CT image segmentation analysis for COVID-19

Chen Zhou[1]

[1]Harold and Inge Marcus department of Industrial Engineering
[1]The Pennsylvania State University,State College,16803

***Abstract*—The novel coronavirus SARS-CoV-2 has posed an unparalleled challenge to various countries and their public health care systems. For COVID -19 diagnosis, computer tomography (CT) has proven to be a vital form of evaluating COVID-patients in terms of progress and thus alleviating the health services' workload. In this paper, we refined the state of the art U-Net and U-Net++ and scrutinize their efficacy for semantic segmentation of pneumonia infected area in the real-world radiological CT data. The dataset includes 929 CT images from patients with confirmed COVID-19 and 929 annotated masks with 4 channels (ground-glass, consolidation, pleural effusions, background). Strong data augmentation and rigorous experimentation are applied to overcome relatively small data size and highly unbalanced data distribution. Dice loss and Focal loss are leveraged as criteria to evaluate the performance of both models. The experimental results indicate that UNet model with 0.01 as learning rate and Adam optimizer along with UNet++ model with 0.001 as learning rate and Adam optimizer reached highest testing accuracy. Therefore, those models are proven to be a promising model for COVID-19 detection and contributes to the development of remedying COVID-19 infection.**

I. Literature Survey

A. Introduction

The coronavirus disease (COVID-19) is an infectious disease caused by the severe acute respiratory syndrome. In December 2019, the COVID-19 was firstly reported in Wuhan city in China and spread rapidly around the world, which has caused inordinate economic impacts on the society for the past year [1] and even threatens people's lives.

Several symptoms are accompanied by COVID-19 including fever shortness of-breath, dry cough and loss of the sense of smell. Further researches also show that except for respiratory systems, the COVID-19 would potentially affect multiple organs in the human bodies including livers, lungs, heart, kidneys and brains. [2][3] For patients with confirmed COVID-19, lungs become the most potentially affected areas in human bodies since the receptor which makes the virus enters the cell by binding to angiotensin-converting enzyme 2 or ACE2 can be found on alveoli. [4] Besides, although most routine lab values of COVID-19 patients could be normal, their unenhanced chest CT images could reveal peripheral, multilobar areas of ground-glass opacity. This sign gives computer tomography (CT) a promising future and makes CT image analysis a motivated task on the diagnosis of COVID-19 pneumonia and its symptoms.

CT image analysis can be briefly interpreted as two missions. The first mission is to identify the positive cases and negative cases in terms of COVID-19 with patients' CT scans. That's technically a binary classification problem. The second mission involves the pixel-level classification which indicates the symptomatic areas of CT images after the positive cases are classified. Analyzing images at the pixel level can be addressed as a semantic segmentation problem. While object detection is popular and powerful around deep learning techniques on computer vision, it shows its limitation on medical image analysis, especially for COVID-19 cases.

B. Motivation

Though multiple tests for COVID-19 are given recently, the accuracy of tests remains a challenging problem. [5] The accuracy depends highly on which test is used, the type of specimen tested, how it was collected and the duration of illness. Lots of factors would eventually affect the final testing results. In this contest, the CT scan could be leveraged as an efficient alternative or auxiliary tool for confirming and controlling the COVID-19 diseases. However, as the confirmed cases increase at an incredible speed at the global scale, it requires lots of effort and money to manually review the data, which is the CT scan images. Besides, it's painstaking for highly-trained specialists to analyze these images where the dominated nonexistence of COVID-19 cases makes the work monotonous and tedious. Therefore, automating this process becomes crucial which not only liberates the expertise from monotonous work but to some extent increases the accuracy of diagnosis. Recently, the world has witnessed the rocket development of artificial intelligence [6] and deep learning technology also has been widely utilized in medical image processes and analysis thanks for its powerful feature extraction. A lot of techniques have been published for analyzing COVID-19 CT images. Dilbag et al. [7] presented a classification model with CNN to automatedly analyze the CT images from patients with COVID-19. The initial model was tuned and multi-objective differential evolution (MODE) is applied to achieve high performance. Amine et al. [8] developed a multitask deep learning model to detect COVID-19 lesions on CT scans and improved state-of-the-art U-Net by leveraging useful information contained in multiple related tasks. Gozes et al. [9] proposed a complicated system where 2D and 3D deep learning models are modified and applied for clinal purposes. Tao et al. [10] utilized the transfer learning technique with AlexNet, GoogleNet, and ResNet backbone to detect the COVID-19 with higher speed. Abbasian et al. [11] also leveraged the transfer learning and evaluated 10 well-known convolutional neural networks to find the highest sensitivity model for characterizing COVID-19 infections. Emitaz et al. [12] introduced a 22-lay CNN architecture and had reached an accuracy of 99.1% for binary classification in COVID-19 detection tasks. A multistage approach including segmentation and classification between COVID-19 infection has been proposed by Ophir et al. [13] U-Net was leveraged to remove the image portion that is not relevant for the detection and Resnet-50 network is modified and applied to deal with the COVID-19 classification problems. The detection of infectious lung areas has been achieved by Shaoping et al. [14] with the VGG model. The model they presented shows excellent performance on categorizing COVID-19 cases from community-acquired pneumonia (CAP) and non-pneumonia.

To summarize, these publications mainly focus on image classification while only a few tasks have explored the segmentation tasks involving indicating symptomatic areas of COVID-19 chest CT images.

II. Problem Definition

In this paper, we aim to detect the symptomatic areas in lung CT images at the pixel level. It involves with semantic segmentation model that produces a heatmap rather than a single classification. With that heatmap, we can locate the infected areas and classify COVID-19 cases.

Figure 1 shows the chest CT scans of patients with COVID-19, which includes ground-glass opacities (GGOS), area of consolidation, pleural effusions and background. While a simple binary classification can tell us whether the CT image is from a patient with COVID-19, we aim to find the specific location of the symptomatic area. To be more specific, our goal is to take the CT image with HU color ($1 \times 512 \times 512$) to generate a segmentation map where each pixel contains a class label represented as an integer ($1 \times 512 \times 512$).

Figure 2 represents a brief structure of the segmentation model with a U-shape. Multiple layers of convolution and downsampling indicate that the model starts at consuming raw pixels to produce specific, detailed detectors for things like texture, and then builds up higher-level conceptual feature detectors. Since the segmentation task needs to produce the masks with the same size of initial images, a U-shape model including encoder and decoder is utilized to upscale the convolutions and get back to the full resolution thus dramatically reducing the computational cost. Meanwhile, skip connections are leveraged to short-circuit inputs along the downsampling path into the corresponding layers in the upsampling path and therefore solve the problem of network converging.

UNet and UNet++ are both U-shape structure networks and have achieved great feats in the area of medical image processing and analyzing. We utilize and refine both UNet structure and UNet++ structure in this study to construct the segmentation model for COVID-19 infection-induced symptoms in lung areas with input as CT scans. Data augmentation technique is proposed in this work to handle the highly imbalanced dataset. A thorough experiment is conducted to find the optimized criterion and training parameters of the segmentation model.

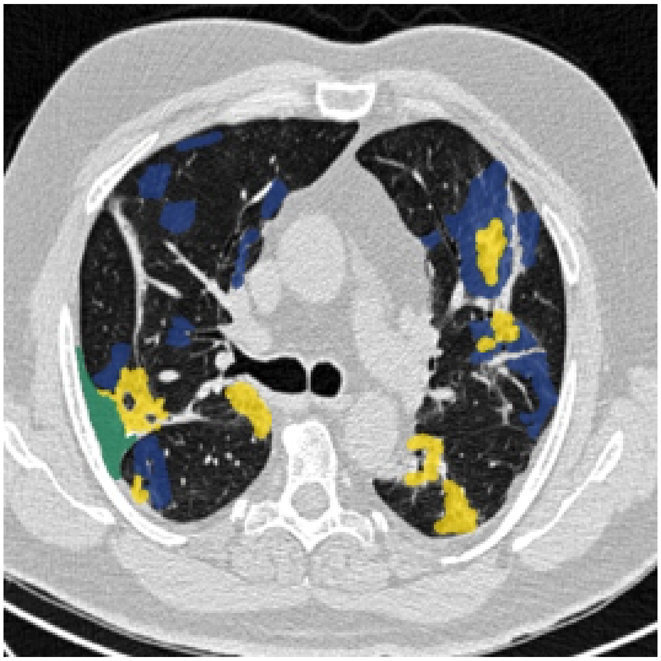


Figure 1. Annotated CT scan of a patient with laboratory proven COVID-19 pneumonia from the Italian Society of Medical and Intervention Radiology (SIRM). The RBG-values were “marked” into the image using a corresponding numbered mask label representing ground-glass opacities (GGOS) in blue, area of consolidation in yellow, pleural effusion in green and background.

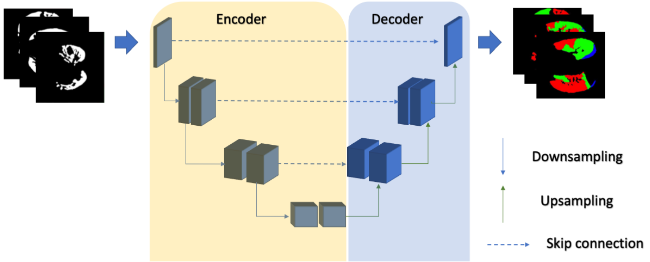


Figure 2. The structure of segmentation model. CT images ($1 \times 512 \times 512$) are firstly processed by repeated layers of convolution and downsampling and end up at the masks with same size of initial images ($1 \times 512 \times 512$).

III. Data Description

The dataset is provided by medicalsegmentation.com and consists of two parts. The first part is the COVID-19 radiological images collected in Italian Society of Medical and Interventional Radiology. the collection contains about 60 cases with example CXRs and single slice CT-images. The second part is 9 segmented volumetric CTs from Radiopaedia. This dataset includes whole volumes and both positive and negative slices (373 out of the total of 829 slices have been evaluated by a radiologist as positive and segmented).

Both of the datasets are annotated by annotators with deep radiology background and proficient annotating skills for accurate annotated labels. The annotated masks contain 4 labels including ground-glass opacities (GGOS), consolidation areas, pleural effusions and background. Figure 3 shows 5 sample CT images and their corresponding annotated labels from the dataset.

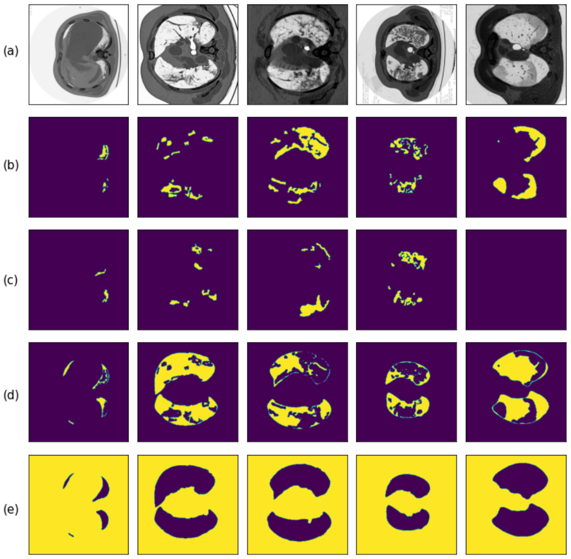


Figure 3. Five random CT images and their annotated labels from the dataset. (a) represents the area of GGOS. (b) represents the area of consolidations. (c) represents the area of pleural effusions. (d) represents the background of the lung.

Both of the datasets have been transferred from the original format of DICOM to JPG. Even though the upper intensity-range was lost during cropping and resizing the images, the datasets contain enough relevant information which is normalized and stored in the original Hounsfield Unit scale (HU). The value of HU ranges from -1024 to around 2000 and anything above 400 is nothing but simply bones with different radiodensity. Therefore, we simply set the threshold ranged from -1000 to 400 while preprocessing the data. The density of the HU value before and after preprocessing is represented in Figure 4. The plot shows the highly disordered HU value distribution, which indicates the highly imbalanced dataset we are dealing with.

IV. Model discussion

This section gives a specific discussion of proposed approaches including data normalization and augmentation model, segmentation model and loss function for highly imbalanced CT image dataset.

A. *Data normalization and augmentation*

Given the fact that full whitening of each layer's inputs is costly differentiable, mini-batch normalization is applied in this work. The advantage of mini-batch comparing with simply normalization is that the transform inserted in the network can represent identity information. The function of simply normalization for each layer with d-dimensional input $x = (x^{(1)} \cdots x^{(d)})$ is defined as:

$$\hat{x}^{(k)} = \frac{\hat{x}^{(k)} - E\left[x^{(k)}\right]}{\sqrt{Var\left[x^{(k)}\right]}} \tag{1}$$

Mini-batch normalization introduces a pair of parameters $\gamma^{(k)}$,$\beta^{(k)}$ for each activation $x^{(k)}$ that can be learned along the original model parameters thus enhancing the presentation power of network. Then the original normalized value is scaled and shifted as:

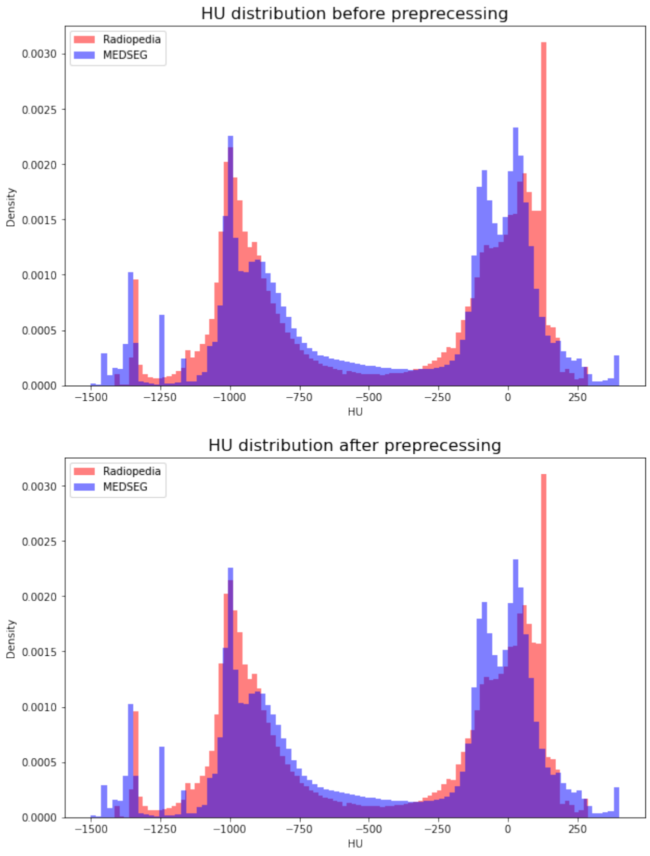


Figure 4. The distribution of HU value before and after preprocessing.

$$y^{(k)} = \gamma^{(k)} \hat{x}^{(k)} + \beta^{(k)} \tag{2}$$

Consider a mini-batch $\mathcal{B} = x_{1 \cdots m}$,the transform is referred as:

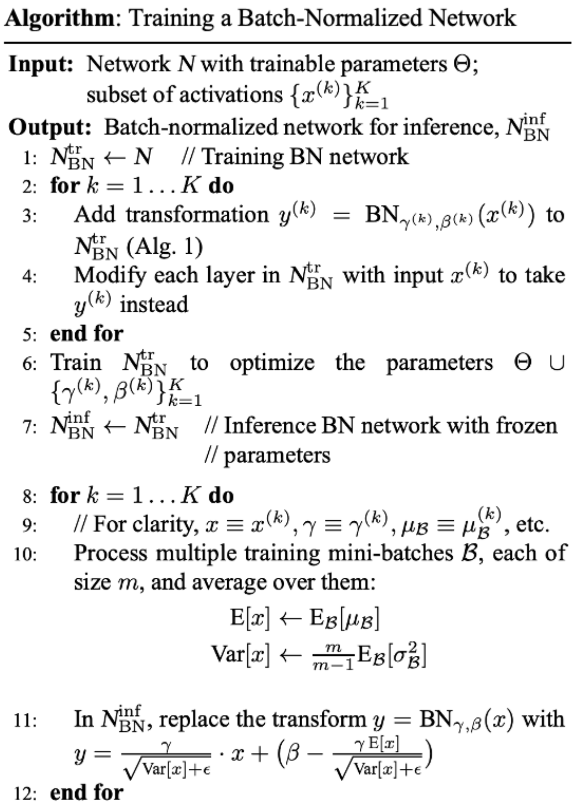
$$BN_{\gamma,\beta} : x_{1 \cdots m} \to y_{1 \cdots m} \tag{3}$$

And the BN transform is defined as:

$$BN_{\gamma,\beta}(x_i) \equiv \gamma \frac{x_i - \mu_\beta}{\sqrt{\sigma_\beta^2 + \epsilon}} + \beta \tag{4}$$

where μ_β is the mini-batch mean and σ_β^2 is the mini-batch variance. In this formulation, ϵ is an added constant to the mini-batch variance for numerical stability. The overall procedure for training batch-normalized networks is summarized in Figure 5.

In addition to normalization, we augment the data by applying synthetic alterations to individual images. By doing this, the effective size of new images is enlarged resulting in the model being forced to increase the ability of generalization. Therefore, the potential overfitting issue could be compromised even with the limited dataset. The affine transformation matrix [16] is adopted in this study to implement four specific types of augmentation including reflecting, shifting, scaling and rotating. I shows the transformation matrix for these



```
Algorithm: Training a Batch-Normalized Network
Input: Network $N$ with trainable parameters $\Theta$;
       subset of activations $\{x^{(k)}\}_{k=1}^{K}$
Output: Batch-normalized network for inference, $N_{\mathrm{BN}}^{\mathrm{inf}}$
$N_{\mathrm{BN}}^{\mathrm{tr}} \leftarrow N$    // Training BN network
for $k = 1 \ldots K$ do
    Add transformation $y^{(k)} = \mathrm{BN}_{\gamma^{(k)},\beta^{(k)}}(x^{(k)})$ to $N_{\mathrm{BN}}^{\mathrm{tr}}$ (Alg. 1)
    Modify each layer in $N_{\mathrm{BN}}^{\mathrm{tr}}$ with input $x^{(k)}$ to take $y^{(k)}$ instead
end for
Train $N_{\mathrm{BN}}^{\mathrm{tr}}$ to optimize the parameters $\Theta \cup \{\gamma^{(k)}, \beta^{(k)}\}_{k=1}^{K}$
$N_{\mathrm{BN}}^{\mathrm{inf}} \leftarrow N_{\mathrm{BN}}^{\mathrm{tr}}$    // Inference BN network with frozen
                                   // parameters
for $k = 1 \ldots K$ do
    // For clarity, $x \equiv x^{(k)}, \gamma \equiv \gamma^{(k)}, \mu_{\mathcal{B}} \equiv \mu_{\mathcal{B}}^{(k)}$, etc.
    Process multiple training mini-batches $\mathcal{B}$, each of size $m$, and average over them:
        $\mathrm{E}[x] \leftarrow \mathrm{E}_{\mathcal{B}}[\mu_{\mathcal{B}}]$
        $\mathrm{Var}[x] \leftarrow \frac{m}{m-1}\mathrm{E}_{\mathcal{B}}[\sigma_{\mathcal{B}}^2]$
    In $N_{\mathrm{BN}}^{\mathrm{inf}}$, replace the transform $y = \mathrm{BN}_{\gamma,\beta}(x)$ with
        $y = \frac{\gamma}{\sqrt{\mathrm{Var}[x]+\epsilon}} \cdot x + \left(\beta - \frac{\gamma\,\mathrm{E}[x]}{\sqrt{\mathrm{Var}[x]+\epsilon}}\right)$
end for
```

Figure 5. The procedure for training batch-normalized networks. [15]

four types of augmentation. z is 0,1 value that specifies the reflecting dimension. t_x specifies the displacement along the x axis, t_y specifies the displacement along the y axis. ss_x specifies the scale factor along the x axis, s_y specifies the scale factor along the y axis. q specifies the angle of rotation.

Table I
TRANSFORMATION MATRIX FOR AUGMENTATION

Affine Transform	Transformation matrix
Reflect	$\begin{bmatrix} z & 0 & 0 \\ 0 & -z & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Shift	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$
Scale	$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Rotate	$\begin{bmatrix} \cos(q) & \sin(q) & 0 \\ -\sin(q) & cos(q) & 0 \\ 0 & 0 & 1 \end{bmatrix}$

B. Segmentation model

Two state-of-the-art segmentation networks (UNet and UNet++) in biomedical image diagnosis are leveraged and implemented in this work. UNet is proposed by Olaf Ronneberger et al. [17] in 2015. The U-Net architecture is shown in Figure 6.

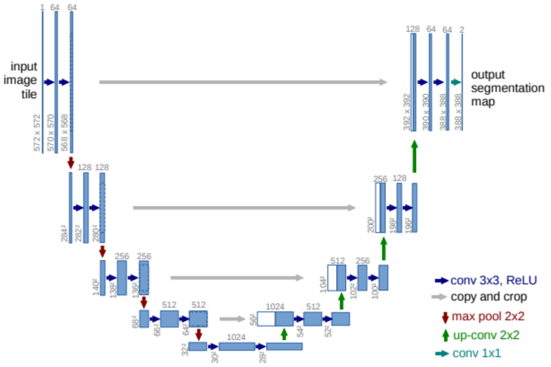


Figure 6. The architecture of UNet. The boxes represent intermediate results and the arrows represent operations between them. In the top is the full resolution (512 × 512), the row below has half resolution, and so on. [17]

The U-shape architecture of UNet consists of contraction path (downsampling), expansion path (up-sampling) and final layers. The contraction path is similar to a typical CNN architecture that consecutively stacks 3×3 convolutions followed by 2×2 max pooling to capture features of the images. Two 3×3 convolutions and a 2×2 up-convolution are followed to upscale the image back to full resolution and enable precise localization. After every 2×2 up-convolution, a concatenation of feature maps is provided with a corresponding layer from the contraction path. These layers receive both the unsampled results of the wide receptive filed layers as well as the output of the earlier fine detail layers by the “copy and crop” bridge connections. As such, even though information reaches a large number of downscaled images, these dropout layers still can provide localization information from contraction path to expansion path. The final detail layers contain both information from the near layers and the first set of full-resolution layers. All the layers enable the segmentation network to produce the output of images that have the same size as the input image and contains localized information represented by a probability at the pixel level.

UNet++ is a continuation of UNet architecture which as well achieves fruitful work in biomedical image analysis. Figure 7 is the illustration of UNet++ architecture.

Compared with UNet architecture, re-designed ship pathways have been added to reduce the semantic gap between encoder and decoder subnetworks. It results in the higher similarity between the semantic level of the encoded feature and that of feature maps waiting in the decoder thus achieving optimization more efficiently. Also, Dense skip connections are used to ensure that all prior feature maps are accumulated and arrive at the current node which results in generating full resolution feature maps at multiple sematic levels. Last, deep supervision is added to make model pruning feasible. With deep supervision, the model complexity can be flexible, which gives more options between model training speed and its performance.

Several adjustments in terms of UNet and UNet++ architectures are adopted in this work for better data representation and model performance. First, batch normalization is passed through the networks as discussed above. Second, since the

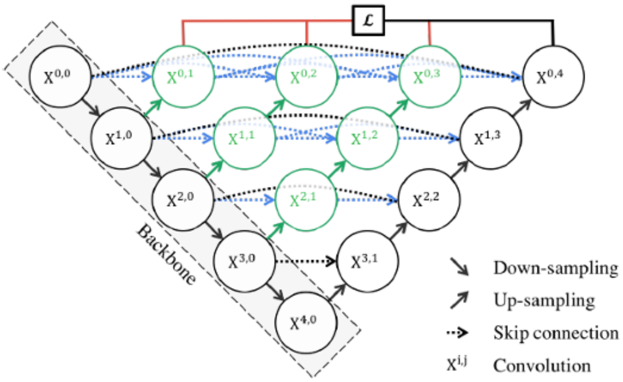


Figure 7. The architecture of Unet ++. It starts with an encoder sub-network or backbone followed by a decoder sub-network. The green and blue arrows represent the re-designed skip pathways that connect the two sub-networks and the use of deep supervision which is represented as the red line. [18]

output values of UNet are unconstrained, an activation layer with sigmoid function is added to restrict the output to the range [0, 1]. Besides, considering the ability of these two models with standard parameters is far beyond the requirement of the size of our dataset, we slightly reduce the depth and number of filters to improve the training efficiency. Finally, the annotated images in our dataset have the same size as the original CT images and contain 4 layers which represent 4 different labels of lung CT scans. Therefore, the convolution layers are adjusted and padded layers are added to meet the output requirements. Our ultimate output is 4 channels, with each pixel of each channel representing the model's estimation of the probability of its corresponding labels.

C. Loss function

The loss functions used in this work are Focal loss [19] and Sørensen–Dice coefficient also known as dice coefficient [20] to handle the imbalance problem of our dataset. Focal loss is also seen as a derivation of Binary Cross-Entropy [21] for highly imbalanced class scenarios. The pixel-wise cross entropy loss is the most commonly used loss function for image classification and also works well for image segmentation. Binary Cross-Entropy is defined as:

$$L_{BCE}(y,\ \hat{y}) = -(ylog(\ \hat{y}) + (1-y)log(1-\ \hat{y})) \tag{5}$$

where $\hat{y}$ is the predicted value by the prediction model.

However, the unbalanced representation in the CT images will make the training dominated by the most prevalent class since the cross-entropy evaluates the class prediction at the pixel level and then averages over all pixels. Therefore, Focal loss is proposed to down-weight positive examples like background and focus training on negative examples thus compromising the problem of class imbalance. The Focal loss is defined as:

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}log(p_t) \tag{6}$$

where $\gamma > 0$, $\alpha_t \in [0,\ 1]$, and p_t is defined as:

$$Fp_t = p \quad if y = 1 \quad otherwise \quad 1-p \tag{7}$$

In this formulation, $(1-p_t)^{\gamma}$ balances the importance of positive/negative examples and α-balanced variant is adopted for higher accuracy.

Since the CT images of lungs are highly unbalanced as shown in Figure 1, both weighted Cross-Entropy loss and Focal loss are leveraged in the segmentation model.

Another loss function applied for this task is the Dice coefficient, which essentially measures the overlap areas between two samples. The Dice coefficient is originally defined as:

$$DICE = \frac{2\,|A \cap B|}{|A| + |B|} \tag{8}$$

where $|A \cap B|$ represents the common elements between sets A and B, and $|A|$ represents the number of elements in set A (and likewise for set B). Also, the Dice coefficient is defined as:

$$DL\left(p_i, g_i\right) = 1 - \frac{2\sum_i^N p_i g_i + \lambda}{2\sum_i^N p_i^2 + 2\sum_i^N g_i^2 + \lambda} \tag{9}$$

where p_i and g_i represent pairs of corresponding pixel values of prediction and ground truth, respectively. λ is added in numerator and denominator to ensure that the function is not undefined in edge case scenarios such as when $p_i = g_i = 0 \ \forall\ i \in N$.

Since Dice coefficient measures the overlap between the prediction and the ground truth, it makes Dice coefficient a valid criterion in segmentation tasks to handle imbalanced data without adding weight.

V. Analysis

Figure 8 illustrates the pipeline of training and test procedure. The dataset is separated into three parts with the purpose of training, validation and testing. To demonstrate more details, 4 layers contained in the masks representing different information leads to the multi-class segmentation problem. Therefore, the output channels produced by segmentation networks are in the form of one-hot encoding tensors with elements representing the possibility of corresponding classes. The multi-class segmentation algorithm is shown in Figure 9. The batch training loss between output and batch of training masks is computed by loss function and backpropagated to the segmentation model to adjust the parameters for each layer. Validation is applied in the procedure to evaluate the performance of the model during the training procedure and testing dataset is used to scrutinize the learning ability of the final model.

Also, both UNet and UNet++ training is assigned with various learning rate and optimizers to find the optimized segmentation model for our dataset. Since the constrained time and heavy learning load for this task, we only choose three levels of learning rate which is 0.01, 0.001 and 0.0001. As for learning optimizers, Adam and Stochastic gradient descent (SGD) are applied since both perform well with respect to the speed of convergence and generation in complicated networks.

VI. Results and conclusions

All the models were implemented in Python with Pytorch libraries. Full working code samples and results can

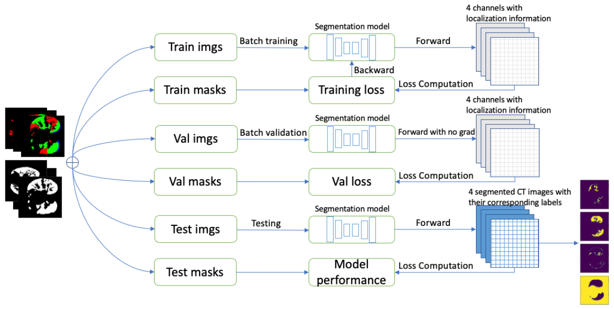
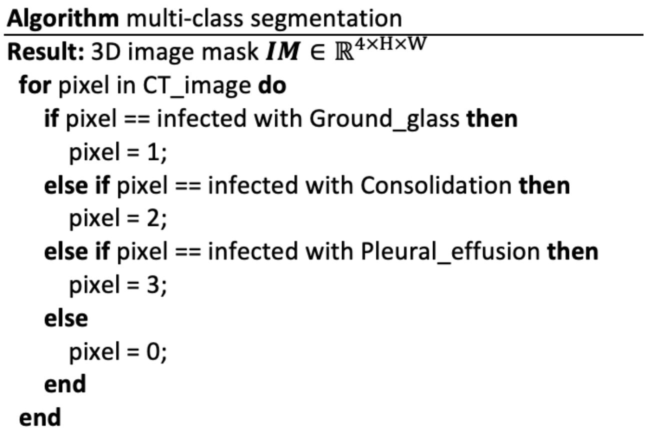


Figure 8. Segmentation model training and testing procedure. The dataset is split into 3 parts for training, validation and testing. The segmentation model in the chart represents the identical segmentation model updated by backward and tested by testing dataset.



```
Algorithm multi-class segmentation
Result: 3D image mask $IM \in \mathbb{R}^{4\times H\times W}$
 for pixel in CT_image do
   if pixel == infected with Ground_glass then
     pixel = 1;
   else if pixel == infected with Consolidation then
     pixel = 2;
   else if pixel == infected with Pleural_effusion then
     pixel = 3;
   else
     pixel = 0;
   end
 end
```

Figure 9. The procedure of multi-class segmentation algorithm.

be found on GitHub (https://github.com/zc534012448/covid-sgementation.git). For network training, all the models are trained with random initial parameters using an NVIDIA Tesla P0 GPU with 32GB RAM provided by Google Colab server. The initial learning rate is 0.01 and then decays to 0.001 and 0.0001. Adam and SGD are used as optimizers in this training. The proposed model will perform the COVID-19 CT segmentation tasks with the CT images as input. An experiment is conducted to evaluate the performance of each training model and find the optimized one.

Figure 10 and Figure 11 demonstrate the experiment results for the UNet model when using Dice loss and Focal loss as criteria. The results with learning rate at 0.001 and Adam as optimizer are not yet converged due to the limitation of iterations and thus are excluded in this experiment. The results indicate the UNet model reaches the lowest training loss and validation loss when setting 0.001 as learning and optimizer as Adam with respect to Dice Loss. The Focal loss plot shows similar results when selecting Focal loss is applied. Therefore, for training and validation, the UNet model reaches the optimized performance with the learning rate at 0.001 and Adam as the optimizer.

The experiment results for the UNet++ model are shown in Figure 12 and Figure 13.

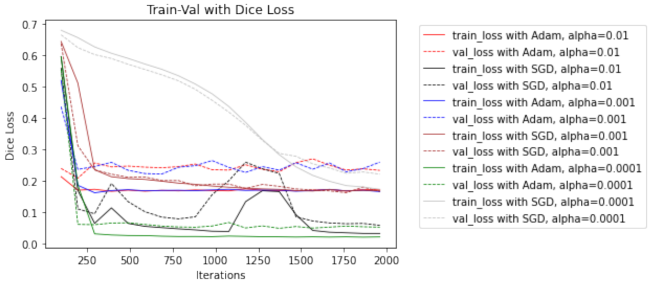


Figure 10. The Dice loss of training and validation for UNet. Each line represents the learning or training curve with specific learning rate and optimizer.

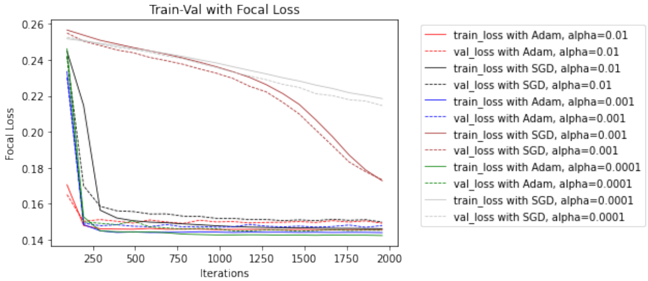


Figure 11. The Focal loss of training and validation for UNet. Each line represents the learning or training curve with specific learning rate and optimizer.

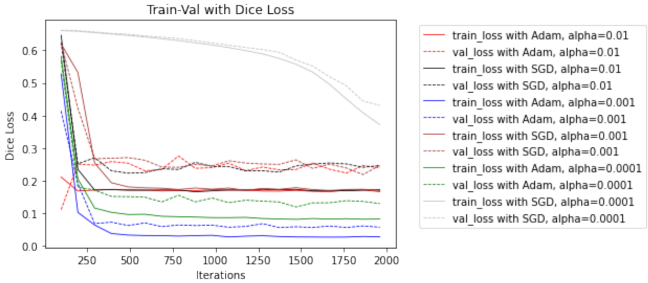


Figure 12. The Dice loss of training and validation for UNet++. Each line represents the learning or training curve with specific learning rate and optimizer.

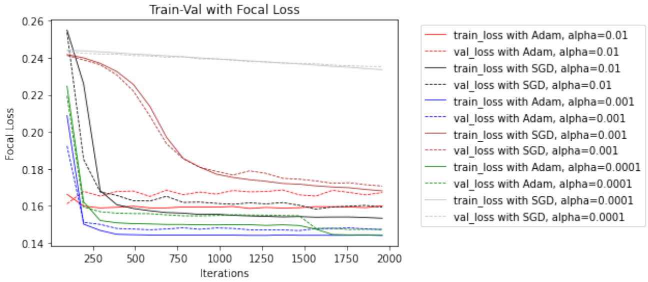


Figure 13. The Focal loss of training and validation for UNet+. Each line represents the learning or training curve with specific learning rate and optimizer.

Like the UNet results, the training loss and validation loss are not yet converged with learning rate at 0.001 and Adam as optimizer due to the limitation of iterations. The highest performance is achieved by the UNet++ model with 0.001 as learning rate and Adam with optimizer. To summarize, the results above firmly demonstrate that both UNet and UNet++ models have superior performance on COVID-19 CT image segmentation tasks with considerably low training loss and validation loss. The testing accuracy of both UNet and UNet++ are shown in II and III, which clearly demonstrate that UNet model with 0.01 as learning rate and Adam optimizer along with UNet++ model with 0.001 as learning rate and Adam optimizer reached highest testing accuracy. These models are testified to be promising models for COVID-19 detection tasks.

Table II
PERFORMANCE SCORES OF UNET

Dice accuracy	Focal accuracy	Learning rate	Optimizer
0.9273	0.8537	0.0100	Adam
0.9513	0.8537	0.0100	SGD
0.9176	0.8549	0.0010	Adam
0.7821	0.8332	0.0010	SGD
0.9516	0.8572	0.0001	Adam
0.8910	0.7883	0.0001	SGD

Table III
PERFORMANCE SCORES OF UNET++

Dice accuracy	Focal accuracy	Learning rate	Optimizer
0.9285	0.8512	0.0100	Adam
0.9139	0.8533	0.0100	SGD
0.9511	0.8541	0.0010	Adam
0.9159	0.8328	0.0010	SGD
0.9456	0.8540	0.0001	Adam
0.6987	0.7686	0.0001	SGD

VII. FUTURE WORK

While lots of efforts were made to accomplish this work, the models are very preliminary segmentation models for medical purposes and the accuracy of output is still not good enough to be utilized clinically. The fact that deep learning is powerful but still limited is acknowledged so that any blind application of deep learning will finally mislead people. As such, more disciplined experimentation and refinement of the model will be considered in future work to make it available in the real-world analysis. In addition, the semantic segmentation model also has its limitations in the form of longer training time, high demand for hardware access and relatively low detection accuracy. Our future efforts will cover these issues and make artificial intelligence a powerful tool for medical image analysis.

VIII. DIFFICULTIES

The most challenging part of this project is that it deals with the highly imbalanced dataset. To compromise this problem, we deliberately selected Dice loss and Focal loss as criteria for model evaluation. Data augmentation is utilized to add slightly modified copies of existing data to overcome the small data problem. And simple normalization is replaced by mini-bath normalization in this work to better interpret the features of data when training the model. Bedside, although this work is completed based on other outstanding projects in GitHub especially for the implementation of UNet and UNet++, we slightly adjusted the previous network structures so that the models can produce expected results. Finally, conducting an experiment to find optimized solutions is extremely time-consuming because of the complicated model and long training time. Thus, we only set 6 treatments for each network with each loss function.

REFERENCES

[1] N. Akbulaev, I. Mammadov, and V. Aliyev, “Economic impact of covid-19,” *Sylwan*, vol. 164, pp. 113–126, 05 2020.

[2] V. O. Puntmann, M. L. Carerj, I. Wieters, M. Fahim, C. Arendt, J. Hoffmann, A. Shchendrygina, F. Escher, M. Vasa-Nicotera, A. M. Zeiher, M. Vehreschild, and E. Nagel, “Outcomes of Cardiovascular Magnetic Resonance Imaging in Patients Recently Recovered From Coronavirus Disease 2019 (COVID-19),” *JAMA Cardiology*, vol. 5, pp. 1265–1273, 11 2020.

[3] C. W. Yancy and G. C. Fonarow, “Coronavirus Disease 2019 (COVID-19) and the Heart—Is Heart Failure the Next Chapter?,” *JAMA Cardiology*, vol. 5, pp. 1216–1217, 11 2020.

[4] T. Bombardini and E. Picano, “Angiotensin-converting enzyme 2 as the molecular bridge between epidemiologic and clinical features of covid-19,” *The Canadian journal of cardiology*, vol. 36, p. 784.e1—784.e2, May 2020.

[5] J. Watson, P. F. Whiting, and J. E. Brush, “Interpreting a covid-19 test result,” *BMJ*, vol. 369, 2020.

[6] Q. Yan, D. Gong, and Y. Zhang, “Two-stream convolutional networks for blind image quality assessment,” *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2200–2211, 2019.

[7] D. Singh, V. Chahar, V. Yadav, and M. Kaur, “Classification of covid-19 patients from chest ct images using multi-objective differential evolution-based convolutional neural networks,” *European journal of clinical microbiology infectious diseases : official publication of the European Society of Clinical Microbiology*, vol. 39, 07 2020.

[8] A. Amyar, R. Modzelewski, H. Li, and S. Ruan, “Multi-task deep learning based ct imaging analysis for covid-19 pneumonia: Classification and segmentation,” *Computers in Biology and Medicine*, vol. 126, p. 104037, 2020.

[9] O. Gozes, M. Frid-Adar, H. Greenspan, P. D. Browning, H. Zhang, W. Ji, A. Bernheim, and E. Siegel, “Rapid ai development cycle for the coronavirus (covid-19) pandemic: Initial results for automated detection patient monitoring using deep learning ct image analysis,” 2020.

[10] T. Zhou, H. Lu, Z. Yang, S. Qiu, B. Huo, and Y. Dong, “The ensemble deep learning model for novel covid-19 on ct images,” *Applied Soft Computing*, p. 106885, 2020.

[11] A. A. Ardakani, A. R. Kanafi, U. R. Acharya, N. Khadem, and A. Mohammadi, “Application of deep learning technique to manage covid-19 in routine clinical practice using ct images: Results of 10 convolutional neural networks,” *Computers in Biology and Medicine*, vol. 121, p. 103795, 2020.

[12] 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*, p. 110495, 2020.

[13] O. Gozes, M. Frid-Adar, H. Greenspan, P. D. Browning, H. Zhang, W. Ji, A. Bernheim, and E. Siegel, “Rapid ai development cycle for the coronavirus (covid-19) pandemic: Initial results for automated detection patient monitoring using deep learning ct image analysis,” 2020.

[14] A. Mohammed, C. Wang, M. Zhao, M. Ullah, R. Naseem, H. Wang, M. Pedersen, and F. Alaya Cheikh, “Semi-supervised network for detection of covid-19 in chest ct scans,” *IEEE Access*, vol. PP, pp. 1–1, 08 2020.

[15] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” 2015.

[16] S. Lee, G. Lee, E. Jang, and W.-Y. Kim, “Fast affine transform for real-time machine vision applications,” vol. 4113, pp. 1180–1190, 09 2006.

[17] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” 2015.

[18] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, “Unet++: A nested u-net architecture for medical image segmentation,” 2018.

[19] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” 2018.

[20] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, “Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations,” *CoRR*, vol. abs/1707.03237, 2017.

[21] Ma Yi-de, Liu Qing, and Qian Zhi-bai, “Automated image segmentation using improved pcnn model based on cross-entropy,” in *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004.*, pp. 743–746, 2004.