COVID-19 Severity Prediction from Lung Radiography Images Using Deep Learning

1st Zitian Tang
College of Engineering
Georgia Institute of Technology
Atlanta, United States
tzt15901481291@gmail.com

1st Danni Chen
College of Engineering
Georgia Institute of Technology
Atlanta, United States
dannichen2000@gmail.com

2nd Peter Lais
College of Engineering
Georgia Institute of Technology
Atlanta, United States
lais.peter.7@gmail.com

3rd Sofiya Vyshnya
College of Engineering
Georgia Institute of Technology
Atlanta, United States
sofiya.vyshnya@gmail.com

4th Lawrence He
College of Engineering
Georgia Institute of Technology
Atlanta, United States
lhe80@gatech.edu

***Abstract*—As of September 23 2021, WHO (World Health Organization) reported 229,858,719 confirmed cases of COVID-19 [1]. The number of COVID-19 positive cases continues to grow around the world, forcing doctors to quickly examine CT scans manually to categorize the severity of disease. Therefore, it is urgent to construct an automated image classification model to help doctors triage patients by automatically estimating COVID-19 severity. We used a 1,110-patient COVID-19 lung CT image dataset provided by medical hospitals in Moscow, Russia. All CT images are categorized into one of the 5 groups: CT-0, CT-1, CT-2, CT-3, and CT-4, with CT-0 corresponding to subjects without COVID symptoms, and numbers 1-4 representing increasing disease severity. For simplicity, CT-4 subjects were combined with CT-3, as the total number of subjects in CT-4 is too small (2 subjects) for robust classification. Our team used transfer learning with the VGG16 network to estimate disease severity from lung CT images. We split the dataset using a 80-20 training/testing ratio. Our model predicts subject severity on a scale of 0-3. As a result, our model did not exhibit state-of-the-art performance in differentiating COVID-positive against negative subjects as well as in predicting the disease severity. Future works could be centered at hyperparameters optimization and data augmentation on the foundation of the current model proposed to reach better performance.**

***Index Terms*—COVID-19, Artificial Intelligence, Transfer Learning, Lung CT**

I. Introduction

To date, COVID-19—the disease caused by the SARS-Cov-2 virus—has led to over 220 million confirmed cases and more than 4 million deaths worldwide [1]. One of the most common methods for SARS-CoV-2 testing is reverse transcription-polymerase chain reaction (RT-PCR), which directly detects the presence of SARS-CoV-2 RNA in respiratory tract specimens [2]. Recently, computed tomography (CT) has been proposed as a faster and more sensitive alternative to RT-PCR [3].

As the number of cases continues to grow, doctors are forced to manually examine multiple CT scans and categorize the severity of a patient's disease in a limited amount of time. This can lead to errors during diagnosis. Recent advances in deep learning have enabled the use of segmentation and classification algorithms to detect COVID in patients based on medical image data [4].

In this work, our team attempted to leverage transfer learning in order to automatically estimate disease severity and triage COVID patients based on axial CT scans of the lung. We evaluated the performance of our model through various metrics and applied our insights to guide our next steps in overcoming the current limitations observed within our approach. Our model can be found on GitHub at: https://github.gatech.edu/plais3/ihs-2021.

II. Methods

A. Dataset

In this study, we used a COVID-19 lung CT image dataset provided by hospitals in Moscow, Russia [5].[1] The dataset consists of 1,110 lung CT images from different subjects categorized into 5 groups: CT-0, CT-1, CT-2, CT-3, and CT-4 (Table I). CT-0 corresponds to subjects without COVID symptoms, and CT 1-4 represents COVID-positive individuals with increasing disease severity. For simplicity, we grouped all CT-4 subjects together with CT-3 patients since the size of group CT-4 (2 subjects) is too small for robust classification.

TABLE I
Class Sizes: Number of patients in each class

	CT-0	CT-1	CT-2	CT-3 + 4
No. patients	254	684	125	47

[1]The data may be accessed publicly at https://healthcaresummit.ieee.org/data-hackathon/ieee-covid-19-imaging-informatics-challenge/.

B. Data Preprocessing

All images in the lung CT dataset were formatted in the Neuroimaging Informatics Technology Initiative (NiFTI) format. To prepare the data for training, we first extracted CT scan slices from the NiFTI files to investigate which slices contain the most distinct evidence of COVID-19 symptoms. Sixteen subjects were randomly selected from the dataset (5 subjects from CT-0, 3 from CT-1, 3 from CT-2, 3 from CT-3, and 2 from CT-4), and every fifth slice of their CT image was plotted. The greatest distinction between COVID and non-COVID subjects, as well as between the various disease severities, was observed in the 25^{th} slice (Figure 1). Therefore, the 25^{th} slice of each NiFTI image was saved in its own NumPy file for later use.

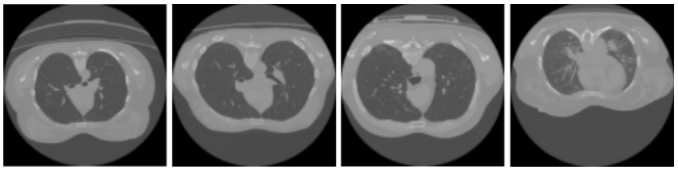

Fig. 1. Sample Sliced Lung CT Scans for Subjects in Category CT-0, CT-1, CT-2, and CT-3

The selected slices were then rescaled and normalized in accordance with the conventions used in the pre-trained VGG16 model available from PyTorch (https://pytorch.org/vision/stable/models.html). To be compatible with the model's input requirements, the one-channel 512×512 slices were converted to three-channel grayscale images of shape $512 \times 512 \times 3$. The slices were split into training and testing data according to a 80-20 training/testing ratio.

C. Model Initialization

In this work, we relied on transfer learning in an attempt to demonstrate the feasibility of using the pre-trained VGG16 model for COVID detection. Previous works in transfer learning have highlighted the favorable performance of ImageNet-trained [7] models on classification of selected categories of medical images, motivating the current project [8] [9]. Since the initial layers of the VGG16 model are already trained, these layers may be 'frozen' (prevented from changing) as the model is trained on a new dataset in order to to significantly reduce the time and computational resources required by the model during training. The VGG-16 architecture is depicted in Figure 2. The input into the first convolutional layer must have dimensions $H \times W \times 3$, where H and W must be at least 224 pixels. The convolutional + ReLU and pooling layers are used for feature extraction, while the dense layers are used for classification. A softmax layer maps the output of the last dense layer to the model output.

To adapt the pre-trained VGG16 model to our COVID classification task, we removed the final dense layer and froze the other layers. We replaced the last dense layer with a new dense layer whose output was a one-dimensional tensor of size $1 \times 1 \times 4$, reflecting our model's intended use for a four-category classification problem.

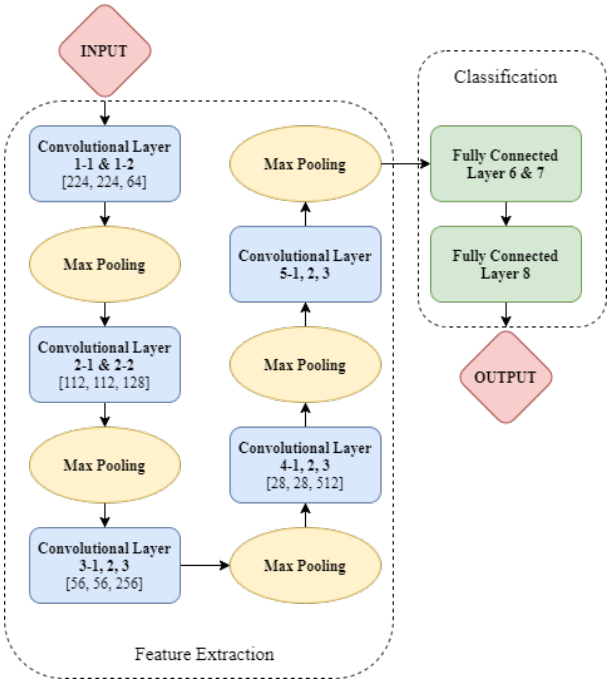

Fig. 2. Architecture of the VGG16 model.

D. Model Training

After we initialized and modified the VGG16 model, we determined the criterion and optimizer that would be used during our model training process. We used cross entropy as the loss function and stochastic gradient descent (SGD) as the optimizer for our model. The optimizer was set to have only the parameters of the classifiers being optimized.

We trained the model using the following hyperparameters: training lasted for 100 epochs with batch size $= 32$, learning rate $= 0.001$, momentum $= 0.9$, and weight decay $= 0.0005$. All training was done using pytorch 1.8.0 on a remote Linux server running Ubuntu 18.04.5 with an Intel Xeon E5-2690 CPU (2.60 GHz) and Tesla K80 GPU (12 GB VRAM) using CUDA version 11.2.

E. Model Testing and Performance Metrics

In order to test the performance of our model, we utilized the area under the receiver operating characteristic (AUROC), the Matthews correlation coefficient (MCC), and the overall accuracy, together with the training and testing loss to measure how well was the model learning. Table II shows the confusion matrix corresponding to our testing results, with the numbers on diagonal in green showing the correct number of images the model predicted, and numbers in red representing the incorrectly classified images. All metrics were calculated using Scikit-Learn 0.24.1.

TABLE II
CONFUSION MATRIX (BASED ON TESTING RESULTS)

		True/Actual Class			
		CT-0	**CT-1**	**CT-2**	**CT-3**
Predicted Class	**CT-0**	13	38	0	0
	CT-1	18	114	4	1
	CT-2	5	20	0	0
	CT-3	2	8	0	0

1) AUROC: In **binary classification**, the area under the receiver operating characteristic (AUROC) tells us what percent of the time our model will correctly assign a higher absolute risk to a randomly selected subject with COVID compared to a randomly selected subject without COVID. The ROC curve is a plot of the false positive rate (FPR) on the x-axis versus the true positive rate (TPR) on y-axis. It shows the trade-off between sensitivity and specificity for various decision thresholds. Mathematically, TPR – or sensitivity – is equal to the fraction of true positives (TP) over the total positives, as shown in equation (1). The FPR is equal to 1 – specificity, where specificity is defined as the fraction of false positives (FP) over the actual number of individuals who do not have the disease, as shown in equation (2). In this project, where we have a **four-category classification** problem, we calculated the AUROC using the one-versus-rest technique, in which we considered CT-1, 2, and 3 as COVID-positive group while CT-0 as COVID-negative group.

$$TPR = \frac{TP}{TP + FN} \tag{1}$$

$$FPR = 1 - \frac{TN}{TN + FP} \tag{2}$$

2) MCC: The Matthews correlation coefficient (MCC) is able to generate promising results, even under imbalanced datasets, because it takes the ratio between positive and negative elements into consideration [6]. In a **binary setting**, the MCC can be calculated as follows. Suppose we define arbitrary variables N, S, and P as shown below (where TP, TN, FP, FN stands for true positive, true negative, false positive, and false negative, respectively):

$$N = TN + TP + FN + FP$$
$$S = \frac{TP + FN}{N}$$
$$P = \frac{TP + FP}{N}$$

MCC can then be calculated according to equation (3).

$$MCC = \frac{TP/N - S \times P}{\sqrt{PS(1-S)(1-P)}} \tag{3}$$

For our **multiclass implementation**, we used the *matthews_corrcoef* method in sklearn.metrics (Scikit-Learn version 0.24.1) to compute the MCC for our multi-class classification model.

III. RESULTS AND DISCUSSION

In this section, we investigate the model performance based on the metrics defined previously. More specifically, we endeavor to see how the model scored under different metrics, from which we may then interpret the strengths and weaknesses of the model.

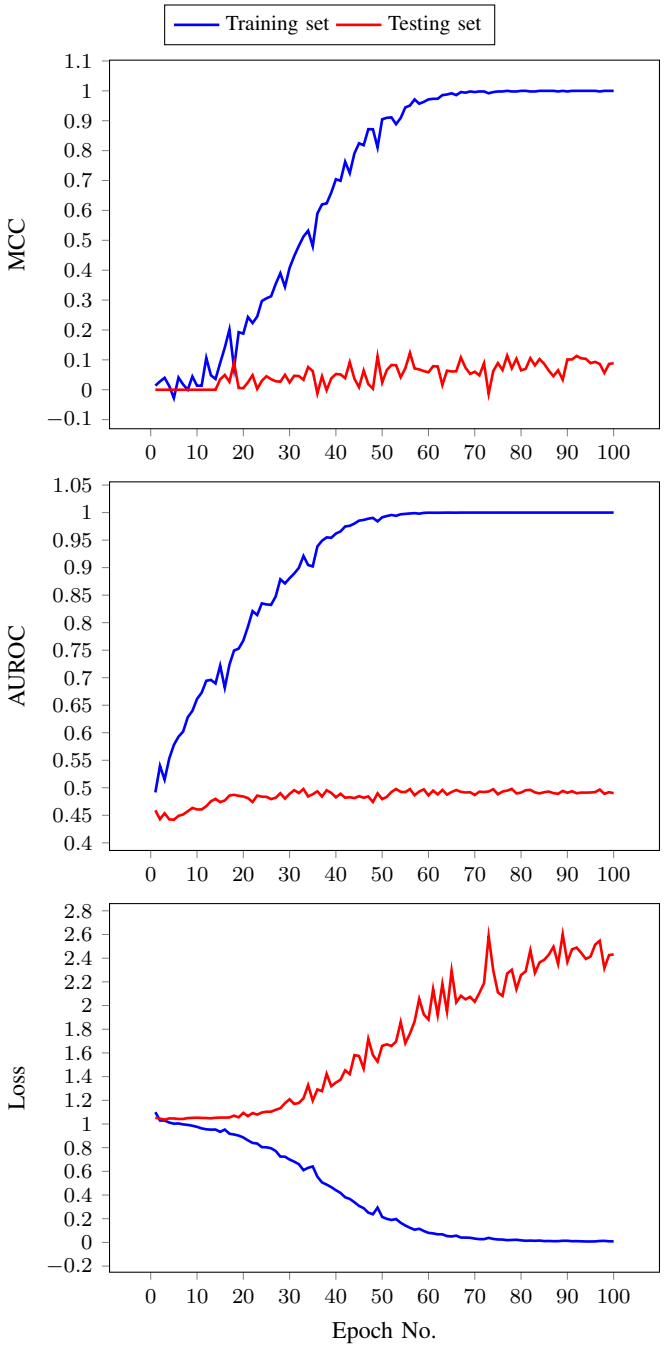

Fig. 3. Performance metrics for the pretrained VGG16 model obtained using the training and testing sets.

Plots of the performance metrics for the model over time can be seen in Figure 3. As can be seen, the MCC for the training set increases from 0 (where the model predicts randomly) to

nearly 1.00, whereas the testing-set MCC fails to improve significantly. The same trend can be seen for the AUC. Overall training loss decreases to near zero, whereas testing-set loss remains relatively constant over time. These metrics collectively indicate a failure of the model to generalize to the testing set. Without additional analysis into model activations (using explainable AI frameworks such as GradCAM) or related parameters, it is difficult to ascertain as to why the model is overfitting to the training set in this way.

Our project started with training a binary classifier which used VGG16 as a base model. For the initial approach, we selected the 25^{th} slice of every subject to be the input data and only the last layer of the network is "unfroze". However, the performance of this approach was not satisfied as the model seemed to be overfitting. In an attempt to combat overfitting, a series of increasingly radical adjustments were made to the model. Our team first tried to have the 23^{th}, 24^{th}, and 25^{th} slices stacked on top of each other instead of a single channel being tiled into a three-channel image. Then we have the 20^{th} to the 25^{th} slices being investigated individually. We have also turned to unfreeze all model parameters instead of only that of the last layer. Moreover, data augmentation was used as random flipping and rotation were done on certain number of images. Unfortunately, none of the approaches mention above has mitigated the problem.

Future work will center around attempting to discover the underlying causes of the model's poor performance [10]. The initial rapid plateau of the model suggests that the learning rate may be set too high, so we believe future investigations into hyperparameter optimization—namely, those optimizations that modify the learning rate—to be justified. Data augmentation may also alleviate potential model focus on noise in favor of more general aspects of COVID-positive images in order to increase the model's ability to generalize to the testing set. This may take the form of either perturbations in the contrast and saturation of each image or the inclusion of additional image slices beyond slice 25 of each NiFTI image within the training and testing pools.

IV. Conclusion

This project applied transfer learning from VGG16 model on COVID-19 testing using lung CT images. We discussed the specific data pre-processing techniques as well as the hyperparameters used for model training. However, our model did not exhibit a strong generalizability from its pre-trained dataset (ImageNet) to the COVID-19 lung CT images dataset. By addressing the approaches we have adopted to alleviate model overfitting in the Results and Discussion section, we believe our proposed model can be served as a useful precedent for others who are constructing pre-trained models on COVID-19 lung CT images. Through hyperparameters tuning and data augmentation, future studies may have the potential to reach more optimal performance.

References

[1] W. Zhao, W. Jiang, X. Qiu. Deep learning for COVID-19 detection based on CT images. Sci Rep 11, 14353 (2021).

[2] W. Wang, et al. Detection of SARS-CoV-2 in different types of clinical specimens. JAMA 323, 1843–1844 (2020).

[3] A. J. Rodriguez-Morales, et al. Clinical, laboratory and imaging features of COVID-19: a systematic review and meta-analysis. Travel Med. Infect. Dis. 34, 101623 (2020).

[4] K. Asipong, S. Gabbualoy, P Phasukkit . "Coronavirus Infected Lung CT Scan Image Segmentation Using Deep Learning." 2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2021.

[5] S. Morozov et al., "Mosmeddata: Chest ct scans with covid-19 related findings dataset," arXiv preprint arXiv:2005.06465, 2020.

[6] D. Chicco, G. Jurman. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics 21, 6 (2020).

[7] Russakovsky, Olga, et al. "Imagenet large scale visual recognition challenge." International journal of computer vision 115.3 (2015): 211-252.

[8] Hon, Marcia, and Naimul Mefraz Khan. "Towards Alzheimer's disease classification through transfer learning." 2017 IEEE International conference on bioinformatics and biomedicine (BIBM). IEEE, 2017.

[9] Tajbakhsh, Nima, et al. "Convolutional neural networks for medical image analysis: Full training or fine tuning?." IEEE transactions on medical imaging 35.5 (2016): 1299-1312.

[10] Yosinski, Jason, et al. "How transferable are features in deep neural networks?." arXiv preprint arXiv:1411.1792 (2014).