Semi-Supervised COVID-19 CT Image Segmentation Using Deep Generative Models

Judah Zammit
7839359
zammitj3@myumanitoba.ca
COMP 4560

Abstract

Semi-supervised learning has seen tremendous advancements in recent years. However, due to the complexity of its label space, those advancements cannot be applied to image segmentation. Despite this, it is this same complexity that makes it extremely expensive to obtain pixel-level label, making semi-supervised learning all the more appealing. This paper seeks to bridge this gap by proposing a novel model that utilizes the image segmentation abilities of the U-Net and the semi-supervised learning abilities of hierarchical graphical models for chest computed tomography(CT) images of patients with the Coronavirus Disease 2019(COVID-19). We compare the performance of our proposed model, the StichNet, to that of several traditional supervised deep-learning methods using two data sets of CT-images, one with pixel level labels and one without. We show that the StichNet is capable of showing what a CT-image would look like with infection of varying severity.

Keywords: COVID-19, image segmentation, semi-supervised learning, lung CT-images, deep learning

1. Introduction

Modern deep learning based, image segmentation techniques tend to require vast amounts of labels to be effective. However, to obtain these labels, it is necessary to have someone sit down and categorize every pixel in an image. This requires a massive amount of human effort. In the case of biomedical images, this is made worse by the fact that it is often necessary to have a panel of experts do the labeling. Therefore, any technique that has the potential to reduce the number of labeled images needed has immense value.

This issue can be seen in the diagnosis and prognosis of patients suspected to have the Coronavirus Disease 2019(COVID-19), using computed tomography(CT) scans of their lungs. A pixel-wise segmentation of these scan, identifying healthy tissue as well as parts of the lungs affected by either typical pneumonia or novel coronavirus pneumonia, can be a powerful tool for diagnosis as well as for identifying how much risk the patient is in, or will be in. Obtaining these segmentations, however, is immensely time-consuming for medical professionals to do by hand. In response to this, there has been work [1, 2, 3] in using deep learning models for image segmentation to automate this process. Despite there being massive data sets of CT scans, these models can only be trained on CT-images that have been hand labeled by skilled radiologists, severely limiting the amount of usable data.

Semi-supervised learning has the potential to alleviate this issue. A semi-supervised model has the ability to learn from both unlabeled and labeled images simultaneously, drastically reducing the number of labeled images needed to achieve satisfactory performance. For this reason, there has been a surge of research into semi-supervised learning in recent years.

Unfortunately, the vast majority of this research has been in the domain of image classification, where the label is simply a single category for each image. The assumption can be made that each of these categories are equally likely to occur. Even though this assumption is very close to the reality, it still allows for easy to compute, closed form calculations. Modern semi-supervised techniques rely heavily on this fact.

A similar assumption *cannot* be made in the case of image segmentation. This is for several critical reasons. Firstly, due to the fact that *each* pixel has a label, the number of unique segmentations is exponentially larger than the number of unique image-level labels. Furthermore, very few of these unique segmentations are realistic. For example, a set of pixels that have been give the *dog* label but are in the shape of a human is not a realistic segmentation. This is important because it completely removes our ability to assume that each unique segmentation is equally likely to occur. Lastly, the label for each pixel is heavily dependent on the labels for the other pixels in the image, removing the possibility of making any independence assumptions. For these reasons, modern semi-supervised techniques tend to fall flat when used for image segmentation.

Though problematic, the issues mentioned above are not at all new. The same issues are encountered while trying to find a distribution capable of modeling images. The variational approach [4, 5] handles this by finding a latent representation of the image as well as a deep learning based, functional mapping between the image and its latent representation. You are then free to make simplifying assumptions about the latent space's distribution without making any assumptions about the images' distribution. This paper proposes a model that will use this technique to find a latent representation for an images' segmentation and then use modern semi-supervised, deep learning techniques with this latent representation instead of the far more complex, image segmentation.

Due to the recent outbreak of COVID-19, chest computed tomography(CT) has been used as an important tool for COVID-19 diagnosis and monitoring its progress. Therefore, we will use the above approach to propose a novel model for CT-image segmentation of suspected and confirmed COVID-19 patients.

In summary, our contributions are:

- We propose a new method for semi-supervised deep learning in domains where the label space is complex and high dimensional, by modeling the label as well as the features
- We propose a semi-supervised, deep learning model well suited for the CT-image segmentation of COVID-19 patients

2. Related Work

2.1. Supervised Image Segmentation

The U-Net [6] is a deep learning based image segmentation model that has seen great success on medical imagery tasks. It utilizes an encoder-decoder style architecture with skip connections between the encoder and the decoder. The structure of this architecture is of particular importance to this paper.

The ResNet [7] is a deep convolutional network that, though origionally used for image classification, has excelled at being used for as an encoder network for *all* tasks that work with images, including segmentation. It consists of an arbitary amount of residual blocks, which are made up of several convolutional layers, connected in the traditional sequential manner as well as novel skip connections between earlier and later layers. Though powerful, the ResNet can take a tremendous amount of time to train. The MobileNetv2 [8] achieves only slightly worse resulst as an encoder, with a significant reduction in training time.

The ResNext [9] further improves on the ResNet by, instead of having a strictly sequential ordering of convolutional layers, allowing for multiple tracks of layers whose outputs are combined periodically.

Zhang, et al. [1] have used several deep learning based, supervised segmentation models [6, 10, 11] to predict a segmentation for a CT-image of a patient's lungs. Fan et al. [3] and Chen et al. [12] both propose novel supervised segmentation models that have been hand crafted to perform well on chest CT images. Though impressive, these models are still limited by the number of CT images with pixel-wise labels.

2.2. Deep Generative Models

The Variational Autoencoder proposes a mathetically rigourus way of learning a generative model that utilizes deep networks. It does this learning the parameters to a deep generative model through assuming a latent variable and using an approximate deep inference model. Though effective, it lacks the expressivity needed to model complex data, such as CT-images. The Ladder Variational Autoencoder [13] seeks to remedy this by introducing a novel training technique that allows for deep hierarchies of latent variables.

2.3. Semi-Supervised Image Segmentation

There is a plethora of papers proposing deep learning models that use image-level labels as a supervisory for the task of image segmentation. They, however, *do not* utilize completely unlabeled images. This task is sometimes referred to as **pure** or **true semi-supervision**, and there are precious few published papers that tackle it.[14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]

The above **purely semi-supervised** models tend to tackle the problem using some form of adversarial training, self-training, clustering or multi-view training. Many papers [24, 23, 21, 18, 15] use an adversarially trained discriminator deep convolutional network to ensure that the prediction of some segmentation model is realistic. This scheme allows them to train their network on unlabeled photos by leveraging the fact that, even if you dont know the ground truth, it should at least belong to the same distribution as the ground truth for the labeled images. The main drawback to this technique is that it can be very difficult to get a model with an adversarial component to converge to a solution.

Other papers [14, 19] use the fact that images–labeled or unlabeled–that have been determined to be similar by some deep learning based, unsupervised clustering algorithm should also be close to each other in various latent and feature spaces. These techniques are dependent on how you define "close" which can be quite difficult for data that is as high dimensional as images, causing the performance of these models to be underwhelming.

Pseudo-labeling [25] is a commonly used semi-supervised learning technique where you train a fully supervised deep network and then use it to make predictions on some unlabeled data. The network is then retrained using the models most confident predictions. However, if this prediction is of low quality, then this scheme will continuously reinforce this bad behavior to disastrous effect. As a result, pseudo-labeling is typically considered the least effective, but simplest, semi-supervised technique. There are several papers [3, 17] that use this general scheme with some significant modifications.

Mondal et al.'s [15] propose a model that ensures that predicted labels are realistic, through adversarial training. It also ensures that the predicted segmentation masks can be used for the reconstruction of the original images. As with the aforementioned adversarial techniques, this model has limited practical use, due to the great implementation difficulties that comes from adversarial training.

As with this paper, many papers [2, 3] seek to utilize unlabeled CT images. Shan et al.[2] use an intriguing human-in-the-loop(HITL) strategy. This strategy entails training a deep learning based, segmentation network on a small data set of pixel-wise labeled data, then using this network to make prediction on a large unlabeled data set. These predictions are then refined by a skilled radiologist and included in the pixel-wise labeled data set. The network is retrained, and this process repeats until satisfactory performance is achieved. Though the labeling effort is significantly reduced, this technique still requires some manual labeling effort. Additionally, many research groups will simply *not* have access to a radiologist.

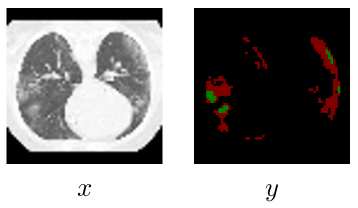
Fan et al.[3] use pseudo-labeling in its most rudimentary form. Despite this, they achieved a sizable increase in segmentation performance compared to their fully supervised baseline. This makes it quite motivating to employ a more sophisticated semi-supervised technique, as pseudo-labeling is far from capable of making full use of these unlabeled images.

3. Proposed Method

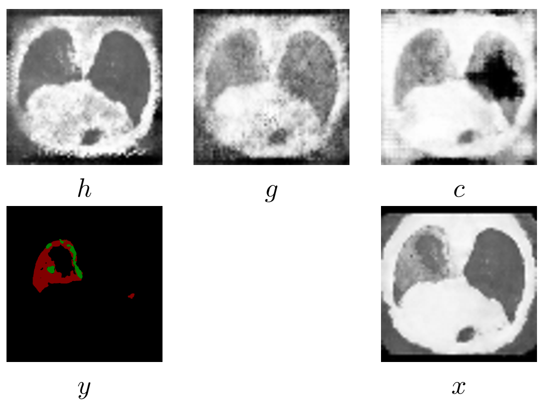
In this section we will introduce the theory, implementation and optimization of the *StichNet*.

3.1. Problem Space

Suppose we have a data set, $D_{UN} = \{(x^{(i)})\}_{i=0}^{N}$, of N CT-images, where $x^{(i)}$ denotes the i^{th} CT-image in the data set. We will assume that $x^{(i)}$ is a high-dimensional vector with entries ranging

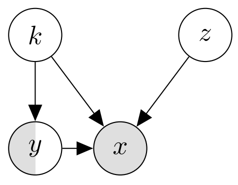


(a) A CT-image(x) with its segmentation(y).



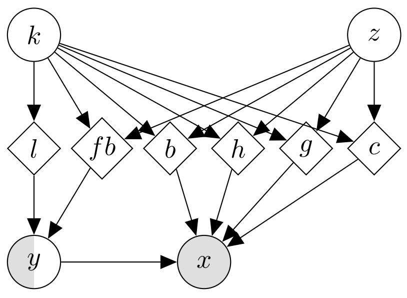
(b) Stylistic generation. The healthy, ground glass opacity and consolidation styles are on the top row, respectively.

Figure 1: Visualization of the data and StichNet's outputs. For segmentations, ground glass opacity is shown in red, consolidation in green and healthy tissue in black.



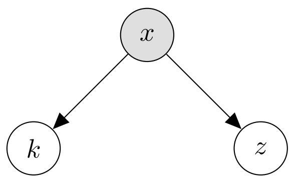
(a) Generative Model.

$$\begin{tikzcd} k \arrow[d] \arrow[dr] & & z \arrow[dl] \\ y \arrow[r] & x & \end{tikzcd}$$



(b) Updated Generative Model.

$$\begin{tikzcd} k \arrow[d] \arrow[dr] \arrow[drr] \arrow[drrr] \arrow[drrrr] \arrow[drrrrr] & & & & & z \arrow[dllll] \arrow[dlll] \arrow[dll] \arrow[dl] \arrow[d] \\ l \arrow[d] & fb \arrow[dl] & b \arrow[d] & h \arrow[dl] & g \arrow[dll] & c \arrow[dlll] \\ y \arrow[rr] & & x & & & \end{tikzcd}$$



(c) Inference Model.

$$\begin{tikzcd} & x \arrow[dl] \arrow[dr] & \\ k & & z \end{tikzcd}$$

Figure 2: Hierarchical graphical models. Latent, partially observed and observed variables are show with clear, half-filled and filled, respectively. Arrows and diamond nodes represent functional mappings.

from zero to one.

Further suppose that we have a data set, $D_{LAB} = \{(x^{(i)}, y^{(i)})\}_{i=0}^{M}$, of M CT-images along with their associated segmentation, $y^{(i)}$. We will assume that $y^{(i)}$ is of the same dimension as $x^{(i)}$ and has entries that belong to the set $\{0, 1, 2\}$.

Here, if the n^{th} entry of $y^{(i)}$ is equal to zero, then this indicates that the n^{th} entry of $x^{(i)}$ is belongs to the healthy class. Furthermore, one and two correspond to the ground-glass opacity and consolidation class, respectively.

This is depicted in Figure 1a, with the healthy class appearing in black, the ground-glass opacity class appearing in red and the consolidation class appearing in green.

3.2. Modeling the Conditional Distribution

We wish to obtain a model capable of taking a CT-image(x) and outputting an accurate segmentation(y). In other words, we wish to approximate the ground-truth $p(x|y)$ conditional distribution.

Though not typically phrased in these terms, supervised deep-learning techniques do this by introducing the following approximation to this distribution:

$$p_\theta(y|x) = CAT(y|f_\theta(x)), \tag{1}$$

where CAT is the Categorical distribution and f_θ is some complex function. Due to their tremendous success on image data, f_θ is typically chosen to be a convolutional neural-network.

These supervised techniques then wish to find the parameters θ that best explains the data we are given. This is done by maximizing the following objective using a numerical approximation algorithm such as gradient descent:

$$J = \sum_{D_{LAB}} log p_\theta(y^{(i)}|x^{(i)}). \tag{2}$$

Phrased in this way, the drawback to these supervised techniques is obvious. They can only use the labeled data set D_{LAB}.

To remedy this, instead of approximating $p(y|x)$, we can model joint distribution $p(x, y)$ and derive the conditional distribution $p(y|x)$ from it. This allows us to use both D_{LAB} and D_{UN} by treating y as a latent variable in the latter case.

3.3. Modeling the Joint Distribution

To effectively model $p(x, y)$, we will assume the existence of two variables latent z and k, instead modeling $p(x, y, z, k)$. Furthermore, we will assume that each of the data points, $(x^{(i)}, y^{(i)}, z^{(i)}, k^{(i)})$, were generated in the following way:

$$\begin{aligned} z^{(i)} &\sim p(z), \\ k^{(i)} &\sim p(k), \\ y^{(i)} &\sim p_\theta(y|k^{(i)}), \\ x^{(i)} &\sim p_\theta(x|z^{(i)}, k^{(i)}, y^{(i)}), \end{aligned} \tag{3}$$

where $p(z)$ and $p(k)$ are assumed follow a Unit Gaussian distribution and $p_\theta(\cdot)$ is assumed to be some distribution parameterized by θ(depicted in Figure 2a).

To generate x we will first generate three stylistic representations–referred to as h, g and c–of x. These stylistic representations of x show you what the image would look like if the entire lung were healthy, ground-glass opacity and consolidation, respectively. We then reconstruct x by choosing the pixel from the style associated with the label predicted by y. Examples of these styles are shown in Figure 1b.

We will use this, as well as the following definitions, to define p_θ:

$$\begin{aligned} p_\theta(\{h, g, c\}|z, k) &= BERN(\{h, g, c\}|\lambda_\theta(z, k)), \\ \Phi(y, h, g, c) &= \begin{cases} h & \text{if } y = 0, \\ g & \text{if } y = 1, \\ c & \text{if } y = 2, \end{cases} \end{aligned} \tag{4}$$

where $BERN$ denotes the Bernoulli distribution and $\lambda_\theta(z, k)$ is some complex function parameterized by θ.

Finally, we define p_θ as

$$\begin{aligned} p_\theta(y|k) &= CAT(y|\pi_\theta(k)), \\ p_\theta(x|y, h, g, c) &= BERN(x|\Phi(y, h, g, c)), \end{aligned} \tag{5}$$

Where $\pi_\theta(k)$ is some complex function parameterized by θ. This give our latent variables a meaningful interpretation. The stylistic information, such as the size, location, general shape of the lung, is contained in the latent variable z, whereas the semantic information, such as which parts of the lung are currently affected by COVID-19, is contained in k.

3.4. Lung Separation

We know that lesions must occur *inside* of the lungs, however the model outlined above has no way of enforcing this. We remedy this by including a fourth label in $y's$ label set.

This label corresponds to healthy portions *of the lung*. This differs from our previous healthy label as that one was used to label areas outside of the lung, as well as inside, as healthy. In summary, we now have four labels, background, ground-glass opacity, consolidation and healthy.

To incorporate this new label in our model, we make the following changes:

- We predict which areas are inside the lung and which areas are outside of the lung using z and k
- We still predict three labels for y, but refer to it as l, as y now contains four labels
- We use this label to immediately predict that the corresponding pixel is healthy when it is outside of the lung, but defer to $l's$ prediction if it is inside of the lung
- We introduce a fourth stylistic representation corresponding to the background class

This is depicted in Figure 2b with diamond nodes denoting deterministic mappings.

We will use the following definitions to redefine our generative model p_θ:

$$\begin{aligned}
p_\theta(l|k) &= CAT(l|\pi_\theta(k)),\\
p_\theta(fb|k) &= CAT(fb|\pi_\theta(z,k)),\\
p_\theta(\{b,h,g,c\}|z,k) &= BERN(\{b,h,g,c\}|\lambda_\theta(z,k)),\\
\Psi(fb,l) &= (fb_0, fb_1 \times l_0, fb_1 \times l_1, fb_1 \times l_2)\\
\Phi(y,b,h,g,c) &= \begin{cases} b & \text{if } y=0,\\ g & \text{if } y=1,\\ c & \text{if } y=2,\\ h & \text{if } y=3.\end{cases}
\end{aligned} \tag{6}$$

Finally, our redefined p_θ is

$$\begin{aligned}
p_\theta(y|k,z) &= CAT(y|\Psi(fb,l)),\\
p_\theta(x|y,b,h,g,c) &= BERN(x|\Phi(y,b,h,g,c)).
\end{aligned} \tag{7}$$

3.5. Optimization of p_θ

We now have a generative model that is well suited to CT-image segmentation. What remains is outlining an effective means for finding the values of θ that best explains our observed data. Concretely, we wish to solve

$$\begin{aligned}
\max_\theta &\textstyle\sum_{D_{UN}} log p_\theta(x^{(i)}) + \sum_{D_{LAB}} log p_\theta(x^{(i)}, y^{(i)})\\
=&\textstyle\sum_{D_{UN}} log \int_z \int_k \sum_y p_\theta(x^{(i)}, y, z, k) dz dk\\
+&\textstyle\sum_{D_{LAB}} log \int_z \int_k p_\theta(x^{(i)}, y^{(i)}, z, k) dz dk.
\end{aligned} \tag{8}$$

The existence of latent variable, and, by extension, the need to integrate over them, makes this objective completely intractable.

We instead optimize a variational lower bound on the log likelihood of p_θ. Concretely, we optimize,

$$\begin{aligned}
&log p_\theta(x^{(i)})\\
&\geq E_{q_\phi(z,k|x^{(i)})}\left[log \frac{p_\theta(x^{(i)}, y^{(i)}, z, k)}{q_\phi(z,k|x^{(i)})}\right]\\
&log p_\theta(x^{(i)})\\
&\geq E_{q_\phi(z,k|x^{(i)})}\left[log \frac{\sum_y p_\theta(x^{(i)}, y, z, k)}{q_\phi(z,k|x^{(i)})}\right].
\end{aligned} \tag{9}$$

Though q_ϕ can be any function of the latent variables, this lower bound is exactly equal to the true log likelihood when q_ϕ is equal to $p'_\theta s$ posterior, $p_\theta(z,k|x,y)$. Therefore, q_ϕ has the interpretation of being an approximation to the posterior. When we implement the q_ϕ, we will keep this fact in mind.

We can further increase tractability by approximating the calculation of the expectation over q_ϕ. We do this by taking a Monte-Carlo sample from q_ϕ and evaluating the expectation with just this sample. This approximation can be made more precise by taking multiple samples and averaging the expectation, but, for our work, we used only one. With this, we arrive at our final objective,

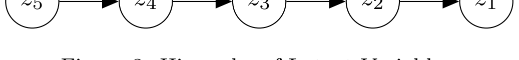
$$\begin{aligned}
&E_{q_\phi(z,k|x^{(i)})}\left[log \frac{p_\theta(x^{(i)}, y^{(i)}, z, k)}{q_\phi(z,k|x^{(i)})}\right]\\
&\approx q_\phi(z^{(i)}, k^{(i)}|x^{(i)})\left[log \frac{p_\theta(x^{(i)}, y^{(i)}, z^{(i)}, k^{(i)})}{q_\phi(z,k|x^{(i)})}\right]\\
&\equiv J_{LAB},\\
&E_{q_\phi(z,k|x^{(i)})}\left[log \frac{\sum_y p_\theta(x^{(i)}, y, z, k)}{q_\phi(z,k|x^{(i)})}\right]\\
&\approx q_\phi(z^{(i)}, k^{(i)}|x^{(i)})\left[log \frac{\sum_y p_\theta(x^{(i)}, y, z^{(i)}, k^{(i)})}{q_\phi(z,k|x^{(i)})}\right]\\
&\equiv J_{UN},\\
&\text{where}\\
&z^{(i)}, k^{(i)} \sim q_\phi(z,k|x^{(i)}).
\end{aligned} \tag{10}$$

3.6. Hierarchy of Latent Variables

As it stands, our generative model takes samples from two Unit Gaussian distributions, applies several complex functions to them, and then combines the outputs of these functions to generate a new CT-image and segmentation. It is worth asking, is this model capable of modeling the highly complex data of CT-images and segmentations?

Currently, the answer to this depends solely on the complexity of the functions. However, we can also increase the complexity of our model by increasing the complexity of our latent variables. Specifically, by introducing a hierarchy of latent variables to replace our Unit Gaussian k and z. We will discuss a hierarchy of variables for z however the same will be done for k.

We replace z with five variables, z_5, z_4, z_3, z_2 and z_1. Each variable depends only on the previous variable, with z_5 being independent(depicted in Figure 3).



$$\begin{tikzcd} z_5 \arrow[r] & z_4 \arrow[r] & z_3 \arrow[r] & z_2 \arrow[r] & z_1 \end{tikzcd}$$

Figure 3: Hierarchy of Latent Variables.

The distribution of these variables are Diagonal Gaussian, with the bottom variable being Unit

Gaussian. Concretely,

$$\begin{aligned}
p_\theta(z_i|z_{i-1}) &= GAUS(z_i|\sigma_{p,i}(z_{i-1}), \mu_{p,i}(z_{i-1})), \\
&\text{where} \\
\sigma_{p,i}(z_{i-1}) &= ResBlock_\theta(Shared) \\
\mu_{p,i}(z_{i-1}) &= ResBlock_\theta(Shared) \\
Shared &= ResBlock_\theta(Transpose(z_{i-1})).
\end{aligned} \tag{11}$$

ResBlock denotes a seven-layer deep block from the ResNet and *Transpose* denotes the transposed convolution, which increases the dimensionality of the latent variable.

3.7. Specification of q_ϕ

It is crucial that our inference network is able to infer high-quality values for our latent variables from the CT-image. Due to its great success in medical image segmentation, we used the U-Net for this task.

The U-Net consist of an encoder, $Enc(x)$, that takes a CT-image and outputs five encoded versions, $e_1, ...$ and e_5, of it. For our work, we used the deep convolutional network, MobileNetV2, as our encoder due to its efficiency. In addition, it consists of a decoder, $Dec(e_1, .., e_5)$, which takes the encoded versions of the CT-image and output a segmentation.

We can, then, define q_θ as

$$q_\phi(z_i|x) = GAUS(z_i|\sigma_{q,i}(e_i), \mu_{q,i}(e_i)), \tag{12}$$

where e_i is the output of $Enc_\phi(x)$ and $\sigma_{q,i}$ and $\mu_{q,i}$ are defined as in section 3.7. Note that k_i is inferred similarly, however, it uses a *different* encoder. This is depicted in Figure 2c.

Due to its success in optimizing hierarchies of latent variables, we use the same technique found in the Ladder Variational Autoencoder. Modifying q_ϕ as follows:

$$\begin{aligned}
q_\phi(z_i|x) &= GAUS(z_i|\sigma_{q,i}, \mu_{q,i}), \\
\sigma_{q,i} &= \frac{1}{\hat{\sigma}_{q,i}(e_i)^{-2} + \sigma_{p,i}(z_{i-1})^{-2}}, \\
\mu_{q,i} &= \frac{\hat{\mu}_{q,i}(e_i)\hat{\sigma}_{q,i}(e_i)^{-2} + \mu_{p,i}(z_{i-1})\sigma_{p,i}(z_{i-1})^{-2}}{\hat{\sigma}_{q,i}(e_i)^{-2} + \sigma_{p,i}(z_{i-1})^{-2}},
\end{aligned} \tag{13}$$

where $\sigma_{p,i}(z_{i-1})$, $\mu_{p,i}(z_{i-1})$, $\hat{\sigma}_{q,i}(e_i)$ and $\hat{\mu}_{q,i}(e_i)$ are defined as in section 3.7. We found this modification to be crucial to the training of our model.

3.8. Specification of λ_θ and π_θ

We have yet to define $\lambda_\theta(z, k)$ and $pi_\theta(k)$–recall that these functions are used to generate the stylistic images and the segmentation, respectively. With the decoder of the U-Net defined, we can now do this:

$$\begin{aligned}
\lambda_\theta(z, k) &= Dec_\theta(C(z_1, k_1), ..., C(z_5, k_5)), \\
\pi_\theta(z, k) &= Dec_\theta(C(z_1, k_1), ..., C(z_5, k_5)), \\
\pi_\theta(k) &= Dec_\theta(k_1, ..., k_5),
\end{aligned} \tag{14}$$

where $C(\cdot)$ concatenates the inputs.

Note that each stylistic image and segmentation requires its own decoder. One might be reasonably concerned that this would result in our model having too many parameters. In our case, however, the vast majority of our models parameters are in the encoder and the mappings between latent variables. The decoder contains a relatively small number. This allows us to use many of them without much cause for concern.

4. Experiments

In this section we define the experimental setting, evaluate the performance of the proposed method on an amalgamated data set using four evaluation metrics, and compare this performance to two other baseline methods.

4.1. Data Sets

For the evaluation of the proposed method we collected a data set of segmented CT-images and unsegmented CT-images. We use the COVID-19 CT Segmentation data set[1] for our segmented data, and the ICTCF data set [26] for our unsegmented data. We use 638 images from the COVID-19 CT Segmentation data set as our training data, 114 images for our validation data, and 117 images for our testing data. All 6654 CT-images of the unsegmented ICTCF data set was used for training. The final set is summarized in Table 1.

4.2. Evaluation Metrics

We employ the following four evaluation metrics: the F1-Score, Recall, Precision, and Intersection over Union(IoU).

[1] http://medicalsegmentation.com/covid19/

Table 1: Med-Seg refers to the COVID-19 CT Segmentation data set and ICTCF refers to the ICTCF data set [26].

Data Split	Source	Segmented	Images	Patients
Training	Med-Seg	Yes	698	39
	ICTCF	No	6654	1338
Validation	Med-Seg	Yes	114	35
Testing	Med-Seg	Yes	117	35

1) The F1-Score (F1): The F1-Score, also called the Dice Coefficient, was used to measure the overlap between the ground-truth infected region(T) and the predicted infected region(P). It is defined as:

$$F1 = \frac{2 \cdot |T \cap P|}{|T| + |S|}, \tag{15}$$

where $| \cdot |$ is the operator that calculates the number of pixels in the given region, and $\cap$ is the intersection operator.

2) Recall (Rec.): The Recall, also called Sensitivity, Hit Rate, or True Positive Rate, was used to measure what proportion of the ground-truth infected region(T) was present in the predicted infected region(P). It is defined as

$$Rec. = \frac{|T \cap P|}{|T|}. \tag{16}$$

3) Precision (Prec.): The Precision was used to measure what proportion of the predicted infected region(P) was present in the ground-truth infected region(T). It is defined as

$$Prec. = \frac{|T \cap P|}{|P|}. \tag{17}$$

4) Intersection over Union(IoU): The Intersection over Union is another way of measuring the overlap between the ground-truth infected region(T) and the predicted infected region(P). It is defined as:

$$IoU = \frac{|T \cup P|}{|T \cap P|}, \tag{18}$$

where $\cup$ is the union operator.

The *Mean* and *Error* of these metrics are defined as follows:

Let $M((x, y))$ be the value of the relevant evaluation metric calculated for the data point (x, y). Then let $Metric = \{M((x_i, y_i))\}_{(x_i,y_i)} \in D_{Val}$, where D_{Val} is the validation data set.

1) Mean: Then the *Mean* is simply $\sum_{Metric} \frac{M((x_i,y_i))}{|Metric|}$.
2) Error: Let *SE* be the standard error of the set *Metric*. Then the *Error* is $SE \times 1.96$.

Note that $Mean \pm Error$ is the 95% confidence interval.

4.3. Compared Methods

```
  Input: Labeled training data $D_{LAB}$ and
         unlabeled training data $D_{UN}$
  Output: Trained U-Net+PL
1 Construct a training data set $D_{Training}$
   using all the labeled CT images from $D_{LAB}$
2 Train our model U-Net+PL using $D_{Training}$
3 repeat
4     Perform testing using the trained model
       U-Net+PL and $K$ CT images randomly
       selected from $D_{UN}$, which yields
       network-labeled data $D_{Net-labeled}$,
       consisting of $K$ CT images with
       psuedo-labels
5     Enlarge the training data set using
       $D_{Net-labeled}$, that is,
       $D_{Training} = D_{Training} \cup D_{Net-labeled}$
6     Remove the $K$ testing CT images from
       the $D_{UN}$
7     Fine-tune U-Net+PL
8 until $D_{UN}$ is empty;
9 return Trained model U-Net+PL
```
Algorithm 1: Psuedo-Labeling Procedure.

We compare our model to the U-Net [6] with a ResNext50 [9]. This model will be referred to as *U-Net.* We also compare our model to the Inf-Net [3] model will be referred to as *SInfNet.*

In addition, we use the well-known semi-supervised technique, psuedo-labeling [25], with the *U-Net* to obtain a semi-supervised baseline model. This model will be refereed to as *U-Net+PL.*

For the *U-Net+PL* model, we follow a procedure similar to Fan et al.'s [3]. We first train the model on a labeled data set in the standard manner. We then obtain this models prediction on 16 random CT-images from an unlabeled data set. These predictions are included in the labeled data set and the model is retrained on this updated data set. This procedure will be repeated until 1600 unlabeled CT-images have been added to the labeled data set. This procedure is described in Algorithm 1.

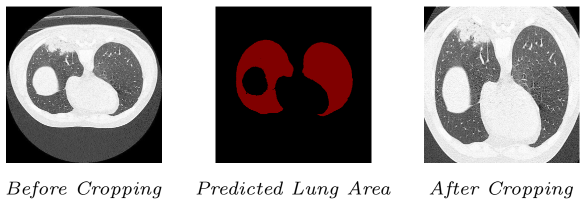


Figure 4: Lung cropping. Predicted lung area is shown red.

4.4. *Experimental Setting*

We ensure a fair comparison by using the same experimental setting for the compared models. These settings are as follows:

- The images in the data set are cropped to include only the lungs. An example of this can be seen in Table 4. This is done by training a *U-Net* model (see section 4.3 for the definition of this model) to predict which pixels belong to the lung. This model is then used to find a bounding box that includes all the predicted pixels. We then crop the images to this bounding box's dimensions. This is depicted in Figure 4.
- The following two data augmentations are used:
  - *Cutout*: A random square section of the image is cutout. The size of this square is randomly chosen to be one of, 64×64, 128×128, 256×256. This augmentation has a 25% chance of being applied
  - *Flip*: The image is randomly flipped horizontally or vertically.
- For speed of training, the images are downsized to 64×64. Note that, to obtain a full resolution prediction, the image is separated into 64×64 squares and predictions are generated for those squares.
- All models use Batch Normalization [27] at all relevant layers.

All models were implemented using Tensorflow [28] and trained using an NVIDIA RTX 2070 graphics card. In addition, the *U-Net* and *U-Net+PL* made use of the segmentation-models [29] library. All data augmentations were done using the albumentations [30] library and our evalutation metrics were either implemented from scratch or with scikit-learn [31].

5. Results and Discussion

The results of our experiments are reported in Table 3. Our model shows negligible improvements over the supervised U-Net baseline. This indicates that our model is learning from the labeled data, but not the unlabeled data.

When trained on only the labeled data, our model predicts styles that are clearly associated with the appropriate lesions, effectively allowing you to see what a CT-image would look like if it were entirely filled with the associated lesion. Because of this, our model seems to perform exactly as expected on the labeled data.

When trained on only the unlabeled data, our model learns unique and meaningful styles, learning a meaningful clustering of the data. This is exactly what we would expect from training with no labeled data.

Based of these two observations, when trained on both labeled and unlabeled data, our model should learn to predict styles that are associated with lesions, for *both* the labeled data *and* the unlabeled data. Our model achieves this on the labeled data, however, on the unlabeled, all the styles are identical and the segmentations are the exact same for every image.

This seems to indicate that the fundamental idea is sound, but that further work needs to be done before the StichNet can outperform the supervised network.

6. Conclusion

In conclusion, we proposed the StichNet, a model that shows promising potential, but will need further improvements before it can outperform the supervised baseline. Without further

Table 2: Quantitative results of GGO, Consolidation, Background, and the overall average on the test data set. The best result is shown in bold.

		Ground-Glass Opacity				Consolidation			
Methods		IoU	F1-Score	Precision	Recall	IoU	F1-Score	Precision	Recall
U-Net	Mean	0.18	0.26	0.41	0.22	**0.26**	**0.35**	0.46	**0.32**
	Error	±0.04	±0.06	±0.09	±0.05	**±0.08**	**±0.10**	±0.12	**±0.09**
U-Net+PL	Mean	0.12	0.18	0.33	0.14	**0.26**	0.33	0.45	**0.32**
	Error	±0.04	±0.05	±0.09	±0.05	**±0.09**	±0.11	±0.12	**±0.11**
SInfNet	Mean	**0.27**	**0.38**	**0.58**	**0.41**	0.22	0.29	**.61**	.31
	Error	**±0.04**	**±0.05**	**±0.07**	**±0.06**	±0.07	±0.08	**±0.10**	±0.08
StichNet	Mean	0.08	0.13	0.33	0.11	0.04	0.06	0.15	0.05
	Error	±0.02	±0.04	±0.07	±0.03	±0.02	±0.03	±0.06	±0.03

		Background				Overall			
Methods		IoU	F1-Score	Precision	Recall	IoU	F1-Score	Precision	Recall
U-Net	Mean	0.75	0.86	0.76	**1.00**	0.40	0.49	0.54	0.51
	Error	±0.02	±0.01	±0.02	**±0.00**	±0.05	±0.07	±0.08	±0.05
U-Net+PL	Mean	0.75	0.86	0.75	**1.00**	0.38	0.46	0.51	0.49
	Error	±0.02	±0.01	±0.02	**±0.00**	±0.05	±0.06	±0.02	±0.05
SInfNet	Mean	**1.00**	**1.00**	**1.00**	**1.00**	**0.55**	**0.50**	**0.73**	**0.57**
	Error	**±0.00**	**±0.00**	**±0.00**	**±0.00**	**±0.04**	**±0.04**	**±0.06**	**±0.05**
StichNet	Mean	0.95	0.98	0.97	0.98	0.36	0.39	0.48	0.38
	Error	±0.01	±0.01	±0.01	±0.01	±0.02	±0.03	±0.05	±0.02

Table 3: Quantitative results of GGO, Consolidation, Background, and the overall average on the validation data set. The best result is shown in bold.

		Ground-Glass Opacity				Consolidation			
Methods		IoU	F1-Score	Precision	Recall	IoU	F1-Score	Precision	Recall
U-Net	Mean	**0.26**	0.35	**0.52**	**0.31**	**0.04**	**0.07**	**0.15**	**0.08**
	Error	**±0.06**	±0.07	**±0.09**	**±0.07**	**±0.03**	**±0.04**	**±0.08**	**±0.04**
U-Net+PL	Mean	**0.26**	**0.36**	**0.52**	**0.31**	**0.04**	**0.07**	**0.15**	0.07
	Error	**±0.06**	**±0.07**	**±0.09**	**±0.07**	**±0.03**	**±0.04**	**±0.08**	±0.04
StichNet	Mean	0.08	0.14	0.28	0.11	0.03	0.05	0.112	0.05
	Error	±0.02	±0.04	±0.07	±0.04	±0.02	±0.03	±0.05	±0.03

		Background				Overall			
Methods		IoU	F1-Score	Precision	Recall	IoU	F1-Score	Precision	Recall
U-Net	Mean	**0.80**	**0.88**	**0.80**	**1.00**	**0.37**	0.43	**0.49**	**0.46**
	Error	**±0.02**	**±0.01**	**±0.02**	**±0.00**	**±0.04**	±0.04	**±0.06**	**±0.04**
U-Net+PL	Mean	**0.80**	**0.88**	**0.80**	**1.00**	**0.37**	**0.44**	**0.49**	**0.46**
	Error	**±0.02**	**±0.01**	**±0.02**	**±0.00**	**±0.04**	**±0.04**	**±0.06**	**±0.04**
StichNet	Mean	0.95	0.97	0.96	0.99	0.35	0.39	0.45	0.38
	Error	±0.01	±0.01	±0.01	±0.01	±0.02	±0.03	±0.04	±0.03

improvements, the StichNet's performance, as well as that of any supervised model, is still limited by the availability of labeled data.

We showed that our model can effectively show what a patient's lung would look like if they had COVID-19 induced pneumonia, even if the patient is completely healthy. It is possible that this could be used to predict the likelihood of an undesirable clinical outcome *if* a healthy patient were to contract the disease, allowing for the identification of vulnerable populations. Future work will be

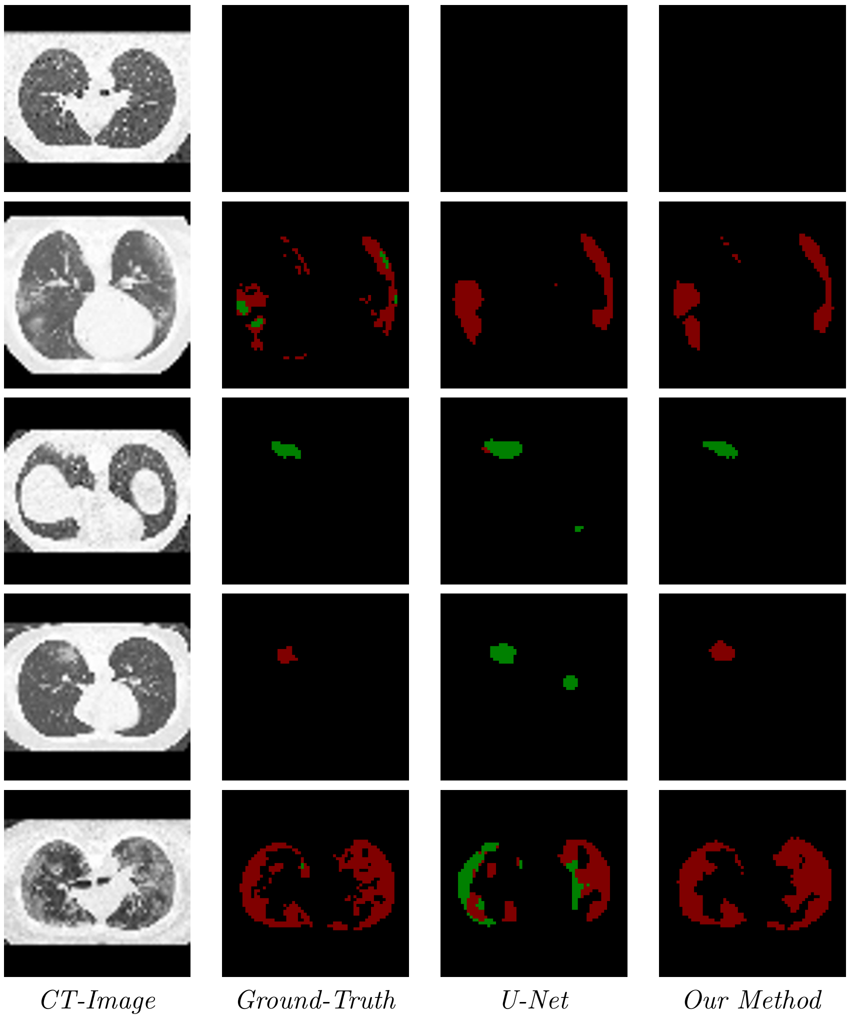


Figure 5: Visual comparison of the segmentation results, where the red and green labels indicate the GGO and Consolidation, respectively.

needed to confirm this.

7. Acknowledgments

I would like to thank Dr. Pingzhao Hu and Dr. Carson Leung for supervising this project and Qian Liu and Daryl Fung for their contributions to it. In addition, I would like to thank Dr. Ruppa K. Thulasiram for allowing me to pursue this project as an industry research project.

References

[1] K. Zhang, X. Liu, J. Shen, Z. Li, Y. Sang, X. Wu, Y. Cha, W. Liang, C. Wang, K. Wang, *et al.*, "Clinically applicable ai system for accurate diagnosis, quantitative measurements and prognosis of COVID-19 pneumonia using computed tomography," *Cell*.

[2] F. Shan, Y. Gao, J. Wang, W. Shi, N. Shi, M. Han, Z. Xue, D. Shen, and Y. Shi, "Lung infection quantification of COVID-19 in CT images with deep learning," 2020.

[3] D.-P. Fan, T. Zhou, G.-P. Ji, Y. Zhou, G. Chen, H. Fu, J. Shen, and L. Shao, "Inf-net: Automatic COVID-19 lung infection segmentation from CT images," 2020.

[4] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013.

[5] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," vol. 32 of *Proceedings of Machine Learning Research*, (Bejing, China), pp. 1278–1286, PMLR, 22–24 Jun 2014.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*,

pp. 234–241, Springer, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[8] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[9] S. Xie, R. Girshick, P. Dollr, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5987–5995, 2017.

[10] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.

[11] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[12] X. Chen, L. Yao, and Y. Zhang, “Residual attention u-net for automated multi-class segmentation of COVID-19 chest CT images,” *arXiv preprint arXiv:2004.05645*, 2020.

[13] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther, “Ladder variational autoencoders,” in *Advances in neural information processing systems*, pp. 3738–3746, 2016.

[14] H. Al-Dmour and A. Al-Ani, “Mr brain image segmentation based on unsupervised and semi-supervised fuzzy clustering methods,” in *2016 international conference on digital image computing: techniques and applications (DICTA)*, pp. 1–7, IEEE, 2016.

[15] A. K. Mondal, A. Agarwal, J. Dolz, and C. Desrosiers, “Revisiting cyclegan for semi-supervised segmentation,” *arXiv preprint arXiv:1908.11569*, 2019.

[16] W. Bai, O. Oktay, M. Sinclair, H. Suzuki, M. Rajchl, G. Tarroni, B. Glocker, A. King, P. M. Matthews, and D. Rueckert, “Semi-supervised learning for network-based cardiac mr image segmentation,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 253–260, Springer, 2017.

[17] I. Radosavovic, P. Dollár, R. Girshick, G. Gkioxari, and K. He, “Data distillation: Towards omni-supervised learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4119–4128, 2018.

[18] Y. Zhang, L. Yang, J. Chen, M. Fredericksen, D. P. Hughes, and D. Z. Chen, “Deep adversarial networks for biomedical image segmentation utilizing unannotated images,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 408–416, Springer, 2017.

[19] C. Baur, S. Albarqouni, and N. Navab, “Semi-supervised deep learning for fully convolutional networks,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 311–319, Springer, 2017.

[20] T. Kalluri, G. Varma, M. Chandraker, and C. Jawahar, “Universal semi-supervised semantic segmentation,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5259–5270, 2019.

[21] S. Mittal, M. Tatarchenko, and T. Brox, “Semi-supervised semantic segmentation with high-and low-level consistency,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[22] J. Peng, G. Estrada, M. Pedersoli, and C. Desrosiers, “Deep co-training for semi-supervised image segmentation,” *Pattern Recognition*, p. 107269, 2020.

[23] W. C. Hung, Y. H. Tsai, Y. T. Liou, Y. Y. Lin, and M. H. Yang, “Adversarial learning for semi-supervised semantic segmentation,” in *29th British Machine Vision Conference, BMVC 2018*, 2019.

[24] N. Souly, C. Spampinato, and M. Shah, “Semi supervised semantic segmentation using generative adversarial network,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5688–5696, 2017.

[25] D.-H. Lee, “Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks,” *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 07 2013.

[26] J. Y. Wanshan Ning, Shijun Lei *et al.*, “iCTCF: an integrative resource of chest computed tomography images and clinical features of patients with COVID-19 pneumonia,” apr 2020.

[27] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proceedings of the 32nd International Conference on Machine Learning* (F. Bach and D. Blei, eds.), vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 448–456, PMLR, 07–09 Jul 2015.

[28] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015. Software available from tensorflow.org.

[29] P. Yakubovskiy, “Segmentation models.” `https://github.com/qubvel/segmentation_models`, 2019.

[30] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, “Albumentations: Fast and flexible image augmentations,” *Information*, vol. 11, no. 2, 2020.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.