Few shot segmentation for COVID-19 infected lung CT slices

A Thesis to be Submitted

in Partial Fulfilment of the Requirements

for the Degree of

Master of Technology

by

SOHAM CHATTERJEE

Roll No : CS2012

under the supervision of

Dr. Sushmita Mitra

Machine Intelligence Unit

Indian Statistical Institute

Indian Statistical Institute, Kolkata

Kolkata - 700108, India

Certificate

This is to certify that the dissertation entitled **"Few shot segmentation for COVID-19 infected lung CT slices"** submitted by **Soham Chatterjee** to Indian Statistical Institute, Kolkata, in partial fulfillment for the award of the degree of **Master of Technology in Computer Science** is a bonafide record of work carried out by him under my supervision and guidance. The dissertation has fulfilled all the requirements as per the regulations of this institute and, in my opinion, has reached the standard needed for submission.

....

Sushmita Mitra

Machine Intelligence Unit

Indian Statistical Institute

Date: 13/07/2022

Declaration

I hereby declare that this thesis report titled "Few shot segmentation for COVID-19 infected lung CT slices" is my own original work carried out as a postgraduate student at Indian Statistical Institute, Kolkata, except the assistance from other sources which is duly acknowledged.
All sources used for this project report have been fully and properly cited. It contains no material which to a substantial extent, has been submitted for the award of any degree/diploma in any institute or has been published in any form, except where due acknowledgment is made.

....

Soham Chatterjee

Roll - CS2012

Master of Technology in Computer Science,

Indian Statistical Institute, Kolkata

Dedication

To Parents.....

Acknowledgements

I would like to express my sincere gratitude to my guide **Dr. Sushmita Mitra** for allowing me to work under her and providing me exciting problems to think upon. Throughout the development of this thesis, she was always available for discussion and guidance and has provided support, patience and optimism.

I would also like to thank **Dr. B. Uma Shankar**, Associate Professor, Machine Intelligence Unit, Indian Statistical Institute, Kolkata, for his valuable feedback.

I would also like to thank **Pallabi Dutta**, Junior Research Fellow, Machine Intelligence Unit, Indian Statistical Institute, Kolkata, for her valuable suggestions and discussions.

I will be failing in my duties if I don't mention friends and people who mattered and conversing with whom did help me during two years at ISI Kolkata. In this period, I have been enriched and nurtured in numerous ways by interacting with many people.

Finally, I would like to express my special gratitude to my parents, who have always encouraged me in all aspects for igniting the love for knowledge within me.

Abstract

The last 2 years have been adversely affected by the COVID-19 pandemic. Doctors usually detect Covid from CT slices from features such as ground glass, consolidation and pleural effusion. These features usually have complex contours, irregular shapes and rough boundaries. With increasing number of cases the workload on the radiologists have increased by leaps and bounds to analyze the lung CT scans for tracking the disease progression in the patient. Moreover manual analysis of the CT scans is also prone to human error. So automated segmentation of infected lung CT slices can help the doctors to diagnose the disease faster. With the advent of deep learning, various approaches have been built to tackle this problem of automated biomedical image segmentation. One such architecture is the U-Net by Ronnenberger et al. [14]. Various other approaches have been proposed which are all variations of the U-Net to achieve better segmentation performance. However, the U-Net and its variations suffer from high model complexity, due to which they easily overfit on limited labelled dataset which is a serious issue in medical image domain. To cater this problem of data scarcity, research in "few shot segmentation" has gained significant importance in the recent years. In this work, we have developed a deep neural network model called Few Shot Conditioner Segmenter Covid (FSCS-cov), an architecture to tackle the problem of segmenting different COVID-19 lesions from limited number of COVID-19 infected lung CT slices using few - shot learning paradigm.

Keywords: Diagnosis using deep learning · COVID-19 · Segmentation · Computed Tomography · Few shot learning

Contents

List of Figures

Chapter 1

Introduction

Coronavirus disease 2019 (COVID-19) is an infectious illness caused by SARS-CoV-2 virus. When people are exposed to respiratory droplets and airborne particles from an infected person, the disease spreads from the infected person to the healthy one. Fever, coughing, headaches, exhaustion, respiratory problems, a loss of taste and smell are possible symptoms. Symptoms can arise as soon as one day and may take up to fourteen days. The RT-PCR test is considered to be the gold standard for diagnosing the illness till date. The rapid spread of the disease has been a great issue of concern for the public as well as the healthcare community. The government has ordered the wearing of face masks or coverings in public places, which can considerably aid in the transmission of the sickness, in order to stop the disease's rapid spread.To diagnose the disease at a fast pace, we need to come up with automated image segmentation techniques.

In the last decade, deep learning has evolved at a massive scale. With the development of deep convolutional neural network(CNN's) architectures, people have cracked some of the greatest challenges in computer vision. Researchers have also come up with various modifications of CNN's such as RCNN by Girshick et al.[4], Fast-RCNN by Girshick[3], Faster-RCNN by Ren et al.[13], YOLO by Redmon et al.[12] to tackle the problem of object detection. However, this problem becomes even more challenging in the medical domain since the lesions present in medical images do not have any regular patterns as in natural images but instead have irregular shapes and complex contours. For example, as shown in 1.1, the lesion corresponding to ground glass opacity(GGO) has a blurred appearance and low contrast.

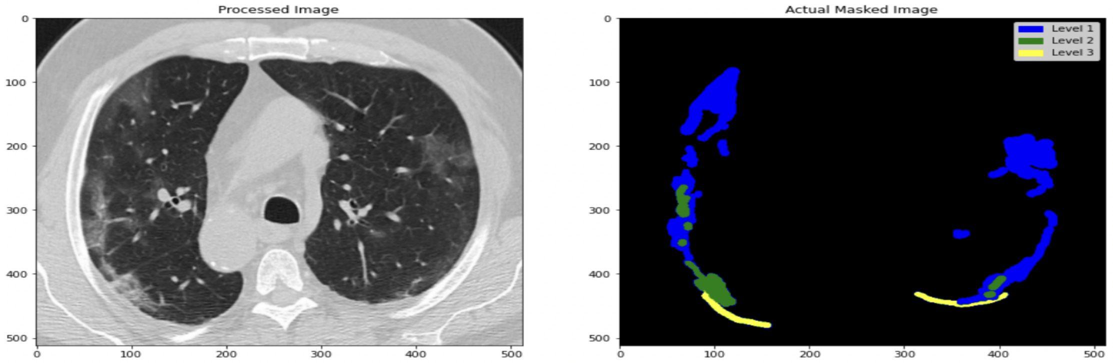

Figure 1.1: A radiologist segmented this CT slice into 3 labels:ground glass(Mask label=1), consolidation(Mask label=2) and pleural effusion(Mask label=3).

1.1 Problem Statement

To solve this problem of image segmentation in medical images, Ronnenberger et al. [14] have come up with the U-Net architecture. Later Oktay et al.[11] came up with attention mechanism in U-Net, which became popularly known as Attention U-Net. Other approaches have also been developed in the recent years which are variations of the U-Net or Attention U-Net.

One major problem with all the above architectures is that that they have high model complexity and require lots of annotated data. They usually overfit in limited data scenarios. However, the availability of labelled data is scarce, especially in the medical domain.

To solve the above problem of learning from limited data, few shot learning paradigm has become quite popular in the recent past. It aims to make predictions on unseen data from very few labelled samples. It is based on the fact that humans can learn generalized patterns and features of any object even after seeing very few images of that object for e.g. human beings can easily identify a dog from its surroundings even after learning how a dog looks like from just one or two images of it. This learning prototype requires very less annotated data to train a model and is also computationally fast.

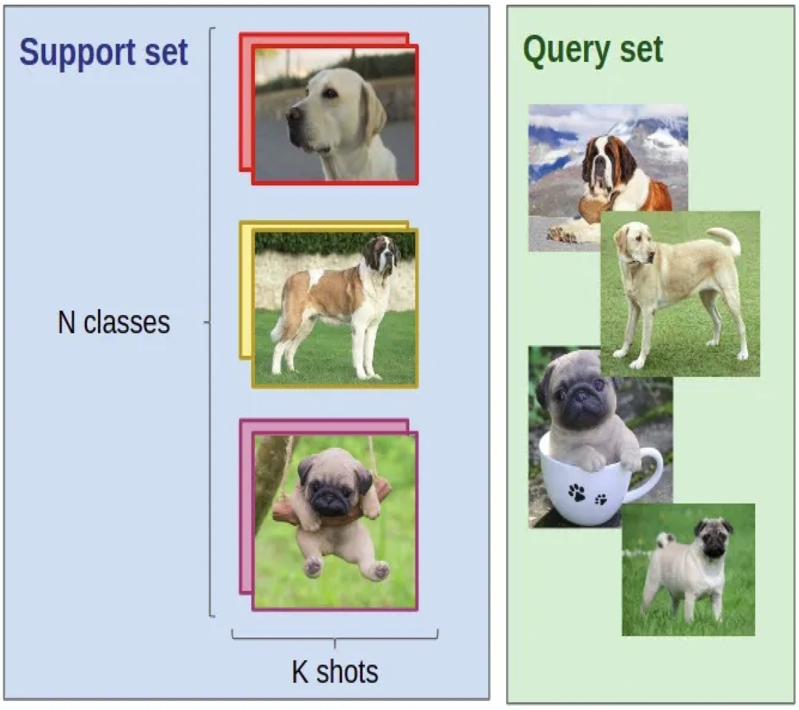

Figure 1.2: Few shot learning

In Few shot learning, as shown in Fig. 1.2, we divide our training dataset into support and query sets. If the number of images in the support set is K, then it is called K-shot learning. While training, the model is trained in episodes. In each episode, the model learns from the K examples of a class and then the model predicts on the query input which is a different image from the support set but is from the same class. Based on this prediction, a loss is calculated and the model is trained on all the N classes in a similar approach. After training, the model generalizes well on unseen data.

So, we are trying to model the problem of image segmentation using few shot learning, i.e.- few shot segmentation. In few shot segmentation tasks, we need to build a model that outputs infection mask corresponding to the input CT slice of the infected patient. We are using CT scans since they capture more anatomical information compared to X-rays. The following section explains all the traditional and recent architectures used for biomedical few shot segmentation.

1.2 Related Work

The defacto model for biomedical image segmentation is the U-Net by Ronneberger et al. [14]. The skip connections introduced in the encoder-decoder framework helped the model to take into account both the high level feature maps as well as the low level feature maps. Skip connections introduced at each stage of information processing helped the model to generalize from the local features as well as the global features which further leads to better localization of region of interest in the output segmentation mask.

Various models down the line are modifications of the U-Net architecture with different blocks added to enhance the information learning process. One such method is the Attention U-Net by Oktay et al. [11] where attention blocks are introduced at each level to focus on certain feature maps which carry more information about the infection. The irrelevant feature maps are suppressed and important feature maps are highlighted. The attention coefficients are used to re-weight these feature maps. Variations of these such as applying attention blocks at different levels of the model architecture and then concatenating the feature maps from these levels with the global feature maps obtained from the final layers of the model and use this combination to make predictions have also been tried.

Later due to large number of architectures people have also come up with a framework called nnU-net('no new net') by Isensee et. al [6] which is a framework to decide the different sets of hyperparameters such as loss function, optimizer etc., configuration of standard U-Net architecture with different variations such as attention gates, residual connections, dilated convolutions etc. using some heuristics. Different sets, each containing a different combination of hyperparameters are prepared. Once this is done, different U-Nets are trained, each with a different parameter set on the same training data. Finally, the ensemble of all these U-Nets, each with a different network configuration is used to determine the dice score coefficient on the training data. Finally, the best configuration is used for prediction on the test data.

People have also come up with Inf-Net (Infection Segmentation Deep Network) by Fan et al. [2] which consists of reverse attention, edge attention modules and a paralleled partial decoder. The parallel partial decoder aggregates the high level feature maps and generates a global map, which guides the reverse attention modules. The reverse attention modules use this global information to accurately label the infection. Together with the edge attention modules, the edge feature maps are generated which are then used to model the contours and boundaries.

Similarly, in few shot segmentation people have also come up with SSA-Net (Spatial self-attention network) by Wang et. al [20] where they have used spatial convolution blocks inside the feature re-extractor and self-attention learning in the feature extractor. The self attention module is used in the feature encoder where it is used to expand the receptive field and extract more contextual information from the deeper layers of the network. Similarly the spatial convolution module is used in the bottle neck between the encoder-decoder architecture, where channel wise convolutions with large kernels are used to extract more spatial information about the rough boundaries and hazy shapes of the lesions.

Another work in this area is Few-Shot U-Net by Voulodimos et al. [19] where they used a U-Net with less training data initially. The results on the test set were evaluated by medical experts. Those images on which the model performed poorly were corrected and then augmented in the training set. The U-Net is again trained on this new set and this process is repeated until the model reaches a decent level of performance.

People have also modelled this problem in terms of contrastive loss such as by Shorfuzzaman et al.[16] where they have used Siamese neural networks by Koch et al.[9] and contrastive loss to design the framework. To capture unbiased feature representations, a fine-tuned pre-trained CNN encder was used.

Attention is an important concept in deep learning. People such as Jetley et al.[7] have used attention estimators at three distinct levels of VGG(Very Deep Convolutional Networks for Large Scale Image Recognition) by Simonyan et al.[17].

This is done in order to capture the lowest, intermediate and highest level feature maps. Finally, each of these feature maps is concatenated with the global level feature map obtained from the final convolutional layer of VGG. The attention coefficients from these three estimators are used for segmentation of input image.

1.3 Organization of Thesis report

The second chapter talks about the models and techniques which we have experimented with. Initially, due to high model complexities and class imbalance in the dataset, the models could not generalize well. Later we developed an architecture which is able to generalize on very less labelled dataset. Details of the architecture have also been explained.

The third chapter talks about the experimental results obtained from our model. Initially it talks about the dataset and its sample distribution. Later it talks about two different strategies of preparing the data in order to train the model. This is followed by qualitative and quantitative analysis of the results obtained from our proposed model. We have tabulated the results from our model and other existing models. We have also done multiple ablation studies to understand the behaviour of our model in different technical settings. We have also documented the qualitative and quantitative results for these studies. Lastly, we have also tabulated the results of our experiments performed as part of ablation studies.

Finally, the last chapter talks about further possibilities and scope of future work.

Chapter 2

Model development

2.1 Models explored

The traditional U-Net by Ronnenberger et al. [14] has millions of trainable parameters due to which it requires large labelled dataset. To solve the problem in limited data domains, people use different augmentation strategies like random cropping, rotations and random change in contrast etc. It is observed that when large amount of data is augmented, then only it performs reasonably well; otherwise it overfits in limited data scenarios. If the amount of data augmented is less, the model still overfits because of high model complexity.

Similarly, experiments carried out on Attention U-Net by Oktay et al.[11] led to the same fate since its complexity is also high. We also observed that the results were slightly inferior compared to the U-Net which could happen due to the reason that the feature maps that were given more importance due to attention gates might have been memorized by the model due to high complexity. It thus failed to generalize well. Another possibility could be higher number of parameters in Attention U-Net as compared to U-Net which might have led to overfitting on the limited number of samples.

Transfer learning has massively become popular in the computer vision domain. Pre-trained networks such as Resnets by He et al.[5] have been massively used for computer vision tasks due to their generalization capability. However, there are no such pre-trained networks in the medical domain. This further puts a limitation on transfer learning in this field. The limitation of these networks is that they were trained on real world object dataset which have well defined shapes and structures. However, these things do not apply in the medical domain as the images deal with histological structures which are semantically different from real world objects. Moreover, the areas representing infection in these medical images have complex shapes and boundaries. Experiments such as using pre-trained resnet50 encoder in the U-Net architecture have given good results if we use large data augmentation; otherwise the model overfits. This is probably because the resnet50 encoder has millions of parameters.

Attention plays an important aspect in deep learning applications since it prioritizes certain feature maps and gives less importance to irrelevant or those feature maps which cater to background information. So, experiments such as cropping the lung portion of the CT scan based on the available lung mask have been tried. Since the infections only reside inside the lung area, so the irrelevant background information is being cropped off in the pre-processing tasks. However such experiments have not been fruitful since the background also adds some contextual information which is important for understanding the contours of the lung and subsequently the infection related information inside it. Loss of background information proves to be detrimental in cases where the infection is present on the inner lining of the lung. The background contains vital spatial information and is also important for understanding the neighborhood context of the lung. In cross domain scenarios where one dataset contains the lung masks and the other dataset doesn't, building a model to predict lung masks in the other dataset and then predicting the infection masks from the cropped images were not successful as the error in predicting lung masks got added to the error in predicting infection masks.

Model hyperparameter tuning has also been tried, such as the optimizer, weight initializers, activation functions but still the above models did not perform well due to high complexity.

2.2 Proposed architecture

In this section, we explain the details of our model architecture, Few Shot Conditioner Segmenter Covid (FSCS-cov) and the functionalities of each block used in it.

2.2.1 Architectural design

FSCS-cov consists of three building blocks: a conditioner arm, a segmenter arm and interaction blocks. Both the conditioner and segmenter arms consist of an encoder-decoder architecture. The conditioner arm processes the support set which consist of a CT image and its corresponding ground truth. The segmenter arm processes the query input which consists of a CT image from the same class as the support set. The interaction blocks obtain contextual and spatial information from the conditioner arm and highlight the feature maps of the segmenter arm.

Architectural details of FSCS-cov are as below:

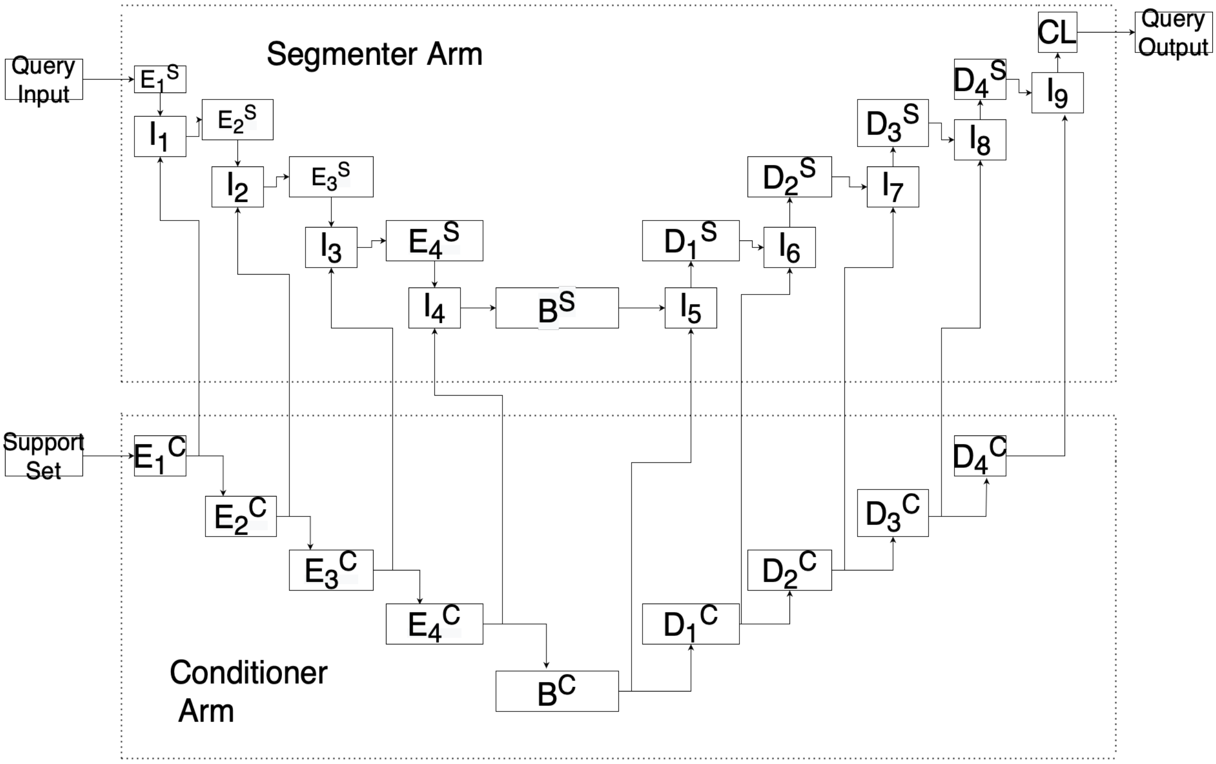

Figure 2.1: Conditioner Segmenter Architecture: E stands for Encoder block, D stands for Decoder block, B stands for Bottleneck block, I stands for Interaction block, CL stands for Classifier Block. C stands for Conditioner arm and S stands for Segmenter arm

2.2.2 Conditioner Arm

The conditioner arm takes in input from the support set which consists of the support image I_s and its corresponding ground truth L_s and generate feature maps which are capable of capturing the necessary areas for segmentation in the query input I_q. It has an encoder-decoder based architecture, where the encoder consists of four encoder blocks and the decoder consists of four decoder blocks with a bottleneck block separating the encoders from the decoders.

As shown in Fig. 2.2 , the encoder block consists of a convolutional layer with stride 1, (5,5)-sized kernels and 16 output feature maps. These are followed by batch normalization and a ReLU activation function. It is then followed by a max-pooling layer with (2,2) kernel size and a stride of 2, which is used to decrease the spatial dimension by half.

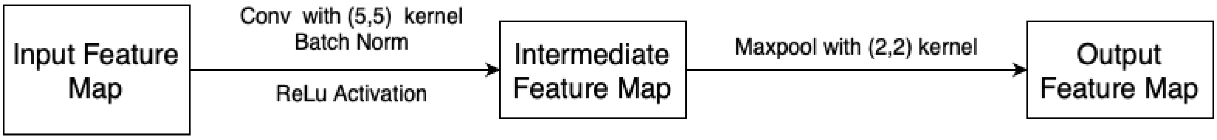

Figure 2.2: Encoder Block

Similarly, as shown in Fig. 2.3, the decoder block consists of an unpooling layer followed by a convolutional layer with stride 1 and (5,5)-sized kernels. These are followed by batch normalization and a ReLU activation function.

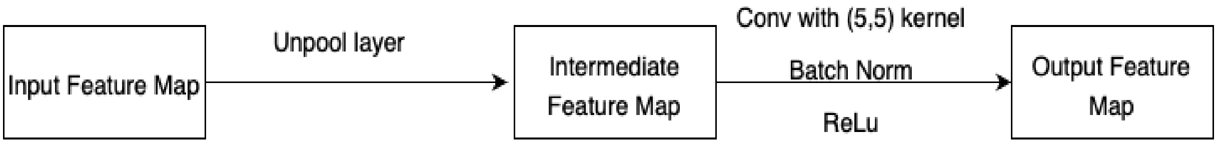

Figure 2.3: Decoder Block

In contrast to the standard U-Net [14] architecture, no skip connections are present between the encoder and decoder blocks. It is because if we use skip connections, the capability of the model to gain contextual information from the support set and use that to predict the mask of the query input is lost.

2.2.3 Segmenter Arm

The segmenter arm is also made up of an encoder-decoder framework.However, it differs in two key aspects. The convolutional layers of the encoder and the decoder in the segmenter consist of 64 output feature maps which is 16 in case of the conditioner. So the segmenter has a higher complexity when compared to the conditioner arm. Moreover, the segmenter outputs a segmentation map, which is fed into the classifier block as shown in Fig 2.4, which is a convolutional layer with kernel size (1,1) followed by a a softmax layer to predict the infection segmentation in query slice.

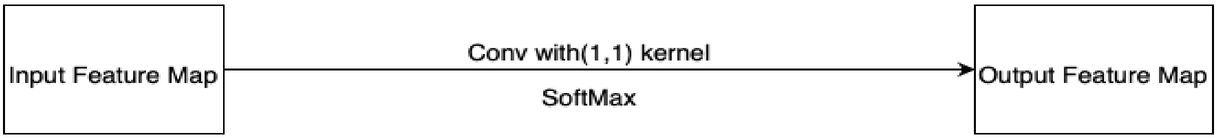

Figure 2.4: Classifier Block

As shown in Fig. 2.5, the bottleneck block consists of a convolutional layer with (5,5)-sized kernels followed by batch normalization and a ReLU activation function.

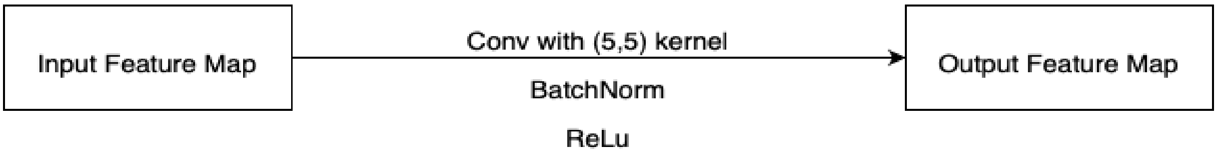

Figure 2.5: BottleNeck Block

2.2.4 Interaction blocks

The interaction blocks play a major role in the segmentation task. From Fig. 2.6, we can see that these blocks take the segmentation related contextual information from the conditioner arm and use it to reweigh the segmentation maps of the segmenter arm, which ultimately help in the segmentation of the query image. These blocks have low complexity so that the computation cost is slightly increased and the training process carries on smoothly without any hindrance in flow of gradients. Each interaction block consists of a convolutional layer with (1,1) kernel size and a sigmoid activation function so that the activation outputs are scaled to $(0, 1)$ and then followed by element wise multiplication to highlight those feature maps which contain important contextual information about the infection.

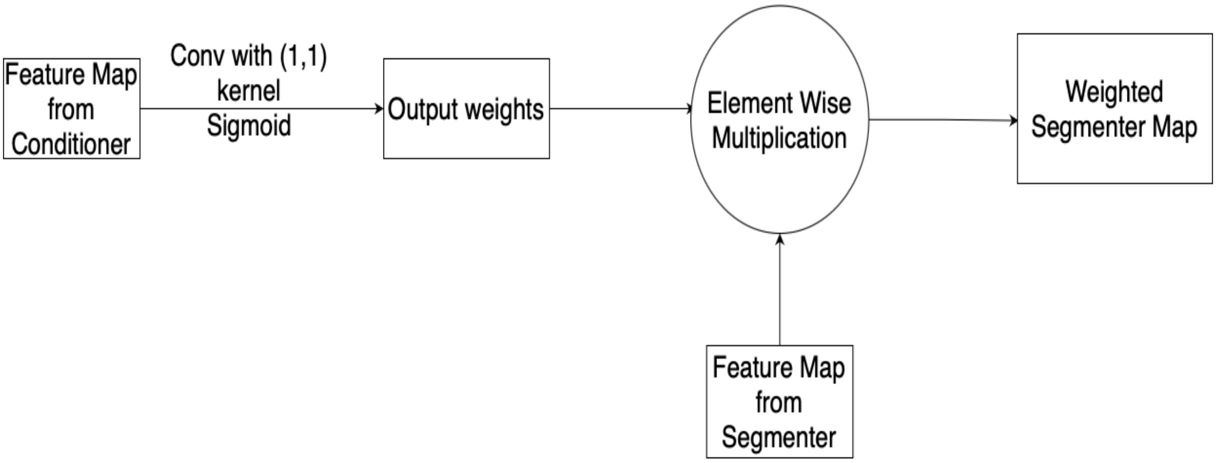
Feature Map from Conditioner
Conv with (1,1) kernel Sigmoid
Output weights
Element Wise Multiplication
Weighted Segmenter Map
Feature Map from Segmenter

Figure 2.6: Interaction Block

2.2.4.1 Attention module as interaction block

Furthermore, it is also observed that if we use attention gates similar to the one used in Attention U-Nets, the results slightly improve. This is because attention coefficients also help in re-weighting important feature maps necessary for the segmenter arm. The feature maps from the conditioner and segmenter each undergo a convolutional layer with (5,5) kernel size, batch normalization and ReLU activation. These intermediate feature maps are then added element-wise; which then undergoes a layer of ReLU, convoutional layer with (1,1) kernel size and sigmoid activation. After resampling, the initial feature maps from segmenter are multiplied element wise with resampled attention weights. The output is the weighted segmenter map.

The architectural details of attention module is as follows:

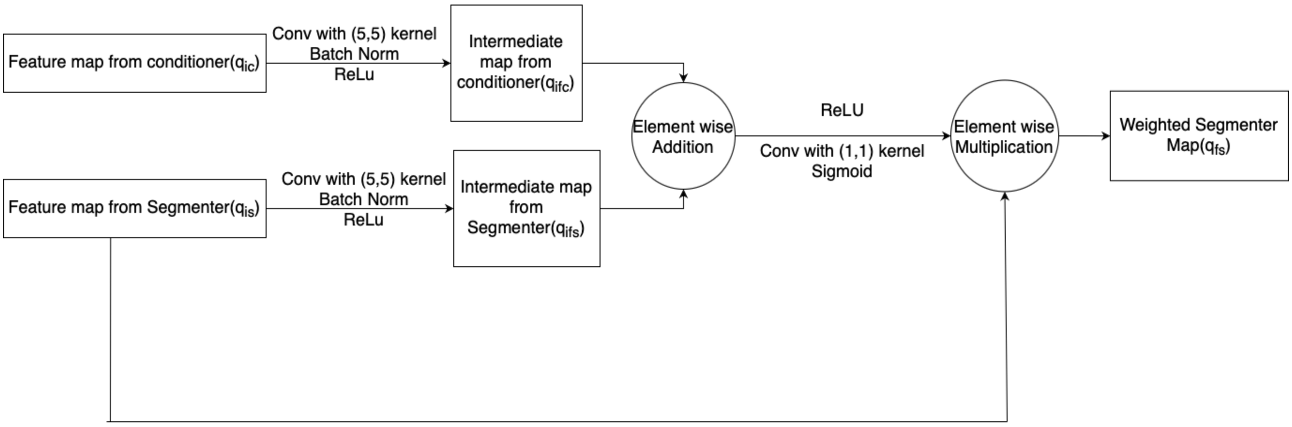
Feature map from conditioner(q_{ic})
Conv with (5,5) kernel
Batch Norm
ReLu
Intermediate map from conditioner(q_{ifc})
Feature map from Segmenter(q_{is})
Conv with (5,5) kernel
Batch Norm
ReLu
Intermediate map from Segmenter(q_{ifs})
Element wise Addition
ReLU
Conv with (1,1) kernel
Sigmoid
Element wise Multiplication
Weighted Segmenter Map(q_{fs})

Figure 2.7: Attention Module used as Interaction Block

In equational form,

$$q_{fs} = q_{is} \otimes \alpha \tag{2.1}$$

where q_{fs} is the final weighted segmenter feature map, q_{is} is the initial segmenter feature map, α are the attention coefficients and $\otimes$ is element wise multiplication operation.

$$\alpha = \sigma(W_{conv1}^{T} ReLU(q_{add})) \tag{2.2}$$

where q_{add} is the element wise additive output of intermediate segmenter feature map and intermediate conditioner feature map, W_{conv1} is the weight matrix of the convolution layer with (1,1) kernel size and σ is the sigmoid activation function.

$$q_{add} = q_{ifc} \oplus q_{ifs} \tag{2.3}$$

where q_{ifc} is the intermediate feature map of conditioner, q_{ifs} is the intermediate feature map of segmenter and $\oplus$ is element wise addition operation.

$$\begin{aligned} q_{ifc} &= ReLU(W^T_{conv51} q_{ic}) \\ q_{ifs} &= ReLU(W^T_{conv52} q_{is}) \end{aligned} \tag{2.4}$$

where q_{ic} is the input feature map of conditioner, q_{is} is the input feature map of segmenter and W_{conv51}, W_{conv52} are weight matrices of convolutional layers with (5,5) kernel size.

2.3 Contributions

Most of the model architectures existing in biomedical image segmentation are variations of the standard U-Net architecture with the number of filters getting doubled at each level in the encoder path and vice versa in the decoder path. Moreover, skip connections are also used in almost all of these models. The skip connections undoubtedly perform better in the standard image segmentation with image and mask as the input and output of these models.

1. Here, as shown in Fig. 1.2, we parallelly make use of the support set and query input while training. So, to cater this, we need two input channels which is the reason why we need two encoder-decoder arms. We refrain from incorporating skip connections within the conditioner and the segmenter part as if we use skip connections here in either or both of the arms, this will aid in effective information flow only within the same arm but will not transfer effective information from one arm to another. Since the segmenter arm makes the final prediction for the query input, so relevant information has to flow from the conditioner to the segmenter side. We can say that the FSCS-cov's segmenter arm makes predictions given that its conditioner arm has also seen and processed the support set effectively in the conditioner arm and only transfers relevant and necessary information to the segmenter arm via interaction blocks. Moreover skip connections lead to increased usage of memory due to higher number of feature maps being given as input to the further convolutional operations.

2. In the interaction blocks, we use spatial convolutions and sigmoid activation which re-weight the feature maps of the segmenter based on their relevance in infection segmentation. So, we use information from the conditioner arm to help the segmenter arm predict the output mask of the query input. If we remove the conditiner arm completely, it becomes a simple encoder-decoder architecture which becomes a U-Net if we add skip connections and double the no of filter at each level.

3. Finally, by incorporating attention gates in the interaction block, then the results are slightly better since the feature maps from the conditioner are used to re-weight the feature maps from the segmenter, which is used for the segmentation of the query input.

So, usually for biomedical image segmentation tasks, U-Net and its variations perform much better. However, in the few shot learning domain, where we need to generalize from very limited training samples and need to use the support and query sets parallelly while training, FSCS-cov fits much better than U-Net and its variations. This is how we came up with architecture and later experimented with information flow via interaction blocks. Spatial convolution seems good since it not only keeps the computational complexity low but also does the job of relevant information transfer from conditioner to segmenter in an efficient way.

Chapter 3

Experimental Results

3.1 Dataset

The COVID-19 CT segmentation dataset can be downloaded from here [1]. It consists of 100 axial CT images collected from more than 40 patients. These images were segmented by a radiologist using 3 labels: ground-glass(mask value=1), consolidation(mask value=2) and pleural effusion(mask value=3). All the 100 samples contain Class 1(Ground glass opacity), class 2(consolidation) is present in 75 of the samples and class 3 (Pleural effusion) is present in only 25 of the samples. Ground glass opacity(GGO) is a radiological term which is used to indicate hazy areas with increased lung opacity through which lung vessels and bronchial structures are still observable. However, in consolidation, these structures are obscured. It occurs when the air that fills the lung airways is replaced with a substance. Pleural effusion is due to the build up of excess fluid between the pleural layers outside the lungs. Generally the symptoms of Covid start with Ground glass opacity which slowly progresses to consolidation and if left untreated can lead to Pleural effusion.

3.2 Data pre-processing

We remove those samples which have a very low infection content,ie- those samples where the number of pixels corresponding to infection occupy a small fraction of the

[1] COVID-19 CT segmentation dataset: `http://medicalsegmentation.com/covid19/`

entire mask are removed. The threshold is chosen at 0.5 percent. Normalization is then performed on the CT slices to scale down the intensities in the range (0,1).

3.3 Training Strategy

We conduct 5-fold cross validation to come up with robust results. To achieve this, we conduct 2 different strategies:

3.3.1 Strategy 1 (3 class approach)

The original dataset contains samples which can contain all the classes,ie- a sample can contain ground glass opacity, consolidation as well as pleural effusion. We observe that there are 25 samples containing all ground glass opacity, consolidation and pleural effusion; 55 samples containing ground glass opacity and consolidation and only 20 samples containing only ground glass opacity. We denote all those samples which contain all ground glass opacity, consolidation and pleural effusion by class A samples. Similarly all those samples which contain ground glass opacity and consolidation by class B samples and the rest which contain only ground glass opacity by class C samples. Now these samples are mutually exclusive.

After removal of outlier samples, we prepare 5 folders, each folder having the same proportion of all the 3 classes(A,B,C) as in the dataset.After we have prepared these 5 folders, we perform 5-fold cross validation with 20 percent in test set, 20 percent in validation and remaining 60 percent in training sets.

3.3.1.1 Class wise sample distribution

Class label	Presence of radiological structures	No of samples
Class A	Ground glass + consolidation + pleural effusion	25
Class B	Ground glass + consolidation	55
Class C	Ground glass	20

3.3.2 Strategy 2 (2 class approach)

Due to class imbalance caused by extremely low proportion of pleural effusion, we follow this approach. Pleural effusion is present in only 25 samples, so we combine those pixels which indicate pleural effusion with those which indicate consolidation. We do this because pleural effusion is generally present in those pixels which are adjacent to consolidation.

Now, we have 80 samples containing both ground glass opacity and consolidation; 20 samples containing only ground glass opacity. We denote these 80 samples which contain both by class A samples and the rest 20 which contain only ground glass as class B samples. Now these samples are mutually exclusive.

We perform 5 fold cross validation in a similar way as we did in Strategy 1.

3.3.2.1 Class wise sample distribution

Class label	Presence of radiological structures	No of samples
Class A	Ground glass + consolidation	80
Class B	Ground glass	20

3.4 Training Configuration

FSCS-cov is trained on Tesla V100 16GB with optimizer as SGD(Stochastic Gradient Descent)[15], initial learning rate set to 0.01 which undergoes exponential decay with increasing number of epochs and the model is trained for 150 epochs. We also conducted runs with Adam optimizer[8], but we saw that the results were more consistent and less fluctuating if we use SGD.

3.5 Loss functions

3.5.1 Dice Loss

Dice Loss as a loss function was first developed by Sudre et al.[18]. It comes from Sørensen–Dice coefficient, which is a measure to check the similarity between two samples. Dice coefficient is defined as:

$$DSC = \frac{2|G \cap S|}{|G| + |S|} \tag{3.1}$$

where G denotes the ground truth and S denotes the predicted set of pixels. It is a measure of overlap between 2 sets. It ranges from (0,1). A dice coefficient of 0 means that there is no overlap between the ground truth and the predicted mask. Similarly a dice coefficient of 1 means complete overlap between the ground truth and predicted mask. Alternatively, the dice coefficient can also be interpreted as:

$$DSC = \frac{2 * \text{TP}}{2 * \text{TP} + \text{FP} + \text{FN}} \tag{3.2}$$

Now, the dice loss is defined as follows:

$$Loss_{Dice} = 1 - \frac{2|G \cap S|}{|G| + |S|} \tag{3.3}$$

where G denotes the ground truth and S denotes the predicted set of pixels. We see that as the dice coefficient increases, the dice loss decreases and vice-versa.

3.5.2 IoU Loss

IoU loss was developed by Zhou et al.[22]. It comes from IoU coefficient, which is a term used to find the amount of overlap of 2 boxes The greater the area of overlap, the greater is the IoU. Simply put, IoU is defined as:

$$IoU = \frac{\text{Area of overlap}}{\text{Area of union}} \tag{3.4}$$

It is formulated as:

$$IoU = \frac{|G \cap S|}{|G \cup S|} \tag{3.5}$$

where G denotes the ground truth and S denotes the predicted mask. It ranges from (0,1). An IoU score of 1 means complete overlap between the ground truth and the predicted mask. Similarly an IoU score of 0 means no overlap. Now, the IoU loss is defined as follows:

$$Loss_{IoU} = 1 - \frac{|G \cap S|}{|G \cup S|} \tag{3.6}$$

where G denotes the ground truth and S denotes the predicted set of pixels. We see that as the IoU coefficient increases, the IoU loss decreases and vice-versa. Alternatively, the IoU can also be interpreted as:

$$IoU = \frac{\text{TP}}{\text{TP + FP + FN}} \tag{3.7}$$

We have also experimented with other loss functions such as Binary Cross Entropy loss by Zhang et al.[21], Focal Loss by Lin et al.[10] and Lovasz Hinge Loss by Berman et al. [1]. However the results were not good, possibly because these loss functions did not converge and also did not correlate with IoU, ie- decrease in Focal loss or Lovasz Hinge Loss had no effect on IoU.

3.6 Evaluation metrics

1. **Dice Similarity Coefficient (DSC):** It is also known as Dice-Sørensen coefficient [2] . It is almost widely used in segmentation. It is a similarity measure function and is used to calculate the similarity of two samples. It is also the harmonic mean of Precision and Recall. It is formulated in Eq. 3.1.

2. **Intersection Over Union (IoU)**: It is also known as Jaccard index[3]. It is a term used to find the amount of overlap of 2 boxes. The greater the area of overlap, the greater is the IoU. It is formulated in Eq. 3.5.

3. **Hausdorff Distance (HD)**: It is also called Pompeiu–Hausdorff distance[4]. It is used to describe the similarity between segmentation result and the ground truth. It is defined as follows:

$$HD = \max\{\max_{x \in G} \min_{y \in S} d(x,y), \max_{y \in S} \min_{x \in G} d(x,y)\} \tag{3.8}$$

where G denotes the ground truth and S denotes the predicted mask. The 95^{th} percentile is taken to avoid the effect of outliers. It measures how far two sets are from each other. It is zero iff both G and S are the same; otherwise it has a finite value. The less the Hausdorff distance, the more similar are the ground truth to its predicted mask.

4. **Mean Absolute Error (MAE)**: It is the average of absolute errors, and it is defined as:

$$MAE = \frac{1}{W * H} \sum_{x=1}^{W} \sum_{y=1}^{H} |S(x,y) - G(x,y)| \tag{3.9}$$

where G denotes the ground truth and S denotes the predicted mask. It is always positive; the lesser the MAE, the greater is the similarity between the ground truth and the predicted mask. A larger value indicates that ground truth is far different from the predicted mask.

[2] https://en.wikipedia.org/wiki/S\OT1\orensen\OT1\textendashDice_coefficient

[3] https://en.wikipedia.org/wiki/Jaccard_index

[4] https://en.wikipedia.org/wiki/Hausdorff_distance

5. **Sensitivity**: It is formulated as:

$$Sensitivity = \frac{\text{True positives}}{\text{True Positives + False Negatives}} \tag{3.10}$$

It is the proportion of true positives that are correctly predicted by the model. A model with high sensitivity will have few false negatives, ie- the ability of a model to correctly identify positive examples. Its value ranges from (0,1). The sum of sensitivity(True Positive rate) and False negative rate is 1. In unbalanced datasets, the more pixels we predict as the true class, the better is the sensitivity.

6. **Specificity**: It is formulated as:

$$Specificity = \frac{\text{True Negatives}}{\text{True Negatives + False Positives}} \tag{3.11}$$

Its value ranges from (0,1). A model with high specificity will accurately identify the majority of the negative outcomes, but one with low specificity may mistakenly classify many negative results as positive. It is also known as True Negative Rate (TNR).

7. **Precision**: It is formulated as:

$$Precision = \frac{\text{True Positives}}{\text{True Positives + False Positives}} \tag{3.12}$$

Its value ranges from (0,1). It checks the number of positive predictions made by the model. Higher precision means the model is good in detecting positive outcomes.

3.7 Experiment 1 - Few shot with all 3 classes

3.7.1 Qualitative Analysis

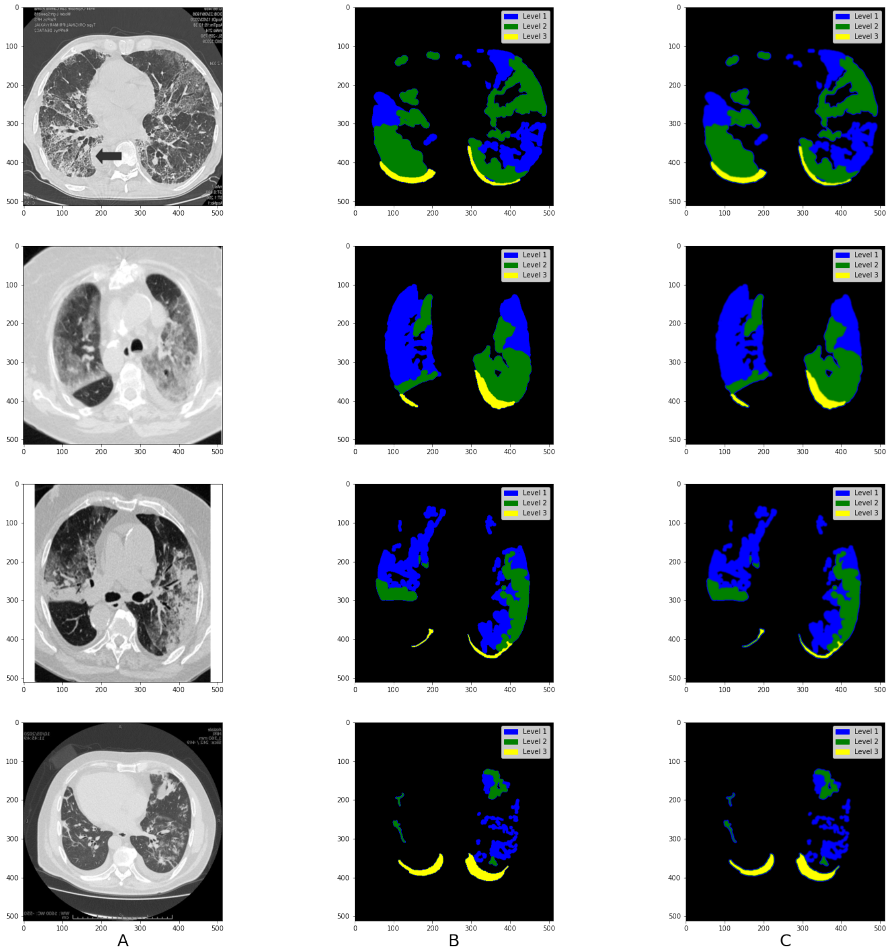

Figure 3.1: (A) sample CT slice , (B) ground truth, and (C) segmentation by proposed FSCS-cov

3.7.1.1 Tabulation - Class Wise pixel count in ground truth vs predicted masks

Class label	Avg. pixel Percentage in Ground truth	Avg. pixel Percentage in Predicted mask
Class 1	6.96	7.95
Class 2	8.08	7.93
Class 3	1.55	1.338
Background	83.39	82.776

3.7.2 Quantitative Analysis

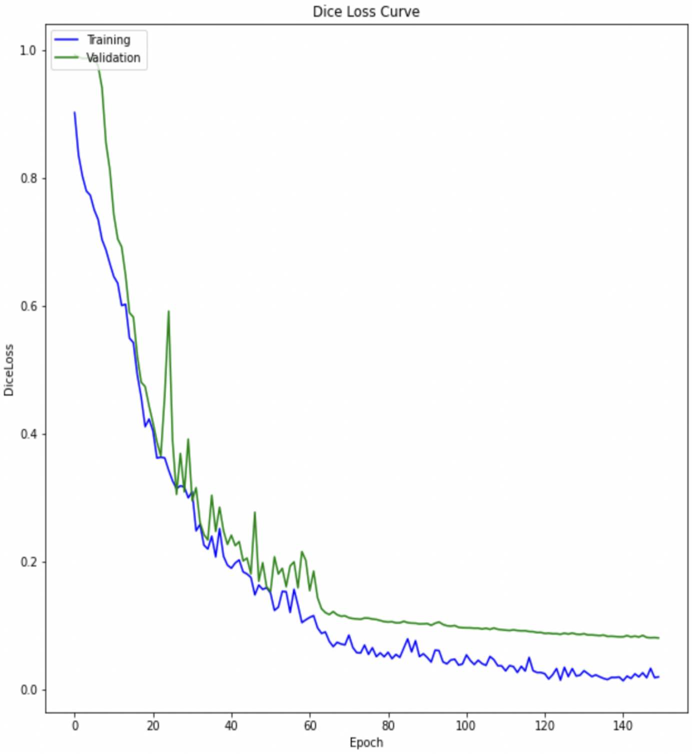

Figure 3.2: Training vs Validation Curve

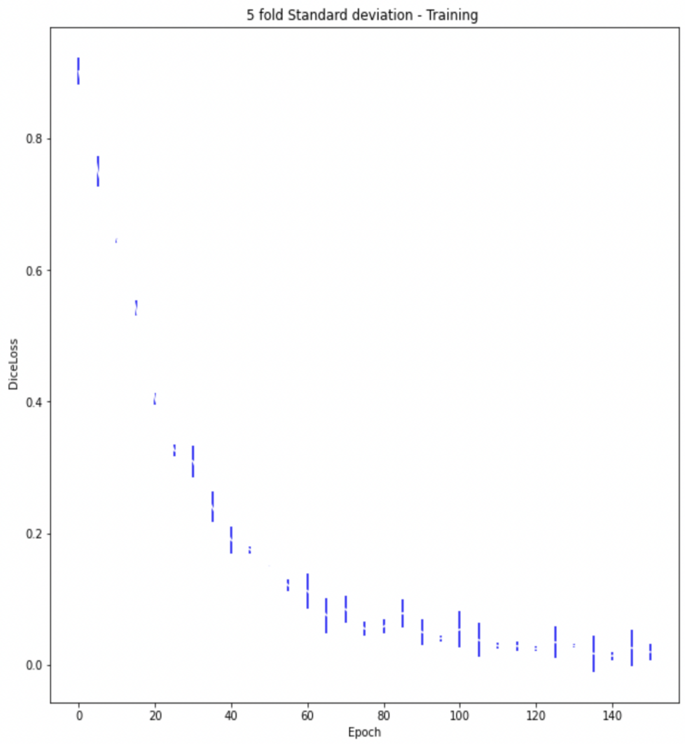

Figure 3.3: 5 fold Standard deviation - Training

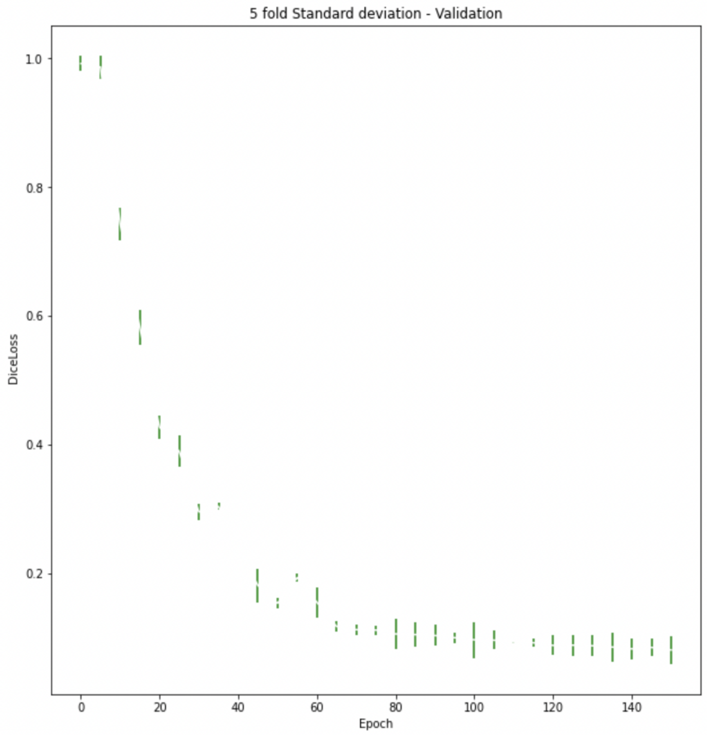

Figure 3.4: 5 fold Standard deviation - Validation

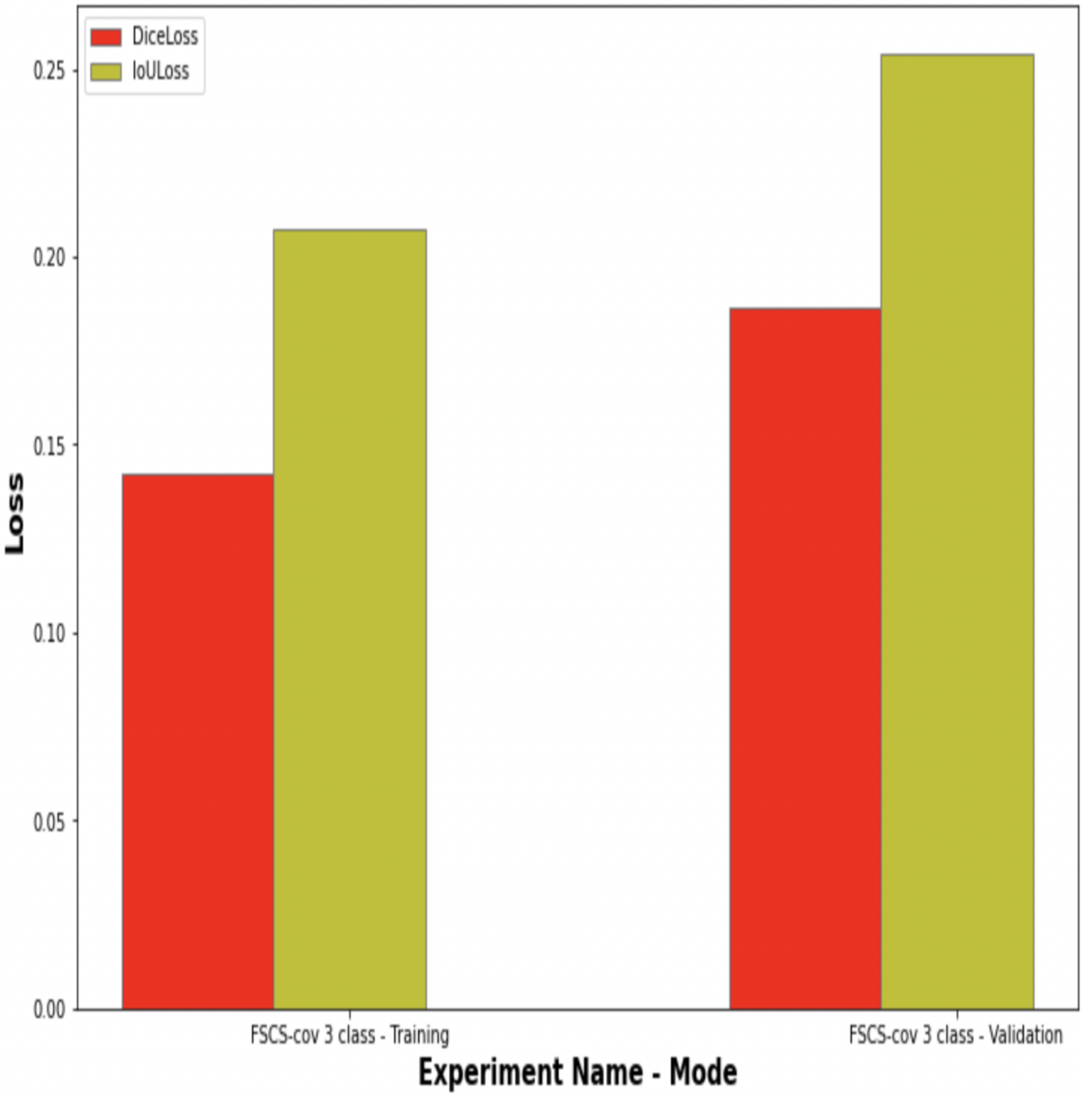

Figure 3.5: Bar Plot - Dice Loss vs IoU loss in Training and Validation

We see that the model's validation performance is better if we use Dice Loss instead of IoU loss. In both training and validation, dice loss is lower in both training and validation cases if we use the same no of epochs.

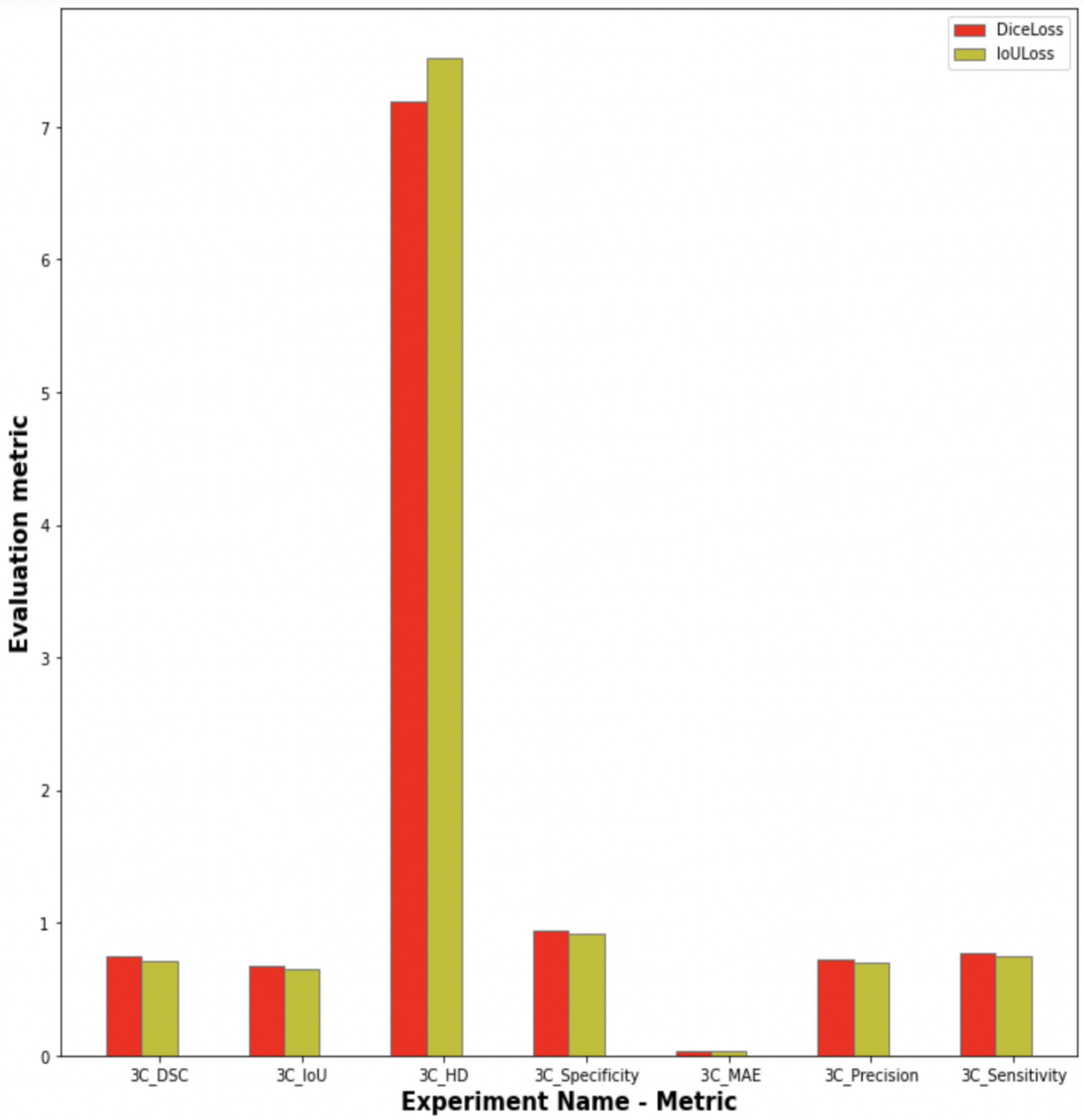

Figure 3.6: Evaluation metrics in test data - DiceLoss vs IoULoss

From the above plot, we see that the model performs better if we use Dice Loss. This is because dice loss is differentiable where as IoU loss is not differentiable since the chain rule of this loss function breaks.

3.8 Experiment 2 - Few shot with 2 classes

3.8.1 Qualitative Analysis

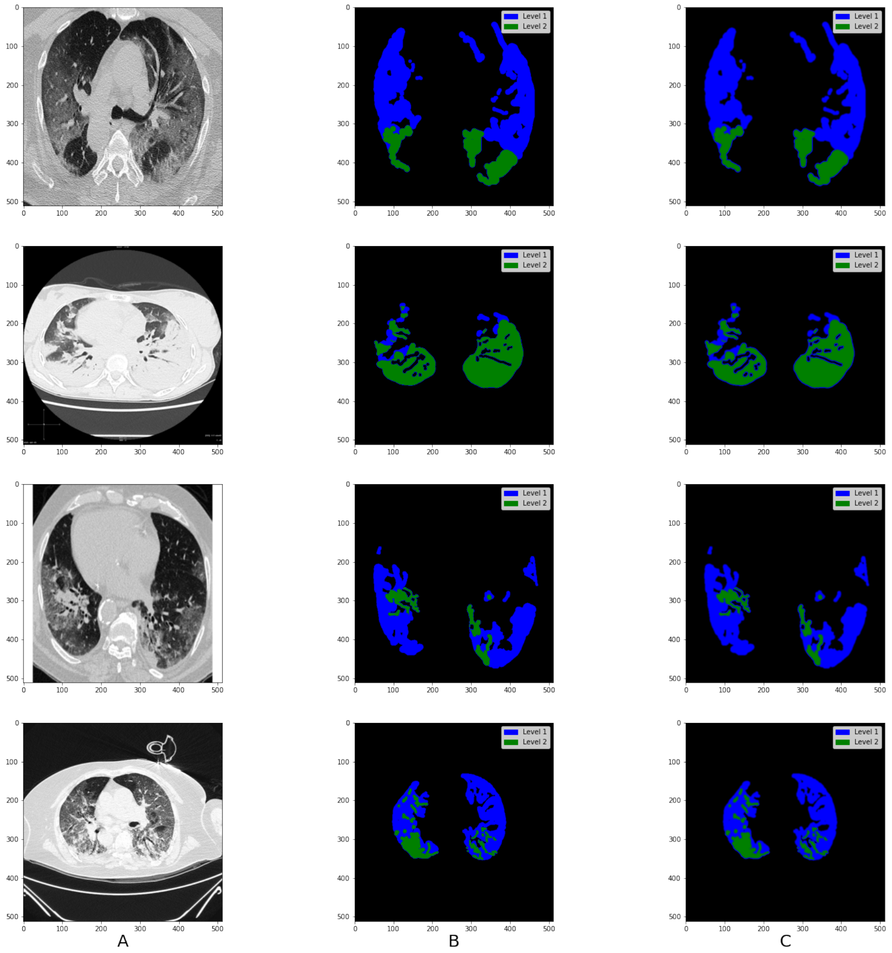

Figure 3.7: (A) sample CT slice , (B) ground truth, and (C) segmentation by proposed FSCS-cov

3.8.1.1 Tabulation - Class Wise pixel count in ground truth vs predicted masks

Class label	Avg. pixel Percentage in Ground truth	Avg. pixel Percentage in Predicted mask
Class 1	9.1	9.95
Class 2	5.79	5.38
Background	85.1	84.66

Here we observe that if we remove the third class, results are better since the percentage of pixels covered by the third class in Ground Truth is extremely less when compared to the other two classes.

3.8.2 Quantitative Analysis

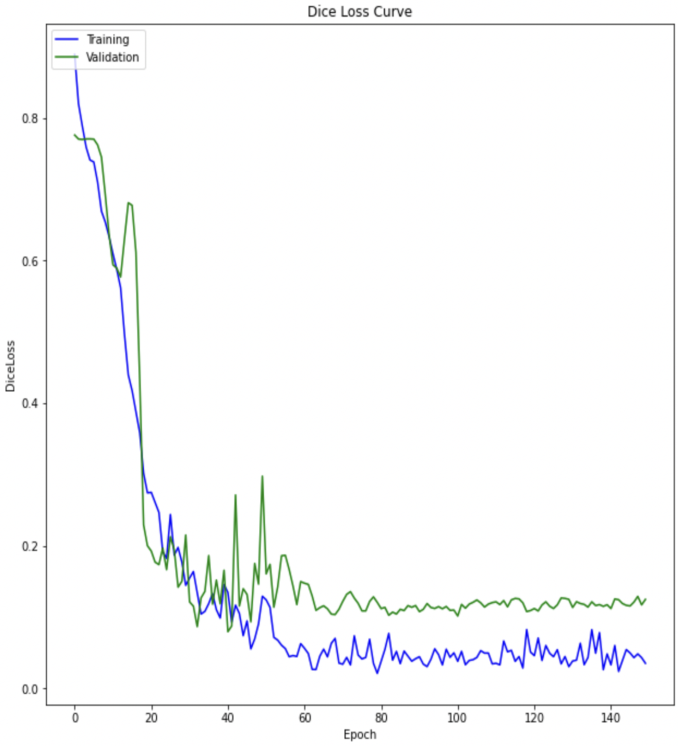

Figure 3.8: Training vs Validation Curve

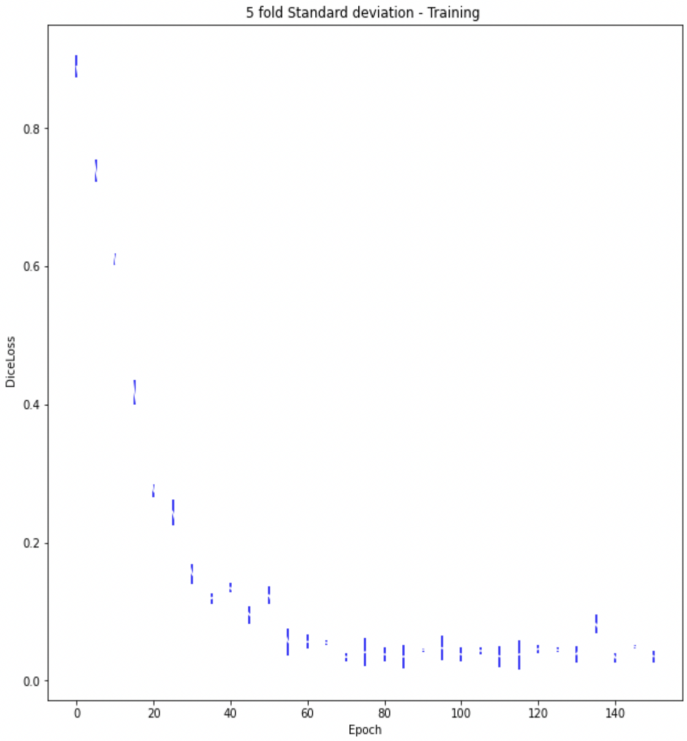

Figure 3.9: 5 fold Standard deviation - Training

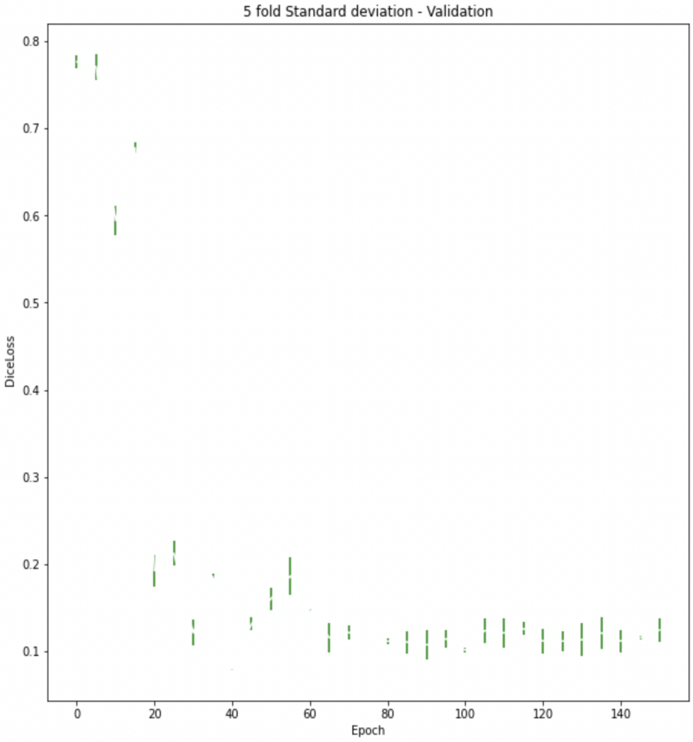

Figure 3.10: 5 fold Standard deviation - Validation

3.8.3 Tabulation - Training vs validation loss

Experiment name - Mode	Mean Dice Loss after 80 epochs ± Standard deviation (σ)	Mean IoU Loss after 80 epochs ± Standard deviation (σ)
FSCS-cov 3 class - Training	0.1422 ± 0.02456	0.2076 ± 0.02843
FSCS-cov 3 class - Validation	0.1865 ± 0.02622	0.2543 ± 0.03082
FSCS-cov 2 class - Training	0.1537 ± 0.01924	0.1872 ± 0.02617
FSCS-cov 2 class - Validation	0.1743 ± 0.02132	0.2217 ± 0.02375

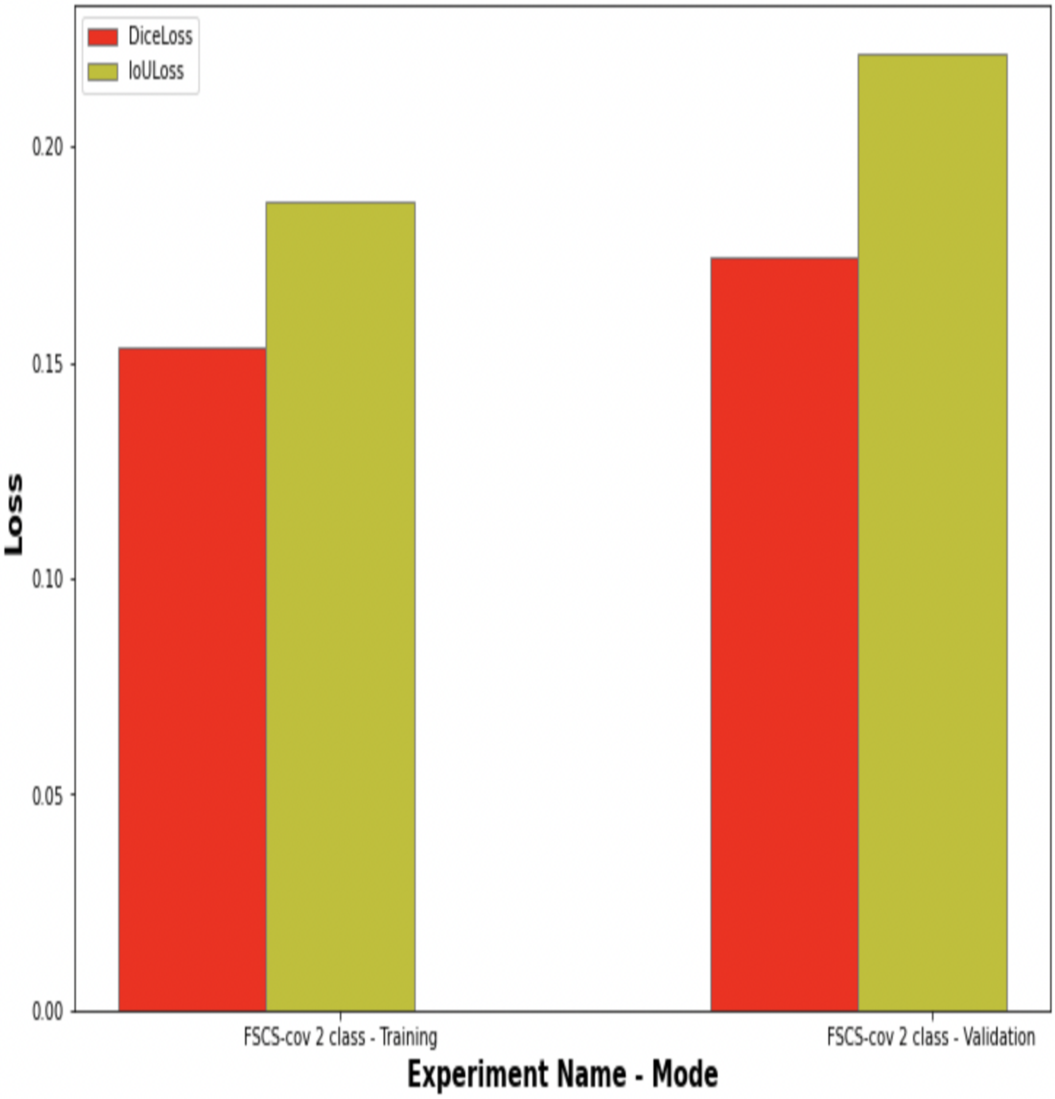

Figure 3.11: Bar Plot - Dice Loss vs IoU loss in Training and Validation

We see that the model's validation performance is better if we use Dice Loss instead of IoU loss. In both training and validation, dice loss is lower in both training and validation cases if we use the same no of epochs.

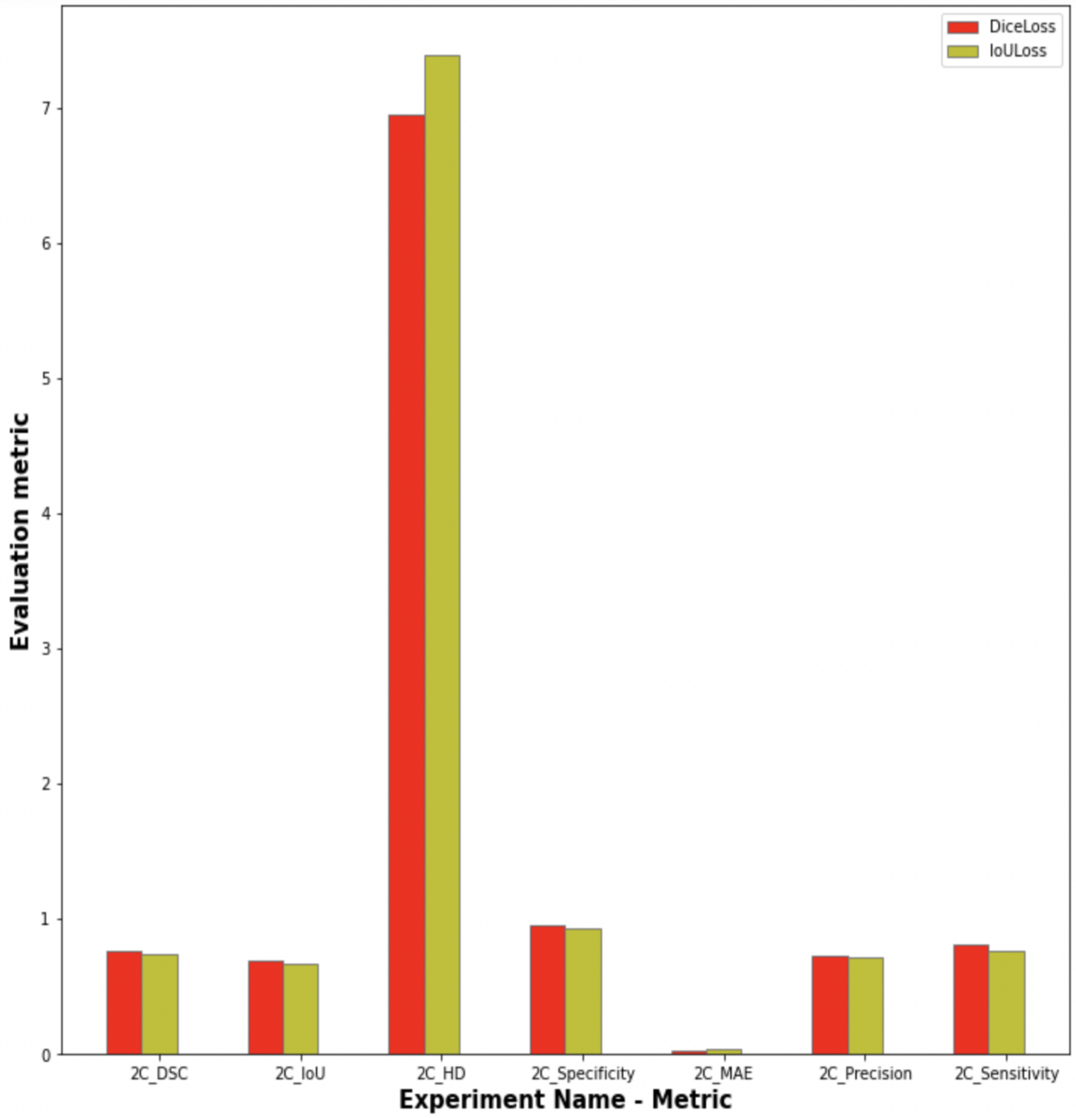

Figure 3.12: Evaluation metrics in test data - DiceLoss vs IoULoss

From the above plot, we see that the model performs better if we use Dice Loss. This is because dice loss is differentiable where as IoU loss is not differentiable since the chain rule of this loss function breaks.

3.9 Tabulation of Results

Model Name	No of Training samples	Mean Dice Score	Mean IoU	Mean HD_{95}	Mean specificity	MAE	Mean Precision	Mean sensitivity
U-Net [14]	480	0.6723	0.6112	8.2343	0.8471	0.1142	0.6421	0.7033
Attention U-Net [11]	480	0.6438	0.5835	10.3465	0.8093	0.1356	0.6303	0.6522
Inf-Net [2]	480	0.7236	0.6834	7.0808	0.9143	0.0311	0.6923	0.7532
nnU-net [6]	480	0.7500	0.6526	7.1841	0.9356	0.0316	0.7471	0.7584
SSA-Net [20]	300	0.7540	0.6698	7.0464	0.9412	0.0305	0.7403	0.7625
FSCS-cov 3 Class(ours)	60	0.7462	0.6732	7.1957	0.9482	0.0295	0.7208	0.7734
FSCS-cov 2 class(ours)	60	**0.7625**	**0.6903**	**6.9482**	**0.9513**	**0.0257**	**0.7266**	**0.8021**

Data augmentation has been performed for the first five models. Here we observe that FSCS-cov's performance is much better in limited training data when compared to the other 5 models. Although SSA-Net performs better than FSCS-cov, but it also uses much more data when compared with FSCS-cov. One of the possible reasons might be that all the other five models follow the U-Net type architecture with increasing no of filters as the depth increases in the encoder and vice versa in the decoder, due to which the model complexity is highly increased. However, in FSCS-cov, the complexity is relativdly much less since each encoder-decoder has 16 filters in the Conditioner arm and 64 filters in the Segmenter arm. Since the model complexity is low, so the generalization capability of FSCS-cov is better compared to other models.

3.10 Ablation Studies

3.10.1 Reducing the no of blocks in both encoder and decoder from 4 to 3

3.10.1.1 Qualitative Analysis

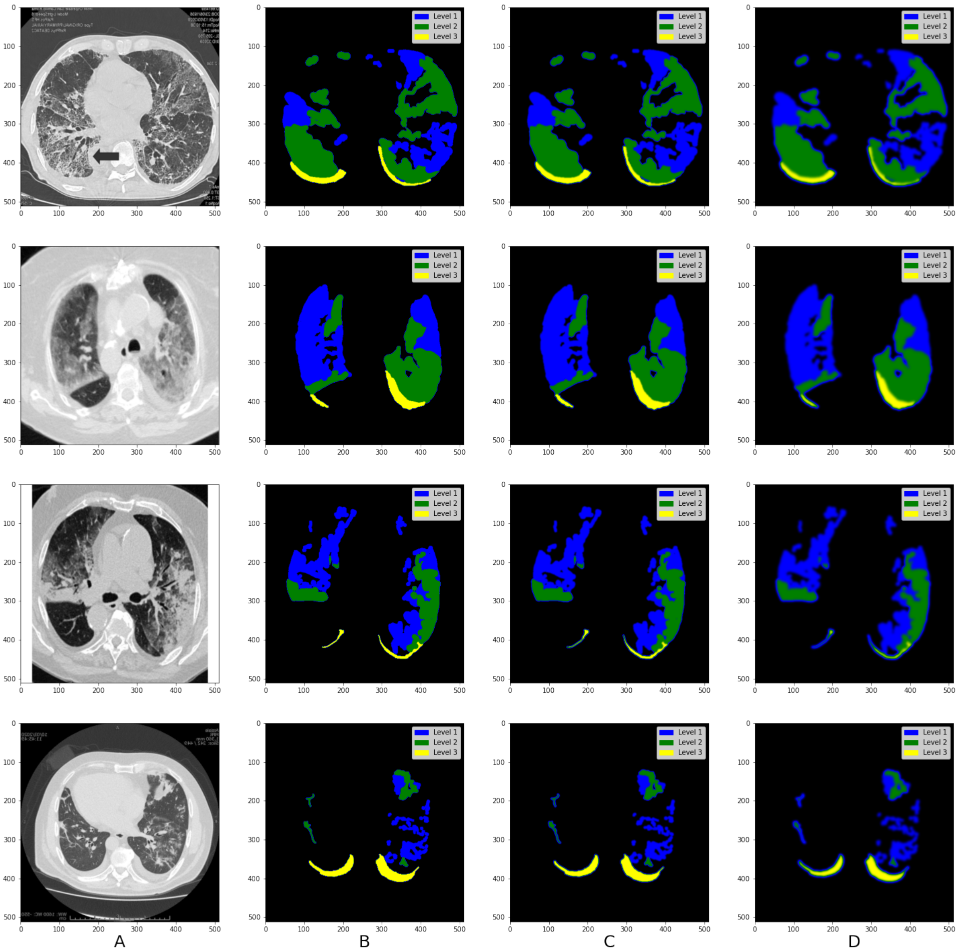

Figure 3.13: (A) sample CT slice , (B) ground truth, (C) segmentation by FSCS-cov and (D) segmentation by reducing encoder and decoder blocks in FSCS-cov

3.10.1.2 Tabulation - Class Wise pixel count in ground truth vs predicted masks

Class label	Avg. pixel Percentage in Ground truth	Avg. pixel Percentage in Predicted mask - FSCS-cov	Avg. pixel Percentage in Predicted mask - reducing encoder and decoder blocks in FSCS-cov
Class 1	6.96	7.95	9.2
Class 2	8.08	7.93	7.76
Class 3	1.55	1.338	1.01
Background	83.39	82.776	82.01

3.10.1.3 Quantitative Analysis

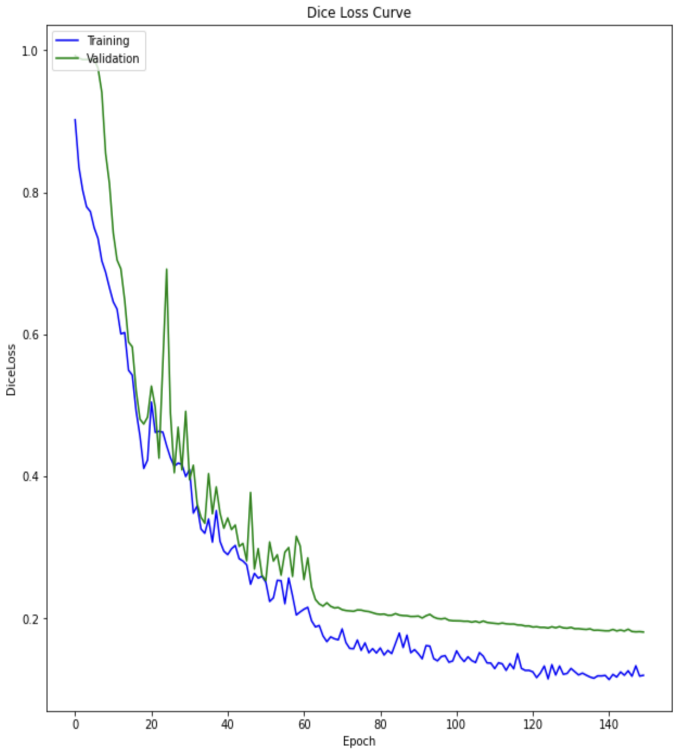

Figure 3.14: Training vs Validation Curve

If we decrease the no of blocks, FSCS-cov takes more time to fit. This happens because if we reduce the no of encoder-decoder blocks, the model has lesser complexity with respect to no of trainable parameters. So, the model takes time to fit the training data. Also, since the model is simplified with respect to computational complexity, so the performance on unseen data deteriorates.

3.10.2 Increasing the no of blocks in both encoder and decoder from 4 to 5

3.10.2.1 Qualitative Analysis

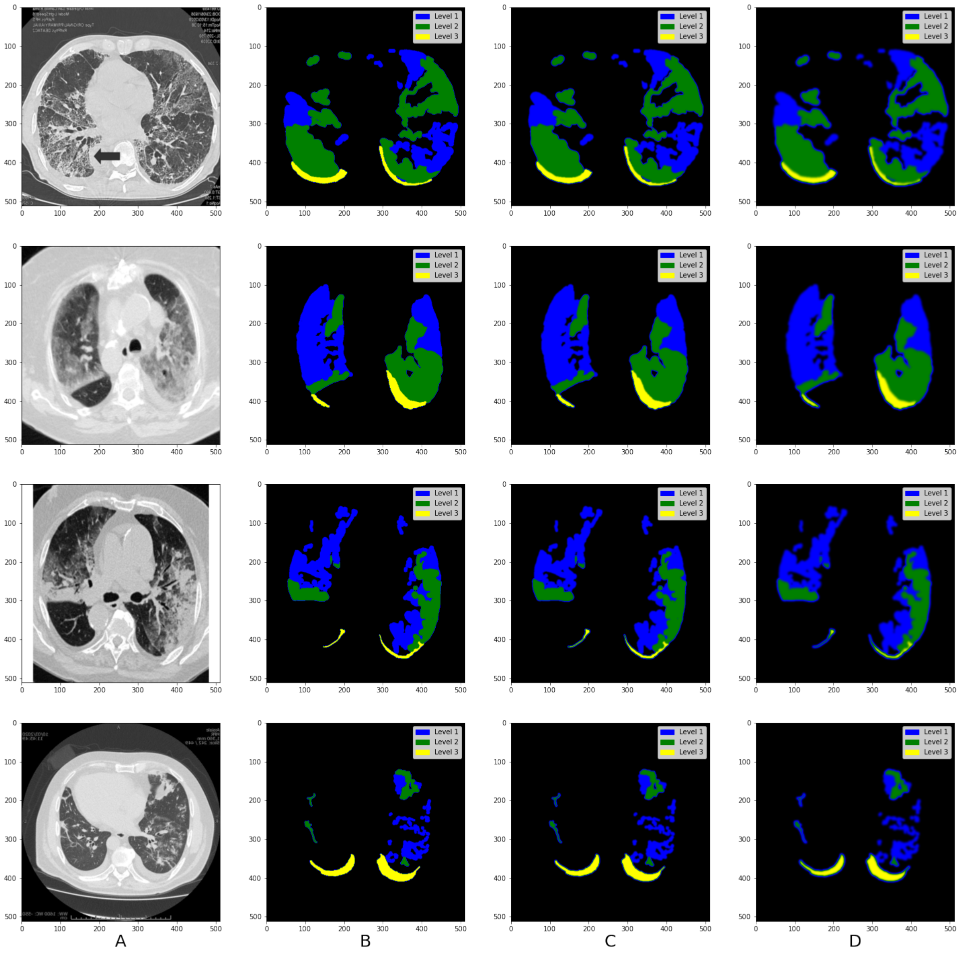

Figure 3.15: (A) sample CT slice , (B) ground truth, (C) segmentation by FSCS-cov and (D) segmentation by increasing encoder and decoder blocks in FSCS-cov

3.10.2.2 Tabulation - Class Wise pixel count in ground truth vs predicted masks

Class label	Avg. pixel Percentage in Ground truth	Avg. pixel Percentage in Predicted mask - FSCS-cov	Avg. pixel Percentage in Predicted mask - increasing encoder and decoder blocks in FSCS-cov
Class 1	6.96	7.95	8.84
Class 2	8.08	7.93	7.8
Class 3	1.55	1.338	1.11
Background	83.39	82.776	82.24

3.10.2.3 Quantitative Analysis

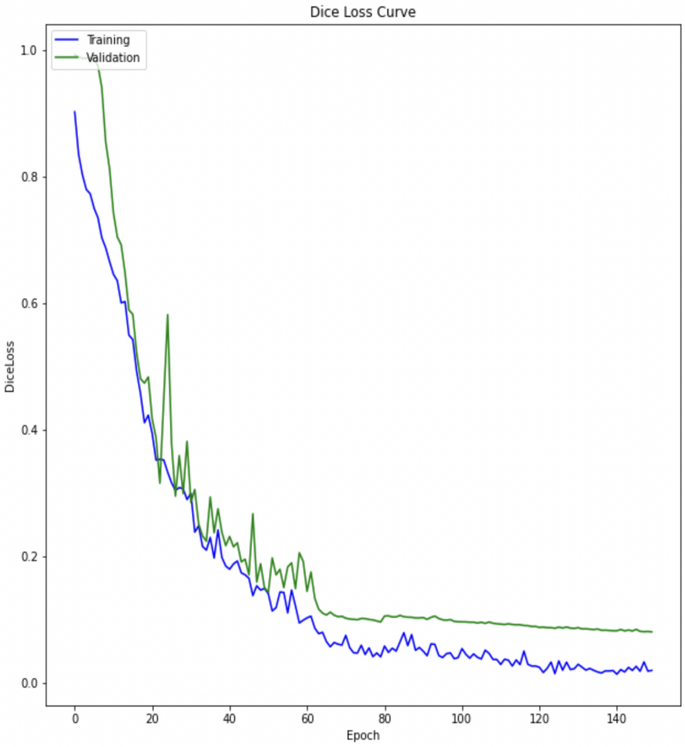

Figure 3.16: Training vs Validation Curve

Here FSCS-cov fits quickly and overfits at 100 epochs. If we increase the no of blocks, the model complexity also increases, so it takes less time to fit the training data and overfits later. So, its performance on training data is good but suffers in test data.

3.10.3 Introducing Attention blocks in interaction module

As shown in Fig. 3.12, we use this attention module in the interaction blocks. It is the same attention module used in Attention U-Net. Architectural details have been discussed in section 2.2.4.1.

3.10.3.1 Qualitative Analysis

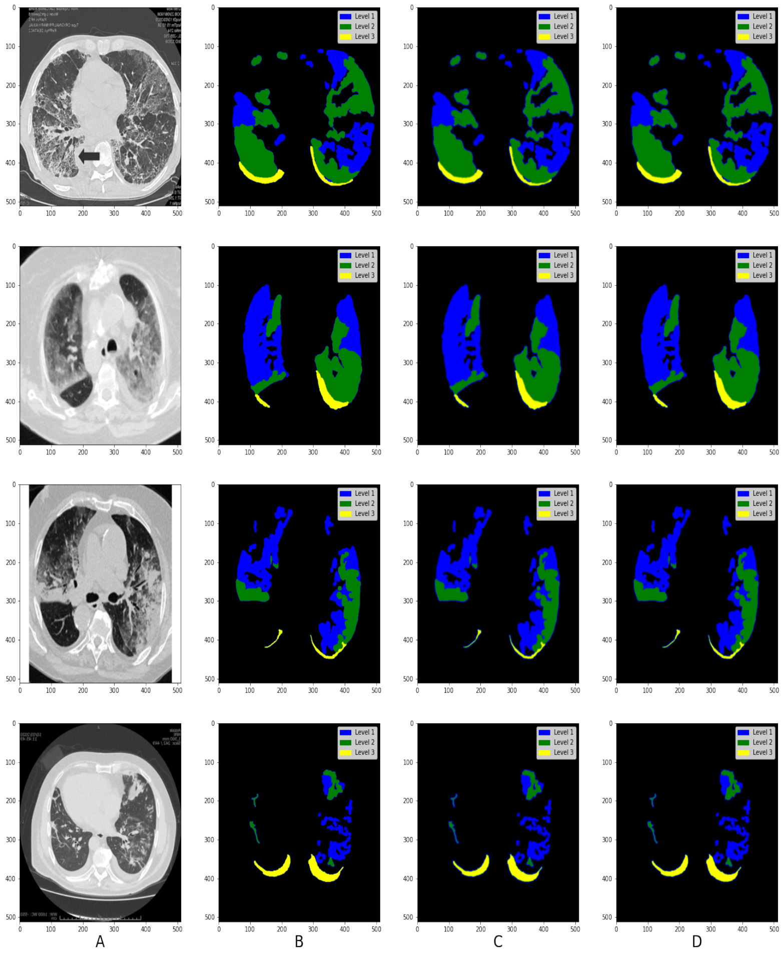

Figure 3.17: (A) sample CT slice , (B) ground truth, (C) segmentation by FSCS-cov and (D) segmentation by introducing attention blocks in interaction module of FSCS-cov

3.10.3.2 Tabulation - Class Wise pixel count in ground truth vs predicted masks

Class label	Avg. pixel Percentage in Ground truth	Avg. pixel Percentage in Predicted mask - FSCS-cov	Avg. pixel Percentage in Predicted mask - introducing attention blocks in interaction module of FSCS-cov
Class 1	6.96	7.95	7.77
Class 2	8.08	7.93	7.951
Class 3	1.55	1.338	1.38
Background	83.39	82.776	82.9

3.10.3.3 Quantitative Analysis

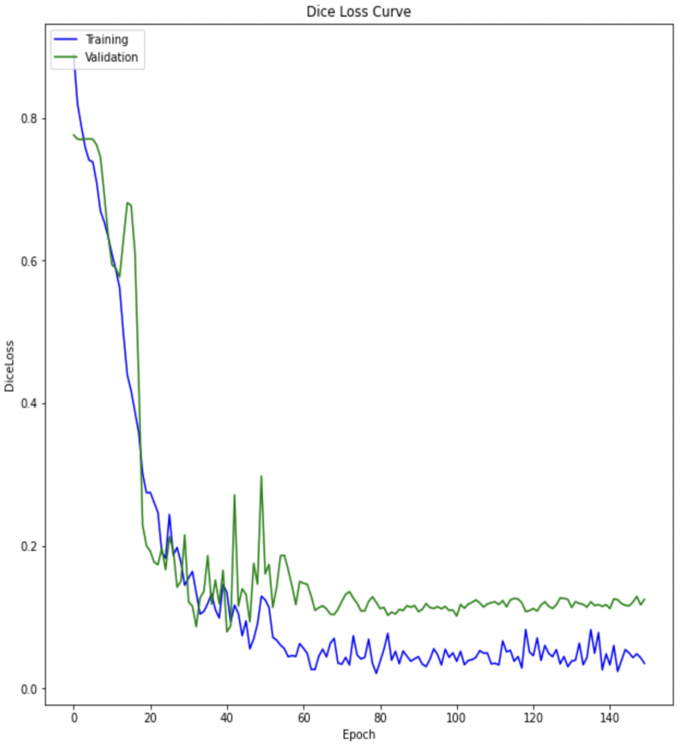

Figure 3.18: Training vs Validation Curve

By using attention blocks, we observe that the results get slightly better. This happens because the purpose of the interaction block is to highlight the feature maps of the segmenter arm based on the feedback received from the segmentation maps of the conditioner arm. The use of attention is to give more weightage to relevant feature maps. So, if we use attention gates directly in the interaction blocks itself, the task of re-weighting feature maps of the segmenter based on the feedback received from the conditioner is done better when compared with spatial convolution.

3.10.4 Increasing the number of training samples

No of samples in training set is increased to 70 percent and in the validation set is reduced to 10 percent. The test set contains 20 percent samples as before.

3.10.4.1 Qualitative Analysis

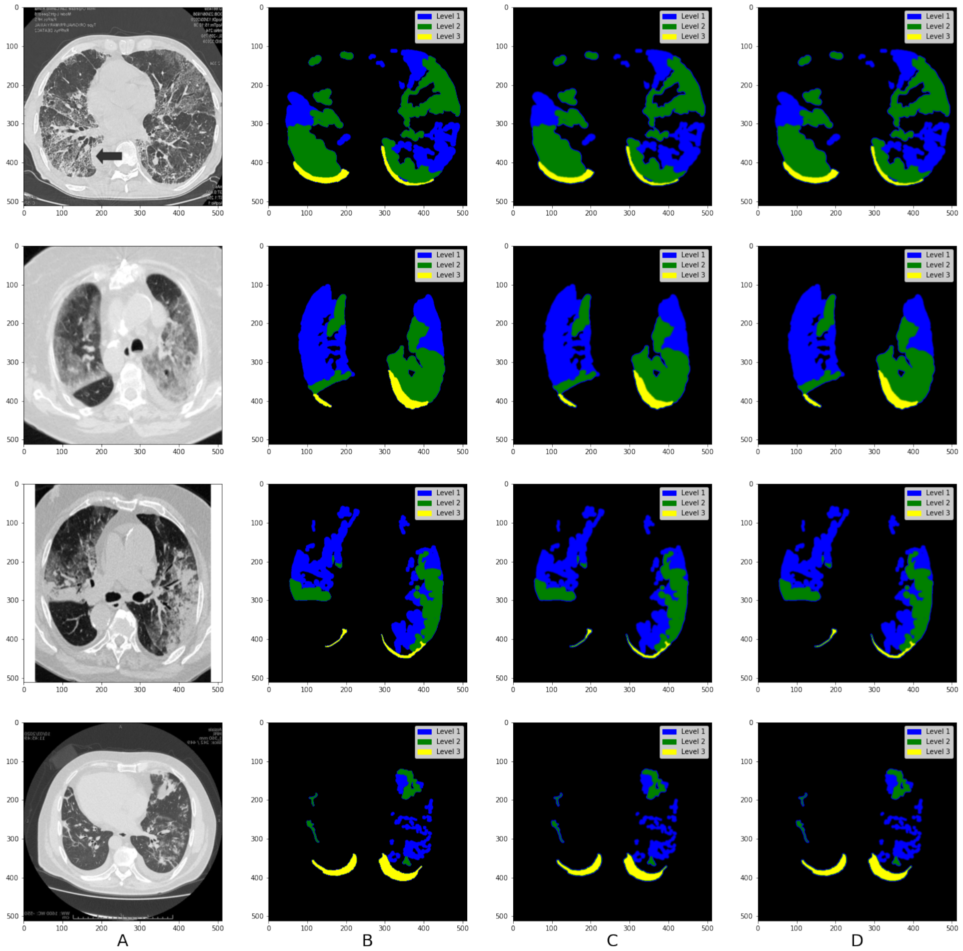

Figure 3.19: (A) sample CT slice , (B) ground truth, (C) segmentation by FSCS-cov and (D) segmentation by increasing training samples

3.10.4.2 Tabulation - Class Wise pixel count in ground truth vs predicted masks

Class label	Avg. pixel Percentage in Ground truth	Avg. pixel Percentage in Predicted mask - FSCS-cov	Avg. pixel Percentage in Predicted mask - increasing training samples
Class 1	6.96	7.95	7.83
Class 2	8.08	7.93	7.927
Class 3	1.55	1.338	1.374
Background	83.39	82.776	82.86

3.10.4.3 Quantitative study

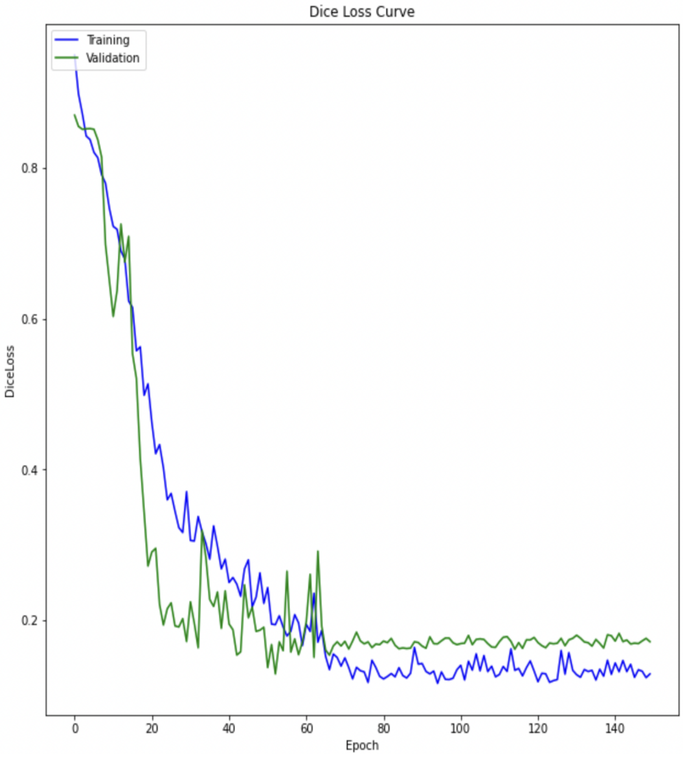

Figure 3.20: Training vs Validation Curve

Results are slightly better due to more number of training samples. This shows that if the model sees more training data, then it learns more about the data distribution. So, it generalizes better on test data.

3.10.5 Tabulation - Training vs validation loss

Experiment name - Mode	Mean Dice Loss ± Standard deviation (σ)
Reducing blocks - Training	0.1648 ± 0.02249
Reducing blocks - Validation	0.1851 ± 0.02863
Increasing blocks - Training	0.1139 ± 0.02574
Increasing blocks - Validation	0.1647 ± 0.02364
Attention blocks - Training	0.1356 ± 0.02857
Attention blocks - Validation	0.1798 ± 0.02359
Increasing Training samples - Training	0.1182 ± 0.02212
Increasing Training samples - Validation	0.1478 ± 0.02056

3.11 Tabulation of Results - Ablation studies

Ablation Experiment	Mean Dice Score	Mean IoU	Mean HD_{95}	Mean Specificity	MAE	Mean Precision	Mean Sensitivity
FSCS-cov 3 Class(ours)	0.7462	0.6732	7.1957	0.9482	0.0295	0.7208	0.7734
FSCS-cov 2 class(ours)	0.7625	0.6903	6.9482	0.9513	0.0257	0.7266	0.8021
Decreasing blocks	0.7225	0.6514	7.6784	0.9421	0.0314	0.715	0.7343
Increasing blocks	0.7355	0.6559	7.4235	0.9452	0.0312	0.7016	0.7611
Attention in interaction module	0.7654	0.6982	7.0112	0.9501	0.0255	0.7367	0.7922
Increasing training samples	0.7489	0.6812	7.1915	0.9496	0.0285	0.7211	0.7821

The results clearly show that if we either increase or decrease the no of blocks in the encoder or decoder, then the model either becomes too complex or too simplified which decreases the dice score. However, if we use attention, then only relevant information flows from Conditioner to Segmenter due to which the dice score increases. Similarly if we increase the no of training samples, the dice score increases because the model understands the underlying data distribution better.

Chapter 4

Discussion

The FSCS-cov architecture is inspired from the standard U-Net[14] with interaction blocks connecting the Conditioner and Segmenter. The spatial convolution in the interaction blocks helps transfer relevant information from the conditioner to the segmenter. We conducted two different types of experiments, one with all the three classes present and another with two classes by merging imbalanced class into the other. The results obtained from two classes are better since the infection proportion is easily detectable because of higher proportion. We also conducted different ablation studies and saw that the results improved if we used attention gates in the interaction blocks or increase the number of training samples. If we either reduce or increase the number of blocks in the encoder-decoder architecture, then the results deteriorate. Thus, using this query-support approach and this architecture, we can make predictions using very few labelled samples, which is a major drawback for U-Net and its variations.

4.1 Future Scope

Future improvements can be performed by modifying the interaction blocks which connect the two arms. Furthermore experiments can also be performed with bottleneck block since it is the medium of information flow from the encoder to decoder. Along with this, semi-supervised learning can also be applied to aid FSCS-cov come with better results in limited data domain scenarios.

Bibliography

[1] M. Berman and *et al.* “The lovász-softmax loss: A tractable surrogate for the optimization of the Intersection-over-Union measure in neural networks”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.* 2018, pp. 4413–4421.

[2] D. Fan and *et al.* “Inf-Net: Automatic COVID-19 lung infection segmentation from CT images”. In: *IEEE Transactions on Medical Imaging* 39.8 (2020), pp. 2626–2637.

[3] R. Girshick. “Fast R-CNN”. In: *Proceedings of the IEEE International Conference on Computer Vision.* 2015, pp. 1440–1448.

[4] R. Girshick and *et al.* “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.* 2014, pp. 580–587.

[5] K. He and *et al.* “Deep Residual learning for image recognition”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.* 2016, pp. 770–778.

[6] F. Isensee and *et al.* “Automated design of deep learning methods for biomedical image segmentation”. In: *arXiv preprint arXiv:1904.08128* (2019).

[7] S. Jetley and *et al.* “Learn to pay attention”. In: *arXiv preprint arXiv:1804.02391* (2018).

[8] D. Kingma and J. Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[9] G. Koch and *et al.* “Siamese neural networks for one-shot image recognition”. In: *ICML deep learning workshop.* Vol. 2. Lille. 2015, p. 0.

[10] T. Lin and *et al.* "Focal loss for dense object detection". In: *Proceedings of the IEEE International Conference on Computer Vision.* 2017, pp. 2980–2988.

[11] O. Oktay and *et al.* "Attention U-NET: Learning where to look for the pancreas". In: *arXiv preprint arXiv:1804.03999* (2018).

[12] J. Redmon and *et al.* "You only look once: Unified, Real-time object detection". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.* 2016, pp. 779–788.

[13] S. Ren and *et al.* "Faster R-CNN: Towards real-time object detection with region proposal networks". In: *Advances in Neural Information Processing Systems* 28 (2015).

[14] 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.* Springer. 2015, pp. 234–241.

[15] S. Ruder. "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747* (2016).

[16] M. Shorfuzzaman and M.S. Hossain. "MetaCOVID: A Siamese neural network framework with contrastive loss for n-shot diagnosis of COVID-19 patients". In: *Pattern Recognition* 113 (2021), p. 107700.

[17] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[18] C. Sudre and *et al.* "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations". In: *Deep learning in Medical Image Analysis and multimodal learning for clinical decision support.* Springer, 2017, pp. 240–248.

[19] A. Voulodimos and *et al.* "A few-shot U-Net deep learning model for COVID-19 infected area segmentation in CT images". In: *Sensors* 21.6 (2021), p. 2215.

[20] X. Wang and *et al.* "SSA-Net: Spatial Self-attention network for COVID-19 pneumonia infection segmentation with semi-supervised few-shot learning". In: *Medical Image Analysis* 79 (2022), p. 102459.

[21] Z. Zhang and M. Sabuncu. “Generalized cross entropy loss for training deep neural networks with noisy labels”. In: *Advances in Neural Information Processing Systems* 31 (2018).

[22] D. Zhou and *et al.* “IoU loss for 2D/3D object detection”. In: *2019 International Conference on 3D Vision (3DV)*. IEEE. 2019, pp. 85–94.