COVID-19 Medical Image Analysis

Chengyuan Cai
Dept. of Computer Science
University Of Virginia
Charlottesville, USA
cc4fy@virginia.edu

Tiancheng Ren
Dept. of Computer Science
University Of Virginia
Charlottesville, USA
tr6cz@virginia.edu

Yuchen Sun
Dept. of Computer Science
University Of Virginia
Charlottesville, USA
ys4aj@virginia.edu

Yanwen Wang
Dept. of Computer Science
University of Virginia
Charlottesville, USA
yw9cj@virginia.edu

***Abstract*—This document is a final project report for CS4774 of team "COVID19 Medical Image Analysis". This report will go through our initial idea, data processing, our approach to the models and newly developed algorithms, and the difficulties we have faced and conclusions of the project.**

I. Motivation

With a huge rising number of patients, developing medical image analysis can help quickly identify suspected COVID-19 cases among patients and reduce the workload of doctors. Batch processing of image data improves the efficiency of diagnosis of suspected patients and reduces the occurrence of misdiagnosis and missed diagnosis. For the huge increase in the number of confirmed diagnoses every day, the development of this project will greatly reduce the burden on medical staffs. In addition, this project will have a strong extensibility. Not only can we use machine learning to realize the diagnosis for COVID-19, we can also use it for the analysis of other medical images, such as x-rays, CT, nuclear magnetic resonance, etc.

II. Background

CT scan images for patients have been used as a reliable way of identifying COVID-19 cases, for there are some consistent characteristics in all COVID-19 positive CT scan images, which are the hazy, patchy white spots in the lung region. However, it has been stated in multiple research papers that the detection and quantification of the disease are too unreliable and inefficient when accessed manually. This brings our attention to developing machine-learning based programs to aid the diagnosis of COVID-19 and other lung diseases in combination with computed tomography. The machine learning algorithms can be better at targeting and scanning these extremely characteristic regions.

III. Related Work

The following are related works regarding machine learning algorithms applied in medical images analysis.

- Machine Learning based image analysis
  The paper describes a method of using a machine learning approach to analyze cell images. The model trained has the capability of identifying different cell morphology and types, and a well trained model becomes the tool for identifying cancers in immunofluorescent or immunohistochemical colored cell samples.

  Kan A. Machine learning applications in cell image analysis. Immunol Cell Biol. 2017 Jul;95(6):525-530. doi: 10.1038/icb.2017.16. Epub 2017 Mar 15. PMID: 28294138.

- Machine Learning based CT Scan analysis
  The paper describes a method of using a machine learning approach to analyze CT scan pictures. Well trained models can be used for identifying symptoms of infections or cancers from the CT scan.

  Feng Z, Rong P, Cao P, Zhou Q, Zhu W, Yan Z, Liu Q, Wang W. Machine learning-based quantitative texture analysis of CT images of small renal masses: Differentiation of angiomyolipoma without visible fat from renal cell carcinoma. Eur Radiol. 2018 Apr;28(4):1625-1633. doi: 10.1007/s00330-017-5118-z. Epub 2017 Nov 13. PMID: 29134348.

- Machine Learning based COVID-19 CT Scan analysis
  These kaggle pages show the work by Ashadullah Shawon and his associates on the CT scan pictures of COVID-19 patients, identifying possible infections.

  https://www.kaggle.com/shawon10/covid-19-diagnosis-from-images-using-densenet121/comments
  https://www.kaggle.com/plameneduardo/sarscov2-ctscan-dataset

IV. Claim and Target Task

The task of our project is to develop a method of identifying possible COVID-19 infections by using a machine learning approach for analyzing and categorizing CT Scan pictures.

V. Proposed Solution

Our project will first process all of those images, and apply five common models that we have learned in this class and use other outside sources. Most importantly, we have developed a brand new algorithm based on weight function which will pay much attention to the white space in the lung's image. After that, we combine all the results that we got and compare each of them by their accuracy and processing speed so that we can give a comprehensive report to the doctors as a reliable resource.

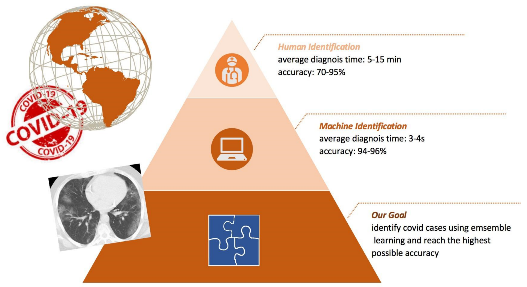

Human Identification
average diagnois time: 5-15 min
accuracy: 70-95%
Machine Identification
average diagnois time: 3-4s
accuracy: 94-96%
Our Goal
identify covid cases using emsemble learning and reach the highest possible accuracy
COVID-19
COVID-19
COVID-19

Fig. 1. Intuitive figure showing the necessity.

VI. Contribution

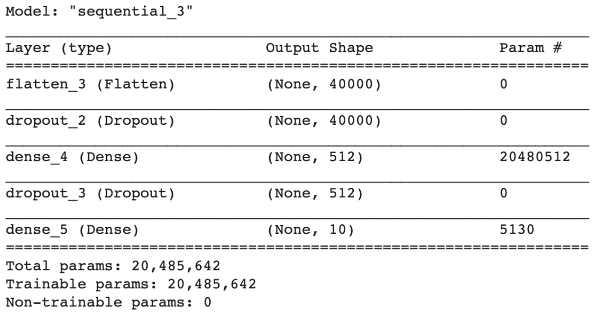
Our project has two major contributions to the related fields and researches as listed below.

- Out of the models we have mentioned in class, found out the best existing one for lung CT Scan Analysis, accuracy and speed wise.
- Developed a new algorithm with the potential of readily assisting doctor in simultaneously diagnosing multiple potential COVID cases through extremely fast analysis of batches of lung CT Scan images.

VII. Implementation

The five common models we have included are MLP, CNN, ResNet, KNN, and SVM.

A. MLP



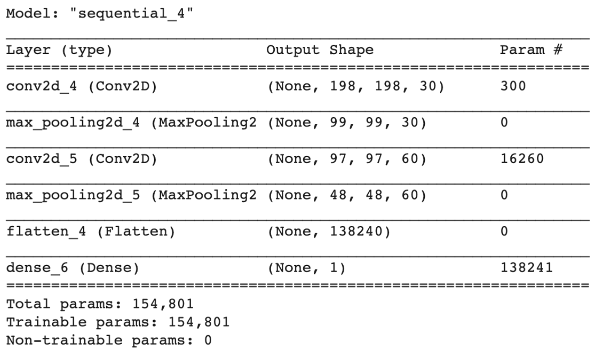
```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
flatten_3 (Flatten)          (None, 40000)             0         
_________________________________________________________________
dropout_2 (Dropout)          (None, 40000)             0         
_________________________________________________________________
dense_4 (Dense)              (None, 512)               20480512  
_________________________________________________________________
dropout_3 (Dropout)          (None, 512)               0         
_________________________________________________________________
dense_5 (Dense)              (None, 10)                5130      
=================================================================
Total params: 20,485,642
Trainable params: 20,485,642
Non-trainable params: 0
```

Fig. 2. MLP model

In the implementation of multilayer perceptron model, a sequential model is being used. The input shape is 200 $*$ 200. We use a flatten layer as the input, and the output layer is a dense layer. The activation function is sigmoid. The hidden layer is a dense layer with relu. Through multiple times of adjusting parameters, we find the best learning rate is 0.001, the batch size is 64 with 30 epochs, the validation split is 0.1. The training accuracy is around 60% and the validation accuracy is about 80%.

B. CNN

We then implemented a convolutional neural network model. We used the following parameters: optimizer = adam, batch size = 64, learning rate = 0.0001, epochs = 30, binary cross-entropy loss function.We decided to add two max-pooling layers, two layers with relu as the activation function, and the output layer with the sigmoid function. We used the sigmoid function because our output should be binary, either yes or no for COVID-19. The training accuracy of our CNN model nearly reached 100%, the validation and test accuracy fluctuated between 80-85%. We could have improved the model by adding more layers. However, it would have taken way too long to train, and defeats the purpose of having an efficient way to predict CT scans.



```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d_4 (Conv2D)            (None, 198, 198, 30)      300       
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 99, 99, 30)        0         
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 97, 97, 60)        16260     
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 48, 48, 60)        0         
_________________________________________________________________
flatten_4 (Flatten)          (None, 138240)            0         
_________________________________________________________________
dense_6 (Dense)              (None, 1)                 138241    
=================================================================
Total params: 154,801
Trainable params: 154,801
Non-trainable params: 0
```

Fig. 3. CNN model

C. ResNet

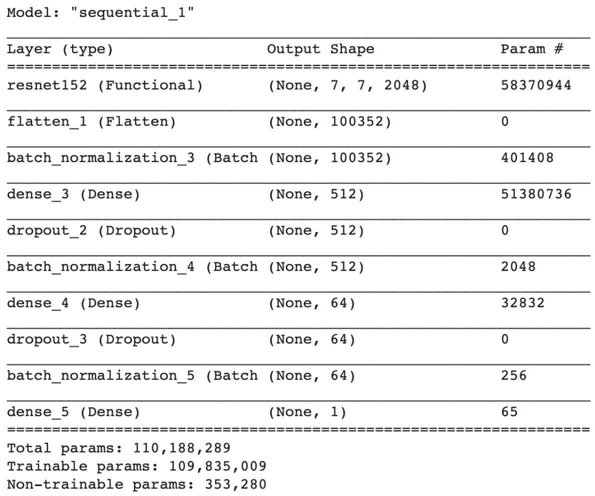
In order to solve complex classification problems, people add more layers in the deep neural network to learn more complicated features and improve performance. However, adding additional layers might lead to the problem of vanishing gradient and the degradation of accuracy. Residual Network helps to solve the degradation problem and achieves better accuracy through residual learning. There are ResNet models of 18, 34, 50, 101, 152 layers. In our experiment, we tried ResNet 50, 101, and 152 and fine-tuned each one. ResNet 50 has a relatively lower accuracy than the other two and the result of ResNet 152 fluctuates more widely. The highest validation accuracy of ResNet 101 could reach 97%. We use the following hyperparameters: pre-trained ImageNet weight, RMSprop optimizer, binary cross-entropy loss function, momentum = 0.9, batch-size = 32, and learning rate = 2e-5.

D. KNN

We also implemented a K nearest neighbors model. This model is by far the most basic out of all the models we implemented. We ran the K nearest neighbor with 8 different K values from 1 to 8. The best K values were 1 and 3. Different from our implementation of the homework, we simply used the KNN classifier provided by sci-kit learn.

E. SVM

The SVM model is implemented as follows. First, we have the images representing the COVID and the non-COVID cases, which are then standardized and read in as numpy arrays,



```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet152 (Functional)       (None, 7, 7, 2048)        58370944
_________________________________________________________________
flatten_1 (Flatten)          (None, 100352)            0
_________________________________________________________________
batch_normalization_3 (Batch (None, 100352)            401408
_________________________________________________________________
dense_3 (Dense)              (None, 512)               51380736
_________________________________________________________________
dropout_2 (Dropout)          (None, 512)               0
_________________________________________________________________
batch_normalization_4 (Batch (None, 512)               2048
_________________________________________________________________
dense_4 (Dense)              (None, 64)                32832
_________________________________________________________________
dropout_3 (Dropout)          (None, 64)                0
_________________________________________________________________
batch_normalization_5 (Batch (None, 64)                256
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 65
=================================================================
Total params: 110,188,289
Trainable params: 109,835,009
Non-trainable params: 353,280
```

Fig. 4. ResNet model

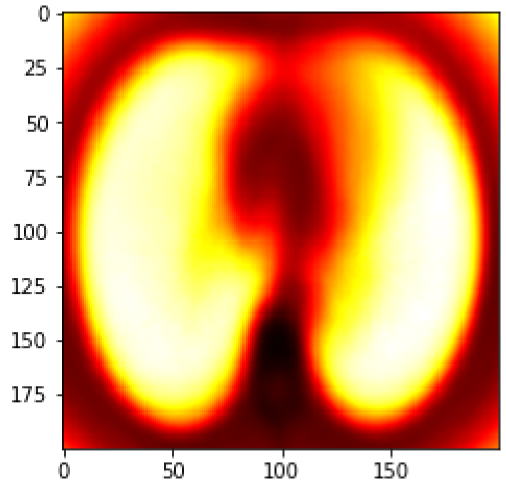


Fig. 5. Overall heat map showing location of the lung cavity.

so that a matrix containing the image information could be created. Then we employed the Sci-kit Learn's SVC module to fit the training data, and use the trained model to predict the test samples. The SVM produces high validation accuracy at around 0.88 but arrives at low test accuracy at around 0.71.

F. Our Model

Finally, we created an innovative model that is different from all the other models we implemented in the previous steps. Our implementation is shown as follows.

We initialize by having the COVID and the non-COVID images, which are then transformed into numpy arrays and information matrix in a similar manner as the implementation for previous models. In the next step, a heat map is generated utilizing the information matrix by summing the grayscale value for each individual pixel and is transformed into a visualizable heat map graph by the Matplotlib module. It is clear from the visual that the resulting heat map specifically marks the regions that are most likely to be the lung cavity in all images.

During the training phase, we randomly select 25 pixels for each image corresponding to the coordinate of the 25 randomly chosen pixels from the highest 90% of heat signatures. Then, using the selected pixel as the center, we expand our scope to two broader blocks for image-specific verification and evaluation: the test block and the evaluation block, respectively. The test block is a scope of 20pixels * 20 pixels area with the selected pixel at the center of the scope, while the evaluation pixel is a 50pixel * 50pixel area with the selected pixel at the center of the scope. For any selected pixel, the verification process would first occur to determine whether the selected pixel location represents the lung cavity region, and decides accordingly whether to discard the evaluation block given the true information of the proportion of the lung that the block represents. The heat map generation and extra verification process for selecting evaluation blocks are necessary due to the fact that not all images from the dataset are aligned. Simply employing a unified heat map would cause severe bias.

The verification process occurs by examining the standard deviation and mean of the test block, since the normal lung cavity region (without lesions) would be characterized by darker background color and few white dots, therefore, resulting in lower mean and lower standard deviations. If the verification determines that the test block belongs to the non-lung cavity region as it is discovering a high mean or standard deviation value from the block, the corresponding 50*50 evaluation block will be discarded from our learning process. However, if the verification determines that the test block is the lung cavity region since the mean and standard deviation falls between the given verification threshold, we keep the 50*50 evaluation block.

The verification process of our model recognizes the lung cavity region pretty accurately since it is observed that our model selects pixels to expand exactly according to the higher weight of the heat signatures aligned to the heat map. In addition, it is observed that our model is good at discarding the regions that actually represent non-lung cavity regions. Admittedly, bias still occurs during the verification process as our model occasionally decides to exclude the lesion regions from images simply because it extremely resembles a normal non-lung cavity region, but the effect of such error has been minimized by wisely selecting the threshold value for the verification process.

For each remaining evaluation block after verification, we evaluate specifically five parameters: the edge number, the mean, the range, the standard deviation, and the blur coefficient, in order for detecting the patchy, hazy white areas that represent lesions. Given the fact that lesions usually cause a lighter background color and produce scattered light gray pixels, it is believed that lesions would increase the mean of the evaluation block, lower the standard deviation of the evaluation block, and decrease the range of the evaluation

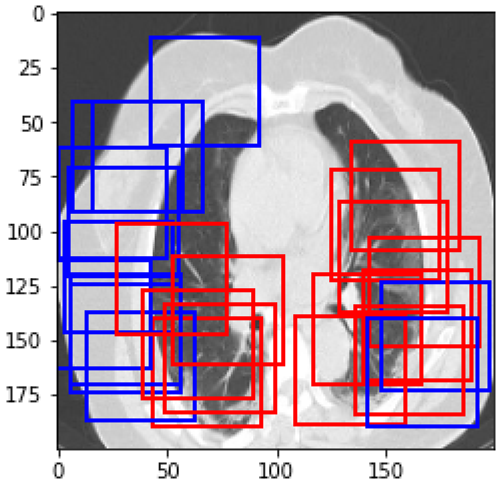


Fig. 6. Verification process visualization.

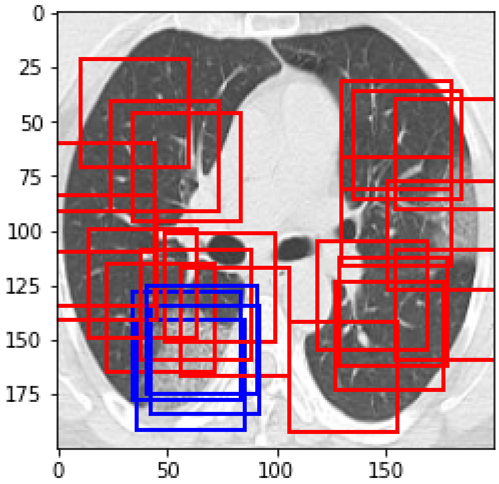


Fig. 7. Biased verification process visualization.

block. Different from the above basic statistical parameters, the edge number of blur coefficient is more complicated. Utilizing the CV2 module, we are able to generate the edge for each image by providing the image in the format of numpy array and the threshold for edge detection. It is seen that the number of edges in an image is very different for the COVID image when compared with the non-COVID image: The COVID image has a significantly higher number of edges due to the lesions. CV2 module has the ability to accurately determine the edges for the lesions, thus contributing to the extra number of edges in the COVID images. On the other hand, the blur coefficient is considered as it is recognized that some of the characteristics of the lesion regions really resemble a blurred picture. By convolving the matrix representing the image with the Laplacian kernel and determining the variance of the result, it is able for us to quantify how blur is the selected evaluation module. As higher the result goes, the more likely the picture is blurred.

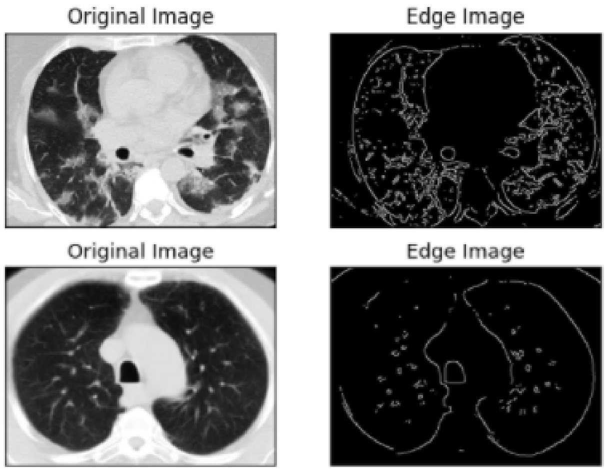


Fig. 8. Edge visualization of COVID image vs. non-COVID image.

G. Technical Challenges

We encountered three major technical challenges.

- Neural Network models demands extremely high resources for running.
  Our experiment included five neural network models: MLP, CNN, and ResNet, which all require complex layer structures in order to achieve high accuracy. However, the resource we have in possession is simply unable to support such high standard.
- Difficulties in assigning the best weight for five evaluation parameters.
  It is difficult to evaluate the relative importance of the five evaluation parameters by direct signing. The best composition of weight must be experimented.
- Disorganized dataset. The images in the dataset are disorganized and are in generally poor-quality. It causes bias for evaluating the COVID and non-COVID parameters in the learning process.

H. Technical Solutions

We developed the following solutions in order to counter each difficulty we experienced as stated in the previous subsection.

- The speed-accuracy trade-off must be faced. We used simple layer structure to evaluate the function and effect for adding each different layer types, and select typical representatives in our experiment. In the same time, we employed Google Colab GPU, which provides limited resources for our experiment. Still the result derived is enough for us to arrive at our conclusion.
- Given that our model is free from the acceleration of GPU, it is able for us to utilize the online running feature of Google Colab to enumerate all possible combinations of weight assignment within a preset range. We created a train and select model function (later removed after the

best weight is found) for selecting the best parameter that produces the highest accuracy.

- We developed a method of standardizing the images by creating the heat map with steps described in the previous subsections. The unified heat map makes it possible to choose the location where it is most likely that the location belongs to accurate lung cavity position in majority of images. In addition, the verification we created is also facilitating this standardization.

I. Technical Novelty

For each selected block, we evaluate specifically five parameters: edge number, mean, range, standard deviation, and blur. These help detecting the white areas in the lung. Such lesions would increase the mean, lower the standard deviation, and decrease the range. After training, we have the mean values for five parameters recorded separately for COVID and non-COVID cases. We compute scores by comparing the new image parameters with the recorded parameters, in the same time assigning different weights when summing those five parameters, and so we have two scores representing the COVID and non-COVID respectively. We compare these two scores to determine the final prediction result for any new image. Our model not only produces high accuracy, but also costs extremely short time to produce results.

VIII. Data Summary

TABLE I
Accuracy and Run Time for each Model

Model	Test Acc	Run Time (min)
MLP	0.7-0.8	10
CNN	0.83	5
Rsenet101	**0.93**	25
KNN	0.71	10
SVM	0.73	20
Our Model	0.85	**1**

From our test runs, we can see that out of all the existing models, Resnet101 clearly has the highest accuracy at 93 percent, however, it took 25 minutes to train around 2500 images, and is by far the slowest model. We have thus contributed by showing the most accurate state of the art model.

We have also achieved our second contribution. Our model runs faster than all five of the models mentioned in class, and it has a test accuracy of 85 percent, only a few percentage points lower than Resnet101. With so many new COVID-19 cases being diagnosed, and medical resources overwhelmed, a decrease in run time from 25 minutes to 1 minute is massive, even if it means we lose some accuracy.

IX. Experimental Results and Conclusion

Overall the deep learning models have better accuracy than the KNN and SVM models. The deep learning models all have their own black box, with hundreds of nodes that are interconnected. Each connection also has its own different weight. So the areas of the lungs that do not change in between the non-COVID and COVID might have a much lower weight than areas that do change, and we believe that this is the main difference which makes deep learning models have much better performances. KNN and SVM models also take all pixels into account. Both models calculate the distance between pixels, and CT scans are going to look similar for a bunch of unaffected areas of the lungs. Therefore, the areas that do matter, or the areas that do change depending on infected or not, may not have that much effect on the predictions.

Our model is different from but also inherits some signatures from both types of models. From KNN and SVM, our model inherits their signature of initially considering all pixels, but only to create a heat map showing the accurate location of the lung cavity. Then, our model also inherits from models like MLP, CNN, and ResNet, since it highly evaluates the key regions, and at the same time assign a higher weight to those key regions. But different from these models, our model is transparent, because it is easier to understand what our model does internally.

It is seen in the data summary section that our model produces accuracy higher than MLP, CNN, KNN, and SVM. The only model in our experiment that has higher accuracy than our model is ResNet101, which takes 25 minutes to complete running. Overall, our model spends the shortest time (ranked 1^{st} among all models in the experiment and achieved the 2^{nd} best accuracy. Therefore it is safe to say that by inheriting the advantages of both types of models, our model produces relatively high accuracy and costs an extremely short time to produce results. Such high accuracy and short running time well satisfied the requirement to produce a tool for readily available assistant diagnosis method.

X. Experimental Analysis

A. Ablation Study

The ablation study of our model reveals three key components of our model: the verification component, the learning component, and the prediction component. We believe that the verification component and the prediction component contributes to a majority of our model, and that the learning component only contributes to a minor part of our model. The reason for this assertion is that the verification component refines the choice of analysis and evaluation by specifying the location of lung cavity in each image. The removal of this verification process reduces the accuracy to around only 0.7. In the mean time, the prediction component is extremely important as well given the weight assignment. Altering the best weight assignment selected by enumerating all possible choices will reduce the final accuracy to a level even below 0.7. The reason we believe that the learning component only contributes to a minor part of the model is that the learning component is essentially an counting component, and it is simply summing all the parameters from all images and averaging them to produce the final parameter. It has minimal effect on accuracy, which is heavily affected by altering the verification threshold and the weight assignment, and also has almost zero effect on time cost, which is heavily affected by the number of pixels selected in the verification component.

Analyzing our experiment reveals the fact that the images provided in the dataset are considerably messy. The reason for this is that images are not aligned so that it is possible to observe the lung cavity position in one image doesn't match the lung cavity position in another image. In addition, some of the COVID images don't contain lesion positions. This is because of the mechanism of CT scans: the machine would scan your body from a top threshold to a bottom threshold, producing a series of continuous images. Therefore, if the images are organized without considering its relative position, it is very likely that the pile of images will contain a considerable amount of useless information, for each different image represents a different part of the lung and a perfectly normal image would definitely show unless the lung cavity of the patient is filled with all sorts of lesions. If we would be able to exclude these normal-looking images from the COVID dataset, it is almost definite that the accuracy of our model will increase dramatically.

B. *Novelty Application*

We are confident that our model is able to be applied to other medical image analysis, including MRI, X-ray, etc., given their similar signature compared with CT Scans. At the same time, we are also hopeful that our model has the capability of being employed in other image recognition and analysis domains such as examining meteorologic maps to identify and predict potential meteorologic phenomenons or even examine captchas images to identify the correct password, but we are absolutely certain that the specific parameters for verification and evaluation must be changed accordingly.

XI. Future Work

Future work includes getting more training data that contains detailed information about lung cavities and trying to improve the accuracy of our model. Besides, to increase the complexity of our model and add functionality, we could also find CT scans dataset labeled with specific pathological changes and try models such as YOLOv5, a fast and compact object detection model, to determine the anatomical location of the lesion. Combining our identifying model with the detection model will improve predictability and generate more helpful information for radiologists.

References

[1] A. Kathuria, "How to Build a COVID-19 Classifier Using PyTorch," Paperspace Blog, 06-Jul-2020. https://blog.paperspace.com/fighting-coronavirus-with-ai-building-covid-19-classifier/.

[2] O. Alwalid, "How to identify COVID-19 on a chest CT scan," Medmastery, 19-Nov-2020. https://www.medmastery.com/guide/covid-19-clinical-guide/how-identify-covid-19-chest-ct-scan.

[3] K. Xu, D. Feng, H. Mi, Deep Convolutional Neural Network-Based Early Automated Detection of Diabetic Retinopathy Using Fundus Image, Molecules. 22 (12) 2017.

[4] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, Proc Adv Neural Inf Process Syst. 1 (1) (2012), 1106 – 1114.

[5] Wernick M., Yang Y., Brankov J., Yourganov G., Strother S. Machine Learning in Medical Imaging. IEEE Signal. Process. Mag. 2010; 27: 25-38. 10.1109/MSP.2010.936730. 25382956

[6] Kukačka J., Golkov V., Cremers D. Regularization for Deep Learning: A Taxonomy. arXiv. 2017. 1710.10686

[7] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition. arXiv. 2015. 1512.03385