Automatic Image Segmentation on COVID-19 Chest X-ray and CT Images

Andrew Paredes, Ashwini Rajasekhar, Revanth Vedula, Yichen Ju
ECE 6780
Georgia Institute of Technology
Atlanta, USA

***Abstract*—The objective of our paper is to identify and segment regions of the lungs which have higher probability of being infected with COVID-19. Doing so will help combat the rise of COVID-19 cases in an attempt to begin the healing process, To achieve our goal, we used a medical image segmentation technique called UNET to train the model by feeding it large amounts of labeled and unlabeled lung CT scans. We used DICE, Precision and Recall metrics to measure the effectiveness compared to other existing state-of-the-art models. Our model performed excellently with the given resources and even has some room for improvement.**

***Index Terms*—COVID-19, UNET, Medical Image Segmentation, CNN, datasets, data preprocessing, dice, precision, recall**

I. Introduction

With the COVID-19 pandemic overrunning the world's medical facilities, automating the detection of the virus in patients has been of the utmost importance. Image segmentation techniques can facilitate the identification of COVID-19 cases by identifying infected tissues in lung CT scans, and distinguishing them from common cases of pneumonia, reducing the overload caused by RT-PCR testing and bringing down requirements for human intervention. RT-PCR, which involves the detection of viral nucleic acid from a sputum/nasopharyngeal swab, requires a very specific piece of testing equipment that is often times in short supply. It also is time-consuming, with tests taking 24-48 hours to confirm. There is also a low sensitivity of the true positive rate, and generally involves an invasive and uncomfortable testing method.

Image segmentation solutions have varied applications in the medical field. These range from the identification of brain tumors, breast cancer, lung cancer, etc. and can be fast-tracked with automated segmentation methods, to speed up the diagnosis, and identify infected areas before the pathogen spreads. This is especially true in the case of COVID-19, as hospitals are stretched thin with the pandemic, and RT-PCR testing has its own flaws. Utilizing medical image segmentation can alleviate the stress on the world's medical system. At this moment in time, the biggest cause for concern is the lack of rapid testing. However, using this methodology, healthcare workers at hospitals can accurately classify whether a patient has COVID-19 induced pneumonia or not based on their preliminary CT scans.

UNET is one of the primary subsets of medical image segmentation. Through both our own project and the work of other published papers, it has shown to be more reliable and accurate than other methods such as the RT-PCR testing. However, a major hurdle for the model seems to be the need for a large amount of datasets of labeled images for training purposes. Nevertheless, as time has passed since the beginning of the pandemic, more COVID-19 related datasets for CT images have been made available. This increase along with the standard data preprocessing methods to increase datasets, has made it easier to alleviate the issue.

II. Related Work

The utilization of UNETs for medical image processing methods has been the norm for a large amount of research in this area. [1] built a UNET convolutional neural network for semantic segmentation. The network was trained on a dataset that originated in China and only contained 19 CT scans. The main new contribution of the paper was utilizing a loss function which is the combination of categorical cross-entropy and the Tversky index. Gathering the results and comparing them against state-of-the-art segmentations at the time of publication, the results compared to ground truth. The network performed well but could be overfitting the data, considering that the dataset they used at the time only contained 19 CT scans. Hence, the lack of datasets was a very prevalent issue in early 2020.

Another approach, taken in [2] addresses the problem of COVID-19 through medical imaging of pneumonia using convolutional neural networks (CNNs). For its CNN implementation, the authors incorporated Multi-Task Learning (MTL) into the standard U-Net architecture with the goal being "to combine several pieces of information from different tasks to improve the performance of the model."

The implementation in [3] was more in line with our own approach, to detect the infected tissue from lung CT scan images. They utilized an 11 layer CNN, with their dataset consisting of 256 x 256 grayscale scans of COVID-19 positive patients, and the output layer of their network was fed with the ground truth of the dataset. Using the common 70-30 split of training and testing data to ensure no data leakage, the infected section of their scans were detected with high accuracy. With 92% of the images in the high-performance region of their plotted ROC curve, they proved that COVID-lesion segmentation is a viable option to be implemented in every hospital.

III. Proposed Model Design

We start off by gathering our datasets we have some that are labeled and some that are unlabeled. We plan to use semi-supervised techniques to expand our dataset by providing labels to the unlabeled portions to prevent overfitting. We will then perform image preprocessing and data augmentation to ensure all the data are unbiased. We will then build a UNET model and train it using standard deep learning gradient descent with an Adam Optimizer. We then test using a 70 - 30 train test split of our dataset, and will also explore 5 fold test validation. These techniques are being used to prevent data leakage when training our model. Next, we evaluate the effectiveness of the model using metrics that have been established in literature such as DICE, precision, and recall. Finally, our final output will be an edge traced 2D image with the segmented regions of the lungs containing and not containing COVID-19.

IV. Data Preprocessing

A. Datasets

The datasets we are using are as follows: COVID-19 CT segmentation dataset including 100 COVID-19 CT images from over 40 patients with pixel-level annotation for the images and ideal for segmentation task; BIMCV COVID-19+ dataset including 1380 CX, 885 DX, and 163 CT studies from 1311 COVID-19+ patients; COVID-19 CT Lung and Infection Segmentation dataset including 20 labeled COVID-19 CT scans labeled with left lung, right lung, and infections. Considering the needs for the unsupervised experiment, we also plan to apply for our work on the following datasets containing unlabeled COVID-19 CT scans COVID-CT-Dataset which has 349 CT images including clinical findings of COVID-19 from 216 patients as well as 397 Non-COVID CT images; COVID-19 Database with 115 COVID-19 CT scans collected from Italian radiological cases; COVID-19 Image Data Collection including 468 COVID-19 chest X-ray and CT images from public sources as well as through indirect collection from hospitals and physician. The dataset was resized, resampled, normalized and clipped to suit the needs of the network and adjust the values for uniformity.

B. Geometric Transformations

We apply a series of homographic augmentations to generate new samples from existing images. Tensorflow's ImageDataGenerator class provides a wide variety of distortions to choose from, and using the Nearest-neighbour interpolation the resultant images retain the necessary information of the scan going through the changes. From the available resources, we choose rotation, shear, height and width shift, rescaling, zoom, and horizontal flips. The program is designed to take random transformations, and apply them to both the scan image and the corresponding masks to ensure the pixel-level ground truth marking still holds accurate.

As we can see in Figure 1, the first column shows us the original scan, normalized and resampled. The second and third column show some examples of possible geometric transforms occurring to the image. The amount by which any transformation takes place (eg,. degrees of rotation, amount of shear, zoom, etc.) is decided by was decided by us through trial and error, to ensure that the important parts of the image aren't lost in the transforms.

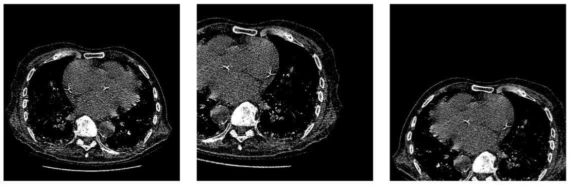

Fig. 1. A series of examples of geometric transformations used for data augmentations.

C. Color Space Transformations

Utilizing the histogram of the scan, we perform adaptive histogram equalization using Contrast Limited Adaptive Histogram Equalization (CLAHE). This particular method works on smaller windows of the image, which are then tiled together using bi-linear interpolation to avoid generating new edges. As we can see from Figure 2, this also vastly improves the contrast quality of the image and enhances the edges, making the localization of the infected area and edge-tracing easier.

We generate new images by working on the lighting of the image. Using OpenCV's addWeighted function we are able to randomly alter the brightness, which gives rise to more images as seen in Figure 3. Following the earlier trend, the parameter responsible to determine how much the brightness is adjusted is chosen at random, from a range that was determined by many trials.

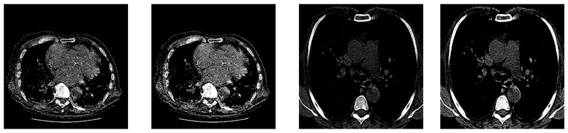

Fig. 2. A set of original images and their histogram equalized counterparts.

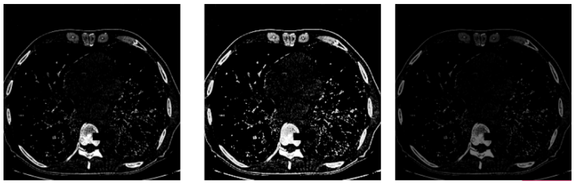

Fig. 3. An original image and the brightness altered counterparts.

D. Noise Injection

Noise injection, as shown in [6] is the addition of a matrix of randomized values, usually taken to be Gaussian Noise. Noise addition has proven to be useful, making the network resilient to environmental noise and learning more robust features. Here, we have chosen to add Gaussian noise to our image, setting the program to take a random mean and variance (within set limits that were determined through many trials, sampled from a random uniform distribution) to make the image grainier, as shown in Figure 4. As we can see, some images have lower noise levels and some have more, showing that the random noise addition works the way we want it to. Therefore, for every image in the dataset, we provide 10 new geometrically augmented images, 1 histogram equalized image, 3 lighting-altered images, and 5 noisy renditions. In total, we increase the size of the dataset by a factor of 20.

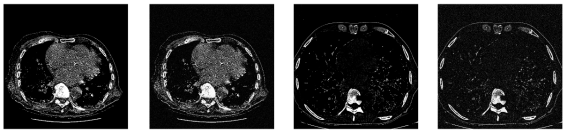

Fig. 4. Samples of images and their noisy counterparts. As we can see, the amount of noise varies, with the first example having less grains of noise and the second example being noisier.

V. Neural Network Architecture

A. UNET

UNET was designed as a form of medical image segmentation. UNET is an encoder-decoder structure with some intermediate copy and crops between the encoder and decoder sections. It converts an image to a vector through a stepwise convolution process. After each layer of filter that is applied to the input image, a feature map is produced which helps identify important aspects of the image. Once the model obtains the end vector, it then rebuilds a new image using the same convolution steps used before, but in a reversed order. Essentially, the same feature maps are used in the decoder and encoder portions of this network. The result is an image that is segmented according to prominent features which can be labeled accordingly to the intended study.

This methodology can be visualized in a ”U” pattern, which is where the name UNET is derived from. Figure 5 below shows the architectural structure of UNET.

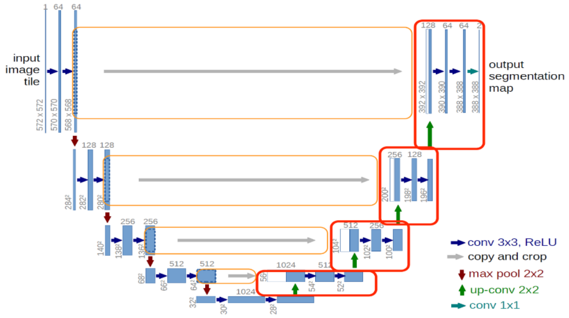

Fig. 5. Example of a UNET Structure that our model was derived from.

B. Loss Function: Sparse Categorical Cross-Entropy

UNET requires a dataset with pixel-wise annotations of the ground truth semantic categories and it must be trained using an appropriate loss function.

When we first started out, we tested between three candidate loss functions: Categorical Cross-Entropy (CCE), Tversky, and Mean Squared Error. From this group, we chose to go with CCE because it is typically used for classifying categories, where every pixel of the image receives a label. Furthermore, it was used extensively in other related papers as well which was a good indicator. However, we found that although we were on the right track with CCE, what we actually needed to use was sparse categorical cross-entropy (SCCE). CCE and SCCE use the same formula, as shown in Equation 1, but the difference lies in the format of the true labels (y_i).

$$L(y_i, q(x_i)) = -\sum_{i=1}^{N} y_i * log(q(x_i)) \tag{1}$$

For CCE, the true labels are usually in the form of a binary vector. On the other hand, for SCCE, the true labels are integers. The main reason we used SCCE over CCE is that our categories are mutually exclusive. Our labels have to be a part of individual categories otherwise they would be misclassified.

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad vs \quad \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}$$

Another benefit of SCCE is that computation becomes more streamlined because it is easier to store the integer categories than vector categories in the memory. That is why SCCE was chosen as the primary loss function used in our model.

C. Evaluation

In a majority of papers on this topic of how to evaluate the model, there were three primary methods that we saw and decided to adopt, as shown in Equations 2-4. The foundation of each stems from the true-positive, true-negative, false-positive, and false-negative classification outputs from the model, as shown in the equations below. Dice is an accuracy metric where the True positives are weighted by a factor of 2. This is done to positively reinforce when the model is behaving well. Precision is a metric that ensures that there are a low amount of false positives. Finally, Recall is a measure of how completely we can segment all categories within the image.

$$Dice = \frac{2TP}{2TP + FP + FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

VI. Results

Our team was able to successfully segment four classes given a CT scan. We shown the output of our model below in Figure 6. We provide plots for DICE Accuracy, Precision, and Recall in Figure 7. Altohugh we are able to segment with a fair about of accuracy when analyzing the categorical accuracies of each class individually we noticed classes 1 and 2 had lower accuracies and recall. We implemented a weighted loss function that weighted the less common classes with higher weights. There may be an exploration that can more efficiently choose the weightings of the individual classes to improve accuracy of the segmentation further.

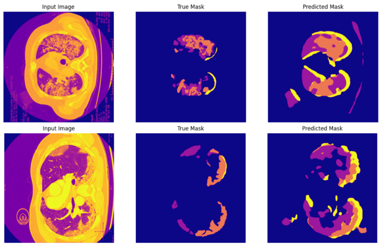

Fig. 6. Mask output images of the model.

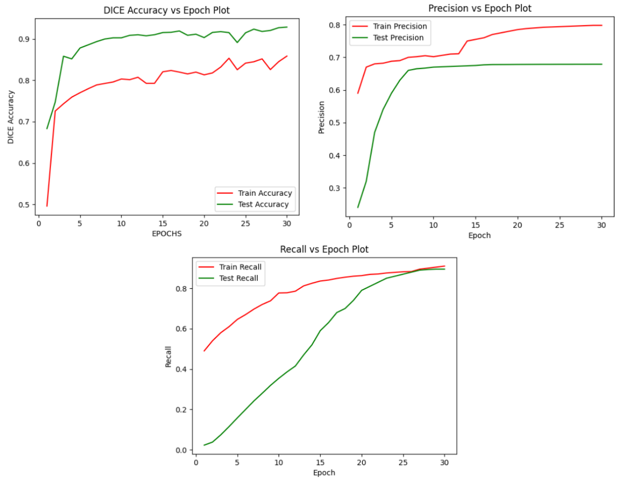

Fig. 7. Evaluation Metrics: DICE Accuracy, Precision, and Recall.

VII. Conclusion

We have built a model which segments four classes previously stated that indicate levels of COVID-19 infection. This model was quite challenging to train as it always wanted to classify all pixels as background. The use of a weighted loss function was utilized in order to enforce that we cared about the other classes more. We still believe this model can be tuned further to achieve a slightly better accuracy and prevent anomalous patches on the outskirts of the image.

References

[1] Müller, Dominik, Iñaki Soto Rey, and Frank Kramer. ”Automated Chest CT Image Segmentation of COVID-19 Lung Infection based on 3D U-Net.” arXiv preprint arXiv:2007.04774 (2020).

[2] Amyar, Amine, Romain Modzelewski, Hua Li, and Su Ruan. ”Multi-task deep learning based CT imaging analysis for COVID-19 pneumonia: Classification and segmentation.” Computers in Biology and Medicine 126 (2020): 104037.

[3] Hassantabar, Shayan, Mohsen Ahmadi, and Abbas Sharifi. ”Diagnosis and detection of infected tissue of COVID-19 patients based on lung X-ray image using convolutional neural network approaches.” Chaos, Solitons & Fractals 140 (2020): 110170.

[4] Oulefki, Adel, Sos Agaian, Thaweesak Trongtirakul, and Azzeddine Kassah Laouar. ”Automatic COVID-19 lung infected region segmentation and measurement using CT-scans images.” Pattern recognition (2020): 107747.

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation” 18 May 2015.