COVID-19 X-Ray Image Classification Using Transfer Learning and Contrastive Learning

Farima Fatahi Bayat, Runchu Ma, Sara Shoouri, Jiahong Xu
{*farimaf, runma, sshoouri, jiahongx*}*@umich.edu*

***Abstract*—As COVID-19 spreads around the world, increasing demand for COVID-19 testing burdens health care workers. Researchers started to utilize machine learning techniques to identify COVID-19 cases using X-rays for fast and more accurate predictions. Motivated by this, a decision fusion method using multiple CNN models has been proposed in this research to accelerate the analysis of chest X-ray images. We have leveraged the power of three pre-trained models (VGG16, InceptionV3, DenseNet121) with freezing of the feature extraction layers to extract the features of the image datasets and trained the last fully connected layers to classify the images. Moreover, We present a Siamese network to integrate the contrastive learning method with a fine-tuned pre-trained DenseNet121 model to capture unbiased feature vectors for final classification. Finally, We have achieved 95.57% accuracy using the idea of Majority voting and segmented-lung test images. We have also used Grad-CAM visualizations to prove the necessity of applying "lung segmentation" to the dataset to have a robust classification model. The codes and weights of the models are available in this Github link.**

***Index Terms*—COVID-19, CNN , Siamese Network, Contrastive learning, Grad-CAM**

I. Introduction

The ongoing pandemic of the COVID-19 virus is posing the most serious healthcare threat to humanity. The World Health Organization (WHO [3]) declares a total of $150,110,310$ confirmed cases of COVID-19, including $3,158,792$ deaths globally as of April 30, 2021. Many tools have been developed to decrease the spread of coronavirus. One effective testing method is *reverse transcription-polymerase chain reaction* (RT-PCR), which is referred to as the "gold standard" of testing. It takes from hours to days for laboratories to process PCR tests. Additionally, the testing procedure is uncomfortable for the patient and dangerous for the operator due to aerosol emission [29]. Another issue is the availability of the testing kits. Health care providers in many developing couturiers have struggled with the lack of effective tools in detecting coronavirus. To address these problems, many researchers have recently attempted to exploit Artificial Intelligent approaches to automate COVID-19 detection [10], [15], [13]. They applied deep learning models on medical images such as Computed tomography (CT) scans and X-ray images of the chest to model the effect of coronavirus on human lungs. These automated approaches can detect the disease in real-time and show the extent of lung involvement. Moreover, it can be used on portable devices; thus, the exposure to healthcare providers and other patients will be minimized.

In this work, we investigate several datasets used for this purpose and select the Kaggle-winning dataset named "COVID-19 Radiography dataset" [11] which consists of Normal, COVID-19, and viral pneumonia chest X-ray (CXR) images. After selecting the dataset, we apply pre-processing methods to the raw images to remove the possible biases such as textual labeling and different scan tool setting of each CXR image. We use a two-step pre-processing, histogram equalization and lung-segmentation [29], to remove mentioned biases. The pre-processing can lower the possibility of learning spurious correlations by neural network models between the images and their assigned labels. After pre-processing the Chest X-Ray images, we take advantage of a decision-fusion-based technique that combines the predicted classes of multiple models to decide the final category of input images. Our model uses four deep neural models. Three of them are Convolutional Neural Networks (CNN), each with different powerful characteristics, and one of them is a Siamese Neural Network (SNN), which uses a few-shot learning approach. CNNs have proven their learning power in computer vision tasks. The Siamese neural network measures the semantic similarity score of a pair of images. In this work, we attempt to use various and powerful deep architectures to design a system that outperforms all baselines in this area. Our last contribution is the use of an Explainable Artificial Intelligence (XAI) technique called Grad-cam. This method helps us to understand the impact of image pre-processing on coronavirus identification. The results show that when we use the raw image in our model, some unnecessary features that may not belong to the

lung regions (such as shoulders) get higher attention at predicting the class label. Since the database has some biases such as various image lightening, textual labels, and a different number of genders in two classes, the model might use the mentioned biases to distinguish the class labels and achieve higher accuracy on the test images when the raw images are fed into the CNN models. However, when the segmented lung images are used, the significant parts of noises and background biases will be removed, resulting in a more realistic and robust model.

II. Related Work

Given the variety of research around COVID-19 detection methods using machine learning, we found several types of research that provide the baseline for our research with similar data used and doing two-class classifications. Alam et al. produced a decision fusion model with histogram oriented gradient and a CNN network that produces a 98.36% accuracy [9]. They used over a thousand images for each category. Goldstein et al. investigated the difference between different CNN networks and achieved the best accuracy of 89.7% with the ResNet50 pre-trained model [13]. They used a relatively balanced dataset with around 1000 images in each category. Panwar et al. developed a model called "nCovidNet," based on the VGG16 network and produced an accuracy of 97.62% [22]. However, their dataset was much smaller than our dataset as they only have around 200 images for each category. Sahlol et al. used the Marine Predator Algorithm to achieve an accuracy of 98.77%. They used an unbalanced dataset with only a few COVID-19 images.

Below, we found some similar studies which are related to our method. Teixeira et al. compared the accuracy of different CNN models with and without lung segmentation [30]. Jadon first used the Siamese Network to evaluate the three-class COVID-19 dataset [18]. However, Jadon did not evaluate the robustness of the model with Grad-CAM or any visualization tools. Our work has combined the Siamese network and three different pre-trained models using the decision fusion approach, which produces a relatively higher accuracy with a large dataset. Also, we use the Grad-CAM to visualize the model to evaluate the system's robustness.

III. Proposed Method

A. Data Collection

After investigating the pros and cons of the datasets, we select the COVID-19 Radiography dataset as it has plenty of high-quality CXR instances for both covid and normal (healthy) cases. This dataset [11], [23], is the winner of the COVID-19 dataset Award, which is awarded by the Kaggle community. The chest X-ray images of the dataset are collected from four major data sources: Italian Society of Medical and Interventional Radiology (SIRM) COVID-19 database [6], Novel Corona Virus 2019 dataset [2], COVID-19 positive chest X-ray images from different articles ([28] and [5]), and COVID-19 Chest imaging at thread reader [4].

The dataset has a total number of 3616 COVID-19 positive CXR images. There are $10,192$ Normal, 6012 Lung Opacity (Non-COVID lung infection), and 1345 Viral Pneumonia images. We apply different pre-processing methods to 3616 COVID and 3616 normal instances of this dataset to generate a balanced set for the rest of this work.

B. Pre-processing

Image pre-processing can be considered an essential step to ensure that the classification model will discriminate different classes without considering biases in the dataset. Image pre-processing contains two main subsections: histogram equalization and Lung segmentation, which are investigated in detail in the following subsections.

*1) **Histogram equalization**:* The purpose of the Histogram equalization is to remove the biases exciting in the dataset. The contrast for X-ray and CT-scan images depends on various reasons, mainly depending on subject contrast or other factors [29]. For instance, in the chosen dataset, "COVID" X-ray images have lower contrast levels than the "Normal" X-ray images. This phenomenon can generate biases in the Neural Network models and make the Neural Network learn the existing bias as distinguishing factors for classifying the dataset.

We have applied Histogram equalization to all images to guarantee a uniform dynamic image to deal with this issue. We have used "Contrast Limited Adaptive Histogram Equalization—CLAHE" [33]. CLAHE method creates non-overlapping sub-images and blocks for a given image, applies Histogram equalization to each block, and finally clips the contrast amplification to a specific value. The clipping of contrast amplification improves the noise reduction on images. There are two hyper-parameters for CLAHE algorithm, which are **tile size** and **clip limit**. Tile size is the neighborhood region's size, and clip limit is the value at which the histogram is clipped. We have found that **tile size=4** and **clip limit=4** give us the best performance according to the lung segmentation.

*2) **Lung segmentation:**:* The lung segmentation for X-ray images can reduce the bias sources such as the texts available on the images or the medical devices

attached to the patients [29]. We have developed a U-net structure for masking out the lung segments [17].

• **U-net architecture:** U-net is a Convolutional neural network developed for biomedical image segmentation, which is fast and relies on a small annotated training dataset [24]. It is widely used in the area of semantic segmentation of cell microscopy images [24], CT-scans [8] and X-ray images [29]. Some other neural networks are proposed [21], [32] based on the U-net.

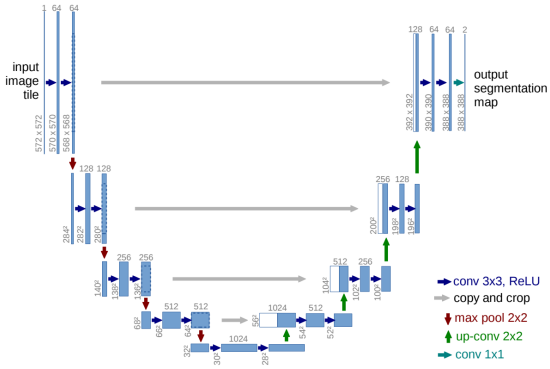

Fig. 1: U-net architecture [24]

The structure of the U-net is shown in Fig. 1. The input of the U-net is a raw image, and the output is a segmentation mask. It consists of two parts: a contracting path (left side) and an expensive path (right side). The contracting path extracts features from the input images as a standard Convolutional neural network. The expensive path halves the feature channels, unsamples the feature maps and concatenates the high-resolution features from the contracting path with the feature maps. The concatenation mitigates the loss of border information due to the down-convolution operation in the contracting path. The output of U-net consists of two channels: background and foreground classes.

• **Training process of U-net and masking out the lungs:**

We have used two datasets, which are publicly available datasets from Montgomery County, Maryland, and Shenzhen Hospital in China [1]. Segmented lung masks are manually labeled for both of these datasets. The overall amount of images is 800, and the whole dataset is randomly divided into train dataset (0.8 of total), validation dataset (0.1 of train dataset), and test dataset. We have applied padding, cropping, and resizing to $512 * 512$ to all of the dataset images. The left and right lung masks have been combined for the images. Fig .2 shows one of the dataset's images with the manually labeled lung segmentation after combing the left and right masks.

As it has been stated, the U-net structure has an encoder and a decoder path. The expansive path that performs the Upsampling of the feature map is combined with high-resolution features from the contracting path to generate the precise lung masks, and this is why we choose the U-net model as our "lung segmentation" model.

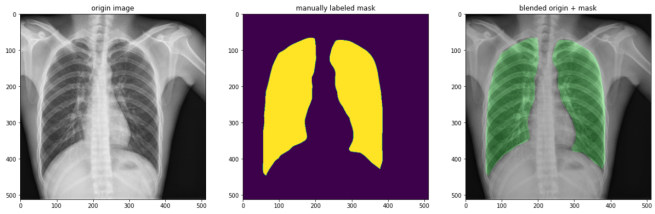

Fig. 2: Left image: Original Image. Middle image: Manually labeled mask. Right image: Combination of mask and original image.

Every image consists of pixels, and we consider the lung mask segmentation as a binary classification for each pixel. As a result, the Softmax function has been applied to each pixel, and the negative log-likelihood loss between the output of Softmax and the labeled mask has been computed as the U-net's loss function.

C. Network Architecture

In this section, we describe the deep architectures that we exploit to address the COVID-19 prediction task. Each deep network is trained separately, as discussed in the following sections. During the testing phase, we employ a decision-fusion-based approach that combines the decision of each model to make a final prediction. The method that the decision fusion model uses for combining the predicted classes of deep networks is called majority voting. The method assigns a final label to an input image based on the majority of the predicted classes. The decision fusion model can lower the misclassification rate of each model and improve the efficiency of the baseline models. Fig. 3 shows the High-level architecture of the proposed method. In what follows, we describe the deep neural networks used in this setting.

*1) **Siamese Neural Network**:* We use deep Siamese Neural Network as one of the approaches for n-shot learning to diagnose COVID-19 cases. The generic high-level architecture of the Siamese network is shown in Fig. 4.

n-shot learning is a sub-field of machine learning which aims to develop models that can be trained with less amount of datasets and provide the required performance. As shown in Fig. 4, a Siamese model

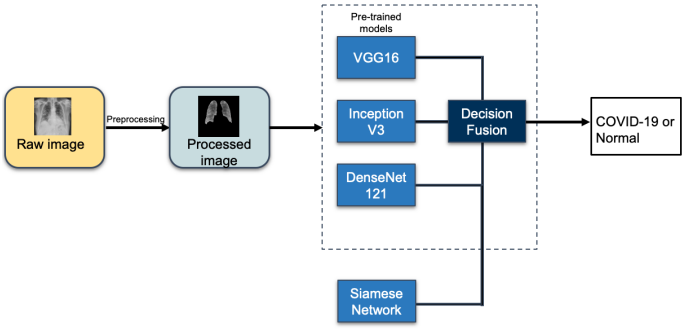

Fig. 3: High-level architecture of proposed decision fusion method.

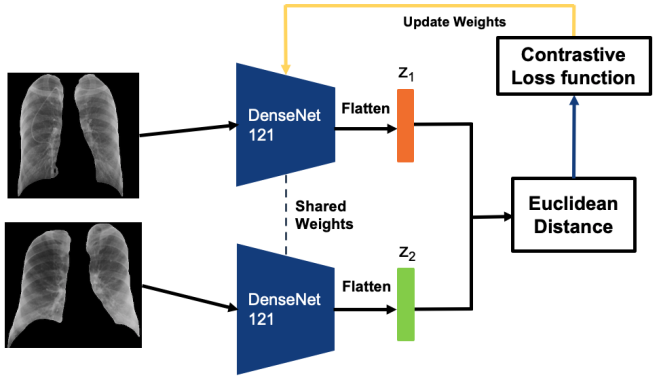

Fig. 4: High-level architecture of Siamese neural network for n-shot COVID-19 classification.

consists of two identical and parallel CNN models that share the same weights. Two different input images are fed into each of the CNN networks to generate various outputs. The goal of the Siamese model is to produce the similarity score for two input images. Ideally, it would be desirable to use a custom CNN model which can achieve the same level of performance in feature encoding as the pre-trained models. However, such a CNN model requires extensive training data to produce rich feature encoding. As our dataset is relatively small, we have utilized the power of a CNN pre-trained model, which in recent times has shown promising results. We use the DenseNet121 model as the underlying network with initial pretraining on ImageNet, as this architecture shows us the best classification accuracy on the chest X-ray images.

Consider we have two input images x_1 and x_2. The first image will be passed through the top CNN feature encoder to generate the output $Z_1 = DenseNet121(x_1)$, which is generated from the last fully connected layer. Similarly, the second image x_2 will be fed into the bottom CNN feature encoder, which is identical to the top one to generate the output $Z_2 = DenseNet121(x_2)$. After that, in the latent space of the feature vectors, we feed Z_1 and Z_2 into the Euclidean function, E, which computes the similarity between two input images. The Euclidean distance can be expressed as follows:

$$E_w = d_w = \|Z_1 - Z_2\|_2 \tag{1}$$

The goal is to make the value for E_w as small as possible for two similar input images and vice versa. In order to achieve this, the value of E_w will be utilized to compute the Contrastive loss function to update the weights in the feature encoding models for improving the extracted vectors. The Contrastive loss is formulated as follows: [20]:

$$Loss = \frac{1-y}{2} * (d_w)^2 + \frac{y}{2} * max(0, m - d_w)^2 \tag{2}$$

The value of y is the actual label, which will be 0 when the two input images are similar and 1 if they are dissimilar, d_w is the Euclidean distance between feature vectors of the input images, and m is the threshold margin. When y equals 0, the loss value will be simplified to the first term only, and d_w will be minimized. On the other hand, if y equals 1, the loss value will be simplified to the second term, and d_w will be maximized to m. Moreover, if the d_w is more than the threshold margin, the second term will be 0.

• **Training Phase of Siamese Network:** We have divided the dataset into training (D_{train}) and test (D_{test}) sets. As a result, the D_{train} contains n sample from each of the two classes and $2 * n$ examples in the training set, in total. At each epoch, a batch of few random samples from a random class are chosen to prepare the support sets. Moreover, another random samples are chosen to create the query sets. After assigning a new label class between each pair of the support set and the query set, the model parameters will be updated to maximize the classification performance on the query images through backpropagation of the calculated Contrastive loss. Since choosing the random support sets and query sets can be any combination of D_{train}, we define an "Epoch Size" variable, which defines the number of choosing the random support sets and query sets at each epoch. The Training phase algorithm for the Siamese network is described in the Algorithm. 1.

• **Testing Phase of Siamese Network:** In the testing phase, for each query of the test dataset D_{test}, we choose 100 random samples of the COVID-19 cases and 100 random samples of the Normal cases from the training dataset D_{train}. Then, we compute the Euclidean distance between each pair of the query test image and the 200 chosen images. Finally, the Mean value of Euclidean

Algorithm 1: Training phase algorithm for n-shot learning

input : Batch size N, margin m, Number of epochs $numEpochs$, "Epoch Size" L, fine-tuned DenseNet121 feature extractor model with parameter θ

output: DenseNet121 feature extractor model with parameters θ

initialization;
for i **to** $numEpochs$ **do**
  for j **to** L **do**
    X_1, X_0=get random batches with size N from D_{train}
    posPairs = getPairsLabel0(X_1, X_0)
    negPairs = getPairsLabel1(X_1, X_0)
    Dis_1=EuclideanDist(posPairs) *Eq. 1*
    Dis_2=EuclideanDist(negPairs) *Eq. 1*
    Loss=Loss(Dis_1, Dis_2, m) *Eq. 2*
    Update model parameters θ with Loss value
  end
end

distances for 100 COVID-19 cases and 100 Normal cases are calculated. The final label of the query test image is the one that has the minimum value of mean Euclidean distance.

*2) **Convolutional Neural Networks**:* For the decision fusion method, we use three Convolutional neural networks. Since our dataset consists of small numbers of X-ray images, we use the transfer learning technique to train each CNN model. It allows us to retrain the final fully-connected layers of a model while freezing all the previous layers. The approach leads to a decrease in the training time and prevents the overfitting problem. The CNNs used in this framework and a summary of their architecture are described in the following paragraphs.

*a) **VGG16**:* VGG16 [26] is a Convolutional neural network that won ILSVR (ImageNet) competition in 2014. It consists of a stack of Convolutional layers (known as feature extractor). Three fully-connected (FC) layers which form the classifier part of the model exist after the feature extractors. The first two feature extractors have 4096 channels each, and the third one performs 1000-way image classification while having 1000 output channels. The network is trained on the ImageNet dataset [12] which contains over one million images from 1000 different categories. To apply transfer learning to this model, we replace the last fully-connected layer with two-channel linear layers that classify the images as either COVID or Normal. Then, we freeze the feature extractor's parameters to retrain only the final FC layer.

*b) **InceptionV3**:* InceptionV3 [27] is another CNN architecture that is more computationally efficient in comparison to VGG16. It was developed by Google in 2015 and trained on the ImageNet dataset. Generally speaking, it consists of a feature extractor (similar to VGG16) that is followed by one fully-connected layer. Similarly, we use transfer learning to retrain the final FC layer and tune the network for our particular task.

*c) **Densenet121**:* Densely-connected Neural networks [14], DenseNets, has increased the depth of the Convolutional neural networks. In a standard CNN, the input images are passed through L layers of the network straightforwardly, as shown in Fig. 5. Each Convolutional layer takes the output of the previous layer and produces a feature map passed to the subsequent layers. In this setting, for L layers, there exist L direct connections.

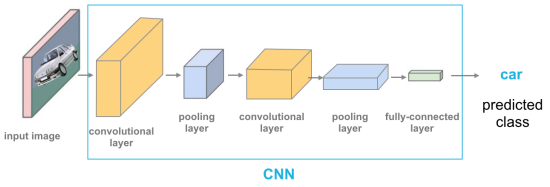

Fig. 5: Standard architecture of Convolutional neural networks

However, a DenseNet architecture consists of many blocks, and in each block, the output of each layer is directly connected to all the subsequent layers. Therefore, for L layers in a block, there are $\frac{L(L+1)}{2}$ connections; hence these layers are densely connected. Fig. 6 illustrates the architecture of DenseNet in one block.

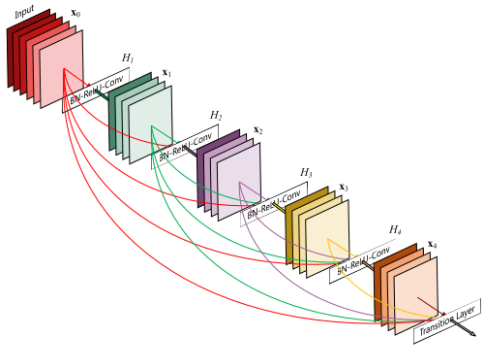

Fig. 6: DenseNet architecture, 1 block

The input of one layer in the concatenation of the output of all its preceding layers. Size of feature maps

for all the layers in one block is the same and it gradually decreases along blocks. The design of DenseNet solves vanishing-gradient problem, strengthens feature propagation, encourages data-reuse, reduces the number of parameters used, and hence presents an extremely powerful learning model. The performance of this model measured in next sections substantiates it's powerful learning ability. To train this network for our COVID-19 detection task, we freeze the convolutional part of the network, modify the last fully-connected layer to output 2 classes, and retrain the modified FC layer.

After training the deep neural models and hyper parameter-tuning them based on their validation performance, we use the decision fusion model, shown in Figure 3 to combine their decisions and make a final prediction on the testing images.

D. Evaluation

This section talks about the methods used for evaluating the lung segmentation model, the CNN model, and the classification results. The evaluation process can be separated into three parts: the evaluation of the lung segmentation model, the evaluation of the binary classification model, and the visual explanation of deep neural networks. The evaluations of the U-net lung segmentation results reveal the differences between the ground truth mask and the generated mask. For the classification model, we need to evaluate the model performance based on the prediction accuracy of each class. Moreover, the visual explanations of the deep neural networks show us how the model is working and extracting the feature to predict the labels. Since we compare the model performance of images with and without the lung segmentation, the visual explanation can help us determine the differences between these two scenarios.

1) ***Lung segmentation model evaluation****:* To evaluate the U-net performance of lung segmentation, we use two various metrics: Dice and Jaccard. Both of these metrics compute the overlap between ground truth and computed mask.

Dice coefficient can be written as follows [17]:

$$DICE = \frac{|S \bigcap GT|}{|S| + |GT|} = \frac{2 * |TP|}{2 * |TP| + |FN| + |FP|} \tag{3}$$

Where GT is the ground truth, and S is the computed mask. Also, TP, FN, and FP are True positive, False Negative, and False Positive results.

Moreover, the Jaccard index is the Intersection Over Union and can be written as follows [16]:

$$J(S, GT) = \frac{|S \bigcap GT|}{|S \bigcup GT|} = \frac{|S \bigcap GT|}{|S| + |GT| - |S \bigcap GT|} \tag{4}$$

True/ Actual	Predicted	
	COVID-pneumonia (C)	Healthy (H)
COVID-pneumonia (C)	P_{CC}	P_{CH}
Healthy (H)	P_{HC}	P_{HH}

TABLE I: Binary model confusion matrix

A model with a higher DICE coefficient and Jaccard index indicates better performance in masking out the lungs.

2) ***Classification model evaluation****:* The model classifies X-ray images into two classes (COVID-19 pneumonia or healthy). Thus, we compute the sensitivity and specificity by generating the confusion matrix. The confusion matrix of the two-class system is illustrated in Table. I. P_{CH} represents the number of COVID-19 pneumonia (C) images incorrectly classified as health cases (H). Similarly, other elements of the matrix are defined. Eq. 5 defines the evaluation metrics.

$$\begin{aligned} Accuracy &= \frac{P_{CC} + P_{HH}}{P_{CC} + P_{HH} + P_{HC} + P_{CH}} \\ Sensitivity &= \frac{P_{CC}}{P_{CC} + P_{CH}} \quad Specificity = \frac{P_{HH}}{P_{HH} + P_{HC}} \\ F_{score} &= \frac{P_{CC}}{P_{CC} + \frac{1}{2}(P_{HC} + P_{CH})} \end{aligned} \tag{5}$$

3) ***Visualization explanation****:* Grad-CAM heatmap [25] is used for the model visualization, which uses the gradient information following into the last Convolution layer of the model to assign different importance values to various parts of images. In Eq. 6 [25], y^c is the prediction score for class c. A^k is the feature map activations of a Convolution layer. α_k^c is the neuron importance weights generated by taking the average of the gradient of a prediction score with respect to the feature map activation of the last Convolution layer over width and height dimensions. Then, the visualization result $L_{\text{Grad-CAM}}$ is generated by a weighted summation of A^k, followed by a ReLU function.

$$\begin{aligned} \alpha_k^c &= \frac{1}{Z} \sum \sum \frac{\partial y^c}{\partial A_{ij}^k} \\ L_{\text{Grad-CAM}}^c &= ReLU(\sum_k \alpha_k^c A^k) \end{aligned} \tag{6}$$

The results can be resized and shown in heatmaps, showing the "attention" level of each part of the image when making the prediction.

IV. Experimental results

A. Results for Lung Segmentation

We have trained the U-net model using the **Adam** optimizer with **learning rate of 0.0005**, **batch size of 4**, and **67 epochs**. We have initialized the weights with the

pre-trained Vgg11 model, as it has a similar structure to the U-net. Moreover, this fine-tuning can help the model to converge faster. Fig. 7 shows three different images of the dataset. The "red area", "green area", and "yellow area" are the predicted mask, ground truth mask, and the intersection area, respectively. Also, the DICE coefficient and Jaccard index for each image have been computed and shown.

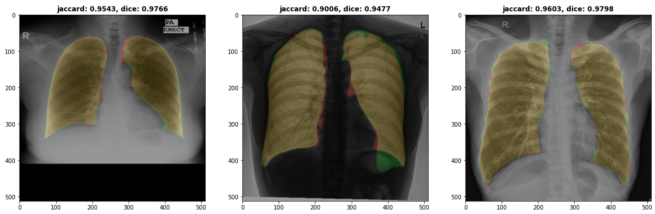

Fig. 7: Obtained results for three different images. Left Image: Jaccard 0.9543, DICE 0.9766. Middle Image: Jaccard 0.9006, DICE 0.9477. Right Image: Jaccard 0.9603, DICE 0.9798.

B. Results for model performance evaluation

We have evaluated the performance of models with lung segmentation and without lung segmentation. Sensitivity, specificity, f-scores, and accuracy are calculated for each model. Table. II presents the four evaluation factors for each pre-trained model, the Siamese network model, and the majority voting models with lung segmentation dataset images. Table. III presents the four evaluation factors for these models without lung segmentation images. For the majority voting models for the lung segmentation section, we evaluated the performance of three combinations of the four models. We found that the combination of VGG16, DenseNet121, and Siamese Network has a significant improvement and generates the best test accuracy.

All four models are trained using **SGD** optimizer with **learning rate of 0.002**, **batch size of 32**, **momentum rate of 0.9**, and **step-size of** 15 for the learning rate scheduler. We have randomly picked 723 images as the test dataset. For the training of InceptionV3, VGG16, and DenseNet121, the **K-fold cross validation** has been applied to the rest of the images to create the validation and training sets. The value of K has been assigned to $k = 4$. For the Siamese Network model, we split the dataset randomly into training and validation sets.

From comparing the Table. II and III, it can be found that models without any lung segmentation perform better than lung segmentation given the selected metrics. However, in the IV-C section, we show that models with lung segmentation are more robust. From our metrics, we also find that the "majority voting" method has higher accuracy than individual models with lung segmentation, achieving 95.57% of accuracy. However, for the model without lung segmentation, the individual model reaches relatively higher accuracy. For example, DenseNet121 without lung segmentation achieves the highest accuracy among all tested models, having an accuracy of 98.89%. Comparing to the baseline models shown in Table. IV, it can be found that for both lung segmented and non-segmented images, our proposed method achieves higher accuracy on the test images. Moreover, our dataset is relatively bigger than the dataset used in the mentioned baselines. There are no prior works on the dataset with similar size of images from our knowledge.

Model	sensitivity	specificity	f-score	ACC
VGG16	93.09%	93.35%	93.22%	93.22%
InceptionV3	90.41%	80.97%	86.01%	85.06%
DenseNet121	95.52%	94.54%	95.05%	94.61%
Siamese Network	97.37%	92.65%	95.02%	94.88%
Decision Fusion (All models)	94.78%	95.54%	95.15%	95.16%
Decision Fusion (VGG16, InceptionV3, DenseNet121)	95.11%	92.00%	93.62%	93.50%
Decision Fusion (VGG16, DenseNet121, Siamese)	**97.13%**	**94.12%**	**95.65%**	**95.57%**

TABLE II: Performance evaluation for data with lung segmentation

Model	sensitivity	specificity	f-score	ACC
VGG16	98.52%	92.49%	95.45%	95.30%
InceptionV3	90.91%	86.65%	88.98%	88.66%
DenseNet121	**99.16%**	**98.63%**	**98.90%**	**98.89%**
Decision Fusion	99.42%	94.99%	97.17%	97.10%

TABLE III: Performance evaluation for data without lung segmentation

C. Results for Visualization Explanation

Fig. 8 shows the Grad-CAM heatmap visualization results of VGG16. The images in the first row show the importance values when the model is trained with the original images. The images in the second row reveal the results with lung segmentation for the same images. Label 0 represents the COVID cases, and Label 1 represents the healthy cases. Fig. 8 shows that the prediction results may rely on some other parts instead of the

Baseline (no lung segmentation)	ACC	Baseline (lung segmentation)	ACC
Emtiaz H [15]	94.2%	Goldstein et. al. [13]	89.70%
Jadon [18]	96.4%	Nur-Alam [7]	93.64%
Manjit K [19]	98.55%	Teixeira [30]	88.00%
Linda W [31]	93.3%	JULIÁN [10]	91.53%

TABLE IV: Baseline Performances for non-segmented (first and second columns) and segmented images (third and fourth columns)

lung regions when the model is trained with the images without lung segmentation. For example, as shown in the fourth figure of Fig. 8, the prediction mainly captures the information on the region of the throat. However, when using the images with lung segmentation, the lung region tends to be more important in making the prediction, as shown in the images on the second row.

This phenomenon indicates that the prediction results based on the raw images are not as robust as the results based on the segmented lung images. Although the accuracy for the data without lung segmentation is higher, the Grad-CAM indicates that the result is not as reliable as the predictions based on the data with lung segmentation.

Fig. 9 shows the Grad-CAM results for InceptionV3 model based on the same cases and Fig. 10 shows the results for Densenet121 network. Localization maps vary among different networks. Nevertheless, when we use the data with lung segmentation, the possibility of making predictions based on lung regions is higher.

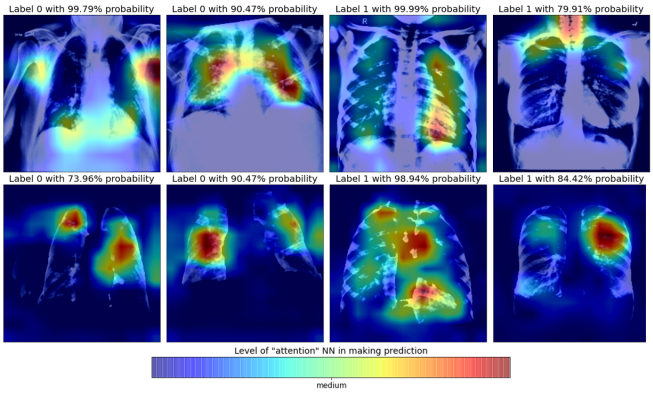

Fig. 8: Grad-CAM visualization result of VGG16

V. Conclusion

Our work has trained and evaluated the performance of VGG16, InceptionV3, and DenseNet121 and the Siamese network on 7200 images in total. The decision fusion method is performed, and it was shown that

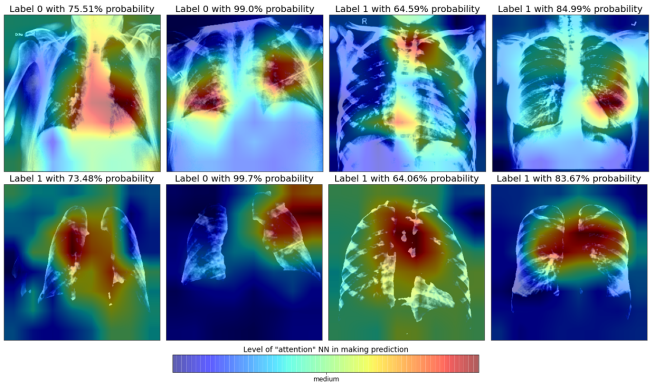

Fig. 9: Grad-CAM visualization result of InceptionV3

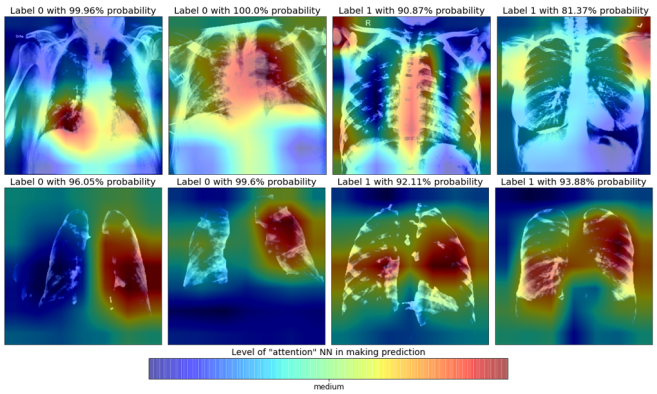

Fig. 10: Grad-CAM visualization result of Densenet121

the combination of VGG16, DenseNet121, and Siamese Network results in the highest accuracy with reliable lung segmentation. We have also compared the model performance using images with and without lung segmentation. Although the models achieve better performance without any lung segmentation, experiments are also done using the visualization tools, Grad-CAM, to show the robustness of the models. We found that models with lung segmentation result in more robustness.

For future works, we would like to increase the size of the images in the dataset to improve testing accuracy. If the large datasets become available, the transfer learning method can be replaced with training deeper CNN models, resulting in more efficient models.

VI. Author Contributions

Farima Fatahi Bayat:

Trained and tuned the Densenet121 model using both raw and processed images; Implemented the decision fusion model (with different variants); Assisted Sara in training Siamese neural network; wrote the final report of the following parts: Introduction, Data collection (III.A), Network Architecture (III.C) [except for the Siamese Neural Network part].

Sara Shoouri:

Wrote the code for histogram equalization, lung segmentation using U-net (worked with Jiahong), K-fold cross validation, Siamese Neural Network, and draft version of Grad-Cam result for InceptionV3 (final version is written by Jiahong for all models). Trained the InceptionV3 model, and the Siamese Network with lung segmentation. Wrote the following parts of the final report: Abstract, Pre-processing of Histogram Equalization (III.B.1), Pre-processing of Training process of U-net and masking out the lungs (III.B.2), Siamese Neural Network (III.C.1), and some parts of Results for model performance evaluation.

Runchu Ma:

Organized the lung segmentation data with and without histogram equalization for pre-processing. Trained and tuned the VGG16 model using both lung segmentation input and without lung segmentation input. Performed calculation for all the performance parameters. Written the related work(II), results for model performance evaluation(IV.B), parts of introduction(I) and conclusion part(V).

Jiahong Xu:

Trained the InceptionV3 model with the dataset without lung segmentation. Generated the Grad-CAM visualization figures for the VGG16, InceptionV3, and Densenet121 model. Tried to apply an open source code of U-net to perform lung segmetion; wrote a daft script for the training process of VGG16. Wrote the final report of the following parts: evaluation (III.D), and results for visualization explanation (IV.C).

REFERENCES

[1] Montgomery County and Shenzhen Hospital. http://openi.nlm.nih.gov/imgs/collections/ChinaSet_AllFiles.zip.

[2] Novel Corona Virus 2019 dataset. https://github.com/ieee8023/covid-chestxray-dataset. [Online].

[3] WHO. https://covid19.who.int/)..

[4] C. imaging. https://threadreaderapp.com/thread/1243928581983670272.html, 2020. [Online].

[5] Radiopedia. https://radiopaedia.org/, 2020. [Online].

[6] SIRM database. https://sirm.org/category/senza-categoria/covid-19/, 2020. [Online].

[7] M. Ahsan, M. Based, J. Haider, M. Kowalski, et al. Covid-19 detection from chest x-ray images using feature fusion and deep learning. *Sensors*, 21(4):1480, 2021.

[8] B. Ait Skourt, A. El Hassani, and A. Majda. Lung ct image segmentation using deep neural networks. *Procedia Computer Science*, 127:109–113, 2018.

[9] N.-A. Alam, M. Ahsan, M. A. Based, J. Haider, and M. Kowalski. Covid-19 detection from chest x-ray images using feature fusion and deep learning. *Sensors*, 21(4), 2021.

[10] J. D. Arias-Londoño, J. A. Gomez-Garcia, L. Moro-Velázquez, and J. I. Godino-Llorente. Artificial intelligence applied to chest x-ray images for the automatic detection of covid-19. a thoughtful evaluation approach. *IEEE Access*, 2020.

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

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[13] E. Goldstein, D. Keidar, D. Yaron, Y. Shachar, A. Blass, L. Charbinsky, I. Aharony, L. Lifshitz, D. Lumelsky, Z. Neeman, et al. Covid-19 classification of x-ray images using deep neural networks. *arXiv preprint arXiv:2010.01362*, 2020.

[14] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks, 2018.

[15] E. Hussain, M. Hasan, M. A. Rahman, I. Lee, T. Tamanna, and M. Z. Parvez. Corodet: A deep learning based classification for covid-19 detection using chest x-ray images. *Chaos, Solitons & Fractals*, 142:110495, 2021.

[16] V. Iglovikov and A. Shvets. Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation. *arXiv preprint arXiv:1801.05746*, 2018.

[17] J. Islam and Y. Zhang. Towards robust lung segmentation in chest radiographs with deep learning. *arXiv preprint arXiv:1811.12638*, 2018.

[18] S. Jadon. Covid-19 detection from scarce chest x-ray image data using few-shot deep learning approach. *Medical Imaging 2021: Imaging Informatics for Healthcare, Research, and Applications*, Feb 2021.

[19] M. Kaur, V. Kumar, V. Yadav, D. Singh, N. Kumar, and N. N. Das. Metaheuristic-based deep covid-19 screening model from chest x-ray images. *Journal of Healthcare Engineering*, 2021, 2021.

[20] M. D. Li, N. T. Arun, M. Gidwani, K. Chang, F. Deng, B. P. Little, D. P. Mendoza, M. Lang, S. I. Lee, A. O'Shea, et al. Automated assessment and tracking of covid-19 pulmonary disease severity on chest radiographs using convolutional siamese neural networks. *Radiology: Artificial Intelligence*, 2(4):e200079, 2020.

[21] X. Li, H. Chen, X. Qi, Q. Dou, C.-W. Fu, and P. A. Heng. H-denseunet: Hybrid densely connected unet for liver and tumor segmentation from ct volumes, 2018.

[22] H. Panwar, P. Gupta, M. K. Siddiqui, R. Morales-Menendez, and V. Singh. Application of deep learning for fast detection of covid-19 in x-rays using ncovnet. *Chaos, Solitons Fractals*, 138:109944, 2020.

[23] T. Rahman, A. Khandakar, Y. Qiblawey, A. Tahir, S. Kiranyaz, S. B. Abul Kashem, M. T. Islam, S. Al Maadeed, S. M. Zughaier, M. S. Khan, and M. E. Chowdhury. Exploring the effect of image enhancement techniques on covid-19 detection using chest x-rays images. *Computers in Biology and Medicine*, page 104319, 2021.

[24] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[25] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision, 2015.

[28] R. M. M. Z. B. M. K. R. I. M. S. K. A. I. N. A.-E. T. R. Muhammad E. H. Chowdhury, A. Khandakar and M. B. I. Reaz. Covid-19 chest x-ray database, 2020. [Online].

[29] E. Tartaglione, C. A. Barbano, C. Berzovini, M. Calandri, and M. Grangetto. Unveiling covid-19 from chest x-ray with deep learning: a hurdles race with small data. *International Journal of Environmental Research and Public Health*, 17(18):6933, 2020.

[30] L. O. Teixeira, R. M. Pereira, D. Bertolini, L. S. Oliveira, L. Nanni, G. D. C. Cavalcanti, and Y. M. G. Costa. Impact of lung segmentation on the diagnosis and explanation of covid-19 in chest x-ray images, 2021.

[31] L. Wang, Z. Q. Lin, and A. Wong. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *Scientific Reports*, 10(1):1–12, 2020.

[32] Z. Zeng, W. Xie, Y. Zhang, and Y. Lu. Ric-unet: An improved neural network based on unet for nuclei segmentation in histology images. *IEEE Access*, 7:21420–21428, 2019.

[33] J. B. Zimmerman, S. M. Pizer, E. V. Staab, J. R. Perry, W. McCartney, and B. C. Brenton. An evaluation of the effectiveness of adaptive histogram equalization for contrast enhancement. *IEEE Transactions on Medical Imaging*, 7(4):304–312, 1988.