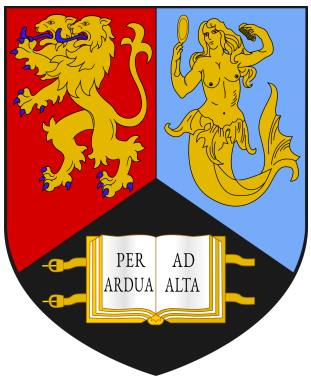
A Deep Residual Segmentation UNet (DRSegUNet) for Automatic Segmentation of Covid-19 on Limited Data

By

Venkata Lahari Balantrapu

Student ID: 2218037



Supervisor: Prof. Hamid Dehghani

A thesis submitted to the University of Birmingham

For the degree of MSc in Artificial Intelligence and Machine Learning

School of Computer Science

University of Birmingham

Birmingham, UK

September 2021

Abstract

The global outbreak novel coronavirus, also known as SARS-CoV-2 or Covid-19, has been declared a pandemic by WHO and has affected millions worldwide, claiming over 4.5 million lives. The never-ending mutations of the virus are leading to different variants of concerns. The gold standard of diagnosis of Covid-19, the RT-PCR(Reverse Transcription Polymerase Chain Reaction), can be overwhelmed during a new variant outbreak due to the supply and demand and significantly low sensitivity. Computed Tomography is one of the high-quality diagnosis tools used to diagnose Covid-19 but is a costly procedure and subject to radiologists' availability. Nevertheless, artificial intelligence can act as a primary tool to perform automatic diagnosis of the infection. However, the study is setback by several challenges, one of which is the unavailability of large datasets required to train the deep learning models. Ground glass opacities(GGO's), the commonly found inflammation in the CT scans, are low-intensity features that the models do not recognize due to insufficient data. To facilitate accurate segmentation of the sensitive GGO's from limited data, a modified Deep Residual segmentation UNet (DRSegUNet) is proposed. The knowledge transfer through skip connections and the usage of deep residual blocks improved the performance of DRSegUNet and made it robust towards vanishing gradients. The proposed model out-performed several state-of-the-art existing 2D models by segmenting the GGO's from a dataset of 100 2D CT images with a dice similarity coefficient of 0.97 and an Intersection over Union of 0.95. DRSegUNet can segment Covid-19 infection from authentic CT images with a sensitivity similar to that of the human. Furthermore, the model can work with limited data in different modalities and can be applied to different areas of research.

Contents

List of Figures

List of Tables

Acknowledgment

I want to express the utmost gratitude to my supervisor Prof. Hamid Dehghani for his constant guidance. His valuable output during different kinds of setbacks helped develop a functional and well-performing tool for segmentation of Covid-19. Furthermore, I would like to thank Dr. Hyung Jin Chang, MSc project coordinator, and my inspector, whose feedback was informative and helpful in enhancing the project. Finally, I would also express my sincere appreciation towards the School of Computer Science for providing us high computing systems and the IT support team for their fast responses to any issues faced while using the systems without which this project would not have been complete. Finally, I am forever grateful to my family for being the ultimate support and motivation.

1 Introduction

1.1 Problem

Coronavirus (Covid-19), a highly contagious disease, stunned the world by causing an unprecedented pandemic since its outbreak in December 2019. Screening a significant number of suspected patients for proper quarantine and treatment were the early measures taken to limit the spread of this virus. As of September 2021, 225,174,442 cases 4,639,867 deaths have been reported across the world [Wu et al., 2020]. The rise in vaccination programs around the world led to the ease of the restrictions. However, the number of cases of Covid-19 worldwide is increasing, and the ever-changing mutations of the virus leading to several side effects Invasive Aspergillosis, Invasive Candidiasis, Invasive Mucormycosis, Invasive Cryptococcosis is a matter of serious concern [Song et al., 2020].

According to the studies thus far, Covid-19 affects numerous organs in the human body, including the heart and lungs. Blood vessels, kidneys, intestine, and brain The virus enters the cells via surface receptor binding ACE2 stands for angiotensin-converting enzyme 2.ACE2 is a receptor located on alveoli, which are tiny air sacs in the lungs of a human [Wadman et al., 2020]. Covid 19 infection leads to inflammation by filling the lungs with fluid or pus, affecting breathing. As a result, the lungs become the virus's starting point [Shi et al., 2021] It is closely associated with Severe Acute Respiratory Syndrome (SARS) and Middle East Respiratory Syndrome (MERS), which result in acute respiratory distress syndrome (ARDS) [Huang et al., 2020][Li et al., 2020a].

1.2 Motivation

The most common diagnostic procedures for Covid-19 are Reverse transcriptase-polymerase chain reaction(RT-PCR) and Biomedical Images such as chest X-Rays and CT-Scans. Although RT-PCR has been the gold standard in diagnosing Covid-19, it has significantly higher false positives towards new variants of concern that do not show any symptoms. The RT-PCR can take up to 6 hours based on the availability of the resources [Peñarrubia et al., 2020] [Dawood, 2020].

As a non-invasive imaging technique, chest computed tomography (CT) can detect specific manifestations in the lung linked with Covid-19; for example, ground-glass opacities and consolidation are the most related imaging findings in pneumonia associated with SARS-CoV-2 infection. As a result, Chest CT is a low-cost, accurate, and efficient approach diagnostic tool for early COVID-19 screening and diagnosis. In addition, it is possible to assess the severity of damage in the lungs and the progress of the patient's condition to aid the improvised treatment plans further [Li et al., 2020b][Pan et al., 2020][Ye et al., 2020].

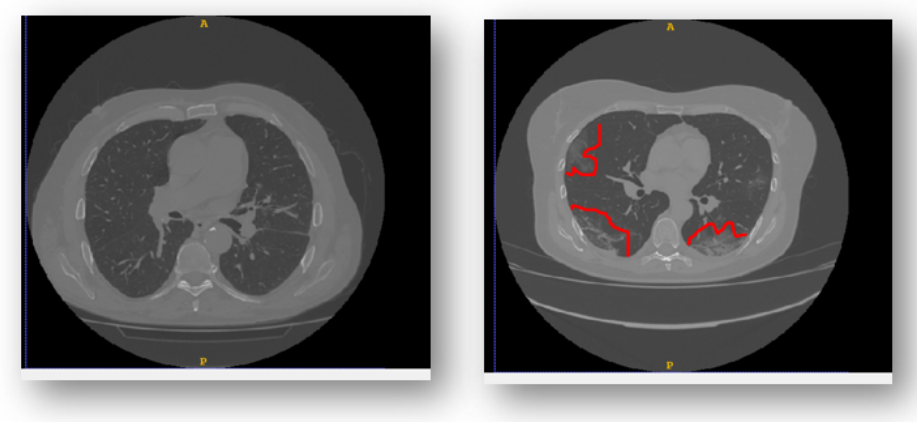


Figure 1: Healthy vs Covid-19 infected CT-scans

As shown in Figure 1 above, The regions marked in red are the ground glass opacities, i.e., inflammation in the lungs. These low-intensity ground-glass opacities(GGO's) make the identification task very challenging. Hence, Accurate segmentation of the ground glass opacity is necessary for the reliable quantification of Covid-19. The segmentation is performed manually by highly trained radiologists. However, the rapid increase in the number of cases during outbreaks has a drastic effect on the radiologists, resulting in a delay in identifying the infection. This delay impacts the patient as a late diagnosis would lead to delayed treatment and increased severity. Therefore, the necessity for automated machines to perform the infected region's segmentation is now more than ever [Gaál et al., 2020]. However, automatically defining infection areas from chest CT scans is complicated due to the significant diversity in position and form across different individuals, as well as the poor contrast of infection regions in CT images [Shan et al., 2020] [Chen et al., 2020].

1.3 Challenges

Segmentation of Covid-19 inflammation (GGO's) is a challenging task as the CT image consists of high-level information of different variations [Fung et al., 2021]. The majority of the CT scan annotations consist of background and infection, where the negative samples are high in number, and the GGO's are less in number. The problem of pixel class imbalance can be solved by choosing an optimal loss function [Fung et al., 2021]. However, with limited data sets, the problem would still be prevalent.

1.4 Proposed Method

Artificial Intelligence acts as a powerful tool for medical image analysis. However, these models often require a high amount of images and annotated masks

to learn. The resources to manually annotate the masks amidst a pandemic are thus limited [Najafabadi et al., 2015]. The challenges mentioned in section 1.3 are the primary motivations to find an alternative solution to tackle annotations and datasets' limitations accurately. In this paper, a modified ResidualUNet deep residual segmentation UNet(DRSegUNet), which combines the features of residual blocks with the encoder-decoder structure of UNet provided with deep connections for ease of flow of gradient information and knowledge among the layers, is proposed. The proposed model in creating deeper networks that can learn from limited data [Zhang et al., 2018b]. This model works well with the main challenge of limited data. We also show that it is beneficial to have multi-class labels when dealing with limited data. However, the challenge of manual annotations is yet to be addressed.

Weakly Supervised Self learning DRSegUNet: An attempt to adapt Self-learning to the model is made where the model is trained recursively on a small dataset of pseudo-generated labels [Zhang et al., 2018a]. The predicted labels from the first level are input to the second level and the knowledge from the first level. The DRSegUNet learns by itself to identify the features. The expected outcome of the paper is to propose a robust model that works with limited data and an approach that can work in the absence of manual annotations.

These methods will be further discussed in detail in Section 4 of the paper

Outline: The rest of the paper is summarized as follows: we first review some related work and contributions to the problem using deep learning methodologies in section 2, Related Work. Then, all the technical background required to understand the proposed model is explained in section 3, Background. In section 4, Methodology, a description of the data and its processing, the process flow and detailed explanation of the model, and the weakly supervised self-learning DRSegUNet will be given. The experiments performed on the model, the approach, and the results will be discussed in section 5, Experimental Setup. Finally, in section 6, we discuss the model's strengths, future work, and the paper's summary.

2 Related Work

In this section detailed analysis of the methods and models proposed previously for segmentation and classification of the problem described in Section1 has been briefed.

2.0.1 Covid-19 Detection and Segmentation using Deep Learning

Over the years, deep learning methodologies have been incorporated into biomedical image analysis due to their efficient performance on feature representations. Ulhaq has presented a survey on computer vision models for Covid-19.et.al [Ulhaq et al., 2020]. Some of the methods used for Covid-19 segmentation and classification are summarised in the table1 below.

Author	Model	Data	Performance	Summary
Jin et al. [Jin et al., 2020]	3D UNet++	1,136 CT images (723 positives for COVID-19) from five hospitals	AUC 0.991, sensitivity of 0.974 and specificity of 0.922.	The model utilized the pre-trained RESNET-50 weights, and it performs lung extraction as pre-processing using UNet++ followed by infection segmentation.
Xu et al. [Xu et al., 2020]	VNET based segmentation model	618 CT samples, 219 Covid-19, 224 - Influenza-A viral pneumonia	Model accuracy 86.7	A 3D segmentation model has been proposed using the RESNET-18 architecture with added location attention mechanism for efficient feature learning.
Shan et al. [Shi et al., 2020]	VB-Net neural network	249 CT images	91.6 dice similarity coefficient	Extensive analysis of deep learning models and proposed VB-Net for automatic lung segmentation and infection from CT scans.
Zheng et al. [Zheng et al., 2020]	UNet	499+131 CT volumes	0.95 ROC AUC and 0.901 accuracy	A pre-trained UNet has been used for segmentation, the segmentation model is used as pre-processing for 3D convolutional neural network for Covid-19 detection.

Continued on next page

Author	Model	Data	Performance	Summary
Ying Song et al. [Shi et al., 2020]	RESNET-50	777 CT images	0.95 AUC, 0.96 sensitivity	The proposed model is a combination of Details Relation Extraction neural network DRENet + ResNet50,with Feature Pyramid Network (FPN)+ Attention module. The DRENet and Pyramid network help in robust feature learning, while the attention module aids in ROI heat maps that eliminate unnecessary features.
Li et al. [Li et al., 2020b]	2D U-Net, COVNet	4356 chest CT images	sensitivity and specificity for COVID-19 are 0.90 0.96 respectively	2D UNet is used to extract ROI in lungs, and a COVNet has been trained using RESNET-50 weights for Covid-19 Detection.

Table 1: Classification and Segmentation methodologies for Covid-19 CT Images

Zhao et al. proposed a new 3D VNet architecture to perform lung extraction with a deformation module to fine-tune the output based on prior shape knowledge. The model achieved a dice score of 0.96.

2.0.2 Self Supervised

Self-supervised learning is one of the methodologies used to work with unlabelled data by incorporating specific tasks to make the model learn the features [Wang et al., 2020].

Chen et al. [Chen et al., 2021] proposed a self-supervised model. The data is cropped into two parts where one part undergoes random flipping while random cropping and distortion are applied to the other part. Representation learning is used to train the model on identifying the parts that belong to the CT image and used to label the images. These labeled images were used in a few-shot classification network. This method would fall prey to situations where the features are mislabelled by query images.

Fung et al. [Fung et al., 2021] Proposed a self-supervised two-stage deep learning model. The backbone INfNet model was incorporated with the advantages

of image in-painting [Bertalmio et al., 2000], focal loss, and look ahead optimizer to segment and diagnose Covid-19. Masks are generated from in-painting and are used to train the model. The performance of the model depends on the missing parts as the missing parts could either contain complex information or unnecessary information [Yu et al., 2018].

Self-Learning: Zhang et al. proposed a self-learning method to tackle the issue of manual annotations using the concepts of pseudo labeling and recursive learning. They pseudo labeled the data and used UNet to perform segmentation [Zhang et al., 2018a].

2.0.3 Attention Mechanism

Sometimes features extracted from the images may not contain any helpful information; in these cases, attention gates are used to identify the informative features. Zhou et al. proposed a modified UNet with an attention mechanism. The proposed model considers only the informative feature representations along spatial-wise and channel-wise and evaluates the model on 437 CT slices. Their model achieved a dice score of 0.83.

D2AUNet is another proposed model constructed using dual attention and dilated convolutions to segment Covid-19 CT scans. The dual attention modules constitute of gate attention module(GAM) and decoder attention module(DAM). The suggested GAM improves the network's skip connections by combining characteristics with semantic-rich gate signals. The suggested DAM is added to the model decoder to enhance decoding quality, particularly while segmenting hazy lesions. To aid the size variation challenge of GGO's dilated convolutions were proposed to gain large receptive fields and better performance. However, the model achieved a dice score of 0.72 [Zhao et al., 2021].However, attention mechanism alone would not be practical for high-resolution images such as CT scans with limited ground labels containing only background and infection.

2.0.4 ResUNet

ResUNet was first proposed by Zhang et al. [Zhang et al., 2018b] for road extraction from aerial images. The proposed model used the advantages of residual blocks in creating deeper networks [Zhang et al., 2018b]. This network was further adapted to the domain of biomedical imaging for brain tumor segmentation and achieved a dice score of 0.83, overcoming the challenges of UNet

Chen et al. [Chen et al., 2020] proposed a residual attention UNet for Covid-19 segmentation. The proposed model consists of attention gates in the decoder to extract the informative features. The model was evaluated on MedSeg, consisting of limited data and achieving a dice score of 0.94.

3 Background

This section consists of the background and all the technical aspects applied in the proposed model, discussed in Section 4.

3.1 K-means Clustering

K-means clustering is an unsupervised approach that calculates the centroids iteratively until an optimal centroid is achieved [Hartigan and Wong, 1979]. The number of clusters' K' is to be provided manually based on the task. The data points are assigned to a cluster where the sum of the squared distance between the data points and the centroid is an optimal minimum. Furthermore, minor variation among clusters means more comparable data points inside the same cluster. K-means uses an Expectation-Maximization technique. The data points are assigned to the nearest cluster using the Expectation step, and the centroid of each cluster is computed using the Maximization-step.

Expectation–maximization (E–M) is a strong technique used in several data science situations. k-means is a particularly simple and straightforward implementation of the method, and we will go over it quickly here. In summary, the expectation-maximization technique entails the following steps: Assume some cluster centers and repeat until convergence is achieved.

- E-Step: Assign points to the cluster center that is closest to the data point.
- M-Step: Set the cluster centers to the mean.

The "E-step," also known as the "Anticipation step," is so named because it requires changing the expectation to which cluster each point belongs. The "M-step" or "Maximization step" is so named because it includes maximizing some fitness function that determines the position of the cluster centers in this example, by taking a simple mean of the data in each cluster. The literature on this method is extensive, but it can be generalized as follows: under normal conditions, each repetition of the E-step and M-step results in a better estimate of the cluster characteristics. [Hartigan and Wong, 1979]

3.2 Model

3.2.1 DeepResUNet

DeepResUNet is proposed by [Zhang et al., 2018b] for the extraction of the road from aerial images. It utilizes the benefits of residual block combined with the segmentation power of UNet.

UNet: UNet is an end-to-end convolutional neural network developed for biomedical segmentation tasks proposed by Ronneberger et al. [Ronneberger et al., 2015]. Convolutional networks are known for their classification tasks

where the images are down-sampled while learning to classify an image belonging to a class. However, Image segmentation tasks require pixel-wise classification and often have equal output and input sizes. So a method to equalize the feature maps to the size of the output is required. To accommodate this, the UNet architecture consists of series of up-sampling blocks, also known as decoders, that provide up-sampled high-resolution maps from the down-sampled feature maps by using transposed convolutions where pixels are added around the existing pixels also utilizing the information from the encoder blocks [Zhang et al., 2018b].

To achieve better results for semantic segmentation of Covid 19, it is critical to use low-level details while retaining high-level semantic information. Furthermore, the ground glass opacities are all low-level information in limited number. However, training a neural network to learn low-level features requires a lot of computational power and vanishing gradients where our ground-glass opacities are lost, especially on a limited data of size 512x512, which cannot be compromised. One solution is to use a pre-trained network and then fine-tune it on the target dataset, as shown in [Long et al., 2015], but the ImageNet does not have any biomedical images, which does not help the problem. Another approach is to use extensive data augmentation, as done in UNet[Ronneberger et al., 2015].

The use of residual blocks in the place of regular convolutional blocks of UNet resulted in a massive improvement in the results.

DeepResUNet:

1. The pre-activated residual blocks are used instead of the normal convolution blocks of the UNet, as shown in fig. This helps in building the deep network and help with training.

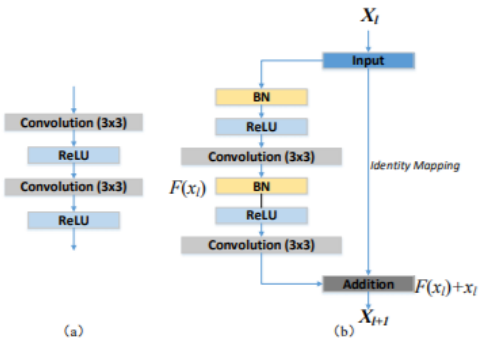


Figure 2: UNet Convolution Block vs Pre Activated ReLU Block

2. Skip Connections: In deep neural networks, skip connections aid in information traversal. Gradient information can be lost as we move through many layers, a phenomenon known as vanishing gradient. The benefits of skip connections are that they pass feature information to lower layers, making it easier to classify minute details. Some spatial information is lost as a result of max pooling. The final layer also has more information; skip connections in the residual block help in ease of propagation.

3. The max-pooling layer is removed, and a stride of 2 is used in the first convolution blocks instead to reduce the feature map by half.

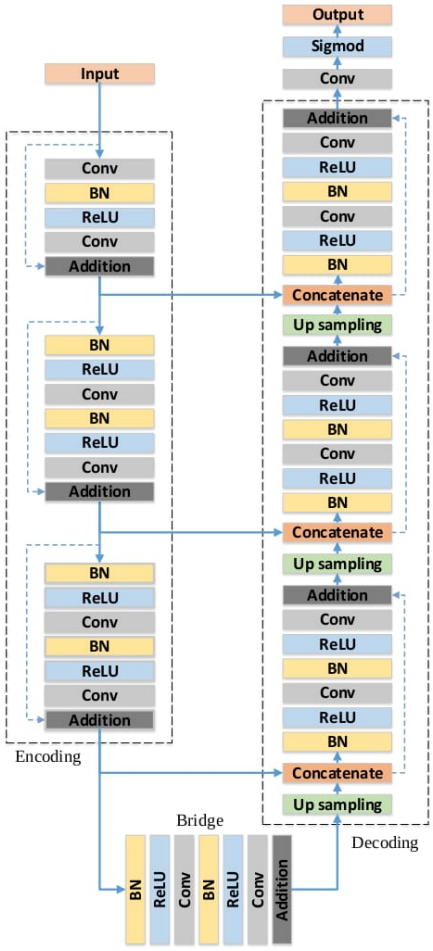


Figure 3: DeepResUNet Architecture

4. The DRUNet follows a 7-level architecture as shown in Figure 3. It contains Zhang et al. [2018b] :

- Encoder Block
- Bridge Block
- Decoder Block

All three parts contain three residual blocks with a kernel size of 3x3, where each residual block consists of two convolutional blocks of size 3x3, and each convolutional block has a batch normalization followed by relu activation a 2D convolution of kernel size 3x3. A convolution of 1x1 is applied on the last layer and a softmax activation to generate a three-channel segmentation output. The architecture of the model is visualized in Figure 3.

ResBlock: As the depth of deep neural networks increases, the performance of the model improves, but it would impede training and may cause a degradation problem (vanishing gradients) [He et al., 2016]. To address this issue, the residual framework (ResNet) was proposed by He et al. for better propagation of the gradient directly from top to bottom of the network during backward propagation [He et al., 2016]. A residual block consists of a set of residual blocks put together.

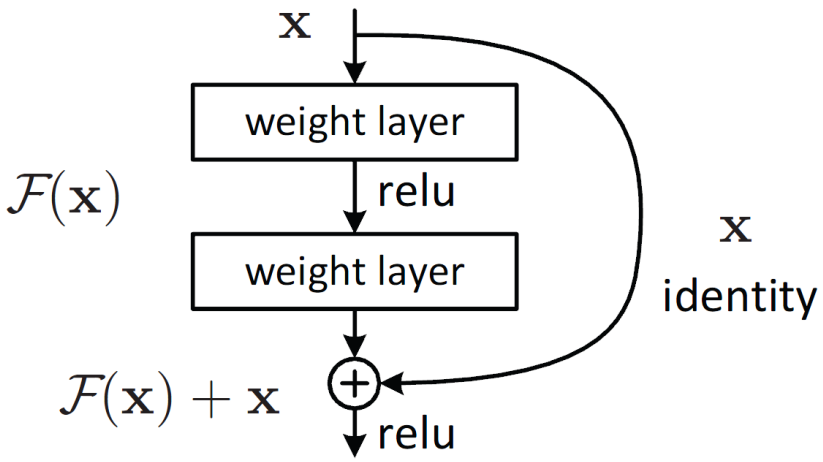


Figure 4: Residual Block

Given an input x_i, the output of an ith residual block x_{i+1}, a residual block can be represented as

$$y_i = h(x_i) + F(x_i, W_i)$$

$$x_{i+1} = f(y_i)$$

Where F(.) is a residual function, $f(y_i)$ is an activation function and $h(x_i)$ is an identity mapping essentially $h(x_i) = x_i$. [Zhang et al., 2018b]

Batch-Normalization: Batch normalization employs a transformation that keeps the mean output close to 0 and the standard deviation of the output close to 1. The mean calculated using the formula

$$\mu = 1/m(\Sigma h_i)$$

where, his the hidden layer activation. This is followed by calculation of the standard deviation

$$\sigma = [\frac{1}{m}(\Sigma(h_i - \mu_i)^2)]$$

The activations are then normalized by subtracting the mean from the input and dividing it by standard deviation

$$h_i norm = \frac{(h_i - \mu)}{(\sigma + \epsilon)}$$

where, ϵ is a smoothing constant used to facilitate ease of the normalization and control division by zero.

Activation: The activation function is applied to every pixel in the feature map. ReLU calculates $f(u) = max(0, u)$. The values are generally thresholded to zero. The significant advantage of ReLU is that it can considerably accelerate the convergence of stochastic gradient descent. When compared to Sigmoid/sophisticated tanh's operations, ReLU is much easier to understand and use in practice. Nevertheless, ReLU units might be weak and even die during training.

There will always be some data points yielding positive values to each given node when training on a suitably sized batch. As a result, the average derivative is rarely near 0, allowing gradient descent to continue. It is possible that when the ReLU unit is updated with forwarding data and backward gradients, its output will always be zero, in which case it will kill the neuron.

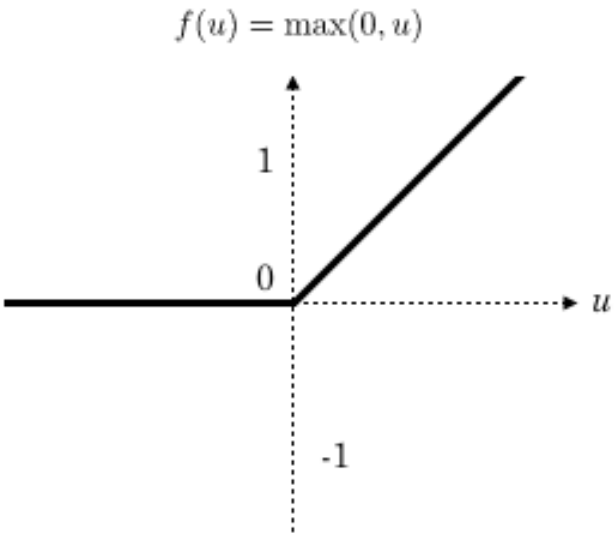


Figure 5: ReLU Activation

2D convolution: The convolution block is used to detect features from an image. It is a process where a kernel matrix of specified size is used to generate feature maps from the image.

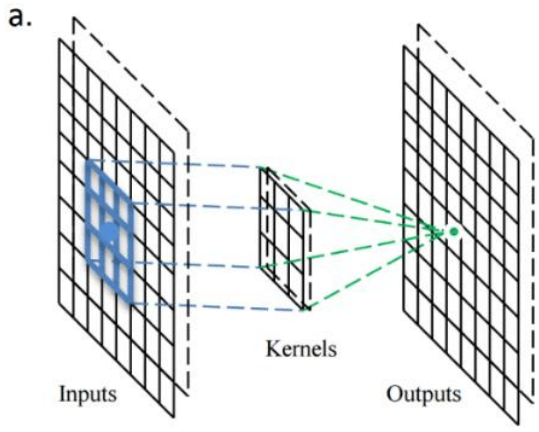


Figure 6: Convolution Illustration with Kernel Size 3

The feature map values are generated using the generic formula

$$F[r, c] = (I * k)[r, c] = \Sigma_i \Sigma_j (k[i, j] * I[r - i][c - j])$$

where I is the image and K is the specified kernel size (3x3) and r, care the rows and columns of the feature map. Each value in the kernel is multiplied by the corresponding values in the image. All the values in the kernel are then summed, and the result is placed in the feature map. In general, on an image of size 6x6 with a kernel size of 3x3, we get a feature map of 4x4. To generate a feature map of the same size, padding is applied, where a border of zeros are placed around the image

$$p = (k - 1)/2$$

where k is a kernel size which is always an odd number, and p is the padding. To reduce the size of the feature maps in a controlled process, a stride is applied in our model a stride of 2 is applied in the first convolution to reduce the size of the feature map [Dumoulin and Visin, 2016].

The output size of the convolution is given by the formula

$$conv2D = 1 + (InputSize - Kernelsize + 2 * Padding)/Stride$$

3.3 Up-sampling

Image segmentation tasks require the output to be the same size as the input, the decoder to increase the size of feature maps obtained from the encoder convolutions [Ronneberger et al., 2015]. There are various methods to up sample the feature maps.

Upsampling2D: The Upsampling2D is a keras function that increases the size of the feature map based on the stride. The stride 2 is increases the size by twice. This is done using nearest neighbor or bi-linear interpolation.

- **Nearest Neighbour:** The nearest neighbor is one of the methods where the size of the feature map is increased by adding the nearest value into the pixels.

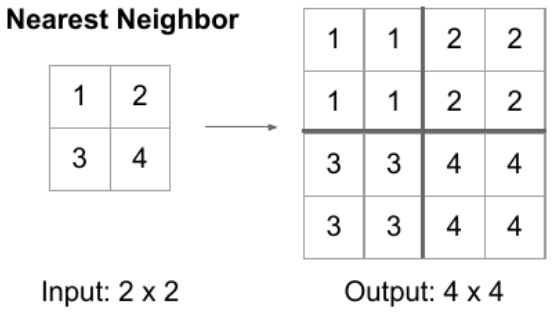


Figure 7: Nearest Neighbour Up-sampling

- **Bi-linear Interpolation:** Bi-linear interpolation is a re-sampling process, which uses the weighted distance average of the nearest pixel values to estimate a new pixel value.

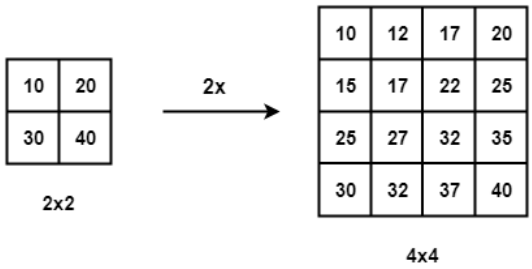


Figure 8: Bi-linear Interpolation Up-sampling

These methods, however, do not learn any semantic meaning while increasing the feature maps. Therefore, often time with this method, loss of gradients/essential features can be seen.

Transposed Convolution: Transposed convolutions, also known as deconvolutions, perform the inverse operation as that of convolution. The convolutions are used to increase the size of the feature maps. The pixel values are added by learning useful features through back-propagation [Dumoulin and Visin, 2016]. The output of the transposed convolutions can be formulated as

$$Conv2Dtranspose = (InputSize - 1) * Strides + kernel - Size - 2 * Padding + OutputPadding$$

.
Skip connections from encoder levels help the transposed convolutions to learn low level features without loosing necessary information[Ronneberger et al., 2015]. However, transposed convolutions are computationally complex compared to basic up-sampling [Dumoulin and Visin, 2016].

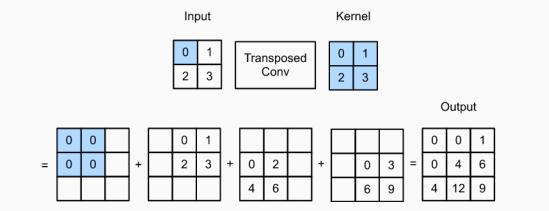


Figure 9: Transposed Convolution with a 2x2 kernel

3.4 Transfer Learning

Transfer Learning is the process of adapting the knowledge obtained from one task onto a relatively closer task as a starting point. The ideas of a domain and a task are involved in transfer learning. A domain D comprises a feature space X and a marginal probability distribution P(X) over the feature space, where $X = x1.....,xn$. For document classification using a bag-of-words representation, X is the space of all document representations, xi is the ith term vector corresponding to some document, and X is the training sample of documents.

3.5 Softmax Activation:

Softmax activation gives a probability of a pixel belonging to a class . It transforms all the values in the input to a range of (0,1).The advantage of softmax over sigmoid is that all the probabilities of a softmax sum to 1 i.e., every pixel can only belong to one class label.The softmax hence is used for multi class classification problems [Nwankpa et al., 2020].

$$\sigma(x)_i = \frac{\epsilon^x{}_i}{\Sigma_j = 1^k \epsilon^x{}_i}$$

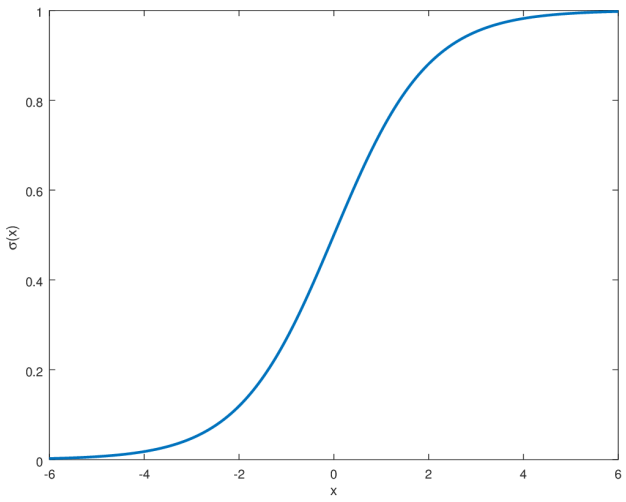


Figure 10: Softmax Activation

3.6 Loss Function:

Categorical cross-entropy is widely used for multi-class classification tasks; it trains a model to output a probability of a pixel belonging to only one of the

C multiple classes. In the specific (and usual) case of multi-Class classification, the labels are one-hot, so only the positive class keeps its term in the loss. There is only one element of the Target vector, which is not zero. So, discarding the elements of the summation which are zero due to target labels, the loss function will be
$Loss = \Sigma_i{}^c y_i log f(y_i)$ where, $f(y_i)$ is the softmax probability of the output y_i

3.7 Evaluation Metrics

We consider four types of predictions while analyzing a simple machine learning model's performance, i.e., true positives, false positives, true negatives, and false negatives.

	Predicted Positive	Predicted Negative
Positive	TP(True Positive	FN(False Negative)
Negative	FP(False Positive)	TN(True Negative)

In the dense prediction task of image segmentation, it is not immediately clear what makes a "true positive" or how we might grade our predictions in general.

Jaccard's coefficient and Dice similarity coefficient are two statistical metrics used to validate the model's performance.

3.7.1 Intersection Over Union

Jaccard's coefficient, also known as Intersection over Union (IoU), is a method for quantifying the percent overlap between our target mask and our prediction output.

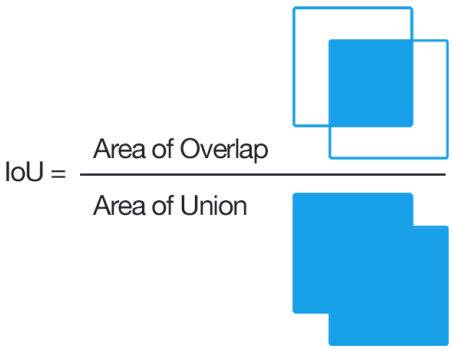


Figure 11: Intersection over Union

In training, the Dice coefficient is commonly employed as a loss function, and this measure is closely connected to it. IoU measures the total number of pixels present in both target- and prediction masks divided by the total number of pixels present in both masks.

For the problem, it is formulated as

$$IoU = \frac{Target \bigcap Prediction}{Target \bigcup Prediction}$$

The intersection $(A \bigcap B)$is made up of pixels from both the prediction mask and the ground truth mask, whereas the union $(A \bigcup B)$ is made up of all pixels from either the prediction mask or the ground truth mask, depending on which one is used. Based on our semantic segmentation prediction, the IoU score is computed independently for each class and then averaged over all classes to produce a global mean IoU score

3.7.2 Dice Score

The Sorensen–Dice index, or simply Dice coefficient or Dice-score, is a statistical instrument that evaluates the similarity between two data sets; it is also known as the Dice similarity coefficient. As a result, it has become one of the most often used tools to validate AI-based picture segmentation algorithms, but it is a much more general idea that can be used for data sets for many different applications.

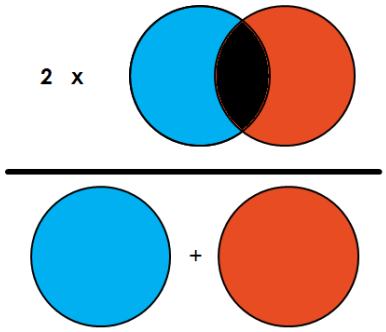


Figure 12: Dice Similarity Coefficient

It is formulated as

$$DiceScore = \frac{2 * (Target \bigcap Prediction) + \epsilon}{|Traget| + |Prediction| + \epsilon}$$

where ϵ is a smoothing term used to stabilize the metric and avoid divide by zero errors.Dice similarity coefficient is often for image segmentation problems.

This is different from accuracy where the objective is to match the values, unlike dice which matches the value + position.

3.7.3 F1-score

The ground glass opacities are pixel class imbalanced, and accuracy is not the best choice to evaluate the performance of these models. An alternative to this would be the F1-score. The F1-score is a harmonic mean of precision and recall.

Precision: Precision effectively describes the purity of our positive detection relative to the ground truth.

$$\frac{TP}{TP + FP}$$

Sensitivity/Recall: Sensitivity, also known as recall, describes the positive detections that are accurately matched to the ground truth among all the positive predictions.

$$\frac{TP}{TP + FN}$$

F1-score: Precision and Recall are linked in a trade-off relationship. As a result, the goal of the F1-score is to include both precision and recall while evaluating a model

$$\frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2 \times precision \times recall}{precision + recall}$$

Dice similarity coefficient is very similar to the F1-score when dealing with binary classification. The F1-score/dice-score is also very closely related to the IoU score. The two measures are always positively linked in the absence of any set "ground truth." For example, if classifier A outperforms classifier B on one metric, it also outperforms classifier B on the other. It is common to infer that the two measures are functionally equal and that the decision between them is arbitrary, but the issue arises when calculating the average score among a group of conclusions. The difference is then quantified by determining how much worse classifier B is than A in each scenario. Even though they may both agree that this one case is poor, the IoU measure tends to penalize single occurrences of incorrect categorization more than the F score numerically. The IoU measure has a "squaring" impact on the errors relative to the F1-score, similar to how L2 penalizes the biggest mistakes more than L1. As a result, the F score is more likely to reflect average performance, but the IoU value is more likely to reflect worst-case performance.

4 Methodology

4.1 Data

The data is obtained from two different open sources, one of which is from medicalsegmentation.com and consists of CT slices from MedSeg and Radiopedia; the second data source is MosMed. MedSeg consists of 100 Axial 2D CT slices collected from more than 40 patients converted from openly accessible JPG images. The images were converted and are segmented by radiologists into four channels, where 0: ground glass opacities, 1: consolidations, 2: lungs, 3: background. Radiopedia consists of whole volumes from 9 different patients (829 slices), out of which 373 positive slices were identified and segmented by the radiologists into the three channels. MosMed dataset consists of axial CT scans from more than 1000 patients, where most cases include complete volumes. Fifty such CT scans are annotated into single-channel masks containing background and ground-glass opacities.

All the CT images are of the same size, 512x512, with a voxel spacing of 1.9x1.9x2.3cm, and the pixel values of these slices are store in a linear density scale knows as Hounsfield Units. The density of every tissue is assigned a Hounsfield Scale where the water has 0 HU value, any tissue denser than water has values greater than 0, and tissues less than water have a value less than zero. Low-density tissues are color-coded darker (blacker), while high-density structures are color-coded brighter (whiter).

Because the human eye can only distinguish a limited number of grey shades, a given image's full range of density values is rarely exhibited. Instead, the tissues of interest are highlighted by limiting the use of visible grey tones to a small area of the image, a process known as "windowing." It is also the method used in normalizing the CT images to highlight regions of interest.

4.2 Data Pre Processing

The HU values of higher density tissues and lower density tissues are all thresholded, i.e., all the HU values greater than 100 are set to 100, and all the HU values less than 1000 are set to -1000. This is called windowing and is used to highlight the GGO's.

$$HU > 100 = 100$$

$$HU < -1000 = -1000$$

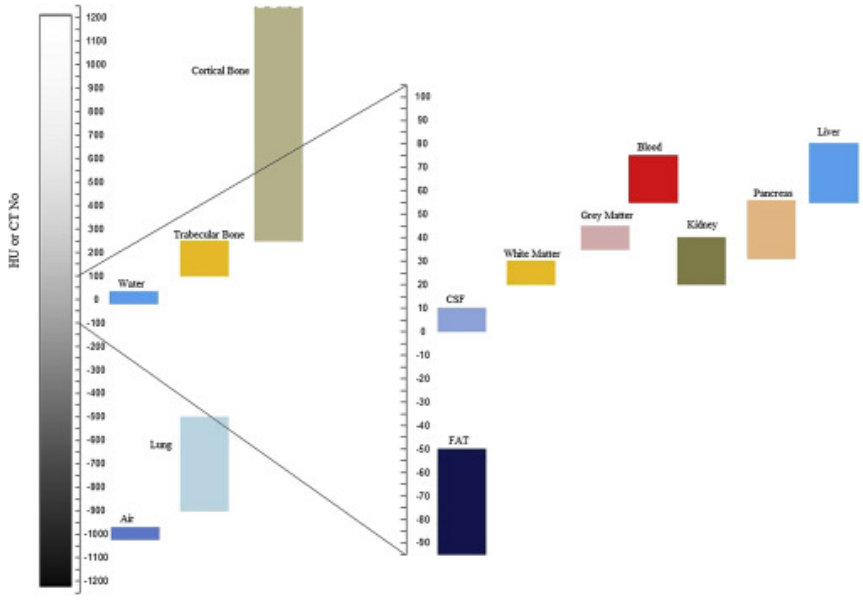


Figure 13: HU values with their gray scale intensities

Z-Score Normalisation: The CT slices are normalized using the Z-score, the Z-score is a scaling variation that represents the number of standard deviations from the mean. The z-score ensures that feature distributions have a mean of 0 and a standard deviation of 1. It is useful when there are a few outliers, but they are not so extreme that clipping is required.

The formula for calculating a point's z-score, x, is as follows:

$$X = (X - \mu)/\sigma$$

4.3 Proposed Model Architecture

The proposed model DRSegUNet consists of 4 parts

- STEM
- ENCODER
- BRIDGE
- DECODER

STEM: The stem consists of two parts where the first part goes through a 3x3 kernel convolution followed by a residual block of stride 1.

The second part performs a convolution on the input followed by a batch normalization. An addition operation is performed on the two outputs to aid in skip connections. The mathematical operations are explained in detail in section 3.2.

operation	connected to	kernel	stride	output
input	CT image (512x512x1)	-	-	512x512x1
conv2d	input	3x3	1	512x512x16
batch-norm	conv2d	-	-	512x512x16
activation	batch-norm	-	-	512x512x16
conv2d-2	activation	3x3	1	512x512x16
conv2d-2	input	3x3	1	512x512x16
batch-norm-1	conv2d-2	-	-	512x512x16
add	batch-norm-1,conv2d-1	-	-	512x512x16

Table 2: Stem of DRSegUNet

ENCODER: The encoder path is also known as a down-sampling path where the feature maps are reduced by using a max-pooling operation [Ronneberger et al., 2015]. The proposed model consists of a pre-activated residual block that utilizes the benefits of residual blocks in the place of normal convolutions [Zhang et al., 2018b]. The encoder block consists of 4 encoders, starting with 32 neurons. Each encoder is built up of two pre-activated residual blocks [Zhang et al., 2018b]. The size of the feature maps is reduced by using a stride of 2 on starting convolutions.

The output of the residual blocks is added to the input from the previous block, where a 3x3 convolution of stride 2 followed by a batch normalization is applied on the input. The residual blocks help in constructing deeper networks while the skip connections ease the flow of gradients. The CT scan consists of high-level information such as bones muscles, and the ground glass opacities are very low-level information which is usually lost in the deeper layers. Hence, deeper connections help the model learn these low-level features as the information from the previous layer is passed. The operational flow of each encoder block is detailed in the tables[3,4,5,6]

operation	connected to	kernel	stride	output
batch-norm-2	add	-	-	512x512x16
activation-1	batch-norm-2	-	-	512x512x16
conv2d-3	activation-1	3x3	2	256x256x32
batch-norm-3	conv2d-3	-	-	256x256x32

activation-2	batch-norm-3	-	-	256x256x32
conv2d-4	activation-2	3x3	1	256x256x32
conv2d-5	add	3x3	2	256x256x32
batch-norm-4	conv2d-5	-	-	256x256x32
add-1	batch-norm-4,conv2d-4	-	-	256x256x32

Table 3: Encoder-1 of DRSegUNet

operation	connected to	kernel	stride	output
batch-norm-5	add-1	-	-	256x256x32
activation-3	batch-norm-5	-	-	256x256x32
conv2d-6	activation-3	3x3	2	128x128x64
batch-norm-6	conv2d-6	-	-	128x128x64
activation-4	batch-norm-5	-	-	128x128x64
conv2d-7	activation-4	3x3	1	128x128x64
conv2d-8	add-1	3x3	2	128x128x64
batch-norm-7	conv2d-8	-	-	128x128x64
add-2	batch-norm-7,conv2d-6	-	-	128x128x64

Table 4: Encoder-2 of DRSegUNet

operation	connected to	kernel	stride	output
batch-norm-8	add-2	-	-	128x128x64
activation-5	batch-norm-8	-	-	128x128x64
conv2d-9	activation-5	3x3	2	64x64x128
batch-norm-9	conv2d-9	-	-	64x64x128
activation-6	batch-norm-9	-	-	64x64x128
conv2d-10	activation-6	3x3	1	64x64x128
conv2d-11	add-2	3x3	2	64x64x128
batch-norm-10	conv2d-11	-	-	64x64x128
add-3	batch-norm-10,conv2d-10	-	-	64x64x128

Table 5: Encoder-3 of DRSegUNet

operation	connected to	kernel	stride	output
batch-norm-11	add-3	-	-	64x64x128

activation-7	batch-norm-11	-	-	64x64x128
conv2d-12	activation-7	3x3	2	32x32x256
batch-norm-12	conv2d-12	-	-	32x32x256
activation-8	batch-norm-12	-	-	32x32x256
conv2d-13	activation-8	3x3	1	32x32x256
conv2d-14	add-3	3x3	2	32x32x256
batch-norm-13	conv2d-14	-	-	32x32x256
add-4	batch-norm-13,conv2d-13	-	-	32x32x256

Table 6: Encoder-4 of DRSegUNet

BRIDGE: The bridge connects the encoder and decoder paths it is comprised of two residual blocks where the convolutions of stride 1 are used. The operational flow is shown in the table 7.

operation	connected to	kernel	stride	output
batch-norm-14	add-4	-	-	32x32x256
activation-9	batch-norm-14	-	-	32x32x256
conv2d-15	activation-9	3x3	1	32x32x256
batch-norm-15	conv2d-15	-	-	32x32x256
activation-10	batch-norm-15	-	-	32x32x256
conv2d-16	activation-10	3x3	1	32x32x256

Table 7: Bridge of DRSegUNet

DECODER: The decoder has an opposite effect as that of an encoder, the feature maps are restored into the original size in the decoder path, the proposed model uses Transposed convolutions of size 3x3 and stride 2 to increase the size of the feature maps. The segmentation task requires an image the same size as the original, with every pixel labeled with a class value. However, during the convolutions in the encoder path, the feature map is reduced using a kernel. To up-sample this to the size of the original image, there are two commonly used techniques.

Upsampling2D is a method in Keras that increases the feature map times the given stride value. However, this is performed using the nearest neighbour or bilinear interpolation and does not learn any valid information, and this could also lead to noisy segmentation. Therefore, Transposed convolutions are chosen as they learn valuable features from the feature maps, as explained in Section

3.3. Furthermore, knowledge from the encoder path is transferred to these convolutions to also help with the gradients. The detailed flow is tabulated below (table[8,9,10,11]).

operation	connected to	kernel	stride	output
transconv2d	conv2d-16	3x3	2	64x64x256
concatenate	transconv2d	-	-	64x64x384
batch-norm-16	add-3	-	-	64x64x384
activation-11	batch-norm-16	-	-	64x64x384
conv2d-17	activation-11	3x3	1	64x64x256
batch-norm-17	conv2d-171	-	-	64x64x256
activation-12	batch-norm-17	-	-	64x64x256
conv2d-18	activation-12	3x3	1	64x64x256
conv2d-19	concatenate	3x3	1	64x64x256
batch-norm-18	conv2d-19	-	-	64x64x256
add-5	batch-norm-18,conv2d-18	-	-	64x64x256

Table 8: Decoder-1 of DRSegUNet

operation	connected to	kernel	stride	output
transconv2d-1	add-5	3x3	2	128x128x128
concatenate-1	transconv2d-1	-	-	128x128x192
batch-norm-19	add-2	-	-	128x128x192
activation-13	batch-norm-19	-	-	128x128x192
conv2d-20	activation-13	3x3	1	128x128x128
batch-norm-20	conv2d-20	-	-	128x128x128
activation-14	batch-norm-20	-	-	128x128x128
conv2d-21	activation-14	3x3	1	128x128x128
conv2d-22	concatenate-1	3x3	1	128x128x128
batch-norm-21	conv2d-22	-	-	128x128x128
add-6	batch-norm-21,conv2d-21	-	-	128x128x128

Table 9: Decoder-2 of DRSegUNet

operation	connected to	kernel	stride	output
transconv2d-2	add-6	3x3	2	256x256x64
concatenate-2	transconv2d-2	-	-	256x256x96
batch-norm-22	add-1	-	-	256x256x96

activation-15	batch-norm-22	-	-	256x256x96
conv2d-23	activation-15	3x3	1	256x256x64
batch-norm-23	conv2d-20	-	-	256x256x64
activation-16	batch-norm-23	-	-	256x256x64
conv2d-24	activation-16	3x3	1	256x256x64
conv2d-25	concatenate-2	3x3	1	256x256x64
batch-norm-24	conv2d-25	-	-	256x256x64
add-7	batch-norm-24,conv2d-24	-	-	256x256x64

Table 10: Decoder-3 of DRSegUNet

operation	connected to	kernel	stride	output
transconv2d-3	add-7	3x3	2	512x512x32
concatenate-3	transconv2d-3	-	-	64x64x48
batch-norm-25	add	-	-	64x64x48
activation-17	batch-norm-25	-	-	64x64x48
conv2d-26	activation-17	3x3	1	512x512x32
batch-norm-26	conv2d-26	-	-	512x512x32
activation-18	batch-norm-26	-	-	512x512x32
conv2d-27	activation-18	3x3	1	512x512x32
conv2d-28	concatenate-3	3x3	1	512x512x32
batch-norm-27	conv2d-28	-	-	512x512x32
add-8	batch-norm-27,conv2d-27	-	-	512x512x32

Table 11: Decoder-4 of DRSegUNet

A convolution of kernel size 1x1 is applied on the output with the use of a softmax activation. The outputs of the data set contain multi-class labels for which sigmoid activation is not the best suitable as sigmoid probabilities are generated independently, i.e., the activation is independent of the other pixel rows [Nwankpa et al., 2020]. However, for multi-class predictions, we need an activation-dependent on the other pixel values where a pixel belongs to only one class. Softmax is an ideal activation function as softmax probabilities sum up to one, as explained in Section 3.5. The model is built on the combination of categorical cross-entropy with softmax activation. The architecture of DRSegUNet is visualized in Figure 14.

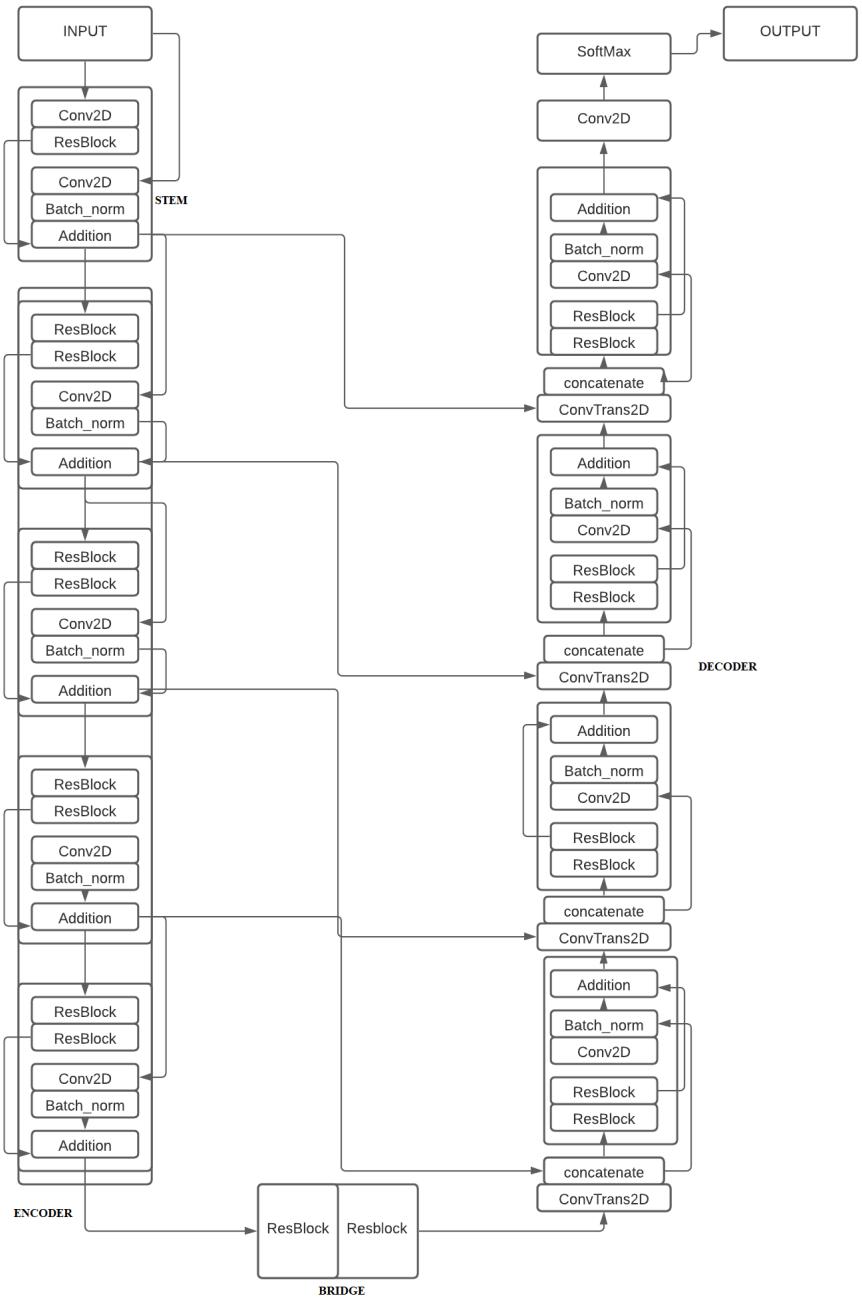


Figure 14: DRSegUNet Architecture

4.4 Weakly supervised self-learning DRSegUNet

In the absence of data with manual annotations, the proposed approach uses pseudo labeling and self-learning concepts, where pseudo labels are generated using a weakly supervised k-means clustering, and the DRSegUNet is trained recursively as shown in the figure. As a result, DRSegUNet improves the model's performance with optimal parameters and has been proven to provide better outcomes with limited data.

4.4.1 Pseudo Labelling

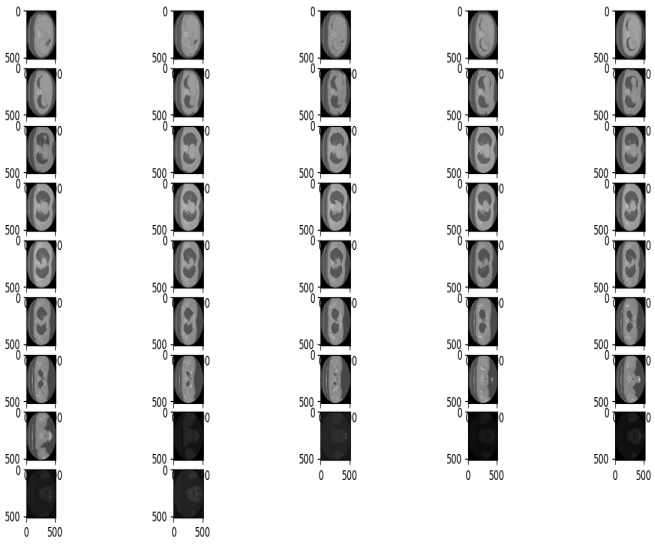


Figure 15: 2D slices of CT scan from a patient

MosMed data set consists of 3D CT images from over 1000+ patients; however, annotations were provided only for 50 images with around 40+ 2D slices in each accommodating up to 2500+ 2D CT scan images. However, not every slice has Covid infection in them. All the 2D slices from a CT scan of a patient are visualized in Figure 15

Data Preparation: Two slices were selected from the whole volumes of the 50 annotated MosMed slices. After careful observation is p is the middle slice

then p-3 and p-2 slices contained a lot of information and were chosen to work with.

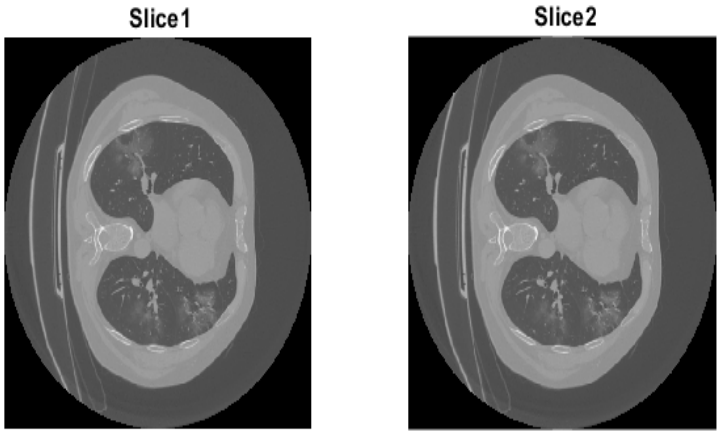


Figure 16: Two extracted 2D slices

Pseudo Labelling Pseudo Labelling was performed on the extracted slices to generate a mask of 3 channels. This is performed in MATLAB by a series of operations.

Step1: The images were normalized with the HU values of air and bone, i.e., -1000,400. A threshold of -170HU has been applied to generate a binary mask of the normalized image.

Step2: bwareafilt (bw,n): bwareafilt extracts n objects from the binary image BW with the largest area, resulting in another binary image BW2 that contains only the objects that meet the criteria. For the obtained mask, n is set to 2 to

obtain the two largest blobs (lungs).

Step3: infill (bw, 'holes'): imfill considers all the background pixels inside the edges as 'holes' and sets their values to the foreground. This is used to avoid information loss.

Step4: The pre-processed mask is applied onto the original image to extract the lungs from the CT-scan retaining only the information in the lungs.

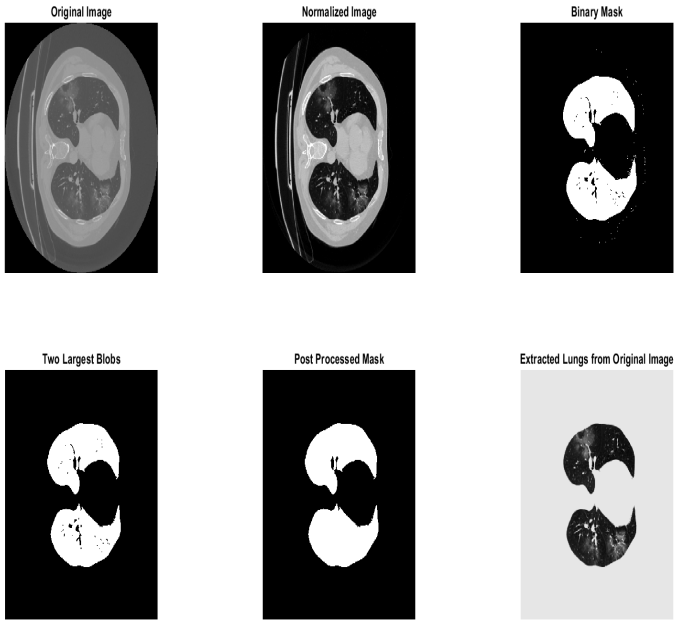


Figure 17: Lung Extraction from CT scans

Step 5: The obtained lung images are pseudo labeled using k-means clustering. For MosMed data, K is set to 3 to obtain three cluster centers background, lungs, and ground-glass opacities. These centers are stored in a 3-channel array, respectively.

Step 6: The images are post-processed by a set of morphological operations such as erosion, dilation, and imclose.

imerode(I,SE) erodes the gray-scale, binary, or packed binary image I, returning

the eroded image, J. SE is a structuring element object or array of structuring element objects 'a', returned by the strel or offsetstrel functions. SE for this problem is a line of length 8 and an angle of 50. This process is then followed by imdilate using the same structuring element. A morphological operation imclose is then used with a disk structuring element of radius 6, which is a dilation followed by erosion.

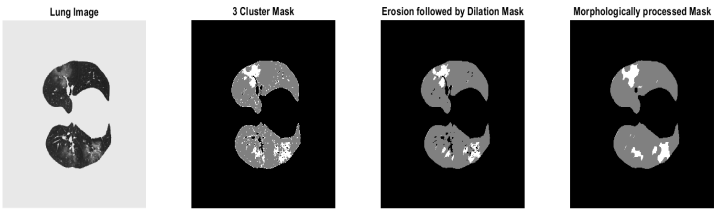


Figure 18: Clustering and Processing the Lung

4.4.2 Recursive Learning

The DRSegUNet is trained in a recursive manner where the first level of the network trains on the pseudo labeled. The trained model will be used to generate a new set of labels that serve as the ground truth for the next level of the model, as shown in Figure 19. An attempt to induce self-learning from pseudo labeled limited data has been adopted in this approach.

The model architecture is the same as described in section 4.3. The first level of the flow DRSegUNet-1 is trained from scratch. The second level uses the concepts of transfer learning. The pre-trained DRSegUNet-1 is fine-tuned where the first two layers of the model are set to non-trainable mode. Then, the rest of the model is trained on the output labels generated from the previous levels.

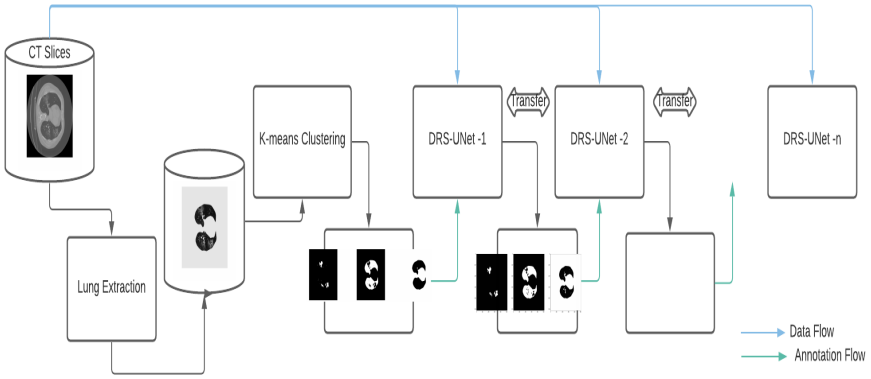


Figure 19: Flow of the approach

4.5 Result Refinement

The softmax function gives probability of a pixel being in a class. These are not labels and are usually between (0,1) as shown in the Figure 10 below .

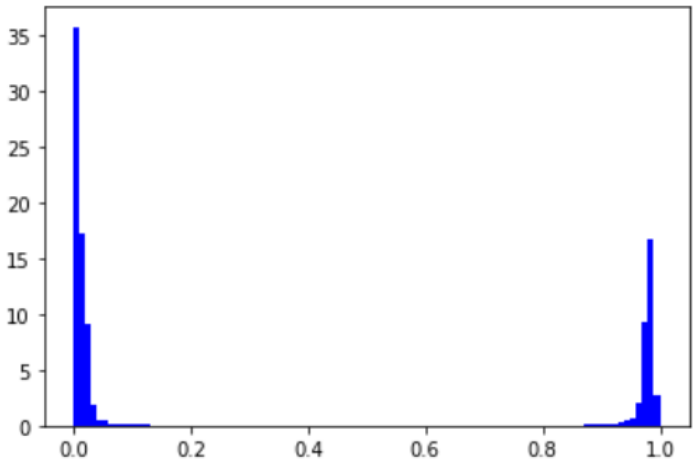


Figure 20: Histogram of Probabilities generated by Softmax

Therefore the output matrix needs to be refined in order to get a labeled mask. A threshold of 0.5 has been set where any probability greater than 0.5 is set to 1 and the others to 0.

$$P_i > 0.5 = 1, P_i < 0.5 = 0$$

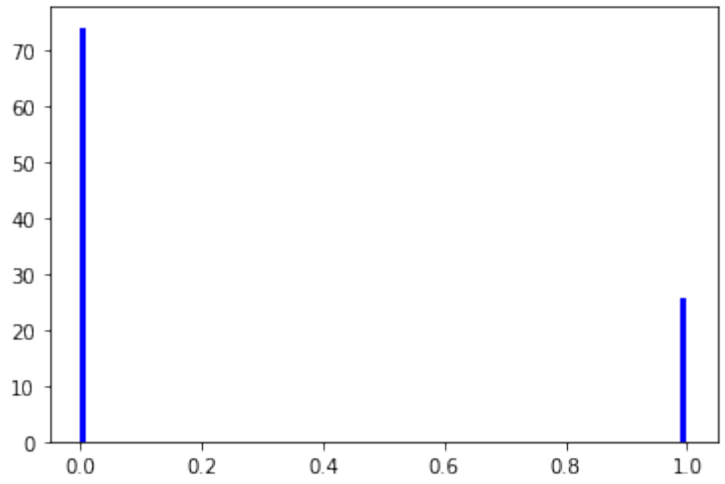


Figure 21: Histogram of Refined Results

5 Experiments and Results

In this section the results of different experiments performed on DRSegUNet and the proposed approach are outlined. The performance of these are evaluated using the metrics mentioned in Section 3.7. The model was implemented using python and keras.

5.1 GPU:

The models were run on Jupyter labs provided by The University of Birmingham and google colab pro+.
Jupyter Labs: This environment is set up on high computing system with CUDA 11.2, Intel(R) Core(TM) i7 (8th Gen),16GB RAM,NVIDIA RTX 2060 GPU with 6GB RAM.
Google Colab Pro+: The cloud on High Ram provides a V100 GPU,53GB of RAM, alongside 8 CPU cores, and is processed on Intel(R) Xeon(R) CPU @ 2.20GHz system.

5.2 DRSegUNet

Dataset: 100 2D CT slices of Covid 19 patients were collected from the Italian Society of Medical and Interventional Radiology and were used to evaluate the performance of the model on the limited data. The slices were all of size 512x512 and high-resolution ct scans with voxel spacing of 1.9x1.9x2.3cm. Each slice

contains information of the lungs stored in the form of Hounsfield(HU) values. The GGO's usually lie in the HU values of (-800,-500) and are considered low-level sensitive information.

Experiment-1: The MedSeg dataset consists 100 images and masks with 4 labelled channels

- 0: Ground Glass Opacities
- 1: Consolidations
- 2: Lungs
- 3: Background

5.2.1 Quantitative Results:

Implementation of DRSegUNet: To optimise the model, the loss function was minimised using the Adam method [Zhang et al., 2018b]. The size of the images were unchanged.

Hyper Parameter Results: A combination of epochs and learning rates were tested on the model.

Epochs	Learning-rate	F1-score	Dice-score	IoU
30	0.01	0.89	0.88	0.87
15	0.01	0.94	0.92	0.86
15	0.00098	0.8	0.85	0.8
40	0.00098	0.9	0.93	0.91
40	**0.0000098**	**0.98**	**0.96**	**0.94**

Table 12: The result of epochs and learning rates

The final model is trained on 30 epochs with a learning rate of 0.0000098. A study on the relation between epochs and learning rate was published [Wilson and Martinez, 2001] describes it as the lower the learning rate, higher training time(epochs). Lower learning rates update the weights slowly, making it require more time to reach the optimal minimum. Higher learning rates led to divergence. The model was able to learn low-level ground-glass opacities. The best model achieved an Intersection over Union of 0.9587, a dice-score of 0.9611, and an F1-score of 0.9735.

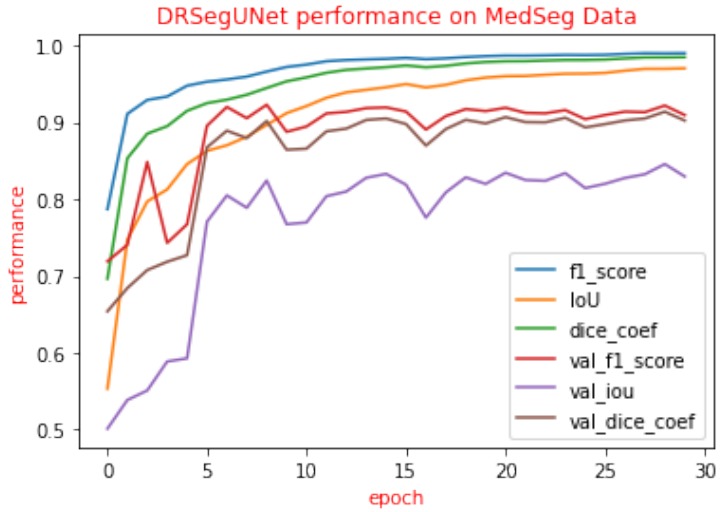


Figure 22: Performance of DRSegUNet on MedSeg

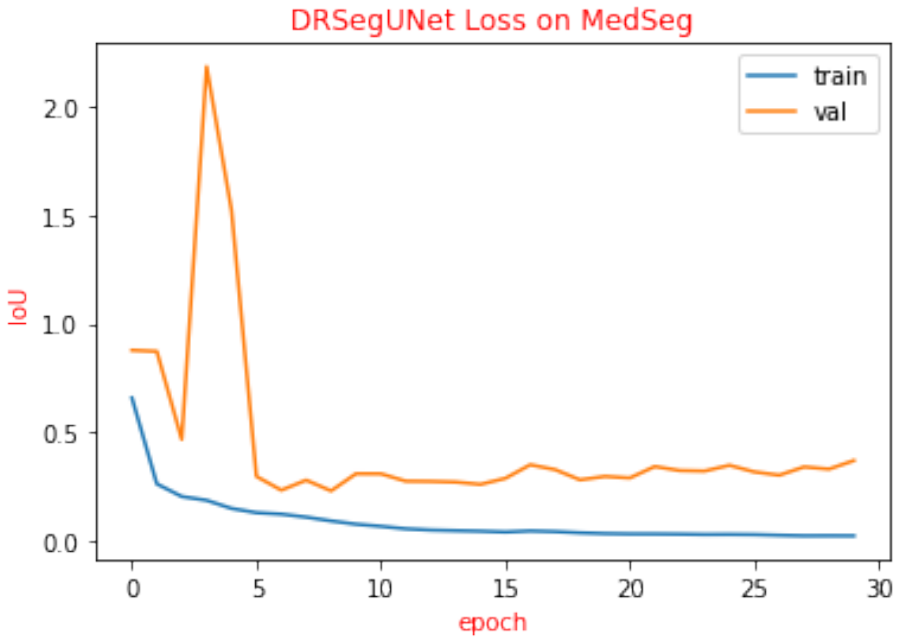


Figure 23: Loss of DRSegUNet on MedSeg

Qualitative Results: Segmentation results cannot only be interpreted by the evaluation metrics. Hence, a CT slice was visualised to show the effectiveness of the model compared to the actual ground truth.

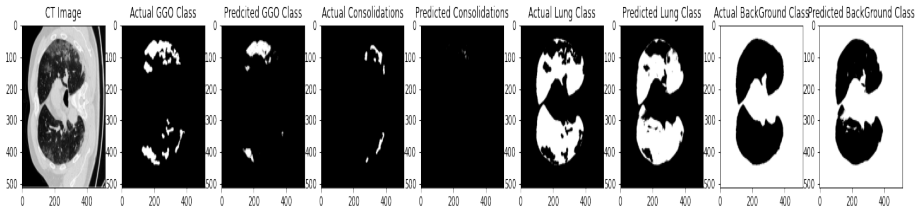


Figure 24: Visualised Segmentation of MedSeg

5.3 Performance of Deep Residual Block:

To assess the performance of the proposed deep residual blocks and to verify its impact, thorough experiments have been conducted with the UNet and the DRSegUNet Residual Blocks against the RadioPedia dataset.

The total number of parameters with 16 start neurons were **2,158,468** for UNet and **5,484,276** for the proposed DRSegUNet. The proposed model resulted in building a network 1.56 times deeper than the state-of-the-art UNet with fewer parameters. Also, the model's performance under transpose convolutions in decoder and UpSampling2D followed by residual blocks is accessed.

The model is trained on the RadioPedia dataset that consists of 829 images collected from 9 patients, out of which around 323 images have the infection. The masks have three labels manually annotated by radiologists, where each image is of size 512x512x1 with similar voxel spacings as MedSeg.

- 0: Ground Glass Opacities
- 1:Lungs
- 2:Background

Hyper-Parameters: A lower learning rate of 0.000098 was set as explained in Section 5.2.1 as lower learning rates aid in efficient learning of complex features and prevent the model from diverging. Figure 22 represents the problem of diverging where the DRSegUNet was trained on a default learning rate of 0.01, and the model started diverging after the 10th epoch. This happens due to a high weights update where the model cannot reach an optimal minimum, so it starts diverging. This is frequent in significantly large problems. Lower learning rates are beneficial since the data has to deal with three channeled outputs and various features from high-resolution CT images.

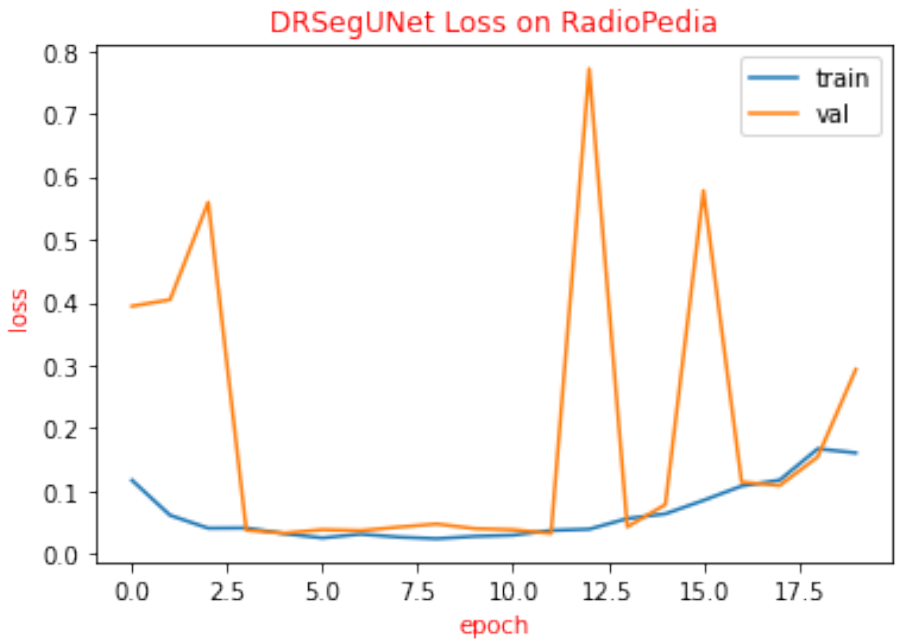


Figure 25: Divergence Problem

Quantitative Results: Three different experiments were performed where the DRSegUNet is trained to evaluate the performance of the Upsampling2D as a method in decoder to increase the feature maps was compared with the Transposed Convolutions. The state-of-the-art UNet has been trained on the same dataset to further highlight the benefits of the residual blocks in DRSegUNet. These were evaluated and the results were tabulated below.

Model	F1-score	Dice-score	IoU
UNet	0.9600	0.9324	0.9132
DRSegUNet(transposed-convolutions in decoder)	**0.9923**	**0.9841**	**0.9735**
DRSegUNet(up-sampling in decoder)	0.9822	0.9689	0.9543

Table 13: Comparison of Different methods on RadioPedia Dataset

Qualitative Results: The segmentation results of the three evaluated methods are visualized in figure 26. All three models showed high quantitative results. However, the UNet suffered from vanishing gradient, and ground-glass opacities are very low-level information. The UNet was able to segment the lungs and the background but had difficulty learning the ground glass opacities. This is due to a shallow network and high-dimensional data with numerous features. While both the up-sampling and the transposed convolutions have similar results with

careful observation, the segmentation of transposed convolutions is closer to the ground truth. Because transposed convolutions tend to learn features, added skip connections help them with vanishing gradient problems.

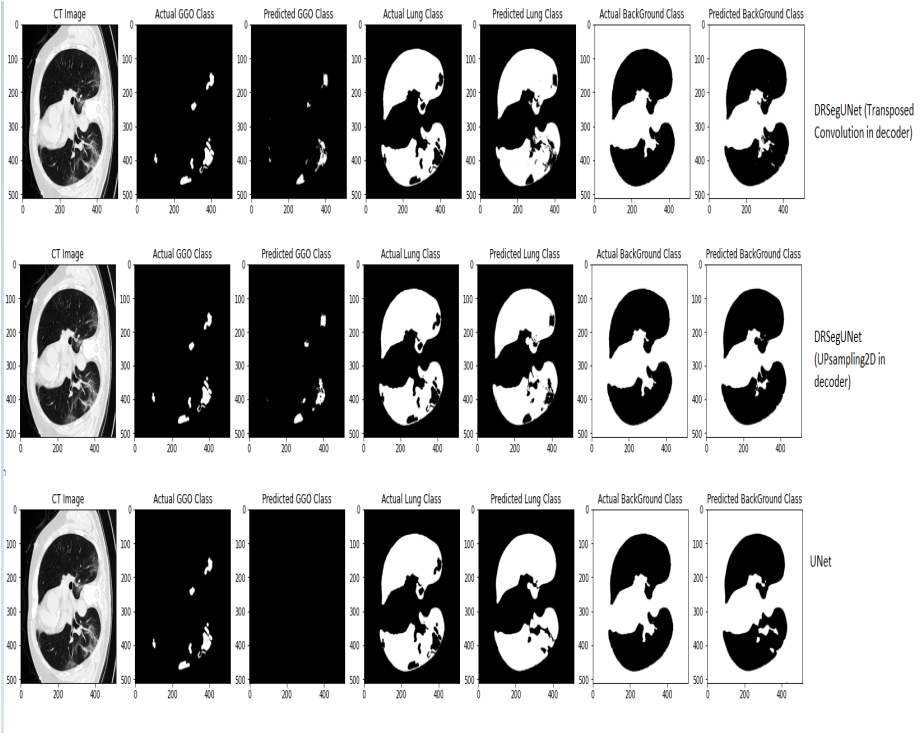


Figure 26: Comparison of Results from different Methods

5.4 DRSegUNet vs other models

Fan et al. [Fan et al., 2020] conducted extensive experiments to compare their proposed model on multi-class segmentation using the MedSeg dataset. The results were compared with DeepLabV3+ (stride 8), DeepLabV3+ (stride 16), FCN8s. Since all the models were run using the same data set as DRSegUNet experiments. We compared our qualitative results with the pre-performed experiments of Fan et al. [Fan et al., 2020].

Qualitative results: DRSegUNet outperformed all the other models as shown in Figure 27 below; DeepLabV3+ had an issue identifying the consolidations, and the ground glass opacities were inconsistent. Furthermore, it can be seen that FCN8s had issues differentiating between consolidations and ground-glass opacities. For example, in CT image 2 in Figure 27, FCN8s predicted even

the consolidations as ground-glass opacities in the right lung, while in CT image 3 in Figure 27, it predicted the ground glass opacities as consolidations. Although Semi-Inf-Net showed remarkable performance compared to the other three models, it could not identify the low-level features, and this issue arises due to vanishing gradients. Given the limited data of 100 images, it is tough for deep learning networks. However, DeepResUNet performed precisely in every segmentation with similar sensitivity as that of a human.

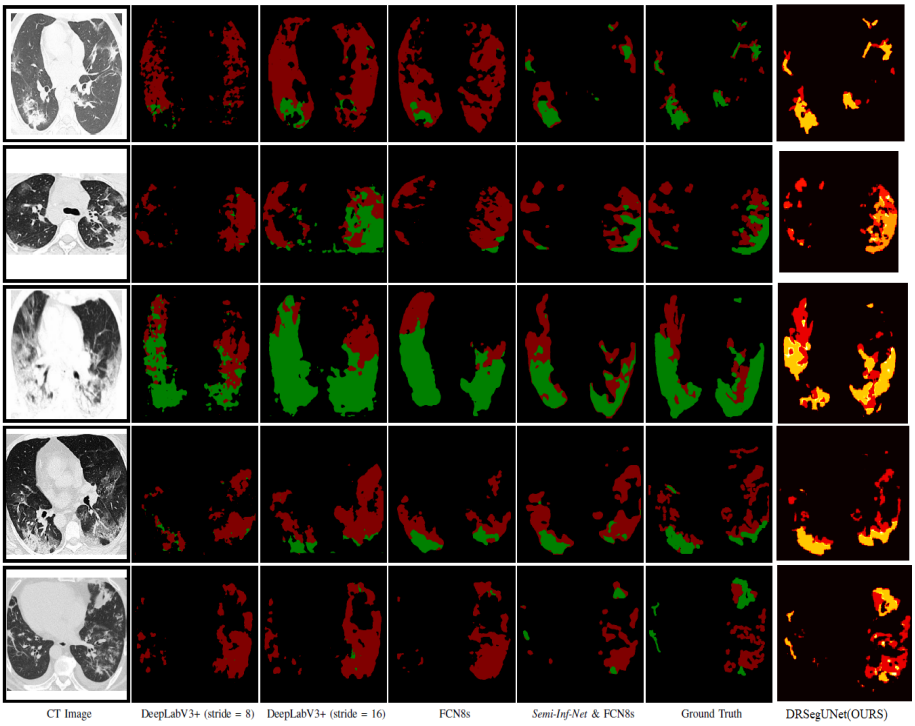


Figure 27: Comparison of Results with the experiments performed by Fan et al. [Fan et al., 2020] where Red = GGO's,Green = consolidations of Fan et al. result Yellow = consolidations of DRSegUNet results

Qualitative Results: A paper published on Covid-19 2D segmentations has performed several experiments on the MedSeg dataset and compared their results with the other existing 2D methods [Saeedizadeh et al., 2020]. The results published by them and the results of DRSegUNet are shown in the table below.

Model	Dice-score
DeepLab-v3+ (stride=8)	0.375
DeepLab-v3+ (stride=16)	0.443
FCN8s	0.471
Semi-Inf-Net+FCN8s	0.646
TV-Unet	0.655
DRSegUNet(ours)	**0.89**

Table 14: Comparison of Different methods on segmenting GGO's on MedSeg Dataset

The model's predictions on GGO's the most important identifiers of Covid-19 was not up-to the par. This performance issue is due to limited data and high level networks where the gradients of the low level features are lost. This issue has been addressed by DRSegUNet and it outperforms all the models with a Dice similarity coefficient of 0.89 only on prediction of GGO labels

5.5 Weakly Supervised Self Learning Approach

Dataset: 100 pseudo labeled 2D CT slices of Covid 19 patients from the 1100 MosMed dataset were evaluated on the model. The slices were all of size 512x512 and high-resolution ct scans.

Implementation: To optimize the model, the Loss function was minimized using the Adam method and implemented in the Keras framework. A total of 100 pseudo labeled training pictures with dimensions of 512 by 512 pixels are provided. These images were selected from the 150 annotated CT volumes of the MosMed dataset. The model has trained from scratch in level1 over 20 epochs and a learning rate of 0.00098. Predictions were made using the trained model on the initial 100 images along with the other 50 MosMed data images. The model is then fine-tuned and trained over a lower learning rate of 0.0000098 over the newly generated labels.

Levels	F1-score	Dice-score	IoU
DRSegUNet-1	0.95	0.94	0.92
DRSegUNet-2	0.93	0.92	0.89

Table 15: Comparison of Performance of different levels

Qualitative Results: It can be observed from Figure 28 below that the model was able to learn by itself. In level 1, despite the noise from the generated pseudo labels, the model was able to identify key features. However, it was not wholly robust to the noise, but the induced self-learning in level 2 enhanced the predictions made by the model, but the loss was high; this is because the

model was trained to optimize the loss function, which penalizes the model with the incorrect predictions. The noise pixels were not predicted, which made the model's loss high. Trying to minimize this loss overthrew the model leading it to predict noisy labels or just no labels, as shown in Example 2 in Figure 28.

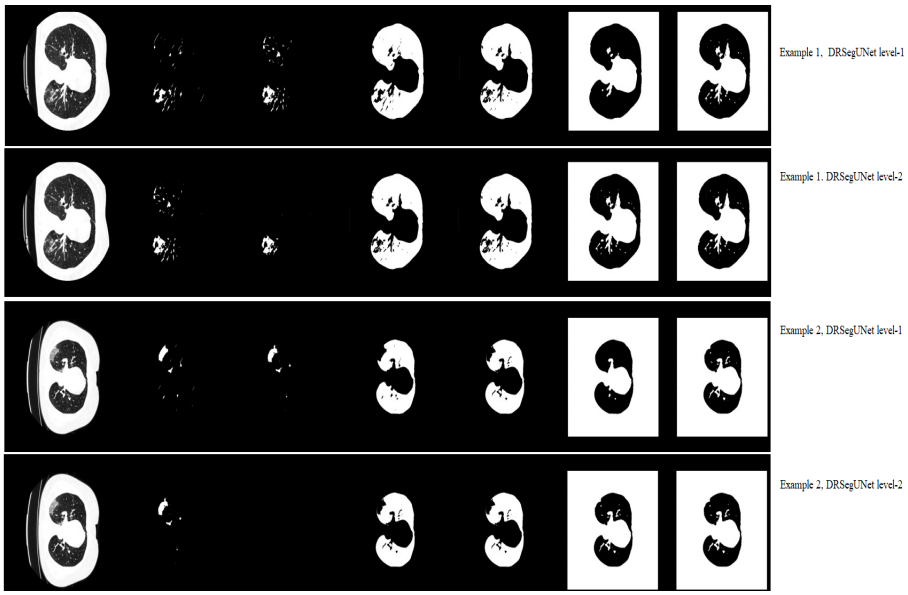


Figure 28: Weakly Supervised Self-learning DRSegUNet Results

6 Discussion

6.1 Strengths

The unavailability of properly annotated datasets is one of the major setbacks observed in all the recent publications of Covid-19. The issues with limited data are also one of the main reasons for the poor performances of 3D models and for the researchers to choose 2D analysis. This challenge has pushed forward the development of deep learning methodologies that work with limited datasets. MedSeg is a widely used dataset containing 100 CT slices and 100 annotated masks of depth 4. However, the low-level sensitive information of Covid-19 GGO's is easily lost while building deeper networks on limited data.

The proposed DRSegUNet model's performance is compared with UNet, DeepLabV3, FCN8s, Semi-Inf-Net, TV-UNet.These proposed models were accurately able to segment lungs and consolidations but were behind in identifying GGO's. DRSegUNet outperformed all the models in segmenting the GGO's with the sensitivity of a human eye. The comparison of the quantitative results of these models

are shown in Table 14.

The accuracy cannot determine the performance of the segmentation models on large problems. However, an accurate idea of the model's segmentation can be established by the Intersection over Union or Dice similarity between the ground truth pixel values and predicted pixel values. The infection pixels are comparatively smaller than the background, making it an imbalance pixel problem. Accuracy would still be high if the model could identify all the black pixels as they are more in number. The proposed model was assessed using the Dice similarity coefficient, Intersection Over Union, and F1-score and gave the results of 0.98 dice score, 0.97 IoU, and 0.99 F1-score. An ablation study has been performed on the decoder block up-sampling components, tabulating the results in Table 13. Transposed convolutions were chosen in the final model as they perform well and learn features while up-sampling.

An attempt at making the DRSegUNet self-learn was one of the experiments proposed in the paper. DRSegUNet naturally performs well on limited data and can learn low-level features. For example, as shown in Figure 28, DRSegUNet was able to identify patterns despite the noise in the pseudo labels. However, the model's behavior was not consistent and required more study before declaring its performance.

6.2 Future Work

The images could not be down-sampled as they led to the loss of information and cropping. Hence, the experiments were conducted on very high computational systems and are incompatible with normal processors, even GPUs with less RAM power. It would also be beneficial to observe the model's performance on mini-batch sizes, which could not be performed due to limited computational resources. The major future prospect would be to optimize the model to work well on normal graphical processor systems without compromising the performance and quality of the results.

The proposed model and the approach focus on the segmentation of the Covid-19 infection. However, in real-time, we would need a system to diagnose the disease and perform segmentation for further examination. Therefore, we aim to incorporate diagnostic systems and segmentation into a single framework in the future.

DRSegUNet: The proposed model, despite its benefits over the state-of-the-art methods, has a scope for improvement. The model was tested on limited data for multi-class segmentation. However, its performance on binary classification was only satisfactory. Therefore, the aim is to tweak the model to make it work with limited binary classifications as most of the annotations available are usually the background and the mask. We aim to test the model on different segmentation tasks and fine-tune it by adding more data to make it a better

openly accessible pre-trained model for lung infection segmentation. This model will be further explored in automatic multi-class segmentation of nuclei based on the color contrasts.

Weakly Supervised Self Learning DRSegUNet: The proposed approach was not completely experimented to its full potential. This approach could be developed into a self-supervised methodology by utilizing unsupervised methods for pseudo labeling and increasing the training data or applying data augmentation techniques to observe the performance on large training sets.

7 Conclusion

In this paper, a modified Deep Residual Segmentation UNet has been proposed to identify low-intensity inflammations from CT scans of Covid-19 patients. A deep residual block has been introduced to provide deeper networks on minimal parameters. The model was carefully assessed through different experiments and achieved a remarkable performance compared to the state-of-the-art and existing systems on limited data. The proposed model has shown a dice score of 0.98 and an Intersection over Union of 0.96. Furthermore, an attempt was made to propose a weakly supervised self-learning on limited data without manual annotations was made. Pseudo-labeling and transfer of knowledge were used to train a model recursively. It was shown that there is a scope to perform segmentation and make a model learn by itself without the annotations. The DRSegUNet can work with several domains and modalities and can be applied in the diagnosis and prognosis. The proposed approach can be further modified to aid in self-supervised learning, eliminating the issue of annotations.

References

M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424, 2000.

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.

X. Chen, L. Yao, T. Zhou, J. Dong, and Y. Zhang. Momentum contrastive learning for few-shot covid-19 diagnosis from chest ct images. *Pattern recognition*, 113:107826, 2021.

A. A. Dawood. Mutated covid-19 may foretell a great risk for mankind in the future. *New microbes and new infections*, 35:100673, 2020.

V. Dumoulin and F. Visin. A guide to convolution arithmetic for deep learning. 03 2016.

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.

D. L. Fung, Q. Liu, J. Zammit, C. K.-S. Leung, and P. Hu. Self-supervised deep learning model for covid-19 lung ct image segmentation highlighting putative causal relationship among age, underlying disease and covid-19. *Journal of Translational Medicine*, 19(1):1–18, 2021.

G. Gaál, B. Maga, and A. Lukács. Attention u-net based adversarial architectures for chest x-ray lung segmentation. *arXiv preprint arXiv:2003.10304*, 2020.

J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108, 1979.

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*, pages 770–778, 2016.

C. Huang, Y. Wang, X. Li, L. Ren, J. Zhao, Y. Hu, L. Zhang, G. Fan, J. Xu, X. Gu, et al. Clinical features of patients infected with 2019 novel coronavirus in wuhan, china. *The lancet*, 395(10223):497–506, 2020.

S. Jin, B. Wang, H. Xu, C. Luo, L. Wei, W. Zhao, X. Hou, W. Ma, Z. Xu, Z. Zheng, et al. Ai-assisted ct imaging analysis for covid-19 screening: Building and deploying a medical ai system in four weeks. *MedRxiv*, 2020.

H. Li, S.-M. Liu, X.-H. Yu, S.-L. Tang, and C.-K. Tang. Coronavirus disease 2019 (covid-19): current status and future perspectives. *International journal of antimicrobial agents*, 55(5):105951, 2020a.

L. Li, L. Qin, Z. Xu, Y. Yin, X. Wang, B. Kong, J. Bai, Y. Lu, Z. Fang, Q. Song, et al. Artificial intelligence distinguishes covid-19 from community acquired pneumonia on chest ct. *Radiology*, 2020b.

J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of big data*, 2(1):1–21, 2015.

C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall. Activation functions: Comparison of trends in practice and research for deep learning. 12 2020.

F. Pan, T. Ye, P. Sun, S. Gui, B. Liang, L. Li, D. Zheng, J. Wang, R. L. Hesketh, L. Yang, et al. Time course of lung changes on chest ct during recovery from 2019 novel coronavirus (covid-19) pneumonia. *Radiology*, 2020.

L. Peñarrubia, M. Ruiz, R. Porco, S. N. Rao, M. Juanola-Falgarona, D. Manissero, M. López-Fontanals, and J. Pareja. Multiple assays in a real-time rt-pcr sars-cov-2 panel can mitigate the risk of loss of sensitivity by new genomic variants during the covid-19 outbreak. *International Journal of Infectious Diseases*, 97:225–229, 2020.

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*, pages 234–241. Springer, 2015.

N. Saeedizadeh, S. Minaee, R. Kafieh, S. Yazdani, and M. Sonka. Covid tv-unet: Segmenting covid-19 chest ct images using connectivity imposed u-net. 07 2020.

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. *arXiv preprint arXiv:2003.04655*, 2020.

F. Shi, J. Wang, J. Shi, Z. Wu, Q. Wang, Z. Tang, K. He, Y. Shi, and D. Shen. Review of artificial intelligence techniques in imaging data acquisition, segmentation, and diagnosis for covid-19. *IEEE reviews in biomedical engineering*, 14:4–15, 2020.

F. Shi, L. Xia, F. Shan, B. Song, D. Wu, Y. Wei, H. Yuan, H. Jiang, Y. He, Y. Gao, and et al. Large-scale screening to distinguish between covid-19 and community-acquired pneumonia using infection size-aware classification. *Physics in Medicine Biology*, 66(6):065031, Mar 2021. ISSN 1361-6560. doi: 10.1088/1361-6560/abe838. URL `http://dx.doi.org/10.1088/1361-6560/abe838`.

G. Song, G. Liang, and W. Liu. Fungal co-infections associated with global covid-19 pandemic: a clinical and diagnostic perspective from china. *Mycopathologia*, pages 1–8, 2020.

A. Ulhaq, A. Khan, D. Gomes, and M. Paul. Computer vision for covid-19 control: A survey. *arXiv preprint arXiv:2004.09420*, 2020.

M. Wadman, J. Couzin-Frankel, J. Kaiser, and C. Matacic. A rampage through the body, 2020.

Y. Wang, J. Zhang, M. Kan, S. Shan, and X. Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12275–12284, 2020.

D. Wilson and T. Martinez. The need for small learning rates on large problems. volume 1, pages 115 – 119 vol.1, 02 2001. ISBN 0-7803-7044-9. doi: 10.1109/IJCNN.2001.939002.

J. T. Wu, K. Leung, and G. M. Leung. Nowcasting and forecasting the potential domestic and international spread of the 2019-ncov outbreak originating in wuhan, china: a modelling study. *The Lancet*, 395(10225):689–697, 2020.

X. Xu, X. Jiang, C. Ma, P. Du, X. Li, S. Lv, L. Yu, Q. Ni, Y. Chen, J. Su, and et al. A deep learning system to screen novel coronavirus disease 2019 pneumonia. *Engineering*, 6(10):1122–1129, Oct 2020. ISSN 2095-8099. doi: 10.1016/j.eng.2020.04.010. URL `http://dx.doi.org/10.1016/j.eng.2020.04.010`.

Z. Ye, Y. Zhang, Y. Wang, Z. Huang, and B. Song. Chest ct manifestations of new coronavirus disease 2019 (covid-19): a pictorial review. *European radiology*, 30(8):4381–4389, 2020.

J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018.

L. Zhang, V. Gopalakrishnan, L. Lu, R. Summers, J. Moss, and J. Yao. Self-learning to detect and segment cysts in lung ct images without manual annotation. pages 1100–1103, 04 2018a. doi: 10.1109/ISBI.2018.8363763.

Z. Zhang, Q. Liu, and Y. Wang. Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 15(5):749–753, 2018b.

X. Zhao, P. Zhang, F. Song, G. Fan, Y. Sun, Y. Wang, Z. Tian, L. Zhang, and G. Zhang. D2a u-net: Automatic segmentation of covid-19 ct slices based on dual attention and hybrid dilated convolution. *Computers in biology and medicine*, page 104526, 2021.

C. Zheng, X. Deng, Q. Fu, Q. Zhou, J. Feng, H. Ma, W. Liu, and X. Wang. Deep learning-based detection for covid-19 from chest ct using weak label. *MedRxiv*, 2020.

8 Appendix

The code is available to access on GitLab at **https://git-teaching.cs.bham.ac.uk/mod-msc-proj-2020/vxb038**

Data:

- MedSeg dataset is available in the folder called data in the GitLab repository. The images are stored in the file **images-medseg.npy** whereas the masks are stored in **masks-medseg.npy**. The RadioPedia masks are also available in the same folder data/ named as **mask1-radiopedia.npy.npz**. However, the images file for radiopedia needs to be downloaded from google drive. **click here to go the drive**. All the data files can also be downloaded from the google drive
- . The MosMed images and files are stored in the google drive **click here to go the drive** where all the MosMed images are stored in the folder named **data2d** while the masks are stored in the **mask**. These are Nifti files.

Dependencies: The model is run on jupyter using python. It also requires some pre-packages to be installed before running the model. These packages can be installed in jupyter using the pip install command. The packages are matplotlib, NumPy, pandas, opencv-pyhton, nibabel, os, scipy, TensorFlow, Keras, glob. Matlab is used to generate pseudo labels. Install Matlab to use the code file. Make sure all the models are the latest version.

Models: All the trained models used to produce the results shown in Section 5 are available in the folder pt-models in the GitLab repository. Model trained on MedSeg data is named as **medSeg**, RadioPedia is named as **RadioPedia**. The weakly supervised self-learning first level is stored as **selflvl1.11** while the second level is stored as **selflvl1.12**.

Code: All the code files are available in the GitLab repository as .ipnyb files. The code for DRSegUNet is in a file named **DRSegUNet.ipnyb** The code for weakly supervised self-learning is in a file named **sl-DRSegUNet**

Implementation: GitLab repository as contains a *ReadMe.txt* file that contains all the information on processing the code.

The pseudo labels were already generated, so there is no necessity to use the Matlab file further. The code for reference has been uploaded on GitLab repository under the filename **processing.m**. In addition, a **matlab-readme.txt** has all the instructions on processing the code.