Deep Learning for Classification of COVID-19 Chest CT Scan: A Transfer Learning Approach

A thesis submitted to Indian Statistical Institute in partial fulfillment of the requirements for the degree of Master of Technology in Computer Science

by
Subhadip Ghosh
CS1904

Under the supervision of
Dr. Pradipta Maji
MIU, ISI, Kolkata

M.Tech(CS) Thesis Completion Certificate

Student: Subhadip Ghosh (CS1904)

Title: Deep Learning for Classification of COVID-19 Chest CT Scan: A Transfer Learning Approach

Supervisor: Dr. Pradipta Maji

This is to certify that the thesis titled "Deep Learning for Classification of COVID-19 Chest CT Scan: A Transfer Learning Approach" submitted by Subhadip Ghosh in partial fulfillment for the award of the degree of Master of Technology is a bonafide record of work out by him under my supervision. The thesis has fulfilled all the requirements as per the regulations of this Institute and, in my opinion, has reached the standard needed for submission. The results contained in this thesis have not been submitted to any other university for the award of any degree or diploma.

09-07-2021

Date

Pradipta Maji

Acknowledgement

I would like to thank my dissertation supervisor, Dr. Pradipta Maji, Professor, Machine Intelligence Unit, Indian Statistical Institute, Kolkata, for agreeing to guide me and helping me with interesting problems. Without his continuous guide and support, this would not have been possible.

Subhadip Ghosh
MTech(CS),2nd year
Roll No.: CS1904
ISI,Kolkata
Friday 9$^{\text{th}}$ July, 2021

Abstract

The latest threat to global health is COVID-19. It has a tremendous diffusion rate and to combat with this pandemic, large scale testing and diagnosis is required. RT-PCR is the most accurate screening for validating COVID-19 infection, but it is highly dependent on swab technique and needs time and resources. Thus, we need to find an alternative way to predict COVID-19. Many researchers already conclude that COVID-19 is very related to Pneumonia and lungs feature of COVID is related to that of Pneumonia. There is ongoing research to detect Pneumonia [13] from Chest CT scans. Lung segmentation can help us to detect pulmonary abnormalities[10].

In this article first we try to segment lungs from chest CT scan and investigate the problems we face for COVID cases in deep learning architectures for lung segmentation. We propose an classical image processing algorithm to detect Lung from chest CT.

As already mentioned that CNN is a great architecture to classify images, we are going to use a deep CNN model for lung classification.

Covid is a new disease and we have to move faster to detect it. Hence, we are going to use transfer learning approach and use knowledge of pneumonia detection to classify COVID-19.

In deep learning weight initialization for deep neural network is a major factor and can lead us to very different performance. In this article we are going to propose an weight initialization technique for transfer learning that can use not only the information about the architecture but also the information of the new class with respect to other known classes.

Contents

1 Introduction

The chest X-ray and thoracic computed tomography (CT) is easily accessible medical imaging equipment. CT image can be used to indicate COVID. As RT-PCR takes up to 24 hours and needs multiple tests for compulsive result, chest CT can be used as a diagnosis tool for rapid screening of COVID-19 patients. However manual diagnosis of chest CT is time consuming and labour-intensive process. A dependable computer aided diagnosis system for COVID-19 may have huge implication for improving the detection efficiency. An AI based model can be used as a preceding test in those areas which are suffering from shortage of RT-PCR tools.

The objective of this work is to identify COVID using chest CT scans. As a prepossessing step, lung segmentation will be a part of this project. The main challenge in lung segmentation is that we do not have mask of covid-19 effected chest CT. So, we need to come up with deep learning model or algorithm to efficiently segment chest CT.

Deep learning techniques have recently achieved impressive results in a variety of computer vision problems, raising expectations that they might be applied in other domains, such as medical image analysis. Deep learning convolutional neural networks has a great potential for building COVID-19 triage systems and detecting COVID-19 patients, especially patients with low severity. Transfer learning is a fast and efficient way to deal with problems that dose not have sufficient data. And as Pneumonia is related to Covid, we are going to use transfer learning on a deep Pneumonia classifier .

1.1 Preliminaries

For Lung segmentation we are going to use Some decoder-encoder network and we are going to propose an alternative algorithm for lung segmentation without the help of machine learning.

For classification we are going to use a basic Convolutional Neural Network with batch normalization. And we are going to use T-SNE[12] as dimension reduction technique.

1.2 Why Lung Segmentation

In medical image processing tasks, data is very valuable and we don't have very large data set compared to other image processing fields. Hence, we have to work with relatively less data and it is a big problem for deep leaning models, as deep CNN is very data hungry. Hence CNN, trained with relatively small data set, tends to prioritize unnecessary features. In this case image pattern outside of lungs. So it becomes less efficient on real world cases. Hence, we expect that if we segment our X-ray and force our CNN to search for features inside lungs, it will perform better.

1.3 Why Transfer Learning

When a new disease occurs, it is very hard to build a model from scratch. And we want to create a model as fast as possible to deal with the current emergency situation. So, we can not wait to get a large data set to train our deep learning model. And we have to use data for other diseases which are intensely related to this new disease. We can prepare this data and create a deep learning model so that whenever we face any emergency situation, we can use this trained model to speed up our processes. Hence, transfer learning is a reliable solution for this problem.

Covid is effecting lungs and researchers have proved that lung feature for Covid positive patients is related to that of Pneumonia. And we have big data sets for Pneumonia classification. So We can transfer our learned features in this case kernels and/or layers to classify COVID.

2 Related Work

2.1 Lung Segmentation

Registration

Image registration is a process where we try to find a correct alignment of images. In the simplest case, two images are aligned. One image is treated as the target image and the other is treated as a source image, the source image is transformed to match the target image. We have an iterative algorithm that transform the source image so that it fits our target image until a local optimum found.

For segmentation we need an image with its mask. We set the image as our source image. Now when a new image cames, we set it as target image and use registration to get the transformation. Lastly, We apply this transformation to get the mask of the new image.

There are many types of registration and for these purposes, we study

1. Atlas-Based Segmentation [8]

2. Shape-Based Segmentation [7]

Deep learning

There are many deep learning architectures that help us to segment images, in this project we study the following 3 deep learning architecture.

1. SEG-Net [1] : We memorize the active indices of the max pooling layers in encoder part, and when we are going to upsamplelling it in decoder we use that indices. So, We can think of a residual connection from encoder to decoder that transmits the activity of indices in every pooling layers

2. U-Net [9]: Here We add residual connections similarly to SEG-Net. But this time we join the outputs of layers of encoder with decoder layers.

3. PSP-Net [14]: In this architecture we use pyramid pooling module over the architecture of U-Net to correctly segment all size objects.

In this project we use U-Net to segment the chest x-ray.

2.2 Deep Learning

In today's world deep CNN is achieving wonders in computer vision tasks. Convolution layers can detect hidden patterns in the image, and this patterns become more and more complex from detecting edges to detecting specific texture, as we go deep inside the CNN. But Deep CNN has its own challenges. Firstly, Deep learning neural networks are very prone to over-fit a training data set with few example. Ensemble is a way for this problem but it is very computationally expensive.Secondly, A Deep CNN is updated layer-by-layer backward from the output to the input that assumes the weights in the layers prior to the current layer are fixed. But after training layers are changed. So the update procedure is forever chasing a moving target. The way to deal with these problems are dropout[11] and batch normalization [5].

Dropout

Dropout is a regularization method. We use dropout to approximate training a large number of neural networks with different architectures. When we apply dropout in a layer, it randomly removes some connections in that layer. and that can be treated as different architecture with different number of nodes. Hence over fitting will be reduced. For example, if the model is overly dependent in one feature and we use a 25% dropout. The expectation of removing that node is in each 4 batch. Hence approximately after every 4 batch that feature is removed and every other feature has their turn to learn.

In our model we are going to use dropout in fully connected portion.

Batch Normalization

In deep learning network, we train our model with a batch at a time, and in batch normalization layer we standardize the activations of the previous layer for the batch. It means that the spread and distribution of inputs during the weight update will not change dramatically. This has the effect of stabilizing and speeding-up the training process of deep neural networks. It makes our model perform well and reduces the requirement of the number of epochs.

In our model we are going to use batch normalization in convolution portion.

2.3 Transfer Learning

Transfer learning[6] is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. The main reason we are seeking help of transfer learning because when this kind of extraordinary situation arrives, we need to work fast and we can not rely on the fact that a deep neural network has to get a large amount of data to perform well. Hence we can create a basic classification for all kind of disease and when these kind of diseases arrive, to create a classifier, first we seek previously known disease common symptoms and effect so that we can under stand that this new disease is related to which of the old ones. And after deciding it, we can use those data as well as the classifier for that disease to create a new model.

Transfer learning generally have 3 step,

1. Select a trained model
2. The trained model can be used as the starting point for a model on the second task of interest. This may involve using all or parts of the model, depending on the modeling technique used.
3. Train the model with new data as well as available data for the previous task.

When we are dealing with computer vision tasks, first few layers detect edges and then some layers should detect shapes in the image and at the end some layers suppose to detect pattern in the image, so in step 3, how far we are going to train our network is a hyper parameter. If we think our trained model is good enough so that we do not need to train the model to detect edges and shapes, we can stop the back propagation mid way to reduce over fit and make it faster to train.

2.4 Weight Initialization

Weight initialization is an important design choice when developing deep learning neural network models. Historically, weight initialization involved using small random numbers, although over the last decade, more specific heuristics have been developed that use information.

We cannot initialize all weights to the value 0.0 as the optimization algorithm results in some asymmetry in the error gradient to begin searching effectively.

Historically, weight initialization follows simple heuristics, such as:

- Small random values in the range [-0.3, 0.3]
- Small random values in the range [0, 1]
- Small random values in the range [-1, 1]

It has been developed over the last decade that weight initialization has become the standard procedure. They may result in a slightly more effective optimization process.

These modern weight initialization techniques are divided based on the type of activation function used in the nodes, such as Sigmoid and ReLU.

Glorot Weight Initialization

The Glorot initialization method [3] is calculated as a random number with a uniform probability distribution in $[-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}]$, where n is the number of inputs to the node. The normalized Glorot initialization method is calculated as a random number with a uniform probability distribution in $[-\frac{\sqrt{(6)}}{\sqrt{n+m}}, \frac{\sqrt{(6)}}{\sqrt{n+m}}]$ where n is the number of inputs to the node (i.e. number of nodes in the previous layer) and m is the number of outputs from the layer (i.e. number of nodes in the current layer)

Though it is a very innovative idea to use the architectural information to initialize weight, it dose not perform well for ReLU activation function.

He Weight Initialization

For nodes with ReLU activation function He initialization [4] works very well compared to Glorot initialization, The He initialization method is calculated as a random number with a Gaussian probability distribution with a mean of 0 and a standard deviation of $\sqrt{(2/n)}$, where n is the number of inputs to the node.

3 Proposed Model

3.1 Lung Segmentation

Lung X-ray must have two regions i.e. two lungs whose texture is different from the background of the image and we want to get that section.

Hence, with a proper filter to detect texture we can transform the given image of a chest X-Ray into an image that has slightly larger value inside chest region than background of the image. In this case mean filter with large stride suffices our requirement.

Now, as we already have a difference in grey level value of the image, we just need to select the edges of the chest region. For this we can apply an efficient edge detection method to get the region of interest for our task. Here we are using Canny edge detection technique and loop filling to get the mask of the X-Ray

Algorithm

We have 2 parameters, k and m. k is a number that we assume as canny edge detection[2] with sigma equals to k will not return anything and m is a tentative number such that a mask of a lung X-Ray will have minimum m number of true pixels.

1. Use a proper filter to differ the grey level value of the image inside and outside the chest.
2. Set jump equals to 0 and σ equals to k
3. Set $\sigma = \sigma - 0.1$
4. Apply Canny edge detection with σ and put it in A
5. Apply dilation on A
6. Apply loop filling on A and put it in B
7. Set $jump = jump + numberoftruepixelinB$
8. If $jump < m$ go to 3.
9. Return A $\oplus$ B

in Figure 1 we can see the states of the image B corresponding to several value of sigma.

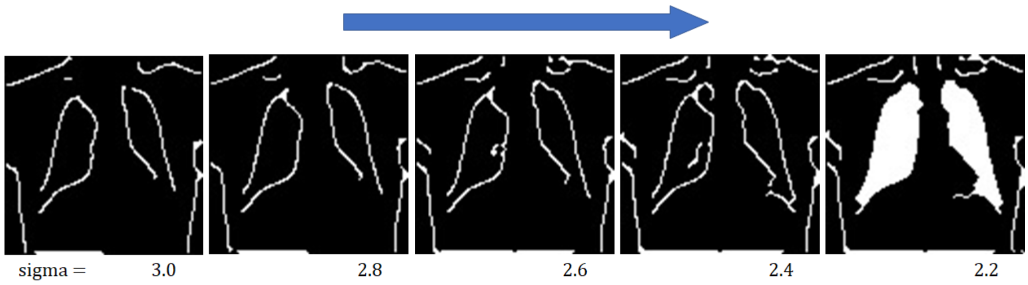

Figure 1: State of the image in various iteration with monotone decreasing sigma

Advantage

- Do not need any data to compute mask
- Independent on the size of the image
- Faster than Deep Learning and image registration

Disadvantage

- Not appropriate for distorted images
- Computing a mask for specific task is tough
- No part of lungs should touch or cross the boundary of the image

3.2 Deep Learning Model

As a deep learning model we use Convolutional Neural Network with batch normalization in convolutional part and dropout in fully connected part of the network. We have 6 convolution layers with kernel size (3,3), (4,4), (5,5), (4,4), (4,4), (3,3) with number of filters are 16, 15, 32, 64, 64, 128 respectively. And we have batch normalization in after every max pooling layer. In fully connected part, we have 3 fully connected layers, with number of nodes 64, 32, 2 respectively. We apply drop out after first two fully connected layers. The architecture of our model is given in Figure 2.

Now, we are going to add another node to the output layer to add another class, in this case Covid.

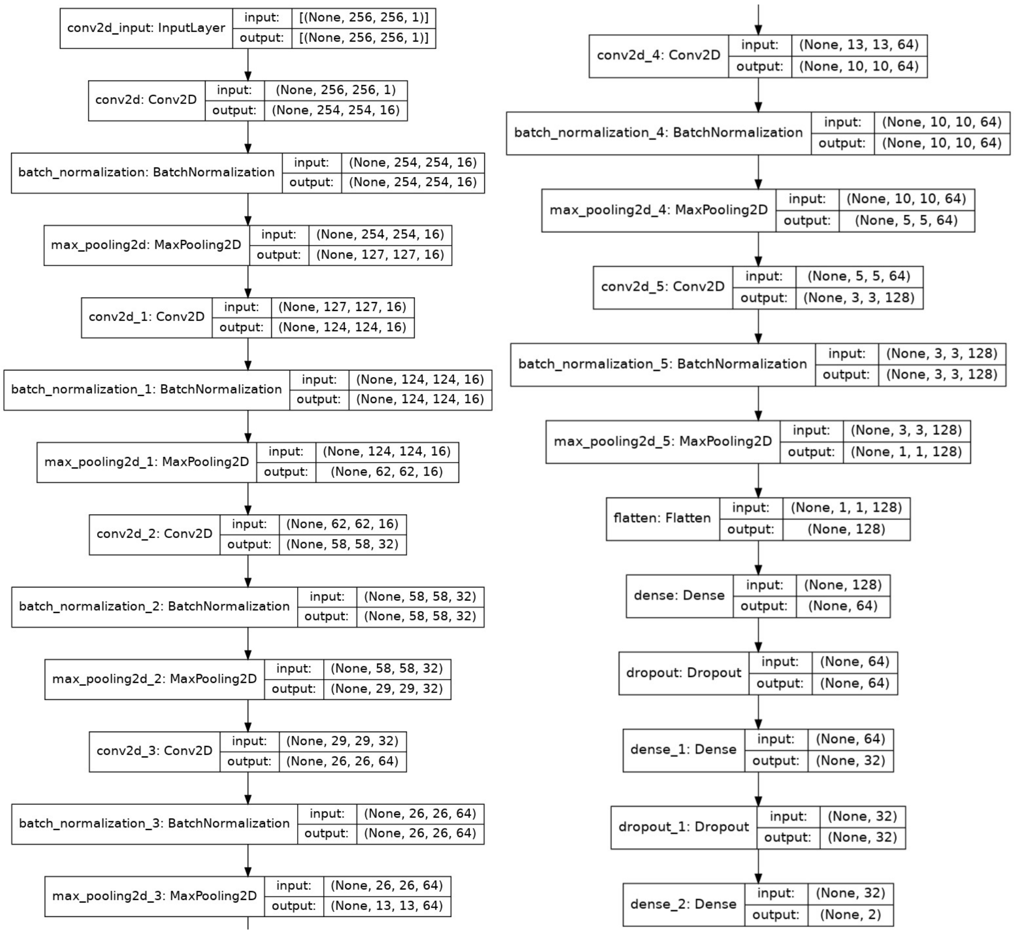

Figure 2: Trained Model

3.3 Transfer Learning

When we add another node in output layer, random weight initialization can lead us to different local optimum, which can result uncertainty in performance. In this section we discuss about a specific weight initialization that searches for relative position of the new class in feature space with respect to trained classes. With the help of that information and weight of the trained classes, we initialize weight of the new class.

Note that, in this case we do not look into the architecture for weight initialization because that information should be hidden in the weights of other classes.

Let us consider a model M with n output nodes and p layers. We want to add another output node in the last layer of the model.

Terminology

For a layer k of a deep learning model,
    $out(k)$ = output vector of the layer k.
    $in(k)$ = input vector of the layer k.

Assumption

1. We have a layer k in the model M such that $dimension(out(k)) \geq n$
2. Each class of the model is related to new class and should help to extract features for the new class.
3. Learned classes are linearly independent, as new class is going to be represented as linear combination of other classes.

Algorithm

1. Get the $out(k)$ of training set which has $(n+1)$ classes.
2. Use a dimension reduction technique to reduce the output into n dimension.
3. Get the centroid of each classes and create a vector $V_{(n,n+1)}$ where last column is the vector for new class. i.e, $V_{(n,n+1)} = [V'_{(n,n)} : k_{(n,1)}]$ where k is the centroid of the new class.
4. Get a vector $x_{(n,1)}$ such that $V'_{(n,n)} \times x_{(n,1)} = k_{(n,1)}$ i.e. we are expressisg the centroid of the new class as linear combination of the other centroids.
5. Get the weight $P_{(dimension(in(k)),n)}$ and bias $b_{(1,n)}$ of the last layer of our model M
6. Construct a new vector $S_{(dimension(in(k)),1)} = P_{(dimension(in(k)),n)} \times x_{(n,1)}$ and $r = b_{(1,n)} \times x_{(n,1)}$.
7. Add a new node in the last layer of our model M with weight S and bias r.

4 Experimental Result

4.1 Data Set

Most of the learning has been done on Kaggle with their available GPU.

To train our model we have used `https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data`. For, transfer leaning purposes we use two data sets.

1. Data Set - 1:

   Pneumonia and Normal: `https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia`

   Covid:

   (a) `https://github.com/ieee8023/covid-chestxray-dataset`

   (b) `https://eurorad.org`

   (c) `https://sirm.org/category/senza-categoria/covid-19/`

   (d) `https://github.com/ml-workgroup/covid-19-image-repository/tree/master/png`

   (e) `https://github.com/armiro/COVID-CXNet`

2. Data Set - 2:

   Pneumonia and Normal: `https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data` (sample of 1500 data in each class)

   Covid: `https://bimcv.cipf.es/bimcv-projects/bimcv-covid19/#1590858128006-9e640421-6711` (sample of 1500 data)

4.2 Lung Segmentation

Proposed Algorithm

Our proposed algorithm works better if we divide the X-Ray into two parts containing one lung in each. We can do this with the help of the following algorithm,

1. Compute a vector v by sum of the image in vertical direction.
2. Set $a = \frac{length(v)}{2}$
3. Move a towards the local maximum value until it reaches one.

4. Return a

This algorithm will give us an index to slice the image to so that we can get two separated lung in two separated images and after that we will use the previously discussed algorithm.

Figure 3: Input, Output and Result of Segmentation by Classical Image Processing Approach

In Figure 3 we are showing an example of the algorithm

U-Net

The U-Net architecture which we have used is shown in Figure 4. An example of this U-Net architecture is shown in Figure 5.

Our proposed method has some advantages and disadvantages over U-Net. In Figure 6 and 7 we show the disadvantages and in Figure 8-11 we show the advantages of our method over U-Net. Figure 6,7 contain example of Pneumonia effected chest CT 8-11 contain examples of Covid effected chest CT.

4.3 Deep Learning

In our experiments in training phase we use RSNA kaggle data set with 50 epoch and in transfer learning phase we use two separate data sets with test-train ratio of 1:1 with 20 epoch and have results for loss, accuracy for testing and training set and precision and recall for Covid class through each epoch of the transfer learning phase.

In our transfer learning algorithm the layer k is a hyper parameter. In our experiment we use two cases, first, we select $flatten$ layer as k and second, second last layer i.e. $dense\_1$ as k.

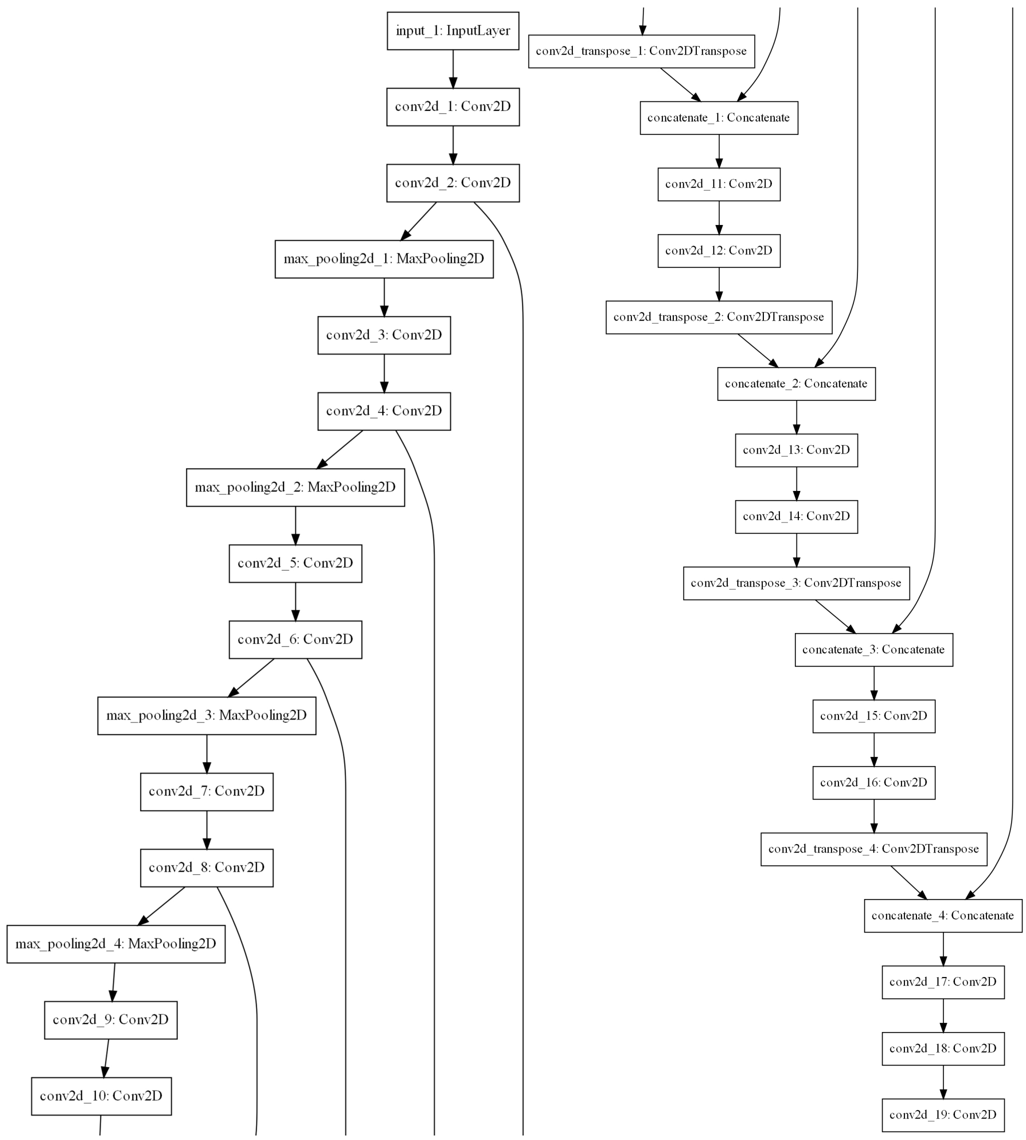

Figure 4: U-Net Architecture

As the number of layers that will be trained in transfer learning is a hyper parameter we select two extreme cases for our experiments. First we train all layers and then we train only last layer.

Visualization of Dimension Reduction

In this project our trained model is 2 dimensional. Hence it is a golden opportunity to visualize the work of our dimensional reduction technique and

Figure 5: Input, Output and Result of Segmentation by U-Net

Figure 6: Input, Output and Result of Segmentation by Classical Image Processing Approach

Figure 7: Input, Output and Result of Segmentation by U-Net

investigate how a 2 class classifier can give us knowledge about a unknown class if that new class is related to known class.

In Figure 12 this example is the result of data 1 from a model that is trained with Normal and pneumonia cases of data 1. In Figure 13 this model is the result of data 1 of a model which is trained with RSNA data set.

Figure 8: Input, Output and Result of Segmentation by U-Net

Figure 9: Input, Output and Result of Segmentation by Classical Image Processing Approach

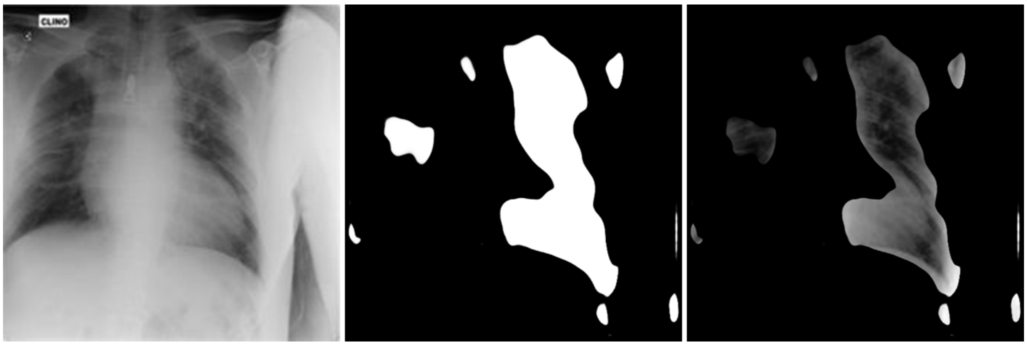

Figure 10: Input, Output and Result of Segmentation by U-Net

Note that in the plots solid points are centroid of their corresponding classes. and in our algorithm, x is nothing but the constant by which we can write the red point i.e. the centroid of the new class as the linear combination of other two points.

Figure 11: Input, Output and Result of Segmentation by Classical Image Processing Approach

Performance of the Algorithm in All Class

Now, we are going to present the performance of our algorithm. We compare our proposed weight initialization technique with Glorot uniform weight initialization technique.

Every plot is corresponding to 10 models where thickness of the plots represent the standard deviation of the data i.e. standard deviation of the 10 points we have.

Hence, in total we have 10 models for each of the 3 training process for 3 class classifier and 1 for 2 class classifier with 2 input type i. e. masked and normal, and for normal we have 2 transfer learning processes. In total we have $2 \times 3 \times 3 \times 10 = 180$ 3 class classifier and $2 \times 3 \times 10 = 60$ 2 class classifier.

In the Figure 14,15 we compare our algorithm for k = flatten layer and second last layer with Glarot initialization technique

Now, let us see the performance of our algorithm in lungs segmented images in Figure 16,17.

All data in Figure 14-17 is shown in the case of transfer learning where we allow all layers to train. In Figure 18,19 we are comparing weight initialization when we learn only last layer of the trained model.

Performance of the Algorithm in New Class

Precision and recall for the new class is shown in Figure 20-24 for data set 1,2.

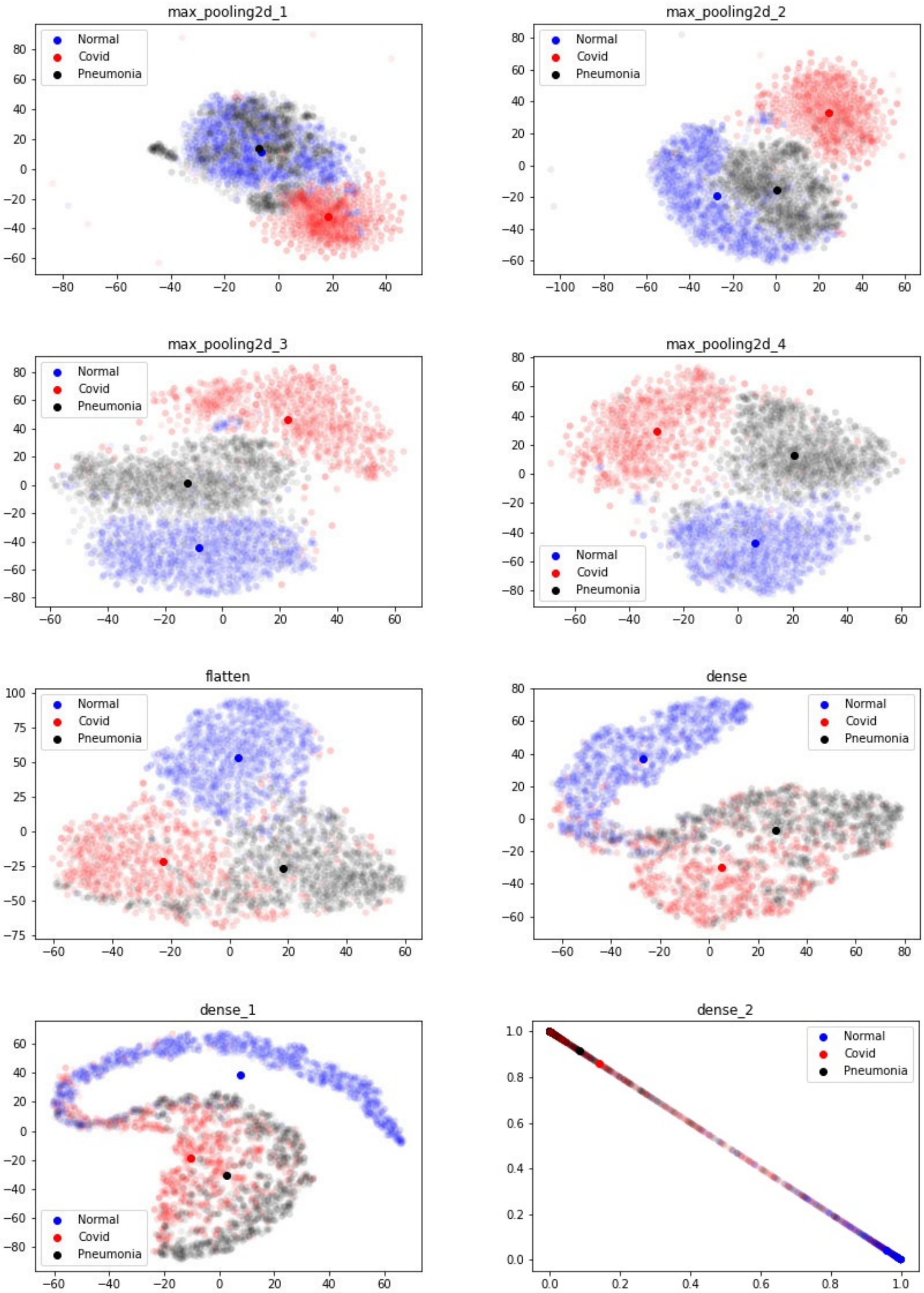

Figure 12: Visualising the information we can get for a unknown class

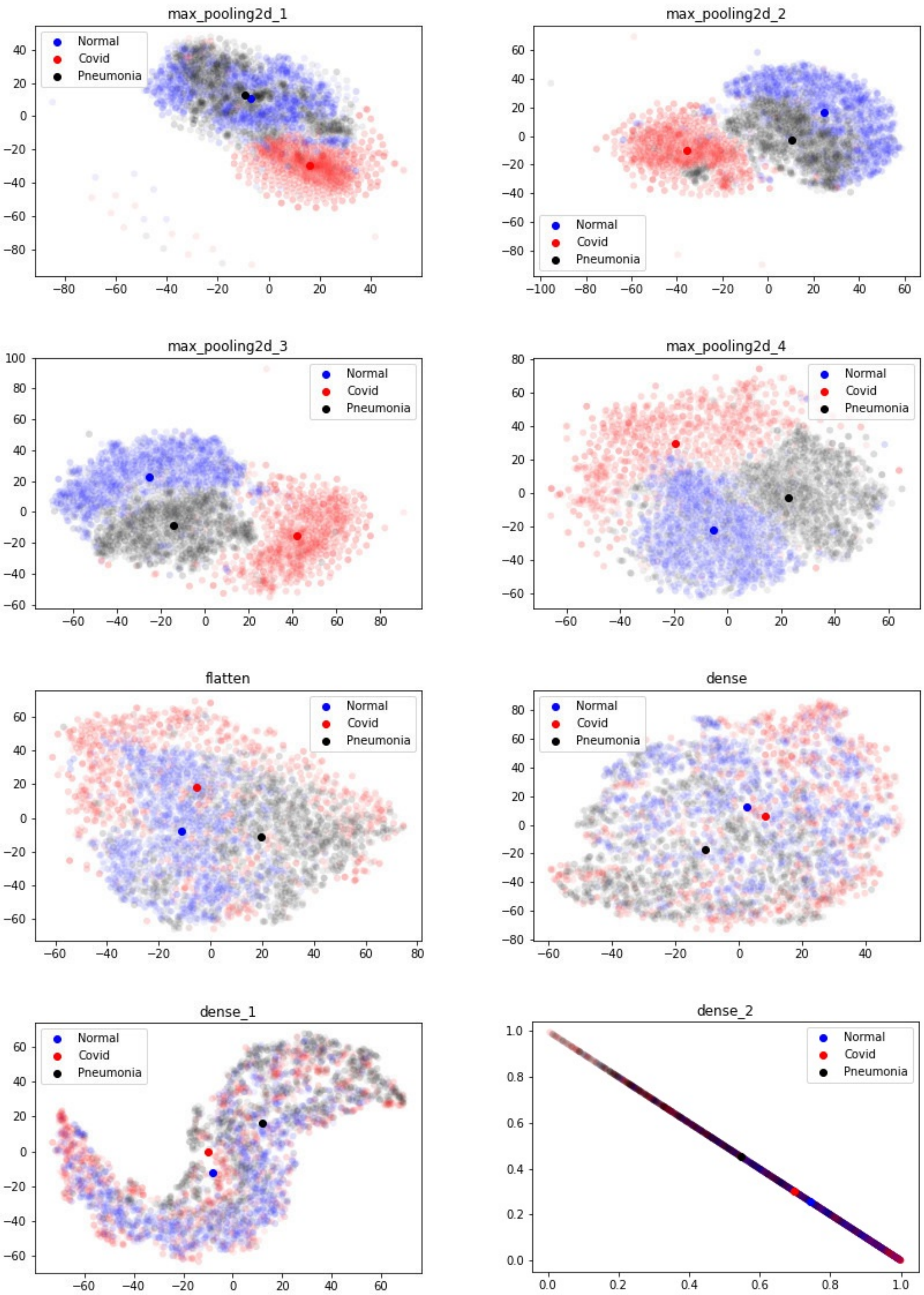

Figure 13: Visualising the information we can get for a unknown class

4.4 Effect of Lung Segmentation in Deep Learning

Lung segmentation should help us in classification. When our data is lung segmented image with entire image, comparison of performance is shown in

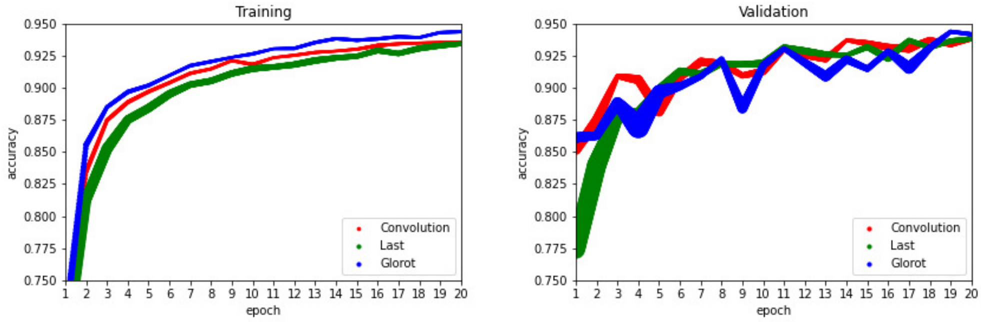

Figure 14: Accuracy corresponding to data set 1

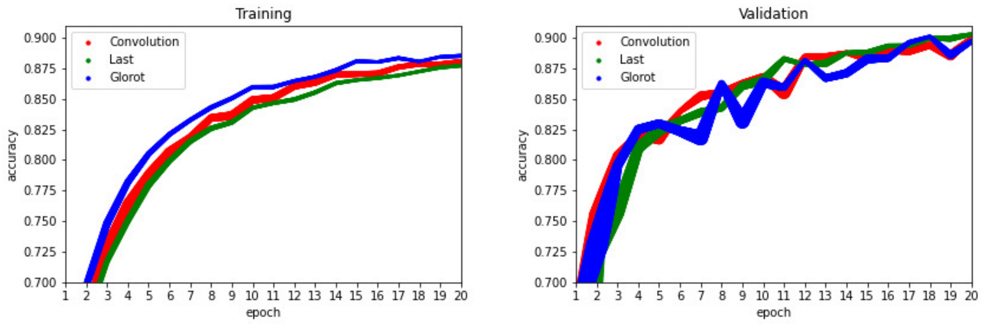

Figure 15: Accuracy corresponding to data set 2

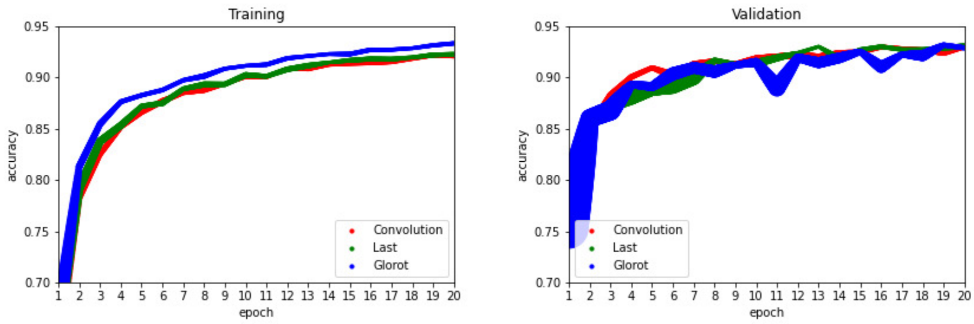

Figure 16: Accuracy corresponding to segmented data set 1

the Figure 24,25.

4.5 Final Result

Final results of our model are shown in table 1 and table 2 when the input image is entire image and lung segmented image with 20 epoch where each

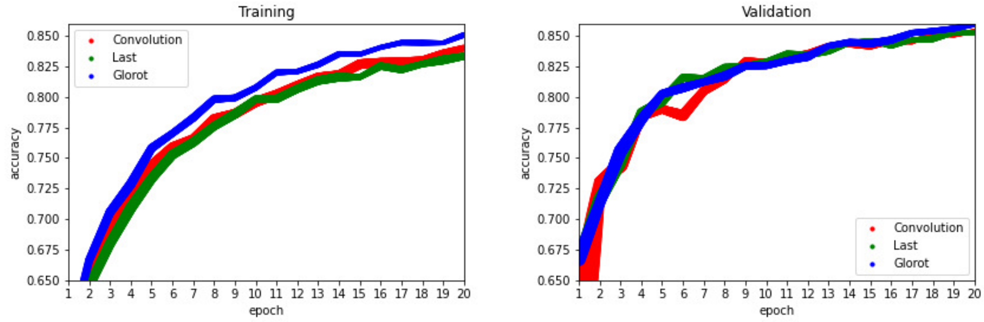

Figure 17: Accuracy corresponding to segmented data set 2

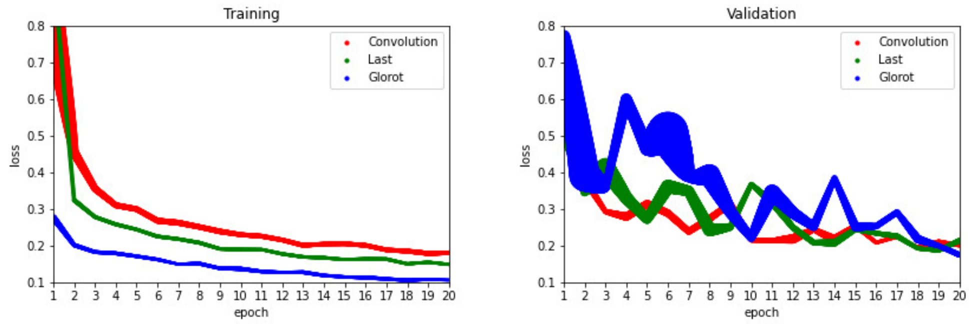

Figure 18: Loss corresponding to data set 1

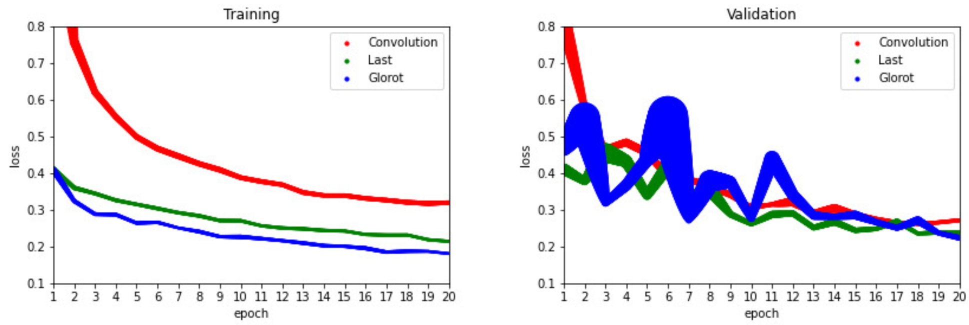

Figure 19: Loss corresponding to data set 2

data is mean among 10 models.

Weight initialization	Confusion matrix Actual class	Predicted class			Accuracy
Data set 1					
Proposed Method for k = flatten layer		Normal	Covid	Pneumonia	0.934
	Normal	637	6	16	
	Covid	5	580	12	
	Pneumonia	79	11	597	
Proposed Method for k = dence_1		Normal	Covid	Pneumonia	0.941
	Normal	632	7	20	
	Covid	8	574	15	
	Pneumonia	57	7	623	
Glorot weight initialization		Normal	Covid	Pneumonia	0.932
	Normal	641	5	13	
	Covid	9	578	10	
	Pneumonia	86	8	593	
Data set 2					
Proposed Method for k = flatten layer		Normal	Covid	Pneumonia	0.9
	Normal	704	39	10	
	Covid	97	614	9	
	Pneumonia	59	5	636	
Proposed Method for k = dence_1		Normal	Covid	Pneumonia	0.899
	Normal	677	47	29	
	Covid	98	602	20	
	Pneumonia	23	4	673	
Glorot weight initialization		Normal	Covid	Pneumonia	0.891
	Normal	624	116	13	
	Covid	29	676	15	
	Pneumonia	50	13	637	

Table 1: Final result of our model when input is entire image

Weight Initialization	Confusion matrix Actual class	Predicted class			Accuracy
Data set 1					
Proposed Methtod for k = flatten layer		Normal	Covid	Pneumonia	0.92
	Normal	575	0	84	
	Covid	8	571	31	
	Pneumonia	20	12	624	
Proposed Method for k = dence_1		Normal	Covid	Pneumonia	0.935
	Normal	599	1	59	
	Covid	9	576	25	
	Pneumonia	23	9	642	
Glorot weight initialization		Normal	Covid	Pneumonia	0.917
	Normal	559	0	100	
	Covid	8	568	34	
	Pneumonia	14	5	655	
Data set 2					
Proposed Methpd for k = flatten layer		Normal	Covid	Pneumonia	0.87
	Normal	589	110	43	
	Covid	90	632	14	
	Pneumonia	17	9	669	
Proposed Method for k = dence_1		Normal	Covid	Pneumonia	0.862
	Normal	594	100	48	
	Covid	116	603	17	
	Pneumonia	10	9	676	
Glorot weight initialization		Normal	Covid	Pneumonia	0.862
	Normal	587	109	46	
	Covid	107	606	23	
	Pneumonia	8	6	681	

Table 2: Final result of our model when input is lungs segmented image

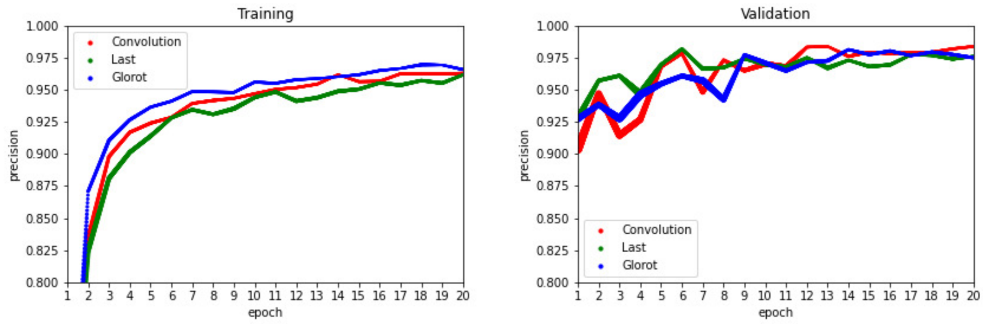

Figure 20: Precision of Covid class for data set 1

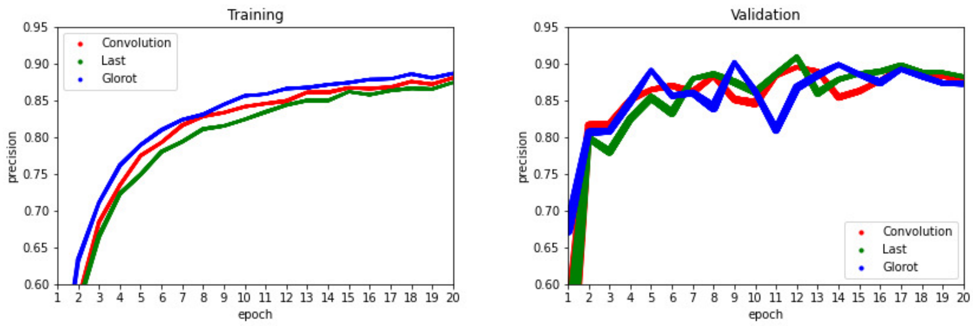

Figure 21: Precision of Covid class for data set 2

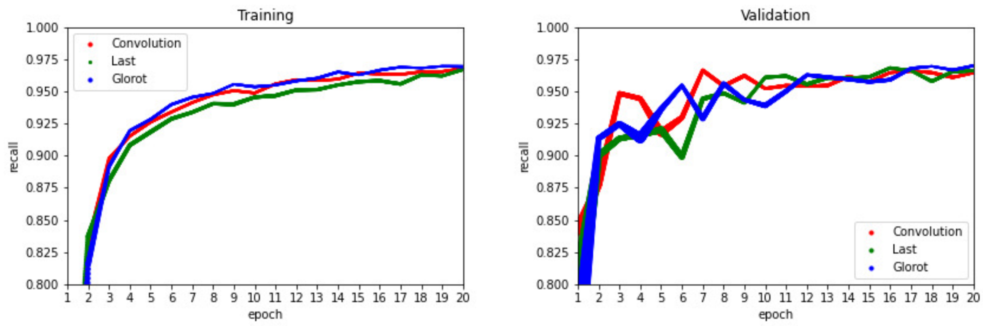

Figure 22: Recall of Covid class for data set 1

5 Discussion

5.1 Computational Limitation

In transfer learning all experiments have been done in two extreme cases. Firstly, we train all layers and secondly where we train only the last layer of

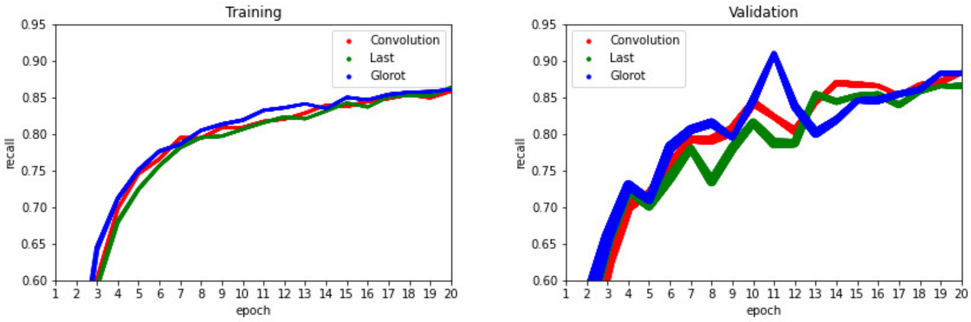

Figure 23: Recall of Covid class for data set 2

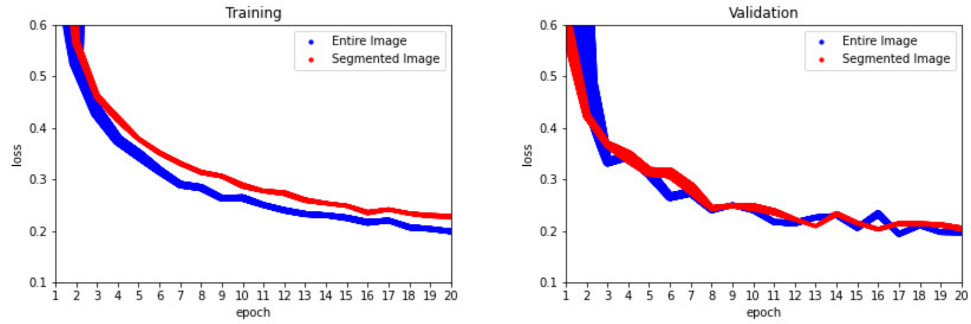

Figure 24: Trained with proposed weight initialization with $k = dense\_1$ technique

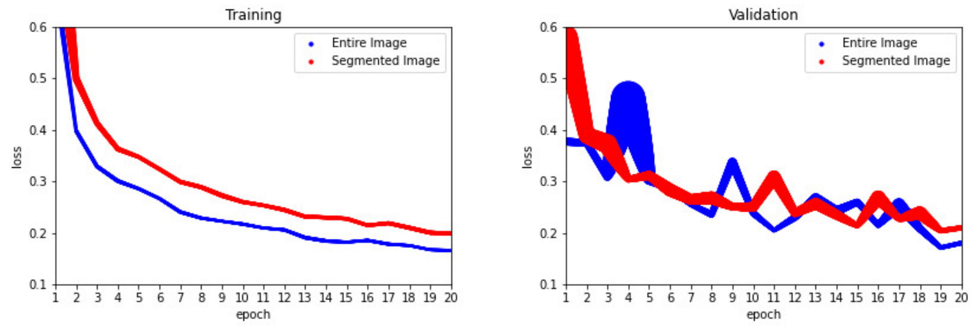

Figure 25: Trained with Glorot weight initialization technique

the trained model. The number of layers that is supposed to be trained in transfer learning is a hyper parameter and due to computational limitation of this project, we fail to set this hyper parameter.

The model we have used is relatively simple with less than 0.2 million parameters and every model is trained with only 20 epoch.

We are forced to reduce training data due to configuration of our system and to train many models so that our result maintains consistency.

5.2 Lung segmentation

Disadvantage of this algorithm over U-Net

This method has disadvantages when we are going to use this for big data set.

- In big data set generally quality of the data is not good and can have noise and distorted images and this algorithm will fail to segment these kind of images as it depends on texture.
- Rotational change of the image can be fatal as the slicing will be very hard to perform. And if we dp not perform slicing,there is a possibility that two lungs may active on two different value of sigma. Hence, image will contain some noise in the lungs that activated earlier.
- Deep learning algorithms take care of shapes, possible symmetry of size and number of blob in the image, where our proposed algorithm fails to do so.

In Figure 5, our classical image processing technique failed to recognise the lower portion of the lungs as the texture of the lower portion was different from the texture of the upper portion of the lungs. But as it dose not satisfy the constrain of true pixel, it goes on and selects the corner loop which forms as the image is taken improperly, which is very common in big data sets.

Advantage of this algorithm over U-Net

When a new disease occurs it is very unlikely to train our lung segmentation model again to perform better in new defective lungs. As a result this deep learning segmentation model never sees this kind of data i.e. Covid effected lungs, sometimes it dose not perform well and fails to get proper structure of lungs. Hence, the performance of deep learning model dips for new classes. In this case it segments quite good in Normal and Pneumonia cases but fails to segment properly in Covid cases.

In Figure 7 U-Net fails to recognise the pattern inside lungs where in Figure 8 our algorithm captures shape from the boundaries of the lungs. In Figure 9 and 10 both algorithm fails to segment the lungs properly. But our algorithm has a minimum true pixel constrain, hence it tends to capture extra information from outside of the lungs. As omitting the portion inside the lungs is very costly, our algorithm tends to work better.

5.3 Motivation for Proposed Algorithm

Our main hypothesis is to obtain an approach that gets some information about an unknown class before training. If the known class is related to the new class, model should give us feature from intermediate layers that can classify the new class. In Figure 12 we can see that 2 dimensional representation of the output from $flatten$ and $max\_pooling2d\_4$ is separating the classes decently and in figure 13 a model trained with completely different data can do the same for $max\_pooling2d\_4$. From the figure it is clear that there is a sweet spot for selecting k, as the the clusters for each class is separated in some of the middle layers of the model.

5.4 Good in Validation

As shown in the plots above, compared to Glorot weight initialization, our algorithm dose not perform well in training set. But our main focus is on the validation set, as our trained model is very efficient in unseen images, our algorithm can utilize that behaviour quite effectively and in many cases it beats Glorot weight initialization.

Due to the randomness of in Glorot weight initialization, the variance of the data is higher than our algorithm, that means there is no guarantee to get an efficient model in each time.

5.5 Performance in Lung Segmented image

It is not prominent in our experiments that lung segmentation helps us in classification though, the difference in performance is drastically reduced in validation. The reason for this can be the failure of the U-net in Covid cases.

6 Conclusion

We all have been affected by the current Covid-19 Pandemic. How ever the impact of the pandemic and its consequences are felt differently depending on our status as individuals and as members of society. As RT-PCR is time consuming and not visible in rural areas an alternate way is required. Availability of Chest CT scan and prominent effect of Covid in lungs lead us to use chest CT to classify Covid. CNN achieves wonders in computer vision tasks. So, we try to classify Covid with deep CNN. As, Covid is very related to Pneumonia and large Pneumonia data set is available, we use transfer learning to classify Covid. We have tried to enhance our classification power by lung segmentation and faced some difficulties to segment Covid cases. We proposed an alternate way but that is not good enough for large data set. We use CNN with batch normalization and dropout to classify Pneumonia and use transfer learning to classify Covid. We discussed the importance of weight initialization in Deep learning networks and proposed an alternative method to initialize weight of the node that we add in transfer learning. In vast experiment it is prominent that through epoch, our weight initialization technique works better. In the end we have an average of 90% along two data set.

7 Future Direction

7.1 Lung Segmentation

In our algorithm we only use the texture and edges of the image. But we fail to use the information about lungs. So we can use the following information,

- Shape of the lungs
- Similarity in height and width of the masks of each lung
- Usage of multiple mask

As pre-processing technique we can use the following method to increase the performance of the algorithm in big data sets,

- Noise cancellation
- Usage of moment vector to make it symmetric along the y-axis
- Usage of more complicated mask to recognise the pattern inside lungs effectively

7.2 Transfer Learning

In our experiments we can see that the classes are well separated before training. For example in Figure 12 and 13 the model is separating the data well till the layer $max\_pooling2d\_4$. Hence there is no need to train the layers prior to the layer $max\_pooling2d\_4$. We can investigate more suitable ways for training.

7.3 Weight Initialization

In our algorithm we use linearly dependency of the classes to initialize the weight of the new class. But this is a naive way to use the information of the known classes to get the weight of the new class. We can search for an efficient way to use this relative position of the classes.

In future, we can investigate the performance of the proposed algorithm on other models and can conclude that this weight initialization is working better than Glorot initialization on every model or this may be a special case.

7.4 Role of Segmentation in Classification

It needs to address that we have to use segmentation more properly, and try to investigate why we are not getting better result for segmented images. A proper segmented data set may give us a better result.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[2] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[3] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[5] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[6] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[7] Kilian M Pohl, John Fisher, Ron Kikinis, W Eric L Grimson, and William M Wells. Shape based segmentation of anatomical structures in magnetic resonance images. In *International Workshop on Computer Vision for Biomedical Image Applications*, pages 489–498. Springer, 2005.

[8] Jonathan Rohrer and Leiguang Gong. Focused atlas-based image registration for recognition. In *2008 15th IEEE International Conference on Image Processing*, pages 1808–1811, 2008.

[9] Olaf Ronneberger, Philipp Fischer, and Thomas 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.

[10] KC Santosh and Sameer Antani. Automated chest x-ray screening: Can lung region symmetry help detect pulmonary abnormalities? *IEEE transactions on medical imaging*, 37(5):1168–1177, 2017.

[11] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[12] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[13] Dimpy Varshni, Kartik Thakral, Lucky Agarwal, Rahul Nijhawan, and Ankush Mittal. Pneumonia detection using cnn based feature extraction. In *2019 IEEE international conference on electrical, computer and communication technologies (ICECCT)*, pages 1–7. IEEE, 2019.

[14] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.