to get a baseline with COVID-Net because it was specifically tuned for COVID-19 detection, and then we added a LSTM layer to see if it improved the results. Fourth, we squeezed out slightly better metrics by threshold moving.

5.1 Results

Overall, our final results were a vast improvement from our draft results, which seemed to have performed as well as a coin flip. COVID-Net and COVID-Net + LSTM were strong performers in comparison to pre-trained models. However, pre-trained models did not trail too far behind in specificity, sensitivity, and AUC.

5.2 Challenges & Future Work

One of the biggest challenges was the turn-around-time for training models. Though we decreased the number of epochs generally, it still took a significant amount of time to train models, adjust parameters, and continue again. Additionally, Google Colab's resource sharing may have been a contributing factor in this as well, since resource availability varies according to user demand and past usage [32].

COVID-Net+LSTM model can be further improved by applying the meta-learning technique as proposed in Ravi et. al. [28]. LSTM layer can be converted into a meta-learner model to learn exact optimization algorithm which, in turn, is used to train COVID-Net or the pre-trained models in the few-shot learning regime.

We also could have fine-tuned pre-trained models more extensively as described by Tajbakhsh, et. al. [8]. Given more time, we believe fine-tuning the pre-trained models may have performed as well as COVID-Net. However, our pre-trained models' final results significantly improved from a best AUC of 0.88 from InceptionV3 in the draft to a best AUC of 0.944 from DenseNet with hyperparameter tuning and data improvements. By learning how to better optimize hyperparameters and how to split data into a validation set, we bettered our results without changing the core of the pre-trained models.

Some future work that would provide beneficial insight would be multi-classification e.g. COVID-19, normal, non-COVID-19 pneumonia, and other lung abnormalities. From there, we could determine if the classifier is having difficulty distinguishing COVID-19 pneumonia from the viral pneumonia images, which has been seen in other studies [7]. For diagnostic purposes, differentiating between non-COVID-19 and COVID-19 pneumonia would not only be beneficial for patient outcomes, but also could lead to greater insight into the disease process itself.

As more COVID-19 images become available, further analysis with more and balanced data should be explored. Despite increasing our dataset, we still encountered a comparatively small pool of COVID-19 images.

6. CONCLUSION

We have made improvements to our models, and introduced COVID-Net and COVID-Net + LSTM since our draft. With changes to our data, hyperparameter tuning, and exploring novel approaches, we were able to compare pre-trained models and more novel approaches. While COVID-Net generally had the best performance, pre-trained models did not perform poorly and were only slightly behind. Given time, we believe all models could be improved with more data and experimentation. As it stands, COVID-Net remains to be a steady standard for COVID-19 classification.

With COVID-19 persisting globally, we believe it is necessary to continue exploring deep learning strategies not only for academic purposes, but also for medical and health care advancement.

7. REFERENCES

[1] Minaee, S., Kafieh, R., Sonka, M., et. al. Deep-COVID: Predicting COVID-19 from chest X-ray images using deep transfer learning. *arXiv*:2004.09363v3;2020.

https://arxiv.org/pdf/2004.09363.pdf

[2] Wang, L., Zhong Q., & Wong, A. COVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. *arXiv*:2003.09871v4;2020.

[3] Islam, Md.Z., Islam, Md.M., & Asraf, A. A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images. *Informatics in Medicine Unlocked* 2020;20:1-11. https://doi.org/10.1016/j.imu.2020.100412

[4] Worldometer. COVID-19 Coronavirus Pandemic. https://www.worldometers.info/coronavirus/

[5] World Health Organization. Timeline: WHO's COVID-19 Response. https://www.who.int/emergencies/diseases/novel-coronavirus-2019/interactive-timeline#!

[6] Maguolo, G. and Nanni, L. A critic evaluation of methods for COVID-19 automatic detection from X-ray images. *arXiv*:2004.12823;2020.

[7] Qiao, Z., Bae, A., Glass, L., et. al. FLANNEL (Focal Loss bAsed Neural Network Ensemble) for COVID-19. *Journal of the American Medical Informatics Association* 2021; 28(3): 444-452.

[8] Tajbakhsh, N., Shin, J.Y., Suryakanth, R.G., et. al. Convolution neural networks for medical image analysis: Full training or fine tuning? *arXiv*:1706.00712v1;2017.

[9] https://www.kaggle.com/tawsifurrahman/covid19-radiography-database

[10] https://github.com/ieee8023/COVID-chestxray-dataset/

[11] https://bimcv.cipf.es/bimcv-projects/bimcv-covid19/#1590858128006-9e640421-6711

[12] https://github.com/ml-workgroup/covid-19-image-repository/tree/master/png

[13] https://sirm.org/category/senza-categoria/covid-19/

[14] https://eurorad.org

[15] https://figshare.com/articles/COVID-19_Chest_X-Ray_Image_Repository/12580328

[16] https://github.com/armiro/COVID-CXNet

[17] https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data

[18] https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

[19] Cohen, J., Morrison, P., and Dao, L. COVID-19 image data collection: prospective predictions are the future. *Journal of Machine Learning for Biomedical Imaging* 2020; 2: 1-38.

[20] He, K., Zhang, X., Ren, S. and Sun, J. Deep Residual Learning for Image Recognition. *arXiv*:1512.03385v1;2015.

[21] https://medium.com/analytics-vidhya/how-to-select-performance-metrics-for-classification-models-c847fe6b1ea3

[22] Iandola, F., Han, S., Moskewicz, M., et. al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5MB model size. *arXiv*:1602.07360v4;2016.

[23] Huang, G., Liu, Z., van der Maaten, L., et. al. Densely connected convolutional networks. *arXiv*:1608.06993v5; 2018.

[24] https://github.com/shervinmin/DeepCovid

[25] https://alquarizm.files.wordpress.com/2019/03/image-4.png

[26] https://voxel51.com/fiftyone/

[27] https://pytorch.org/tutorials/beginner/finetuning_torchvision_models_tutorial.html

[28] Ravi, S., Larochelle, H. Optimization as a model for Few-shot learning.

https://openreview.net/references/pdf?id=BJoM2P45e

[29] https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33

[30] https://github.com/iliasprc/COVIDNet

[31] https://github.com/mdbloice/Augmentor

[32] https://research.google.com/colaboratory/faq.html

[33] Szegedy, C., Vanhoucke, V., Ioffe, S., et. al. Rethinking the Inception Architecture for Computer Vision. *arXiv*:1512.00567v3;2015.

[34] Chowdhury, M.E.H., Rahman, T., Khandakar, A., Mazhar, R., Kadir, M.A., Mahbub, Z., Islam, K., Khan, M., Iqbal, A., Al-Emadi, N., Reaz, M.B.I, Islam, M. T. Can AI help in screening viral and COVID-19 pneumonia? *IEEE Access*2020;8:132665-132676.

[35] Rahman, T., Khandakar, A., Qiblawey, Y., Tahir, A., Kiranyaz, S., Kashem, S.B.A., Islam, M.T., Maadeed, S.A., Zughaier, S.M., Khan, M.S. and Chowdhury, M.E. Exploring the Effect of Image Enhancement Techniques on COVID-19 Detection using Chest X-ray Images. *Computers in Biology and Medicine* 2021;132.