
COVID-19 Lung Segmentation and Quantification

Saisivani Narahari

School of Natural Sciences and Mathematics
University of Texas at Dallas
Richardson, TX 75080
sxn172330@utdallas.edu

Daniel Ralston

Department of Mathematics
Bowdoin College
Brunswick, ME 04011
dralston@bowdoin.edu

Weiqing Gu

McAlister professor of mathematics
Harvey Mudd College
Claremont, CA 91711
gu@g.hmc.edu

Abstract

The novel coronavirus disease 2019 (COVID-19) has caused a health crisis around the world unlike any other that the current generation has experienced. A lot is unknown about the virus, so scientists and researchers are actively working to find a solution that would put an end to the crisis. The current methods used for testing are time consuming and limited in quantity. They fail to effectively identify the extent to which the infection has spread in the lungs. Thus, there is an urgent need for a quick, reliable, and automated way to identify the infected region and to quantify the extent of the infection. We propose applying a U-Net based model to automatically segment the infected regions and use geometric models to find the volume of the infection in the lungs. Our method was evaluated on two datasets comprised of 20 COVID-19 positive patients. The segmentation model achieves a DICE score of 0.8722 on testing data, suggesting promising segmentation efficiency.

1 Introduction

Coronavirus refers to a family of several viruses that cause mild to moderate upper respiratory tract illnesses. Coronaviruses appear predominantly in animals such as cats, pigs, camels, and bats [22]. In various cases, the viruses have transferred from animals to humans in what is called a spillover event. These viruses have made an appearance in the human body time and time again since first recorded in 1960's, most of which caused symptoms similar to a common cold [22]. Three out of the seven known viruses, however, have developed to become serious widespread illnesses. SARS coronavirus (SARS-CoV), which causes severe acute respiratory syndrome, emerged in 2002 and disappeared in 2004. MERS coronavirus (MERS-CoV) transferred from camels and was first identified in 2012. This virus continues to have sporadic but localized outbreaks. The third virus is SARS-CoV-2. It causes coronavirus disease 2019 (COVID-19) and emerged from China in December 2019. It was declared a global pandemic by the World Health Organization on March 11, 2020 and continues to spread across the world [1]. Due to the lack of information on this virus, the community of scientists and researchers are actively working on developing COVID-19 diagnostics and a vaccine. It is crucial to understand how this disease spreads in the human body to develop a prevention method [1]. Our focus in this paper lies in COVID-19 in an attempt to gain a better understanding of how the disease affects human beings.

The exact origin of SARS-CoV-2 is yet to be identified. The first case of COVID-19 was reported in Wuhan City, China, in December 2019. According to the World Health Organization (WHO), it was found that some of the earliest cases of the virus had a link to a wholesale food market in Wuhan, while some did not [25]. Environment samples taken from the market showed the existence of SARS-Cov-2. The sequenced virus isolated from human cases around the world suggests that the virus has an ecological origin in bats [8]. However, there is no evidence to suggest that there was a direct bat to human transmission of the coronavirus. It is possible that the transmission happened through an intermediate host, which has not been identified yet [2]. There has been speculation that the virus was manufactured in a lab, but according to WHO, the genomic features of the virus suggest that it is not man-made and, in fact, has a zoonotic source [25]. Various investigations are underway to understand the source of the outbreak in China, including environmental sampling from markets and farms in areas where the cases were first identified and detailed studies of patients who were infected in December 2019. The results of these studies are crucial to prevent spillover events in the future [25]. Coronaviruses get their name from their distinctive spiked structure. The spikes with rounded tips on the surfaces resemble the sun's atmosphere, otherwise known as corona [34]. Similar to the other coronaviruses, SARS-CoV-2 has a spherical shape with proteins called spikes on the surface that have the ability to latch onto the receptors on the human cell surfaces called angiotensin-converting enzyme 2 (ACE2) [12]. Once the virus attaches itself to the cells, it undergoes a structural change which makes it possible for the viral membrane to fuse with the cell membrane. This allows for the viral genes to enter the host cells and reproduce [12]. According to the National Health Institution (NIH) recent research shows that SARS-CoV-2 spikes are 10 to 20 times more likely to bind to ACE2 on human cells than the spikes from the 2002 SARS virus [12]. This enables the SARS-CoV-2 to spread more rapidly and easily from person to person, making it very dangerous. Various researchers are working on developing a vaccine by targeting the spike protein in SARS-CoV-2.

SARS-CoV-2 spreads primarily from person to person through droplets that are released when an infected person sneezes or coughs. The incubation period of SARS-CoV-2 varies for each person and can be between 2 to 14 days [13]. Due to the lack of symptoms, people that are infected have the chance of spreading the virus to many others before they are aware of the infection. The incubation period along with the ability of the virus to spread rapidly have made it very difficult to contain SARS-Cov-2. The number of COVID-19 cases continues to significantly rise everyday in countries like the United States, Brazil, Canada among others at the time of the research. Due to the alarmingly growing number of cases, an urgent need was developed for an automatic, fast, and reliable computerized method for diagnostics. In this paper, we use machine learning techniques to automatically identify the infectious regions of the lungs, organ that is most affected by SARS-Cov-2, in a chest scan. The automatic detection of the infection through a chest scan also limits the amount of contact between the medical professionals and the patients, which reduces the chances of the virus spreading. Patients are isolated when the chest scan is being taken. Once the chest scans are taken and the infectious areas are automatically detected, it can be sent to the radiologist. This allows the radiologist to examine the scans remotely and decide the plan of treatment.

While this is an exciting and active area of research, there are inevitably a variety of challenges that arise. An initial, and somewhat unavoidable challenge in the field is the computational power needed to harness complicated networks and their parameters. Additionally, due to the recent and unknown rise of the virus, there is still a lack of data in vast quantities. As time goes, this is becoming less and less of a constraint, although most online data sets for medical images only contain a few hundred photos, rather than the thousands needed for industrial machine learning. Furthermore, many of the compiled datasets that are available do not contain ground truth labeling by expert radiologists, making them difficult to use for training and validation in supervised segmentation tasks.

2 Related Research

While COVID-19 is a novel virus, substantial research has been done to understand its properties, symptoms, and consequences. In the case of image segmentation, to the our knowledge, there is no peer reviewed work published at the time of this writing. However, there are independent researchers who have applied advanced machine learning techniques to the limited COVID-19 data that contains proper truth labeling. These researchers mainly draw from Brain Tumor Segmentation research that employ convolutional neural network to achieve the task of object segmentation. Our focus lies in

using convolutional neural networks to automatically identify the COVID-19 infected regions of the lungs in a CT scan.

2.1 Brain Tumor Segmentation Research

Building off of previous recent successes of convolutional neural networks, Isensee et. all in "Brain Tumor Segmentation and Radiomics Survival Prediction: Contribution to the BRATS 2017 Challenge" develop a modification of the popular U-Net architecture for autonomous MRI brain segmentation [15]. By preprocessing MRI intensity values to allow for a more accurate training set and validation set, the authors avoid initial bias from each patient individually by subtracting the mean and then dividing by the standard deviation of the brain region. In addition, they remove outlier intensities by clipping the images at $[-5,5]$ and rescaling values from $[0,1]$, labeling any non-brain region with a value of zero. In one Python implementation of this paper by GitHub user Angad Bajwa, the OpenCV package—specifically the 'cv2' module—is used to accomplish the task of cropping the image. Bajwa uses the 'numpy' library to normalize the images after they have been cropped. However, rather than using normal batch normalisation which normalizes across input features in a specific training example, the authors use instance normalization which normalises across each channel in each training example, meaning that the instance normalization layer is applied at testing time.

The authors used a network inspired by Ronneberger's U-Net to perform semantic segmentation [28]. U-Net is said to have a descending path that acts as an encoder, and an ascending path that acts as a decoder. This is the 'U' that gives it its name. As this is a multi-class segmentation problem, the authors used a softmax output activation layer. The U-Net models consist of multiple levels of what could be described as 'U-Net blocks.' Each convolution corresponds to an increased abstraction with lower spatial resolution but higher dimensional feature representation. As with many U-Net implementations, the authors use the ReLU function to increase non-linearity at each level, using a negative slope of 10^{-2} . This paper and implementation are significant and useful, both because of their modern and successful results and also because of the relative ease and simplicity of the Python code and libraries needed to reproduce the results.

Similarly to Isensee et all, Noori et all in their paper, "Attention-Guided Version of 2D U-Net for Automatic Brain Tumor Segmentation," use a slightly modified U-Net architecture with attention mechanism to achieve similar results [23]. More importantly however, the authors implement the N4ITK Algorithm, which is a variant of the popular N3 algorithm used for bias field correction. Essentially, in image data, low frequency intensity non-uniformity (also known as bias, in homogeneity, illumination non-uniformity, or gain field) can be present. This means that in some images, the way the image is taken can influence the machine learning algorithm. By using B-splines to smooth intensity fields, the N3 algorithm was able to approximate intensities of an image. While previous methods use a highly localized approximation using a single level 32×32 element B-spline mesh, by using a hierarchy of B-spline mesh resolutions, better global approximations can be fitted to images. Thus, by incorporating these analyzation techniques, bias in images can be better discovered.

Bui et. all [6] in "3D Densely Convolutional Networks for Volumetric Segmentation" take on the challenge of volumetric brain image segmentation aiming to separate the brain tissues into non-overlapping regions such as white matter (WM), gray matter (GM), cerebrospinal fluid (CSF) and background (BG) regions. However, the low contrast between the regions can cause misidentification. The researchers address this issue by using a novel deep network architecture based on densely convolution network. They begin with combining local and global predictions through concatenation of feature map of fine and coarse dense blocks. This allows the capturing of multiscale contextual information. They use the traditional DenseNet architecture but replace the pooling layers with convolutional layers of stride two, preventing the loss of spatial information caused by pooling layers. They also use a bottleneck model with compression. This method reduces the number of learned perimeter, resulting in increased computational efficiency and accuracy compared to existing methods.

Wang et al. propose a method to identify multi-motal MRI scans with brain tumor and segment the tumor into three regions: whole tumor, tumor core, and enhancing tumor core [31]. They use a triple cascaded framework of fully convolutional neural network to accomplish this task, which is designed to convert the multiclass segmentation problem into three binary segmentation problems in the order of their hierarchy. They use three networks to segment subsections of the brain. WNet is used to segment the whole tumor from the patients' MRI scans. The output of this layer becomes the

input of the second network, TNet. TNet segments the tumor's core and the output, image region inside the bounding box of the tumor's core, is used as the input for the third network, ENet. ENet segments the enhancing tumor core. Each of the networks use an anisotropic structure, which creates a balance between receptive field, model complexity and memory consumption. The researchers also employ multimodal fusion to reduce the noise in the segmentation results, which is crucial. The cascading framework reduces false positives, as each network works off of the previous network's output. The framework is also computationally efficient in inference time, although it takes longer time for training and testing due to the fact that it is not end-to-end.

Researchers Buda, Saha, and Maciej take brain segmentation a step further in [5] by developing a model that yields high quality segmentation of lower-grade(LGG) in brain MRI's that would automatically identify tumor genomic sub-types. This technique would make tumor identification faster, inexpensive, and free of inter-reader variability. The researchers identified three shape features that were important in the context of LGG and analyzed their relationship with different tumor molecular subtypes as following: bounding ellipsoid volume ratio with RNASeq, miRNA, CNC, and COC; the relationship of Margin fluctuation with RNASeq; and the relationship of angular standard deviation with IDH/1p19q, RNASeq, Methylation, CNC, and COC, resulting in 10 specific hypotheses. To assess the statistical significance of these association, the Fisher exact test was employed, and the researchers concluded that fully automatically-assessed imaging features of low grade gliomas are associated with tumor molecular subtypes. The strength of these associations was shown to be moderate. The methods employed in this research can be used in developing a model that differentiates between SARS-CoV-2 and pneumonia in chest scans.

Mlynarski et al. propose deep learning with mixed supervision in [20]. Most of the work in segmentation research relies on manually segmented images. Manual segmentation of images is time-consuming and requires high level of medical expertise. Thus, there is a need for a training method that has less computational costs. They work with a dataset that has both fully annotated images and weakly annotated images that have image-level labels indicating the presence or absence of a tumor. The researchers use a CNN-based segmentation model, which can be trained using weakly annotated images in addition to fully annotated images. They further propose to extend segmentation networks, such as U-Net, with an additional branch performing image-level classification. The goal is to exploit the representation learning ability of CNNs to learn from weakly annotated images while supervising the training using fully annotated images to learn features relevant for the segmentation task. This model provides a significant improvement over the standard supervised approach (U-Net trained on fully annotated images) when the number of fully annotated images is limited.

Similarly, Radford et al. propose an alternative method to supervised learning in computer vision applications [26]. The researchers introduce a class of CNNs called deep convolutional generative adversarial networks (DCGANs), that have certain architectural constraints, and demonstrate that they are a strong candidate for unsupervised learning. They propose that one way to build good image representations is by training Generative Adversarial Networks (GANs), and later reusing parts of the generator and discriminator networks as feature extractors for supervised tasks. In this method, the researchers use all convolutional neural net to replace deterministic spatial pooling functions with strided convolutions, allowing the network to learn its own spatial downsampling.

Esteves et al. address the problem of 3D rotation equivariance in the convolutional neural networks in their research [10]. They propose the first neural network based on spherical convolutions, in addition to introducing pooling and parameterization of filters in a spheres domain. In their method, they define a block as one spherical convolutional layer, which is followed by optional pooling and non linearity. A global weighted average pooling is applied to the final layer to produce an invariant descriptor. The main architecture is divided into two branches, one for distance and the other for surface normals. This architecture has better performance when compared to the traditional two input channels. Each branch consists of eight spherical convolutional layers, and the pooling and feature concatenation of the two branches happens every time the number of channels is increases per layer. Weighted Global Average Pooling is performed after the last layer. The researchers in this work apply Spherical CNN to 3D objects classification, retrieval, and alignment. This method has potential applications in COVID-19 infection segmentation due to its 3D nature.

2.2 COVID-19 Lung Segmentation Research

Researcher Rohit Verma created a GitHub repository containing packages to segment COVID-19 infection sites, as well as segmentation for the entire lung [30]. Most notably, Verma uses U-Net and U-Net++ (a variation off of the original design that seeks to reduce the "semantic gap between the feature maps of the encoder and decoder sub-networks") to accomplish the task, yielding slightly higher results for the U-Net++ architecture. This repository demonstrates the importance of relying on previous techniques developed for brain tumor segmentation, highlighting the adaptability of such architectures. Please note that we employ a lot of the techniques discussed in this repository and base our U-Net model from Verma's work, hence we credit the author for the results we were able to obtain for the segmentation task of our project. Expanding off of this work, our project uses advanced geometrical techniques to quantify infection caused by SARS-CoV-2.

Another notable development in the field of applying machine learning to COVID-19 lung data is the open source COVID-Net project, available on GitHub [32]. According to the authors of COVID-Net, it is one of the first "open source network designs for COVID-19 detection from chest x-ray images at the time of initial release. First, the contributors assembled a data set of approximately fourteen thousand chest x-ray images, comprising of lung scans of patients with pneumonia, an ailment that COVID-19 has been known to bring on. Furthermore, to the best of our knowledge, this data set contains one of the largest publicly available COVID-19 positive cases, making COVID-Net an important milestone in applying machine learning techniques to combating the virus. While we don't plan to use the same techniques as the authors do since this repository doesn't focus on image segmentation, we credit this project to leading us to the data set we will be using, as well as providing foundational information on how to generally approach the problem of COVID-19 infection segmentation.

3 Project Description

We will be accomplishing the task of automatic segmentation of infectious lung regions using deep machine learning techniques. Using OpenCV, we develop a way to calculate the volume of the infection in the lungs in an attempt to quantify the spread of the disease. There has been immense success with convolutional neural networks (CNN) in term of image identification and segmentation. Specifically, we will use a bio imaging packaging called U-Net, which has been very effective in brain segmentation research. We will be basing our work off of several important papers on brain tumor segmentation. Our work will be distinct due to the chest scan dataset we are working with, as well as our application of specific machine learning techniques to the problem of COVID-19 .

4 Method

4.1 Dataset

We require a dataset that contains images of lungs along with expert-annotated ground truth labels identifying the infection caused by SARS-Cov-2 for supervised training. At the time of the research, there is a lack of COVID-19 data with expert-annotated images. We were able to acquire a dataset that answers our segmentation problem, although it is small. This dataset includes 20 patients' chest CT scans and has been acquired from Radiopaedia, online collaborative radiology resource and Coronacases.org [24][35][17]. Only the scans of the patients that were diagnosed with COVID-19 with high certainty were included in this dataset, although not all slices contain evidence of COVID-19. We are aware of the short-comings of using a small dataset for training, in terms of obtaining results with moderate accuracy when the model is applied on a large group of new data. We will continue to work on improving the model as more COVID-19 data becomes available for public research.

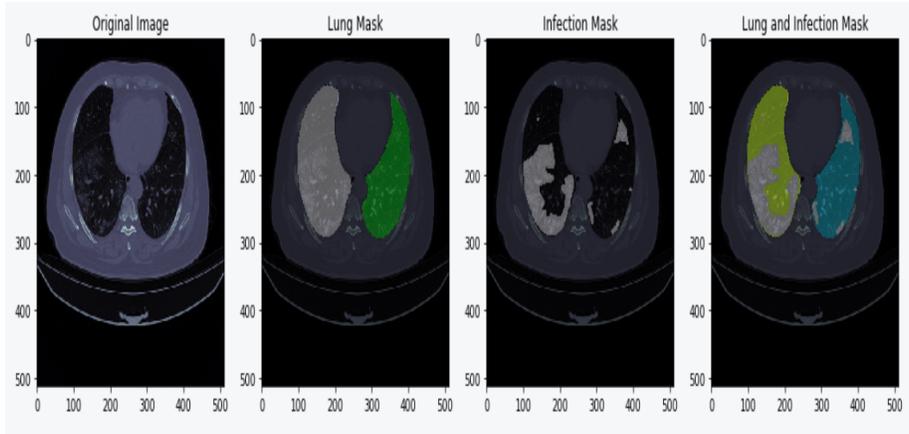


Figure 1: Sample data
[24][35][17]

4.2 Data Preprocessing

It is crucial to preprocess the available data to able to obtain the most accurate and desired results. We begin the preprocessing by identifying the slices without any traces of lung or infection and separating them from the rest of the data. The separated slides are later used for developing a model for empty mask prediction. Second, we enhance the CT scan images using Contrast Limited Adaptive Histogram Equalization (CLAHE). It amplifies the contrast between pixels that might otherwise look similar, thus enhancing the image. This preprocessing step is vital to a segmentation problem because the model relies on differentiating between neighboring pixels. Third, we crop the CT scans to obtain the region of interest using Otsu Binarization. This step is taken to prevent the excess area from interfering with the training of the model. The final step in the preprocessing stage is data augmentation. Data augmentation significantly improves the diversity and amount of the data that is available for training. Transformations such as shift, zoom, and rotate are applied randomly to the available data to create new data.

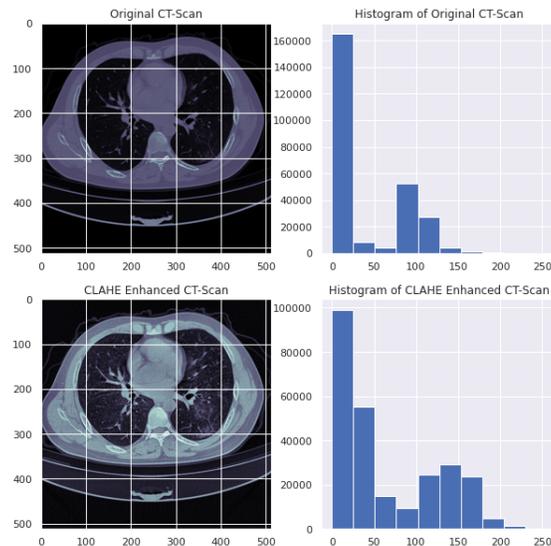


Figure 2: Pixel Distribution between Enhanced and Un-enhanced Scans

4.3 Semantic Segmentation and U-Net

Semantic segmentation is a method of processing image data and classifying the image in pixel level through a process called dense prediction [16]. Normally, processing an image as a data set involves using filters, and obtaining gradient and color information. However, by utilizing recent advances in Deep Learning, semantic segmentation has gotten significantly more efficient. For example, current models use a probabilistic framework for labeling current pixels and developing a relationship between nearby pixels and pixels with a similar color. Neural network structures can also account for hundreds of other probabilities; for example, if there is a "basketball net" class within a photo, then there is more likely to be a "basketball" class within the photo as well, i.e. when there is a basketball net in a photo, there is also more likely a basketball. Applied to COVID-19 positive CT scans, semantic segmentation in Deep Learning allows for an image to be inputted, and then based on color, pattern, opacity, and medical probabilities, the image can be classified into healthy and potentially infectious segments efficiently.

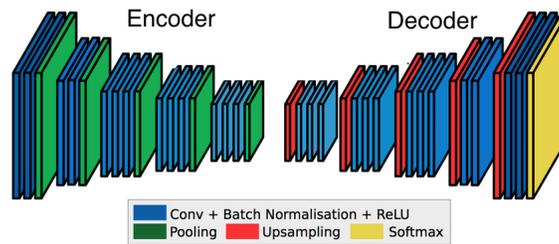


Figure 3: Semantic Segmentation layering [11]

U-Net uses a Fully Convolutional Network Model to accomplish semantic segmentation. The architecture was developed by Olaf Ronneberger specifically for biomedical image segmentation [18]. The architecture consists of down sampling (encoder) and up sampling (decoder). Down sampling allows for better recognition on a pixel level, but comes at the cost of the location. Up-sampling is necessary to identify the location. The encoder path captures the context of the image through a stack of convolutional and max pooling layers. The up-sampling path uses transposed convolutions to identify the precise location of the object in interest, outputting a segmentation mask. The U-Net structure only contains convolutional layers and does not contain any dense layers, so the user can input an image of any size [18].

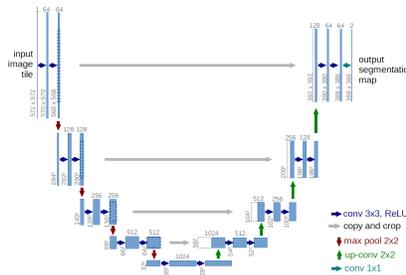


Figure 4: U-NET architecture [29]

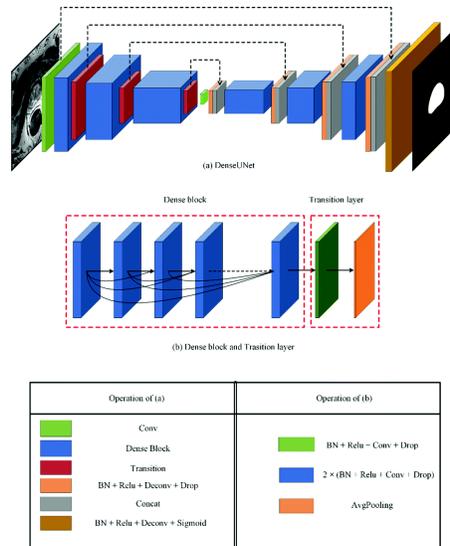


Figure 5: U-NET layering [19]

4.4 Training and Testing

The model is compiled using an Adam optimizer, a common and more effective computer algorithm that builds off the stochastic gradient descent procedure, combining the advantages of both Adaptive Gradient Algorithm and Root Mean Square Propagation. From the previous work done on this project, we obtain that a learning rate of 0.0005, a batch size of 32, and 80 epochs is optimal for our data set. While these metrics may be slightly unusual, including the fact that we use a higher batch size and a relatively low learning rate, this makes sense for our limited data set. To avoid over fitting for our small dataset, a phenomenon that occurs when a model can predict a data set too well and may be recognizing coincidences rather than correlations, using a larger batch size ensures that with each update of our model, a larger amount of data will be recognized and accounted for, giving a more holistic training method. However, this increase in batch sizes means that we must decrease the number of epochs, since otherwise we would be still having our data “seen” by our model too frequently. Finally, by choosing a relatively low learning rate, it helps ensure that the model trains more slowly. Although this limits the amount of progress the model can make with such a limited dataset, it helps to ensure minimal over-fitting.

In training our model, as inspired by previous work, we split the data set into 70 percent for training purposes and 30 percent for validation purposes. This is a relatively common percentile divide. However, it is important to note that we have preformed most of our testing using random samples from this data set. While this is unusual, we wish to ensure that given the lack of data present for this model, all possible CT slices are used to train the model to make predictions that are as accurate as possible.

4.5 Cross-Validation

Cross validation is a method used in machine learning to evaluate the skill of a particular model on unseen data. There are many different ways of applying cross validation, but the general procedures, taken from [33] are as follows:

- Shuffle the data set randomly.
- Split the data set into k groups. Note that this is where the k in k -Fold Cross Validation comes from.
- For each of these unique k groups:
 - Take the group as a hold out or a data set
 - Take the remaining groups as a training data set
 - Fit a model on the training set and evaluate it on the test set

- Retain the evaluation score and discard the model
- Summarize the skill of the model using the sample of model evaluation scores

Furthermore, each data sample is given to a group and will remain in the given group, meaning that each data group can be used to train the model $k - 1$ times. This last time the model is tested on this group. Another important point worth noting is that each group is sometimes described as a “fold.” Most commonly in the Python programming environment, the Scikit-Learn library implements k -Fold Cross Validation.

A challenge of k -Fold Cross Validation is that the researcher must choose an accurate value of k to separate the data. Sometimes an integer of 10 or less is used, with, according to [33], the two most common being values of 10 and 5. However, choosing one’s k value must ensure that both the training and the testing groups of data samples can statistically represent much of the data set. In other words, since each group should already be obtaining a sample way of modeling a representative portion of the data, finally comparing all values together ensures that an extremely holistic model is evaluated. In some cases the value of k may be chosen to be the size of the dataset. This ensures that each test sample can be used with a fold, and thus this is aptly named leave-one-out Cross-Validation. It is also worth noting that it is usually desirable to split the data into groups (folds) that contain the same number of samples, meaning that the most even modeling may occur.

4.6 Dice Score

The Dice Coefficient, also known as the Sørensen-Dice Coefficient, is a statistical measure of precision, commonly used in the field of machine learning to measure the score of image segmentation. In particular, the Dice score penalizes for any positives that an algorithm does or does not find. Thus, it is of particular interest to this project. This measurement is very similar to the Jaccard distance, which is defined as

$$1 - J_c(A, B)$$

where

$$J_c = \frac{A \cap B}{A \cup B}$$

is the Jaccard coefficient, also known as the Intersection over Union metric. The Dice coefficient is defined as

$$D_c = \frac{2|A \cap B|}{|A| + |B|}$$

where $|\cdot|$ represents the cardinality of sets A and B . However, in a more computer science related metric, the Dice Coefficient can be represented with Boolean data as

$$D_c = \frac{2TP}{2TP + FP + FN}$$

where TP is a true positive, FP is a false positive, and FN is a false negative. An important note here is that the difference function for the Dice coefficient, defined as

$$d = 1 - D_c$$

does not satisfy the triangle inequality and thus is not a distance metric that can be used in a metric space.

4.7 Volume of Segmented Infection Regions

After running a U-Net architecture over COVID-19 volumetric data, we then analyzed the actual area of each labeled infection area in millimeters squared, to obtain post processing data that could lead to statistical correlations. While there are many different methods of calculating such area of the segmented regions—we explored methods involving k -mean clustering, alpha shape calculation, and convex-hull triangulation—we settled on simply calculating the number of pixels contained in the segmented region.

This approach has several benefits. Since the U-Net architecture is programmed to return a mask containing only black and white (a white pixel corresponds to an infected area), this method of obtaining area proved easy and computationally efficient. Secondly, another data point we are

interested in calculating is the volume of infected areas within the lungs of patients infected with COVID-19. Since most of our data points include scaling measurements, it then becomes relatively easy to scale our pixel values to match the metric of millimeters squared; then, simply multiplying by the thickness of each slice in the volumetric CT scan yields an approximation of the volumetric area.

$$\text{Volume} = \sum \left(\frac{\text{Area of Top Slice} + \text{Area of Bottom Slice}}{2} \right) \times \text{Height}$$

There are some draw backs to using this method of calculating the area of segmented regions by using a graphical, rather than a mathematical method. The first is simply the nature of a mathematical calculation; by using an advanced algorithm to approximate the area of a segmented region, interested shapes, pasterns, and various other interesting data points may be uncovered, which could yield further information on the way the infection may spread throughout the lung, important for our statistical analyzations.

However, a larger problem that plagues any of our considered methods of analyzing the area of a segmented region regards scaling. In order for our U-Net architecture to be able to process an image—a slice from a volumetric data set—we must resize images to be the proper dimensions. Thus, whenever we are re-scaling our data we must be clear on whether it is being cropped or resized. In our case we have ensured that we are only cropping our data, so that all spatial orientation can be preserved. However, as more and more data from around the world becomes available, simply cropping an image may no longer feasible, and thus advanced measures to keep track of spatial changes must be implemented. Additionally, given the variation of patients, even our relatively small data set from one source contains wide variation on the thickness of slices given, making such a scaling problem even more difficult.

5 Results

5.1 DICE Coefficients

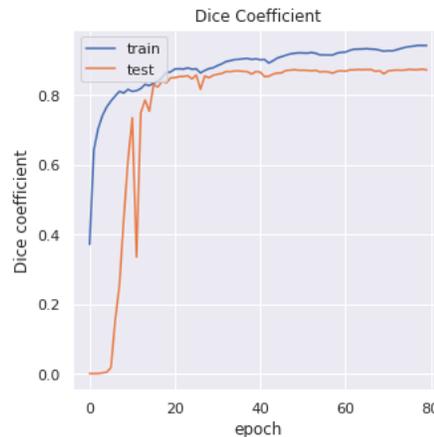


Figure 6: Dice Coefficient Training and Testing

Figure 6 displays the DICE coefficient curve for training and testing. The training curve begins with a DICE coefficient of 0.4. There is a steep increase from 1 to 10 epochs, suggesting that most of the learning happened here during the training. After 10 epochs, there is a steady, almost linear, increase in the coefficient. After 80 epochs, the training DICE coefficient is 0.9380. The testing DICE coefficient begins with close to zero. There is steep increase between 5-20 epochs, after which it flattens. After 80 epochs, the testing DICE coefficient is 0.8722. This shows that the model has a high accuracy rate when predicting the infection masks

The DICE loss curves in figure 7 for training and testing are similar and complements the DICE coefficient curve. DICE loss coefficient is another way to measure the precision of the model. Our goal is to minimize the DICE loss curve to obtain a model that has a high accuracy in prediction.

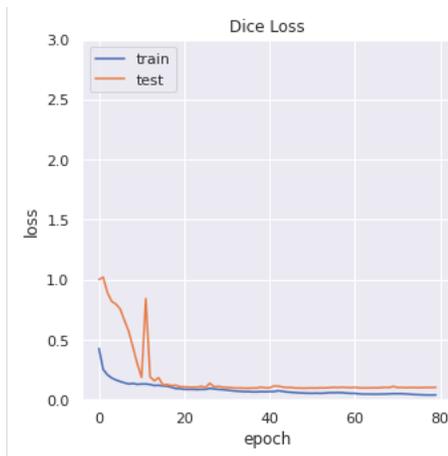


Figure 7: Dice Coefficient Loss and Testing

After 80 epochs, the DICE loss training is 0.0397 and for testing it is 0.1035, which again shows that the model has high accuracy.

5.2 Volumetric Results

Patient	Infection Volume (in mm ³)
1	136352.7262
2	187589.9636
3	166686.7488
4	169484.7466
5	182675.2370
6	159180.8654
7	140129.6510
8	168604.5253
9	162845.7202
10	192705.6705

Figure 8: Volume of SARS-CoV-2 in Patients' Lungs

Figure 8 shows the volumetric calculations of the infection for ten COVID positive patients in the dataset, for which we were able to find the pixel-to-millimeter ratio. This model only work for scans where there is a pixel-to-millimeter conversion because pixels cannot be standardized. Unfortunately, there is not enough data available to the public with this information. Volume of the infection for each patient ranges from 136352.73 mm^3 to 192705.67 mm^3 which is approximately 4.6 to 6.5 fluid ounces.

5.3 Prediction Masks

Figure 9 shows the CT scan slices of the lungs, the model's prediction of the infectious areas, and expert-annotated mask of the infection. The similarity between the prediction and actual masks suggests that the model predicts SARS-CoV-2 with high accuracy, which supports the conclusion drawn from the DICE curves. The model however, does not detect the rigid edges of the infection. It outputs a smoother segmentation mask. This is something that could be improved in the future.

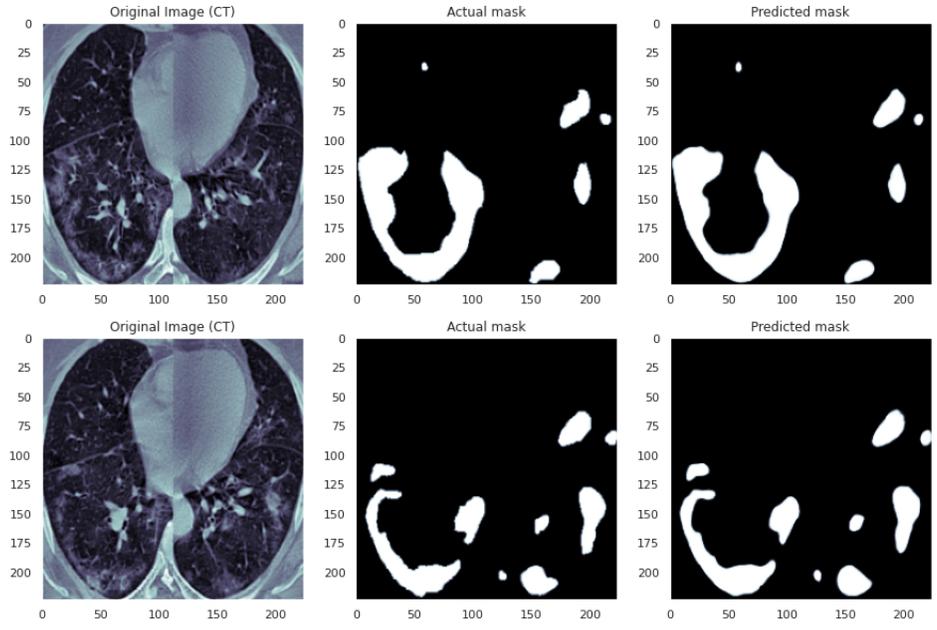


Figure 9: Sample Actual and Prediction Masks

5.4 Classification Threshold

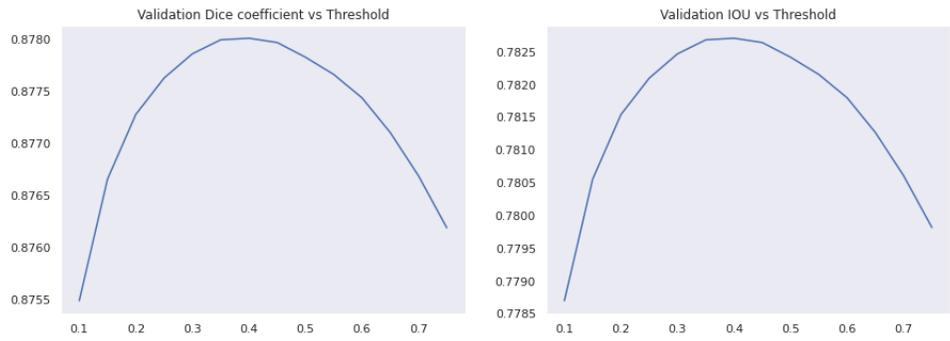


Figure 10: Classification Threshold Compared with Dice and IOU Values

This graph charts the classification threshold (the independent variable, which is a measure of when to convert a predicted probability into a class label, or in other words at what probability threshold the model will classify a region) when compared to the dice loss and IOU scores. Note that the optimal IOU and dice scores are achieved when the classification threshold is between 0.4 and 0.5. The standard classification threshold is normally 0.5, since if the model predicts a more than 50 percent probability then it is more likely a class object. For our model, the optimal classification threshold looks to be about 0.41.

5.5 Comparison of 3-Fold and 4-Fold Cross Validation

3-fold Dices dataframe				4-fold Dices dataframe				
Rows indices: Thresholds, Column indices: Split Number				Rows indices: Thresholds, Column indices: Split Number				
	1	2	3	1	2	3	4	
0.30	0.955636	0.955493	0.931729	0.30	0.967673	0.968324	0.967564	0.948251
0.35	0.955924	0.955752	0.931824	0.35	0.967852	0.968491	0.967749	0.948343
0.40	0.956157	0.955881	0.931924	0.40	0.967965	0.968611	0.967878	0.948406
0.45	0.956242	0.955924	0.931901	0.45	0.968049	0.968690	0.967945	0.948440
0.50	0.956256	0.955925	0.931817	0.50	0.968106	0.968744	0.968002	0.948489
0.55	0.956183	0.955850	0.931724	0.55	0.968093	0.968774	0.968039	0.948460
0.60	0.956037	0.955695	0.931541	0.60	0.968082	0.968746	0.968015	0.948404
0.65	0.955777	0.955424	0.931277	0.65	0.968021	0.968693	0.967968	0.948332
0.70	0.955435	0.955014	0.930958	0.70	0.967912	0.968606	0.967856	0.948176
0.75	0.954926	0.954364	0.930472	0.75	0.967727	0.968416	0.967670	0.947997

Figure 11: Dice Comparison

3-fold Ious dataframe				4-fold Ious dataframe				
Rows indices: Thresholds, Column indices: Split Number				Rows indices: Thresholds, Column indices: Split Number				
	1	2	3	1	2	3	4	
0.30	0.915160	0.914893	0.872667	0.30	0.937436	0.938655	0.937221	0.901795
0.35	0.915686	0.915369	0.872837	0.35	0.937770	0.938969	0.937568	0.901960
0.40	0.916112	0.915606	0.873014	0.40	0.937984	0.939193	0.937810	0.902075
0.45	0.916267	0.915688	0.872974	0.45	0.938142	0.939342	0.937935	0.902137
0.50	0.916294	0.915689	0.872831	0.50	0.938249	0.939444	0.938041	0.902223
0.55	0.916160	0.915552	0.872669	0.55	0.938224	0.939500	0.938112	0.902171
0.60	0.915892	0.915269	0.872351	0.60	0.938204	0.939449	0.938066	0.902069
0.65	0.915418	0.914775	0.871894	0.65	0.938091	0.939350	0.937979	0.901938
0.70	0.914792	0.914028	0.871339	0.70	0.937888	0.939186	0.937769	0.901658
0.75	0.913860	0.912845	0.870497	0.75	0.937542	0.938830	0.937421	0.901334

Figure 12: IOU Comparison

3-fold precision dataframe				4-fold precision dataframe				
Rows indices: Thresholds, Column indices: Split Number				Rows indices: Thresholds, Column indices: Split Number				
	1	2	3	1	2	3	4	
0.30	0.945219	0.945942	0.922529	0.30	0.960685	0.961352	0.960312	0.940954
0.35	0.947955	0.948604	0.925096	0.35	0.962161	0.962695	0.961799	0.942553
0.40	0.950464	0.950984	0.927484	0.40	0.963420	0.963917	0.963102	0.943986
0.45	0.952704	0.953190	0.929736	0.45	0.964587	0.965108	0.964246	0.945360
0.50	0.954800	0.955301	0.931843	0.50	0.965751	0.966227	0.965381	0.946745
0.55	0.956829	0.957366	0.933975	0.55	0.966850	0.967333	0.966556	0.948096
0.60	0.958869	0.959412	0.936103	0.60	0.967937	0.968402	0.967634	0.949408
0.65	0.960995	0.961508	0.938271	0.65	0.969016	0.969528	0.968838	0.950769
0.70	0.963192	0.963748	0.940605	0.70	0.970241	0.970720	0.970064	0.952172
0.75	0.965637	0.966128	0.943170	0.75	0.971592	0.972109	0.971438	0.953871

Figure 13: Precision Comparison

In these figures, we compare using 3 and 4 Fold Cross Validation. As shown above, while 4-Fold Cross Validation does yield marginally better results, the closeness of the scores indicates that 3-Fold Cross Validation may be optimal for this problem and data set.

6 Conclusion

In this paper, we present and discuss the application of a U-Net Convolutional Neural Network to a volumetric CT-Scan COVID-19 lung data set. The model interprets 2-D slices from our data set

and uses an Adam Optimizer and Dice Loss function for training purposes, using 70 percent of the data set for training purposes and 30 percent of the data set for validation. We explore the results of 3 and 4 fold cross validation; however, we predict that as more data becomes available validation methods may change, as well as the current model architecture. Furthermore, we expand on previous work done in the field by calculating the approximate volume of infected regions, and plan to conduct statistical analysis as more data becomes available about the patient, including data on gender, age, and location.

7 Future Work

As more COVID-19 data becomes available to the research community, we will work on improving our model's accuracy in predicting the SARS-CoV-2 infected regions of the lungs. Furthermore, once we have access to patient data such as the survival rate, pre-existing conditions, lifestyle habits, statistical models can be developed to evaluate the relationship between the damage done to the lungs and the patient demographics. This will help healthcare professional and the public gain a better understanding of how SARS-CoV-2 affects human bodies.

Another more urgent problem at hand is the automatic COVID-19 detection in chest x-rays or CT scans using deep learning. This is a difficult problem because in a chest x-ray or a CT scan, pneumonia and SARS-CoV-19 look very similar to the untrained eye. Classification of COVID-19 has been an active area of interest in the deep learning research community, with researchers developing new convolutional neural networks, such as COVID-Net and DarkCovid-Net, tailored to address this classification problem. However, there is a long way to go before these techniques are production ready. Due to the shortage of testing kits in the united states and many other countries, this research would be extremely helpful in rapid testing and diagnosis of individuals.

References

- [1] National Institute of Allergy and Infectious Diseases. *Coronaviruses*. Tech. rep. May 2020.
- [2] American Veterinary Medical Association. *COVID-19 Origin and Spread*. Tech. rep. May 2020.
- [3] *Brain Tumor and Spinal Tumor Program*. Available at <https://www.urmc.rochester.edu/neurosurgery/services/brain-spinal-tumor/conditions/low-grade-glioma.aspx#:~:text=Low>.
- [4] *Brain Tumor Education*. Available at <https://www.abta.org/about-brain-tumors/brain-tumor-education/#:~:text=Brain..>
- [5] Mateusz Buda, Ashirbani Saha, and Maciej A. Mazurowski. "Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm". In: (2019).
- [6] Toan Duc Bui, Jitae Shin, and Taesup Moon. "3D Densely Convolutional Networks for Volumetric Segmentation". In: (2017).
- [7] *Cancer Information*. Available at <https://www.nfcr.org/cancer-information/>.
- [8] Jon Cohen. *A WHO-led mission may investigate the pandemic's origin. Here are the key questions to ask*.
- [9] Ivana Despotović et al. *MRI Segmentation of the Human Brain: Challenges, Methods, and Applications*. 2015. URL: <https://www.hindawi.com/journals/cmmm/2015/450341/>.
- [10] Carlos Esteves et al. "Learning SO(3) Equivariant Representations with Spherical CNNs". In: (2018).
- [11] Divam Gupta. *A Beginner's guide to Deep Learning based Semantic Segmentation using Keras*. 2019. URL: <https://divangupta.com/image-segmentation/2019/06/06/deep-learning-semantic-segmentation-keras.html>.
- [12] National Institutes of Health. *Novel coronavirus structure reveals targets for vaccines and treatments*. Tech. rep. Mar. 2020.
- [13] Erica Hersh. *How Long Is the Incubation Period for the Coronavirus?*
- [14] *High Grade Gliomas*.
- [15] Fabian Isensee et al. "Brain Tumor Segmentation and Radiomics Survival Prediction: Contribution to the BRATS 2017 Challenge". In: (2018).
- [16] Jeremy Jordan. *An overview of semantic image segmentation*. <https://www.jeremyjordan.me/semantic-segmentation/2018/5/21>.
- [17] Ma Jun et al. *COVID-19 CT Lung and Infection Segmentation Dataset (Version Verson 1.0)*. <https://zenodo.org/record/3757476#.XwedcWpKg6g>.

- [18] Harshall Lamba. *Understanding Semantic Segmentation with UNET*. <https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b472019/2/17>.
- [19] Suiyi Li et al. *Cascade Dense-Unet for Prostate Segmentation in MR Images*. 2019. URL: https://link.springer.com/chapter/10.1007/978-3-030-26763-6_46.
- [20] Pawel Mlynarski et al. “Deep learning with mixed supervision for brain tumor segmentation”. In: *Journal of Medical Imaging* 6 (2019).
- [21] nathancy. *Calculate the area of the masks (in pixels) in grey scale images with python*. <https://stackoverflow.com/questions/58068315/calculate-the-area-of-the-masks-in-pixels-in-grey-scale-images-with-python>.
- [22] National Foundation for Infection Diseases. *Coronaviruses*. Tech. rep. May 2020.
- [23] Mehrdad Noori, Ali Bahri, and Karim Mohammadi. “Attention-Guided Version of 2D UNet for Automatic Brain Tumor Segmentation”. In: *IEEE* (2019).
- [24] Paiva O. *Helping Radiologists To Help People In More Than 100 Countries! | Coronavirus Cases - Coronacases.org*. Accessed: 2020-03-20.
- [25] World Health Organization. *Coronavirus disease 2019 (COVID-19) Situation Report – 94*. Tech. rep. Apr. 2020.
- [26] Alec Radford, Luke Metz, and Soumith Chintala. “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”. In: (2016).
- [27] MD Robert Fenstermaker. *Low-Grade Glioma*. Available at <https://www.roswellpark.org/cancertalk/201806/low-grade-glioma> (2018/06/18).
- [28] Olaf Ronneberger, Philip Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: (2015).
- [29] Olaf Ronneberger, Philip Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. URL: <https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/>.
- [30] Rohit Verma. *One-Stop-for-COVID-19-Infection-and-Lung-Segmentation-plus-Classification*. June 2020. URL: <https://github.com/deadskull7/One-Stop-for-COVID-19-Infection-and-Lung-Segmentation-plus-Classification/blob/master/README.md>.
- [31] Guotai Wang et al. “Automatic Brain Tumor Segmentation using Cascaded Anisotropic Convolutional Neural Networks”. In: (2017).
- [32] Lechen Wang and Alexander Wong. “COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images”. In: 2020.
- [33] Wikipedia. *Cross-validation (statistics) — Wikipedia, The Free Encyclopedia*. [http://en.wikipedia.org/w/index.php?title=Cross-validation%20\(statistics\)&oldid=966151425](http://en.wikipedia.org/w/index.php?title=Cross-validation%20(statistics)&oldid=966151425). [Online; accessed 30-July-2020]. 2020.
- [34] Shawna Williams. *A Brief History of Human Coronaviruses*.
- [35] Glick Y. *Viewing Playlist: COVID-19 Pneumonia | Radiopaedia.Org*. <https://radiopaedia.org/playlists/25887>. Accessed: 2020-04-20.
- [36] Maria Yiallourous. *High-Grade Gliomas (HGG) - Brief Information*.