Deep learning for COVID-19 lung pathology segmentation

DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology
in
Computer Science

by

Gurdit Singh Bedi

[Roll No: CS-1912]

under the guidance of

Dr. Sushmita Mitra

Professor
Machine Intelligence Unit

Indian Statistical Institute
Kolkata-700108, India

July 2021

CERTIFICATE

This is to certify that the dissertation entitled **Deep learning for COVID-19 lung pathology segmentation** submitted by **Gurdit Singh Bedi 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.

_____

Dr. Sushmita Mitra
Professor,
Machine Intelligence Unit,
Indian Statistical Institute,
Kolkata-700108, India.

Acknowledgments

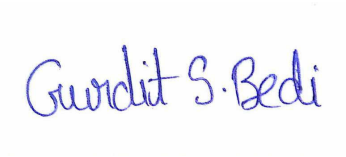
I would like to show my highest gratitude to my advisor, *Prof. Dr. Sushmita Mitra*, Machine Intelligence Unit, Indian Statistical Institute, Kolkata, for his guidance and continuous support and encouragement. He has literally taught me how to do good research, and motivated me with great insights and innovative ideas.

I would also like to thank *Subhashish Bannerjee*, Senior Research Fellow, Machine Intelligence Unit, Indian Statistical Institute, Kolkata, for his valuable suggestions and discussions.

My deepest thanks to all the teachers of Indian Statistical Institute, for their valuable suggestions and discussions which added an important dimension to my research work.

Finally, I am very much thankful to my parents and family for their everlasting supports.

Last but not the least, I would like to thank all of my friends for their help and support. I thank all those, whom I have missed out from the above list.



Gurdit Singh Bedi,
Indian Statistical Institute,
Kolkata - 700108, India.

Abstract

COVID-19 pandemic has impacted billions of lives and created a challenge for the healthcare systems. Detection of pathologies from computed tomography (CT) images offers a great way to assist the traditional healthcare for tackling COVID-19. Pathologies such as ground-glass opacification and consolidations are region of interests which the doctors use to diagnosis the patients. In this work, we have developed and tested various segmentation model using transfer learning to find such pathologies. U-Net [15] is the foundation of the models which we have tested. Along with U-Net we have changed the encoder section of the said model, to various classification models such as VGG, ResNet and MobileNet. As these model have won ImageNet Challenge, there core component have been used for feature extraction and usage of their pretrained weights will help in faster convergence. A small subset of studies which has been annotated with binary pixel masks depicting regions of interests in MosMedData [12] Chest CT Scans dataset have been used to train the segmentation model. The best segmentation model achieved a mean dice score of 0.6029.

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

Contents

Chapter 1

Introduction

Coronavirus disease 2019 (COVID-19) is a contagious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The first known case was identified in Wuhan, China in December 2019. Transmission of COVID-19 occurs when people are exposed to virus-containing respiratory droplets and airborne particles exhaled by an infected person. Those particles may be inhaled or may reach the mouth, nose, or eyes of a person through touching or direct deposition (i.e. being coughed on). Symptoms of COVID-19 are variable, but often include fever, cough, headache, fatigue, breathing difficulties, and loss of smell and taste. Symptoms may begin one to fourteen days after exposure to the virus. Several testing methods have been developed to diagnose the disease. The standard diagnostic method is by detection of the virus' nucleic acid by real-time reverse transcription polymerase chain reaction (rRT-PCR), transcription-mediated amplification (TMA), or by reverse transcription loop-mediated isothermal amplification (RT-LAMP) from a nasopharyngeal swab. Preventive measures include physical or social distancing, quarantining, ventilation of indoor spaces, covering coughs and sneezes, hand washing, and keeping unwashed hands away from the face. The use of face masks or coverings has been recommended in public settings to minimize the risk of transmissions.

1.1 Problem Statement

In this problem we have to build an algorithm which can detect pathologies such as ground-glass opacification and consolidations in lung CT of a human. For this we have used MosMedData [12] Chest CT Scans dataset provided by Research and Practical Clinical Center for Diagnostics and Telemedicine Technologies of the Moscow Health Care Department. The dataset is volumetric in nature. It contains 50 samples for which annotations relating to pathologies are provided. The final model will be able to segment out the pathologies when given an input of a CT scan as input.

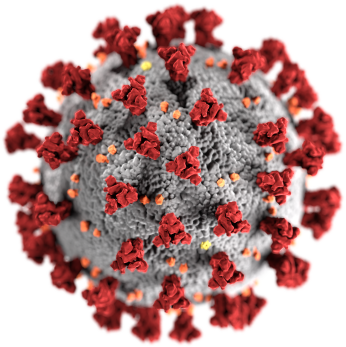
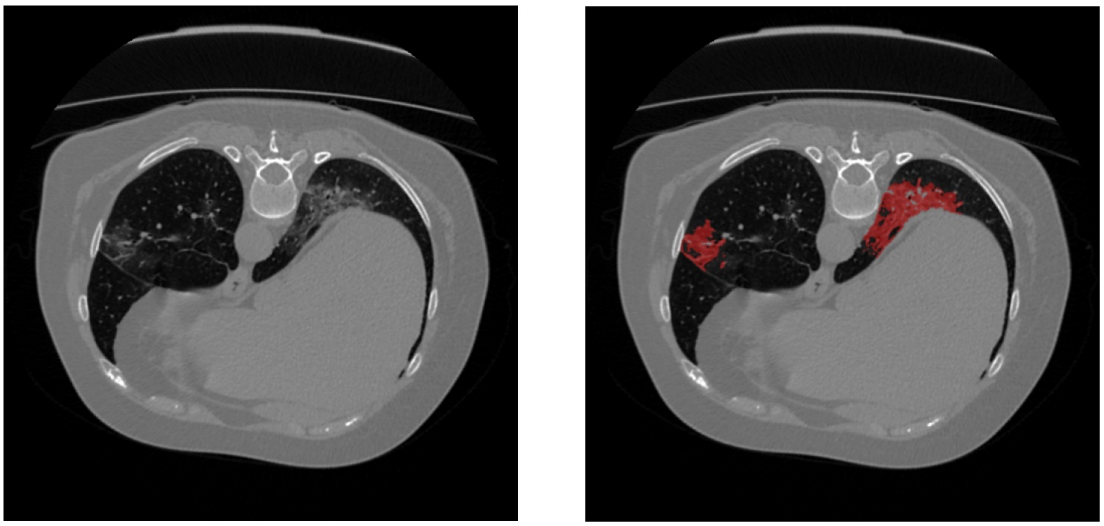


Figure 1.1: An illustration of the virus created at the United States Centers for Disease Control and Prevention (CDC) which reveals ultrastructural morphology exhibited by coronaviruses.



(a) Slice of the CT scan.

(b) Pathologies overlayed on the corresponding slice.

Figure 1.2: Example of the input which the model will get at the time of training.

Chapter 2

Related Work

MosMedData [12]: Chest CT Scans With COVID-19 Related Findings Dataset was submitted in May, 2020. The aim is to segment out the pathologies which are ground-glass opacifications and consolidation. These segmentation findings are great significance for further diagnosis and treatment of COVID-19 patients. U-Net [15] is the most used model when it comes to Medical Image Segmentation. So, it can be observed that most of the related work in relation to this problem is based on U-Net also. In [8], [18], [11], [13] and [4] they have trained U-Net and/or its variations. While [18] have trained a custom U-Net, [11] has expressed the result as a benchmark. [4] have trained 2D U-Net and 3D U-Net both. In MiniSeg [14] they have used hierarchically stacked spatial pyramid of dilated depthwise separable convolutions and feature pooling for lightweight multi-scale learning. They claim that this has made use of comparatively fewer parameters compared to other segmentation models, 83K in this case. In the study of Chen et al., they proposed Residual Attention U-Net [1] for multi-class segmentation. Inf-Net [2] has used parallel partial decoder is used to aggregate the high-level features and generate a global map and uses this global along with reverse attention to make the predictions. They have also presented with a semi-supervised segmentation framework to alleviate the shortage of labeled data. Further, [19] have used conditional generative model, to generate for data and then used this data to train 2D U-Net and 3D U-Net.

Chapter 3

Dataset

The dataset that we have used is MosMedData [12] Chest CT Scans dataset provided by Research and Practical Clinical Center for Diagnostics and Telemedicine Technologies of the Moscow Health Care Department. The dataset is volumetric in nature. In total in contains lung CT for 1110 patients. Each of a sample can belong to any one of the following categories:

1. **CT-0**: Normal lung tissue, no CT-signs of viral pneumonia.
2. **CT-1**: Several ground-glass opacifications, involvement of lung parenchyma is less than 25%.
3. **CT-2**: Ground-glass opacifications, involvement of lung parenchyma is between 25 and 50%.
4. **CT-3**: Ground-glass opacifications and regions of consolidation, involvement of lung parenchyma is between 50 and 75%.
5. **CT-4**: Diffuse ground-glass opacifications and consolidation as well as reticular changes in lungs. Involvement of lung parenchyma exceeds 75%.

Out of 1110 samples, 50 samples studies belonging to class **CT-1** have been annotated by the experts of Research and Practical Clinical Center for Diagnostics and Telemedicine Technologies of the Moscow Health Care Department. During the annotation for every given image ground-glass opacifications and regions of consolidation were selected as positive (white) pixels on the corresponding.

The data is provided in a directory which has further 5 more directories each of a given category. Each file in these categories has been provided in compressed NIfTI-1 format, `nii.gz`. The annotation information of 50 samples is provided in a separate directory.

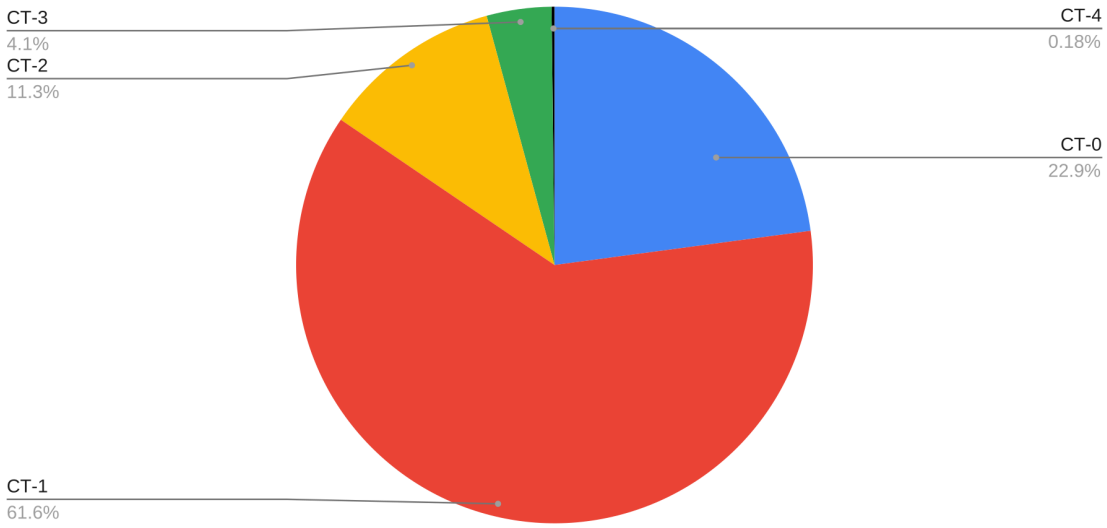


Figure 3.1: Piechart of Data.

3.1 Training Data Distribution

For the segmentation task the data has been divided in 7:1:2 ratio, resulting in 35 samples for training Data, 5 samples for validation data and 10 samples for testing. Since each sample is volumetric, we will be decomposing it in slices. The resulting number of samples are summarized in the Table 3.1:

	#Sample	**#Slices**
Train	35	1428
Validation	5	212
Test	10	409

Chapter 4

Data Preprocessing

Data Preprocessing is that step in which the data gets transformed, or encoded, to bring it to such a state that now the machine can easily parse it. In other words, the features of the data can now be easily interpreted by the algorithm.

4.1 CT Images

The dataset is saved in the form of `NIFTI`(Neuroimaging Informatics Technology Initiative) format. It is format used to save Neuroimaging data. As the Neuroimaging data is volumetric in nature, the authors of the dataset have stored the CT scan data in `NIFTI` format.

4.2 Radiodensity and Hounsfield Scale

Radiodensity is the relative inability of electromagnetic radiation(X-ray and radiowave) to pass through a particular material. The Hounsfield scale is a quantitative scale for describing radiodensity. Water has a value of zero Hounsfield units (HU), tissues denser than water having positive values, and tissues less dense than water having negative values.

4.3 Volumetric data to slices

The model which we have used takes an 2D single channel image. As the data provided is volumetric, each sample is decomposed into slices with the slice plane being coronal.

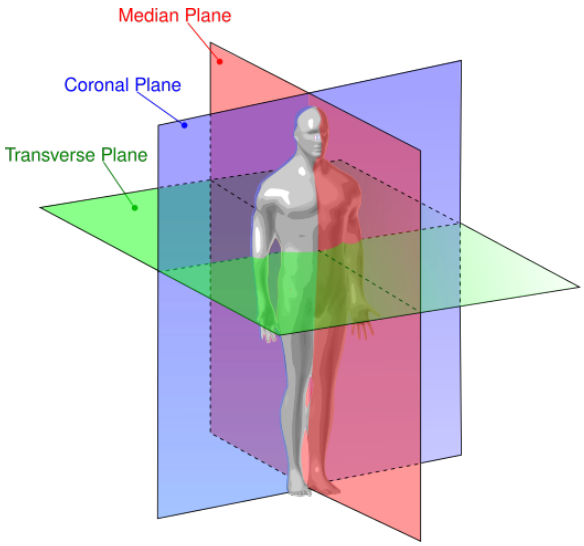


Figure 4.1: Anatomical planes in a human.

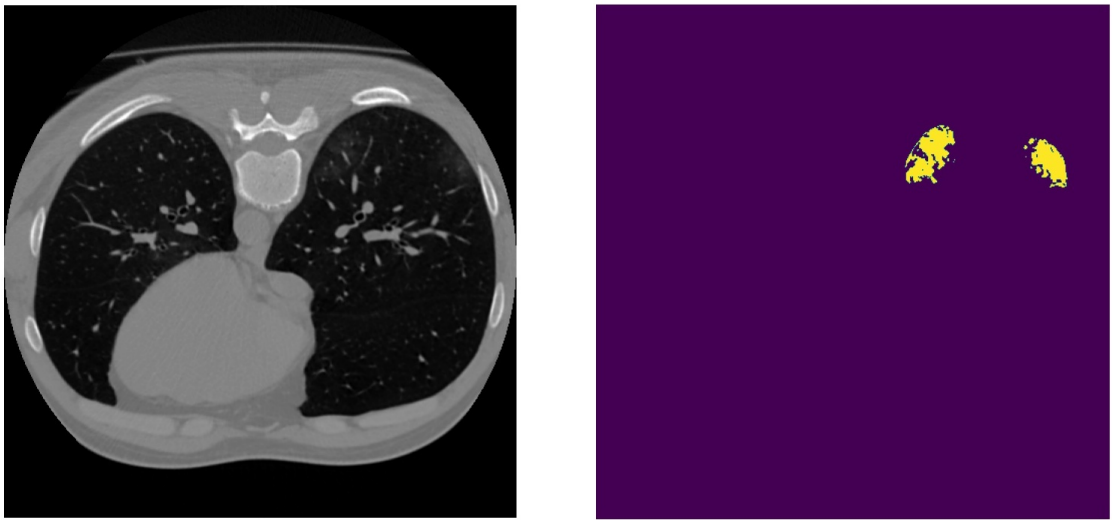
4.4 Data Normalization

In Lecun et al. [9] it has been suggested to normalize the input as it leads to faster convergence. Hence, in this case also the data has been normalized.

4.5 Steps

In summary the following are the data preprocessing steps:

1. Read the input data and mask data from `nii.gz` files.
2. Clip (limit) the hounsfield values in an input data to -1000 to 1000.
3. Divide each value in the input data by 1000. Now each value in the input data is between -1 and 1.
4. Make tuple of input data and mask data, to feed it into the network.



(a) Slice of the CT scan.

(b) Corresponding mask of the slice which marks the pathologies.

Figure 4.2: Example of the input which the model will get at the time of training.

Chapter 5

Image Segmentation

Image segmentation is a process in which each point of an image (2D or 3D) is label to certain category. This category is generally predefined. Image segmentation can be divided into two broad categories:

1. **Semantic Segmentation**: Semantic segmentation is the process of classifying each pixel belonging to a particular label. It doesn't differentiate across different instances of the same object. For example if there are multiple trees in an image, semantic segmentation gives same label to all the pixels of all of the trees, instead of labelling each tree's pixels differently.

2. **Instance Segmentation**: Instance segmentation differs from semantic segmentation in the sense that it gives a unique label to every instance of a particular object in the image.

5.1 Medical Image Segmentation

Medical image segmentation has an essential role in computer-aided diagnosis systems in different applications. The vast investment and development of medical imaging modalities such as microscopy, dermatoscopy, X-ray, ultrasound, computed tomography (CT), magnetic resonance imaging (MRI), and positron emission tomography attract researchers to implement new medical image-processing algorithms. Image segmentation is considered the most essential medical imaging process as it extracts the region of interest (ROI) through a semiautomatic or automatic process. Medical image segmentation is generally semantic in nature. It divides an image into areas based on a specified description, such as segmenting body organs/tissues in the medical applications for border detection, tumor detection and segmentation, and mass detection.

(a) Example of semantic segmentation: a street scene would be segmented by pedestrians, bikes, vehicles, sidewalks, and so on. Here each tree or a vehicle is not treated uniquely.

(b) Example of instance segmentation: Uniquely identifying vehicles which are cars, motorcycles, buses, and so on. Each vehicle is treated uniquely.

Figure 5.1: Visual difference between Semantic Segmentation and Instance Segmentation.

5.2 Loss function for Image Segmentation

The Dice coefficient is widely used metric in computer vision community to calculate the similarity between two images. The Dice score coefficient (DSC) is a measure of overlap widely used to assess segmentation performance when ground truth is available. For a binary segmentation task Dice Loss can be expressed as:

$$DiceLoss(P,T) = 1 - \frac{\sum p_n t_n + \epsilon}{\sum p_n + t_n + \epsilon}$$

where T is the ground truth for the segmentation with values t_n, and P is the prediction by the models having values p_n. The ϵ term is used here to ensure the loss function stability by avoiding the numerical issue of dividing by 0.

5.3 Metrics for Image Segmentation

1. Dice Score: The dice score is related to dice loss in the following way.

$$DiceScore(P,T) = 1 - DiceLoss(P,T) = \frac{\sum p_n t_n + \epsilon}{\sum p_n + t_n + \epsilon}$$

Chapter 6

Medical Image Segmentation using U-Net

U-Net is a convolutional neural network that was developed for biomedical image segmentation at the Computer Science Department of the University of Freiburg [15]. The network is based on the fully convolutional network [10] and its architecture was modified and extended to work with fewer training images and to yield more precise segmentations. U-Net has been prime inspiration for various medical image segmentation networks which came afterwards. The architecture of U-Net is shown in Figure 6.1.

6.1 Architecture

U-Net model has two main parts, encoder and decoder. The encoder reduces the spatial dimensions in every layer and increases the channels. On the other hand, the decoder increases the spatial dims while reducing the channels. The tensor that is passed in the decoder is usually called bottleneck. In the end, the spatial dims are restored to make a prediction for each pixel in the input image.

1. **Encoder**: It consists of the repeated application of two 3×3 convolutions. Each conv is followed by a ReLU and batch normalization. Then a 2×2 max pooling operation is applied to reduce the spatial dimensions. Again, at each downsampling step, we double the number of feature channels, while we cut in half the spatial dimensions.

2. **Decoder**: Every step in the expansive path consists of an upsampling of the feature map followed by a 2×2 transpose convolution, which halves the number of feature channels. We also have a concatenation with the corresponding feature map from the contracting path, and usually a 3×3 convolutional (each followed

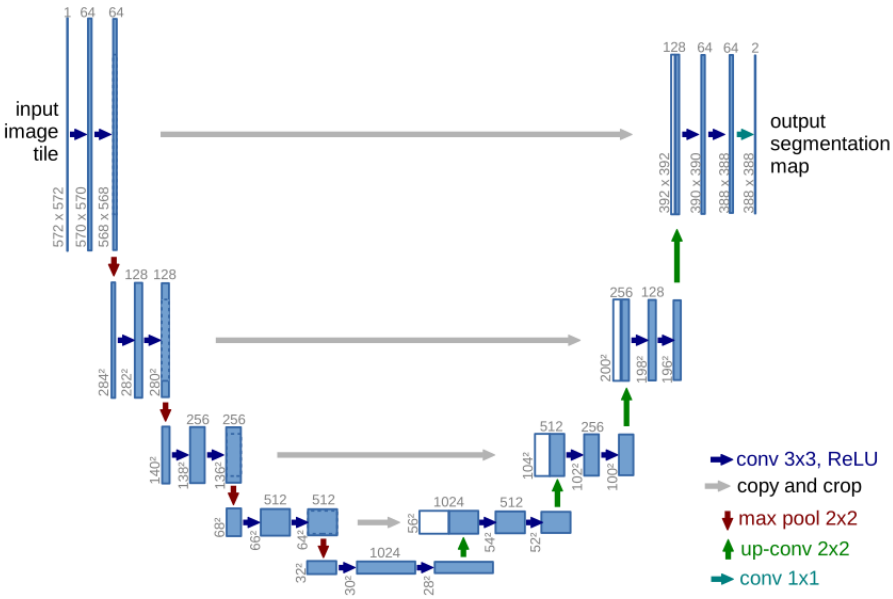


Figure 6.1: U-net architecture. Each blue box corresponds to a multi-channel feature map. The number of channels is denote don top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations. source: [15].

by a ReLU). At the final layer, a 1×1 convolution is used to map the channels to the desired number of classes.

6.2 Training

The training has been done after the preprocesing step as described in Chapter 4. It has been trained using Adam Optimizer with learning rate of 0.0001. The model has been trained for 61 epochs. Early stopping with patience value 5, has also been used to train the network.

6.3 Results

After evaluating this model on the testing set, mean and maximum dice score is 0.602920 and 0.809378 respectively.

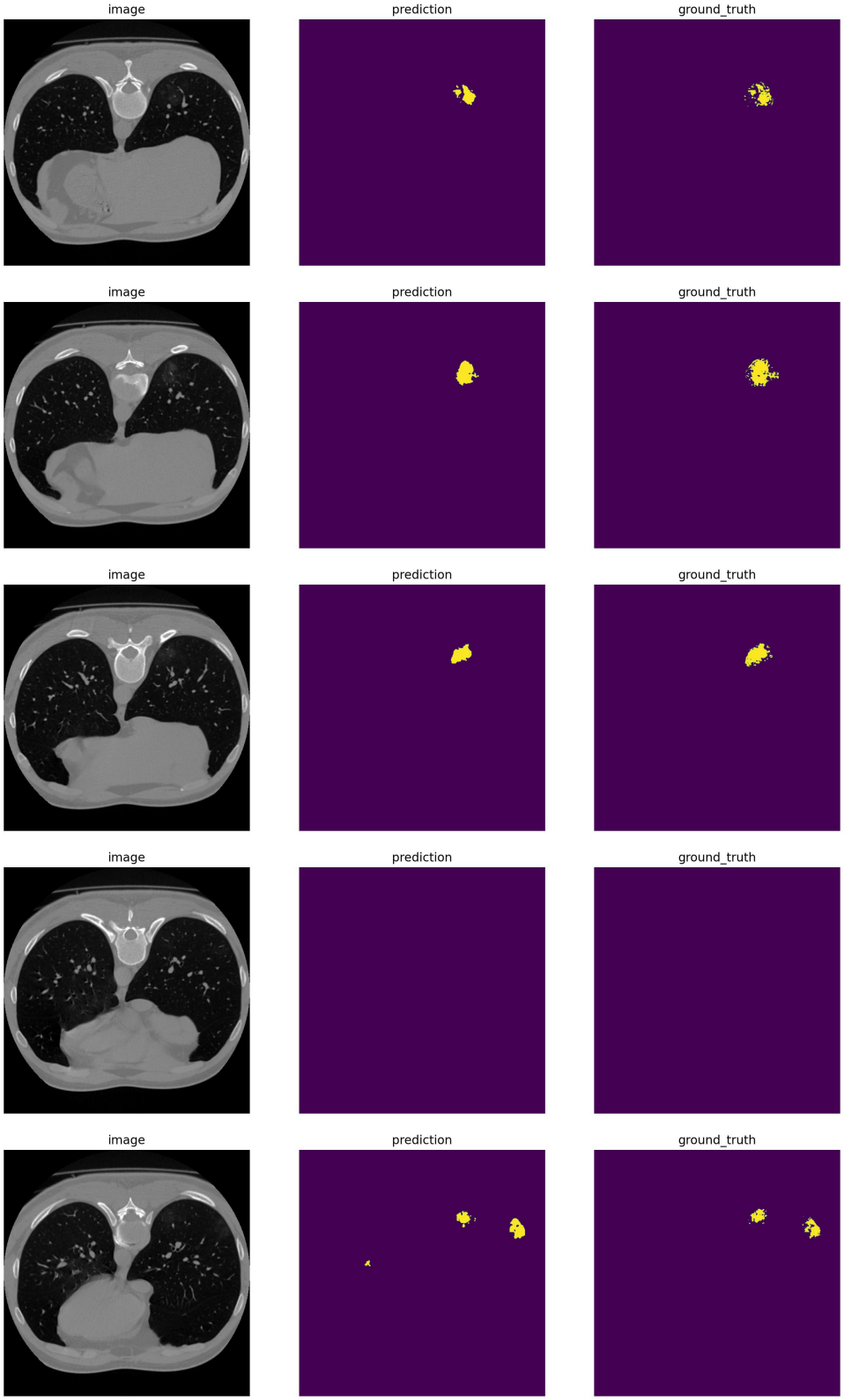


image
prediction
ground_truth
image
prediction
ground_truth
image
prediction
ground_truth
image
prediction
ground_truth
image
prediction
ground_truth

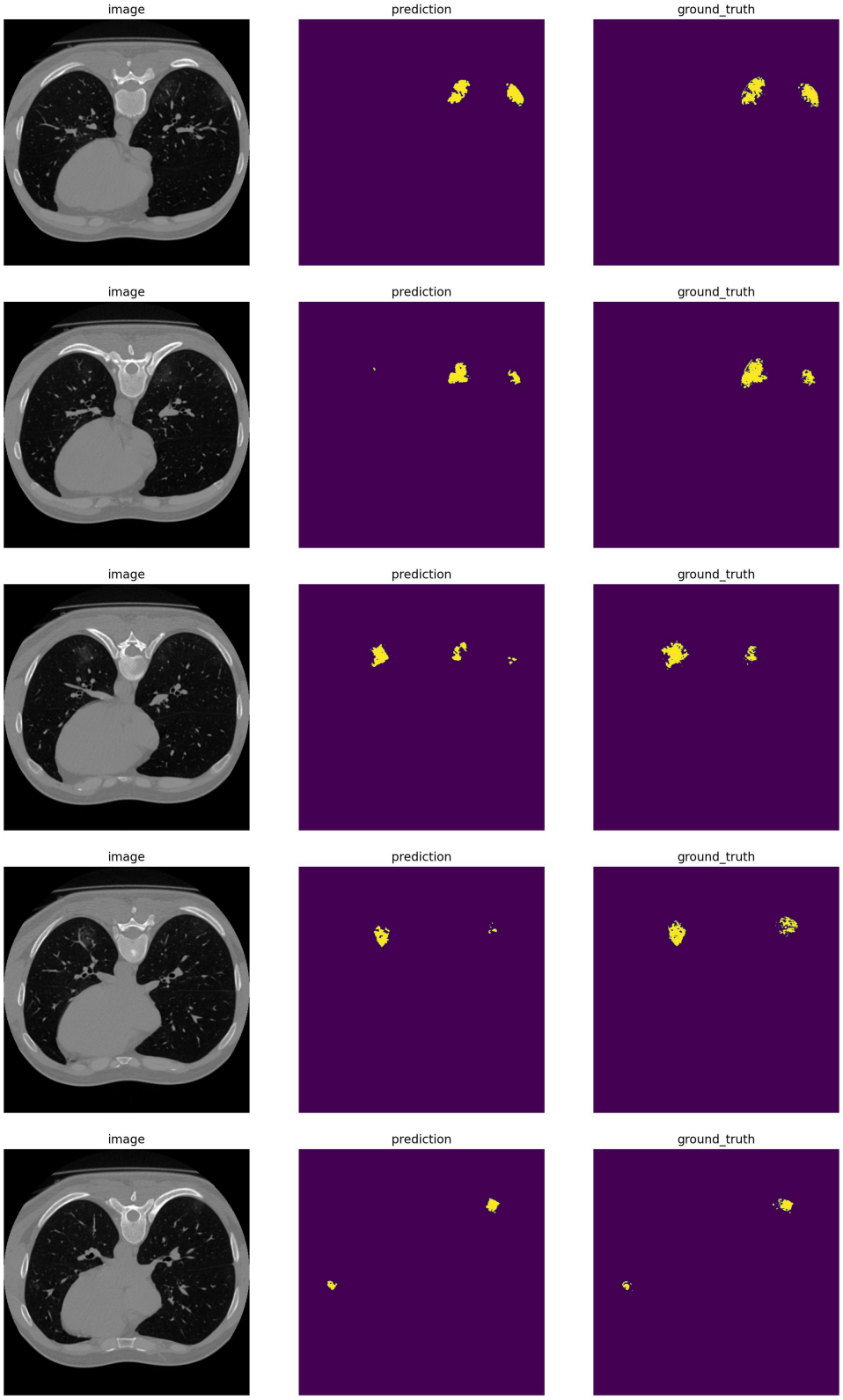


Figure 6.2: Prediction as done by trained U-Net model. The slices number 10 to 19 has been shown, out of 40 slices, of this example. For each row, leftmost image is the CT scan, middle image is the prediction made by the model, and the rightmost image is the ground truth as given in the dataset.

Chapter 7

Medical Image Segmentation using U-Net with VGG19 as encoder

VGG are a series of model proposed by Visual Geometry Group, University of Oxford. VGG19 is a variant of VGG model. ImageNet project is a large visual database designed for use in visual object recognition software research. ImageNet Large Scale Visual Recognition Challenge(ILSVRC) is an annual competition in which a subset of images from ImageNet competition are used and the challenge is to classify images into 1000 categories. VGG model secured the first and the second places in the localization and classification tasks in ImageNet ILSVRC-2014.

7.1 Architecture of VGG19

VGG consists of 19 layers (16 convolution layers + 3 Fully connected layer). The architecture of VGG19 is:

1. Conv (64) repeated twice.
2. MaxPool
3. Conv (128) repeated twice.
4. MaxPool
5. Conv (256) repeated 4 times.
6. MaxPool
7. Conv (512) repeated 4 times.
8. MaxPool

9. Conv (512) repeated 4 times.
10. MaxPool
11. Fully Connected (4096)
12. Fully Connected (4096)
13. Fully Connected (1000)
14. SoftMax

The convolution layer kernel size is 3×3 with stride 1 with appropriate spatial padding to preserve the spatial resolution of the input. The number in bracket beside the conv operation represent the number of output channels from that layer. Each convolution operation is followed by an RELU activation. Max pooling layer kernel size is 2×2 with stride 2.

7.2 U-Net with VGG19 encoder

In this experiment, we have replaced the encoder of the U-Net with the VGG19 model (up till the last convolution layer). As VGG was a model trained on ImageNet challenge it will be able to have better feature extraction.

7.3 Training

The training has been done after the preprocesing step as described in Chapter 4. Model has been trained using Adam Optimizer with initial learning rate of 0.0001. The model has been trained for 32 epochs. Early stopping with patience value 5, has also been used to train the network. The encoder of this model had pretrained weights from ImageNet challenge prior to training.

7.4 Results

After evaluating this model on the testing set, mean and maximum dice score is 0.597698 and 0.815499 respectively.

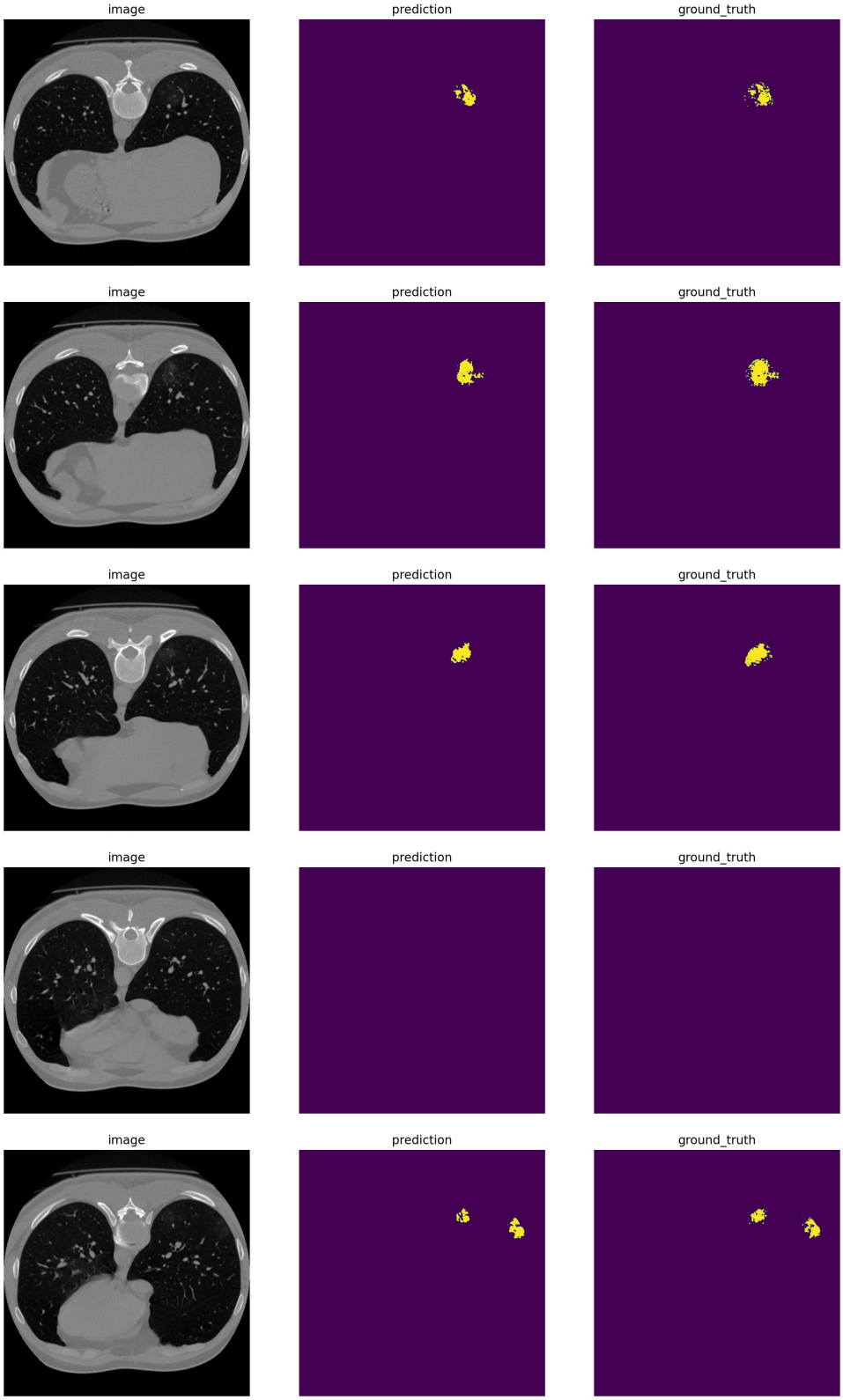


image
prediction
ground_truth
image
prediction
ground_truth
image
prediction
ground_truth
image
prediction
ground_truth
image
prediction
ground_truth

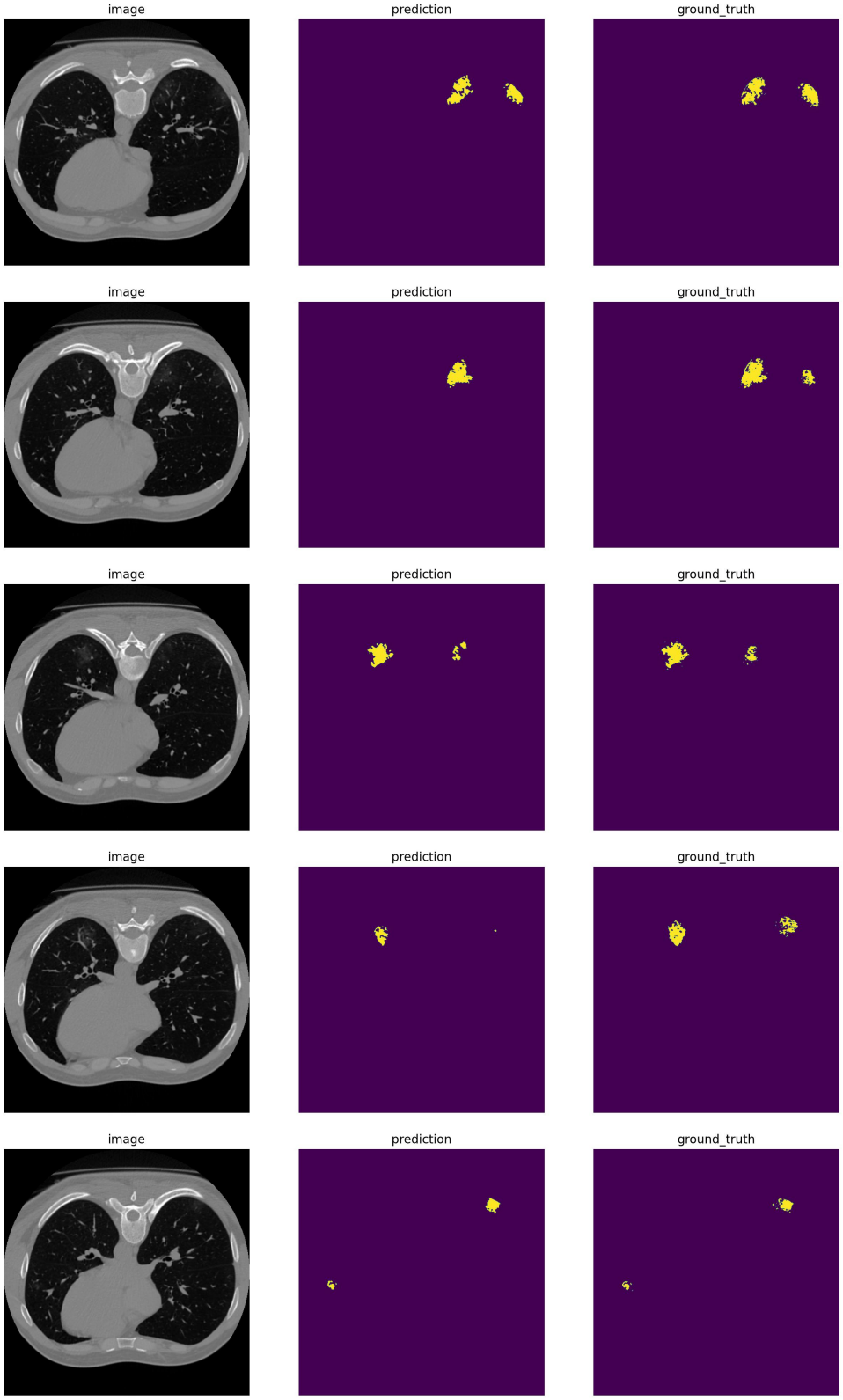


Figure 7.1: Prediction as done by trained U-Net with VGG19 encoder model. The slices number 10 to 19 has been shown, out of 40 slices, of this example. For each row, leftmost image is the CT scan, middle image is the prediction made by the model, and the rightmost image is the ground truth as given in the dataset.

Chapter 8

Medical Image Segmentation using U-Net with Resnet34 as encoder

8.1 Resnet

ResNet is stands for a residual network. They were introduced in 2015 by He et al. [5]. Previous to introduction to ResNet networks didn't used to be that deep, for example VGG-19 [17] had 19 layers. The initial model of ResNet is Resnet34 which as the name suggested had 34 layers. It was observed that training more deeper networks is hard. There are models such as resnet50, resnet101, resnet152 which are far more deeper. With deeper network the problem of vanishing/exploding gradients arises [3], which hamper convergence from the beginning. When deeper networks are able to start converging, a degradation problem has been exposed: with the network depth increasing, accuracy gets saturated and then degrades rapidly.

To overcome this problem, Microsoft [5] introduced a deep residual learning framework. Instead of hoping every few stacked layers directly fit a desired underlying mapping, they explicitly let these layers fit a residual mapping. The formulation of $F(x) + x$ can be realized by feed forward neural networks with shortcut connections. Shortcut connections are those skipping one or more layers shown in Figure 8.1. The shortcut connections perform identity mapping, and their outputs are added to the outputs of the stacked layers. ResNet won the 1st place on the ILSVRC 2015 classification task.

In the experimentation in [5] they have shown:

1. Deep residual nets are easy to optimize, but the counterpart simply stacked network exhibit higher training error when the depth increases.

2. Our deep residual nets can easily enjoy accuracy gains from greatly increased depth, producing results substantially better than previous networks.

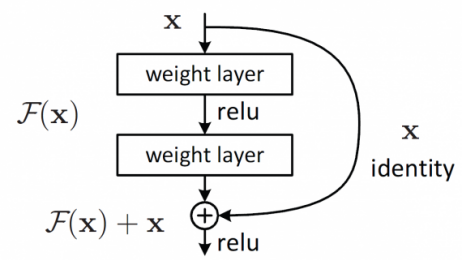


Figure 8.1: Residual learning: a building block.

8.2 Architecture of resnet34

The architecture of Resnet34 is:

1. Convolution layer with kernel size 7×7 and stride 2 with padding 3.
2. Max pooling with kernel size 3×3 and stride 2.
3. Double Convolution Block with number of channels in the output is 64. This is repeated 3 times. Labelled conv2_1, conv2_2, ..., conv2_3.
4. Double Convolution Block with number of channels in the output is 128. This is repeated 4 times. Labelled conv3_1, conv3_2, ..., conv3_4.
5. Double Convolution Block with number of channels in the output is 256. This is repeated 6 times. Labelled conv4_1, conv4_2, ..., conv4_6.
6. Double Convolution Block with number of channels in the output is 512. This is repeated 3 times. Labelled conv5_1, conv5_2, ..., conv5_3.
7. Average Pooling
8. Fully Connected Layer of size 1000 with softmax activation.

Unless specified a convolutional layer kernel size is 3×3 with stride 1 and padding 1. The double convolution block implies two convolution layer one after another. Each convolution layer is followed by batch normalization [7] and RELU activation, except that the second convolution layer in each double convolution block. The first layer conv3_1, conv4_1, and conv5_1 are responsible for down-sampling using convolution with a stride of 2. Residual shortcut connections are formed by using the following rules:

1. The identity shortcuts can be directly used when the input and output are of the same dimensions (solid line shortcuts in Fig 8.2).

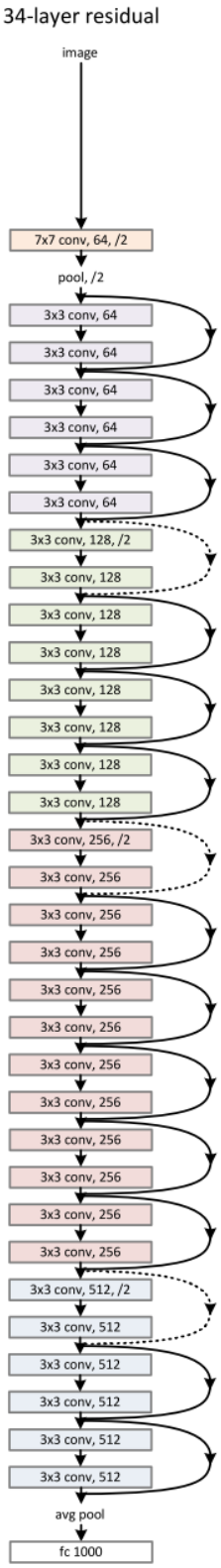


Figure 8.2: Resnet34 architecture.

2. When the dimensions increase (dotted line shortcuts in Fig. 8.2), 1x1 convolutions are done to match the dimensions. For both options, when the shortcuts go across feature maps of two sizes, they are performed with a stride of 2.

8.3 U-Net with resnet34 encoder

In this experiment, we have replaced the encoder of the U-Net with the Resnet34 model (up till the average pooling layer). It has been done in the spirit that as Resnet is a deeper model and it will be able to have better feature extraction as compared to its predecessors.

8.4 Training

The training has been done after the preprocesing step as described in Chapter 4. Model has been trained using Adam Optimizer with initial learning rate of 0.0001. The model has been trained for 22 epochs. Early stopping with patience value 5, has also been used to train the network. The encoder of this model had pretrained weights from ImageNet challenge prior to training.

8.5 Results

After evaluating this model on the testing set, mean and maximum dice score is 0.592900 and 0.803339 respectively.

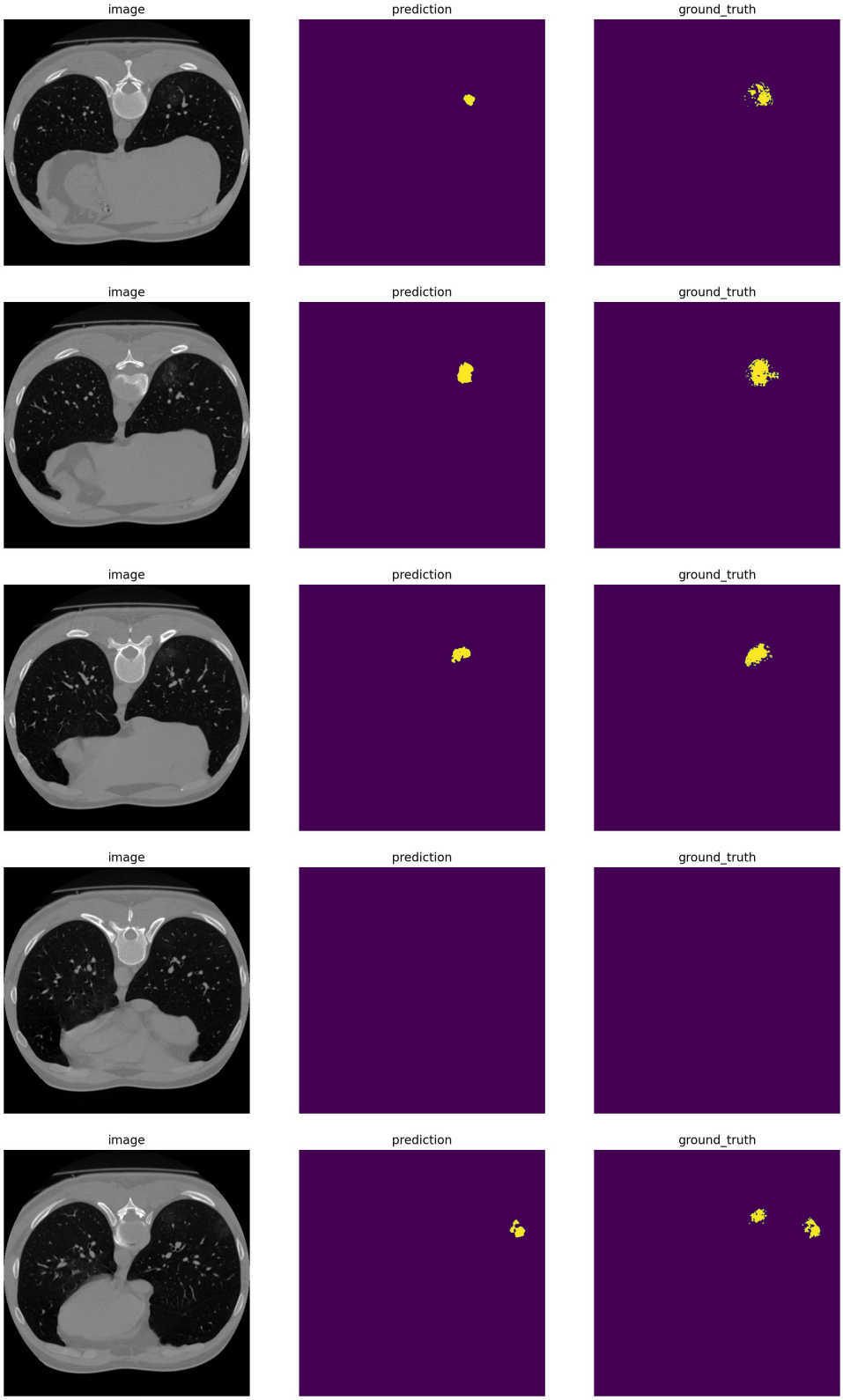


image
prediction
ground_truth
image
prediction
ground_truth
image
prediction
ground_truth
image
prediction
ground_truth
image
prediction
ground_truth

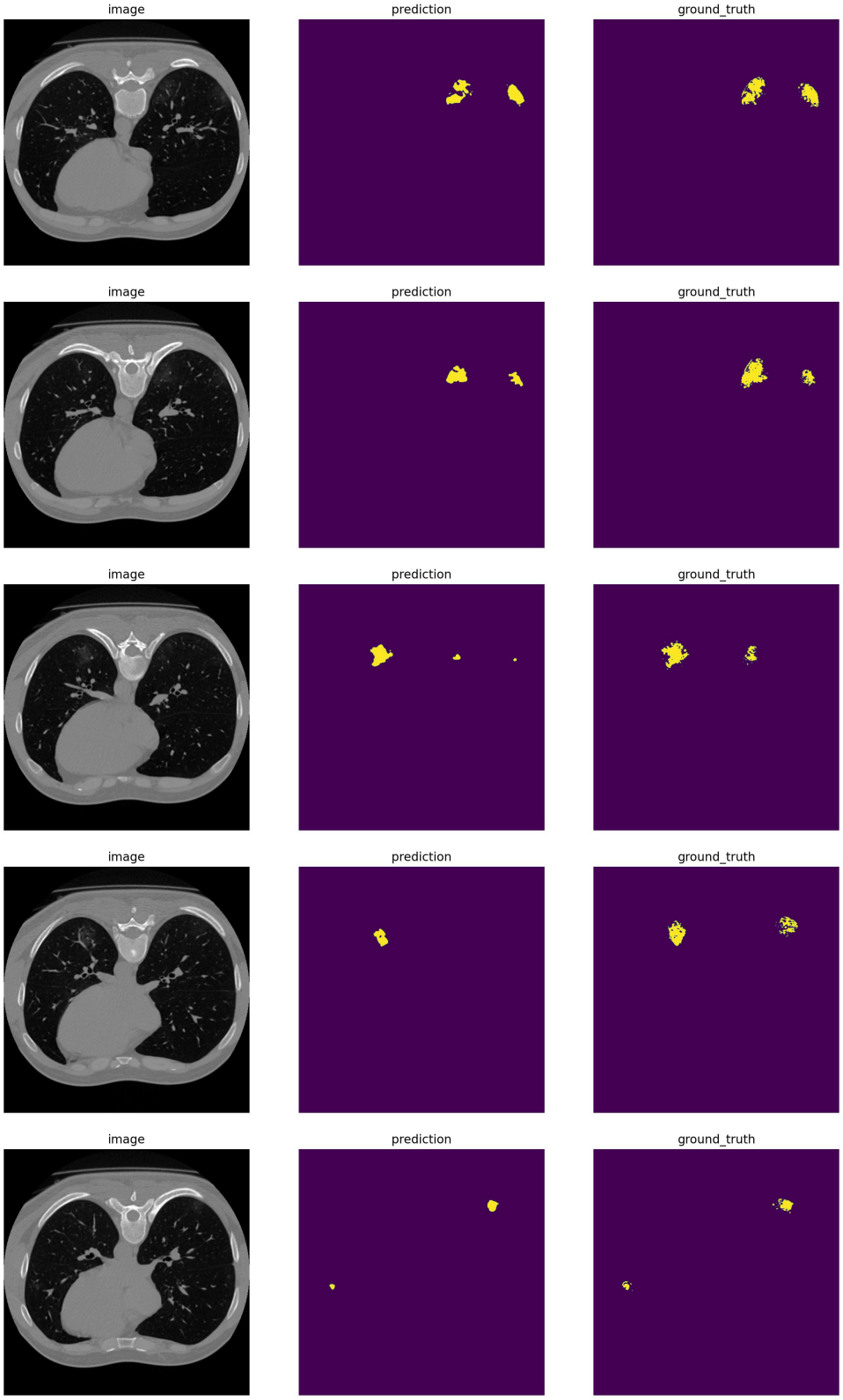


Figure 8.3: Prediction as done by trained U-Net with Resnet34 encoder model. The slices number 10 to 19 has been shown, out of 40 slices, of this example. For each row, leftmost image is the CT scan, middle image is the prediction made by the model, and the rightmost image is the ground truth as given in the dataset.

Chapter 9

Medical Image Segmentation using U-Net with MobileNetV2 as encoder

MobileNetV2 [16] is model architecture refined from MobileNets [6] also called MobileNetV1. The core idea of MobileNetV1 is that convolutional layers, which are essential to computer vision tasks but are quite expensive to compute, can be replaced by so-called depthwise separable convolutions. This will lead to development of lightweight model.

9.1 Depthwise separable convolution - Building block of MobileNetV1

Depthwise separable convolutions is the core building block of the MobileNetV1 [6]. Recall a regular convolutional layer applies a convolution kernel to all of the channels of the input image. It slides this kernel across the image and at each step performs a weighted sum of the input pixels covered by the kernel across all input channels. But in depthwise separable convolutions, the kernel are applied on per channel basis i.e. after this convolution operation the number of the channels will remain same. The depthwise convolution is followed by a pointwise convolution which is the same as a regular convolution but with a 1×1 kernel.

To analysis the difference in the computation cost, consider the $D_F \times D_F \times M$ input F is mapped to $D_G \times D_G \times N$ output G where F is the number of channels of the input which has height and width of D_F, N is the number of channels of the output which has height and width of D_G. Also assume, standard convolution operation is of size $D_K \times D_K \times M \times N$, where D_K is the spatial dimension of the kernel and has stride one and zero padding. The computation cost will be of the standard convolution operation will be,

$$N \times (D_G \times D_G \times (D_k \times D_k \times M))$$

In case of Depthwise convolution having M filters will have computational cost of

$$M \times (D_G \times D_G \times (D_k \times D_k))$$

and case of pointwise convolution which results in an N channel output will have computational cost of

$$N \times (D_G \times D_G) \times M$$

So, combining the above result we get the, the reduction in the computation cost is:

$$= \frac{M \times (D_G \times D_G \times (D_k \times D_k)) + N \times (D_G \times D_G) \times M}{N \times (D_G \times D_G \times (D_k \times D_k \times M))}$$
$$= \frac{1}{N} + \frac{1}{D_k^2}$$

9.2 Bottleneck residual block - Building block of MobileNetV2

Bottleneck residual block is build upon this depthwise separable convolution. In this block there are 3 major components, first is "expansion" layer, which is similar to pointwise layer but the number of output channels are to be greater than the number of input channels. Second layer, is the same depthwise convolution as described in the preceding para. The third and last layer is "projection" layer which is similar to pointwise layer but the number of output channels are to be less than the number of input channels in this case. So, the block first expands the number of channels using pointwise convolution followed by depthwise convolution and then finally reduces the number of channels using pointwise convolution again. The expansion in the block is governed by a hyperparameter called expansion factor. It has been chosen to be 6. In cases, when the number of input channels are equal to the number of output channel of the block, there is a residual connection similar to ResNet [5]. Also, each convolutional layer is followed by batch normalization.

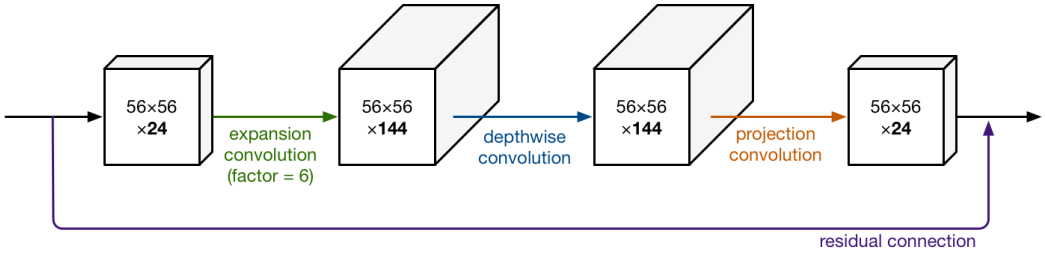


Figure 9.1: Bottleneck residual block.

9.3 Architecture of MobileNetv2

The architecture is described in Table 9.2. Each line in the table represents a sequence of identical layers(stride differs) repeated n times. The number of output channels is fixed for a sequence. All the layers of a sequence have stride 1 except the first layer in the sequence which has a stride s. All the spatial convolution use 3×3 kernels. The expansion layers use the expansion layer factor t.

Operator	t	c	n	s
conv2d	-	32	1	2
bottleneck	1	16	1	1
bottleneck	6	24	2	2
bottleneck	6	32	3	2
bottleneck	6	64	4	2
bottleneck	6	96	3	1
bottleneck	6	160	3	2
bottleneck	6	320	1	1
conv2d 1x1	-	1280	1	1
avgpool 7x7	-	-	1	-
conv2d 1x1	-	k	-	

Table 9.2: Architecture of MobileNetV2.

9.4 U-Net with MobileNetv2 encoder

In this experiment, we have replaced the encoder of the U-Net with the Resnet34 model (up till the fully connected layer). The last two layers avgpool and conv2d 1x1 have been removed and thus the last layer left is connected to the decoder.

9.5 Training

The training has been done after the preprocesing step as described in Chapter 4. Model has been trained using Adam Optimizer with initial learning rate of 0.0001. The model has been trained for 29 epochs. Early stopping with patience value 5, has also been used to train the network. The encoder of this model had pretrained weights from ImageNet challenge prior to training.

9.6 Results

After evaluating this model on the testing set, mean and maximum dice score is 0.539929 and 0.776971 respectively.

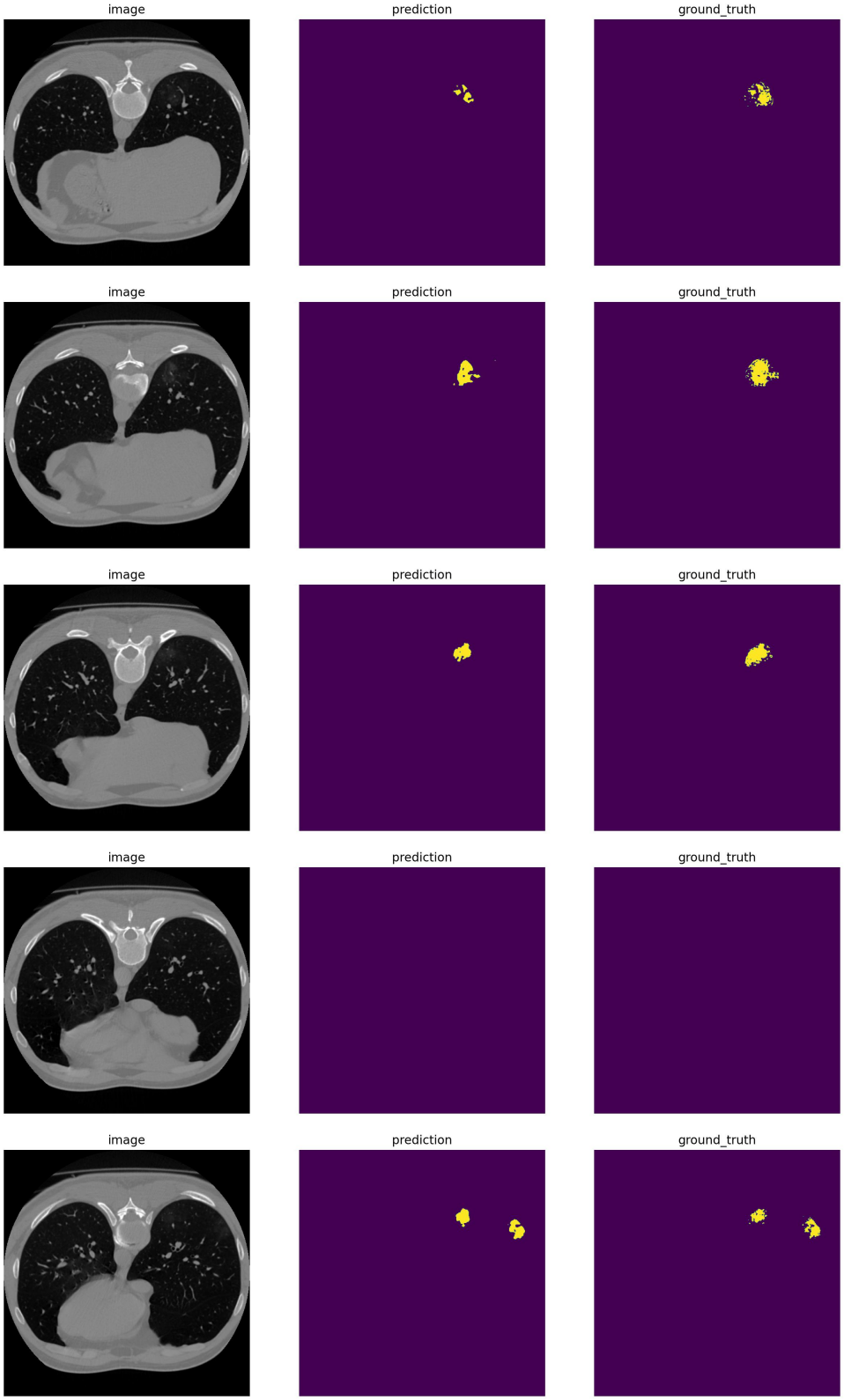


image
prediction
ground_truth
image
prediction
ground_truth
image
prediction
ground_truth
image
prediction
ground_truth
image
prediction
ground_truth

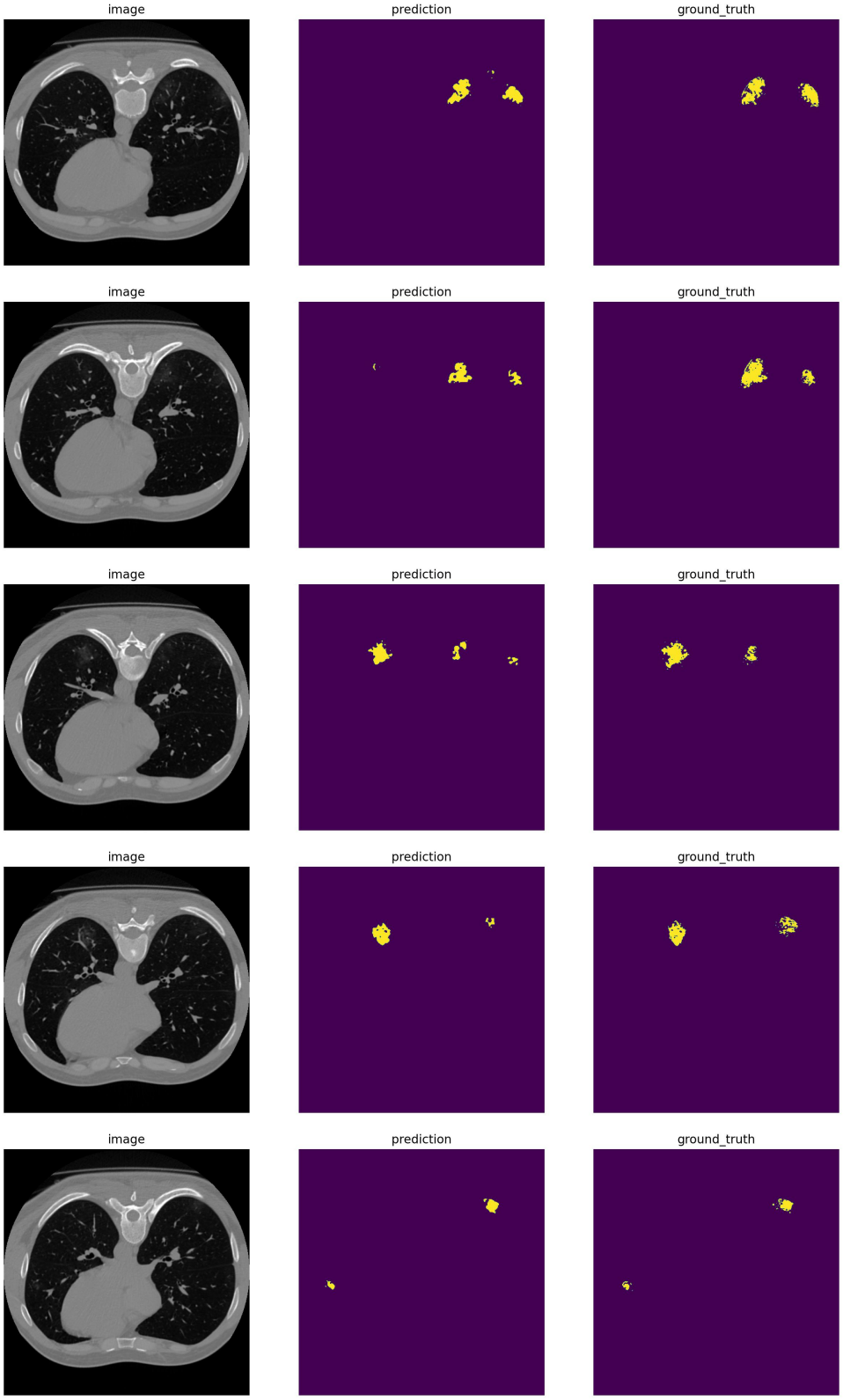


Figure 9.3: Prediction as done by trained U-Net with MobileNetV2 encoder model. The slices number 10 to 19 has been shown, out of 40 slices, of this example. For each row, leftmost image is the CT scan, middle image is the prediction made by the model, and the rightmost image is the ground truth as given in the dataset.

Chapter 10

Comparison of the proposed models

Model	**Dice Score**	
	Mean	Max
U-Net	0.602920	0.809378
U-Net with vgg19 encoder	0.597698	0.815499
U-Net with resnet34 encoder	0.592900	0.803339
U-Net with mobilenet encoder	0.539929	0.776971

Some specific slices from test set are shown. Slices are chosen in such a way that various positives and negatives cases can be covered.

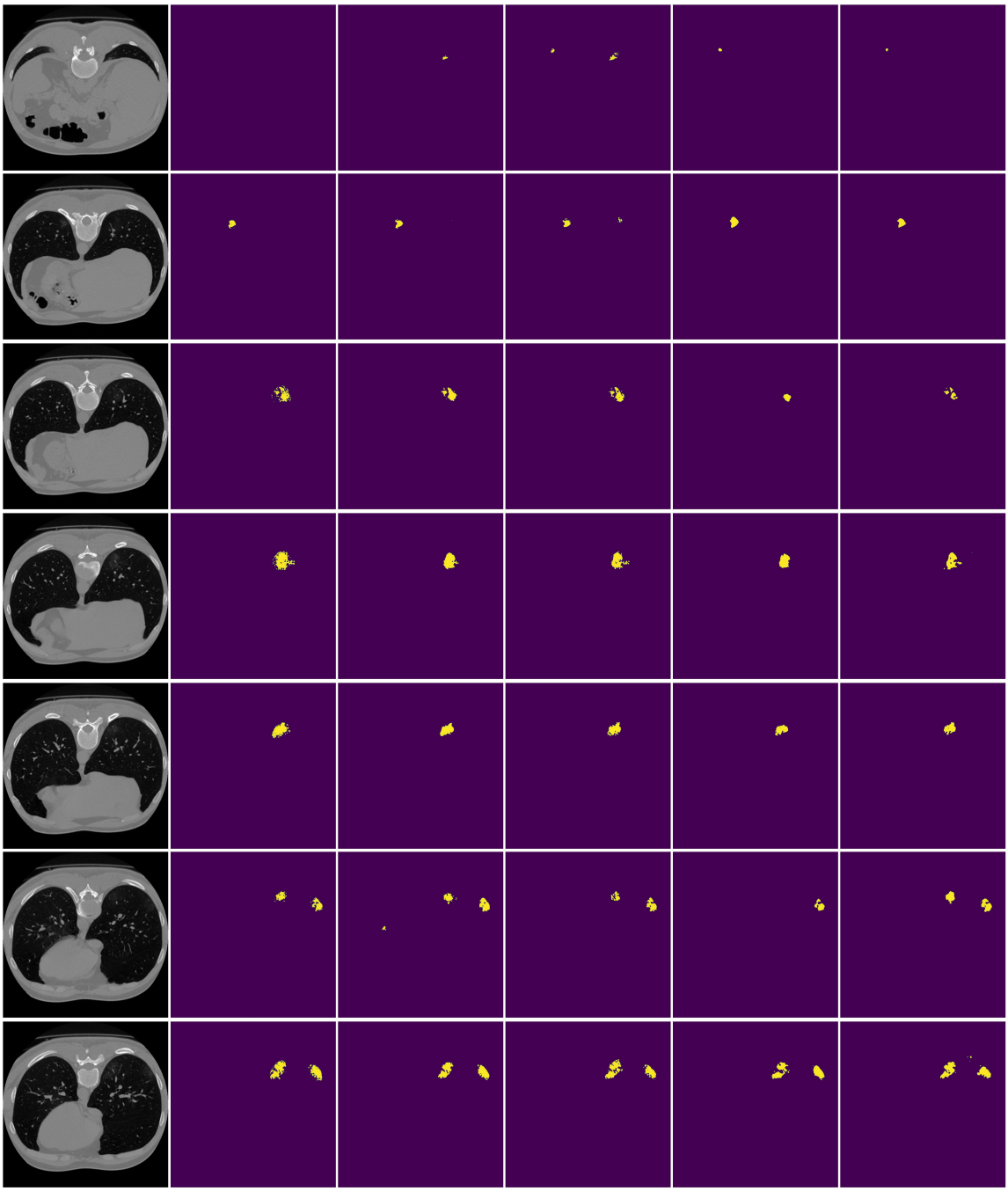


Figure 10.1: The slices numbered 2, 9, 10, 11, 12, 14, 15 has been shown. The images in each row from left to right are CT scan, ground truth, U-Net prediction, U-Net with VGG19 prediction, U-Net with Resnet34 prediction, U-Net with MobileNetV2 prediction.

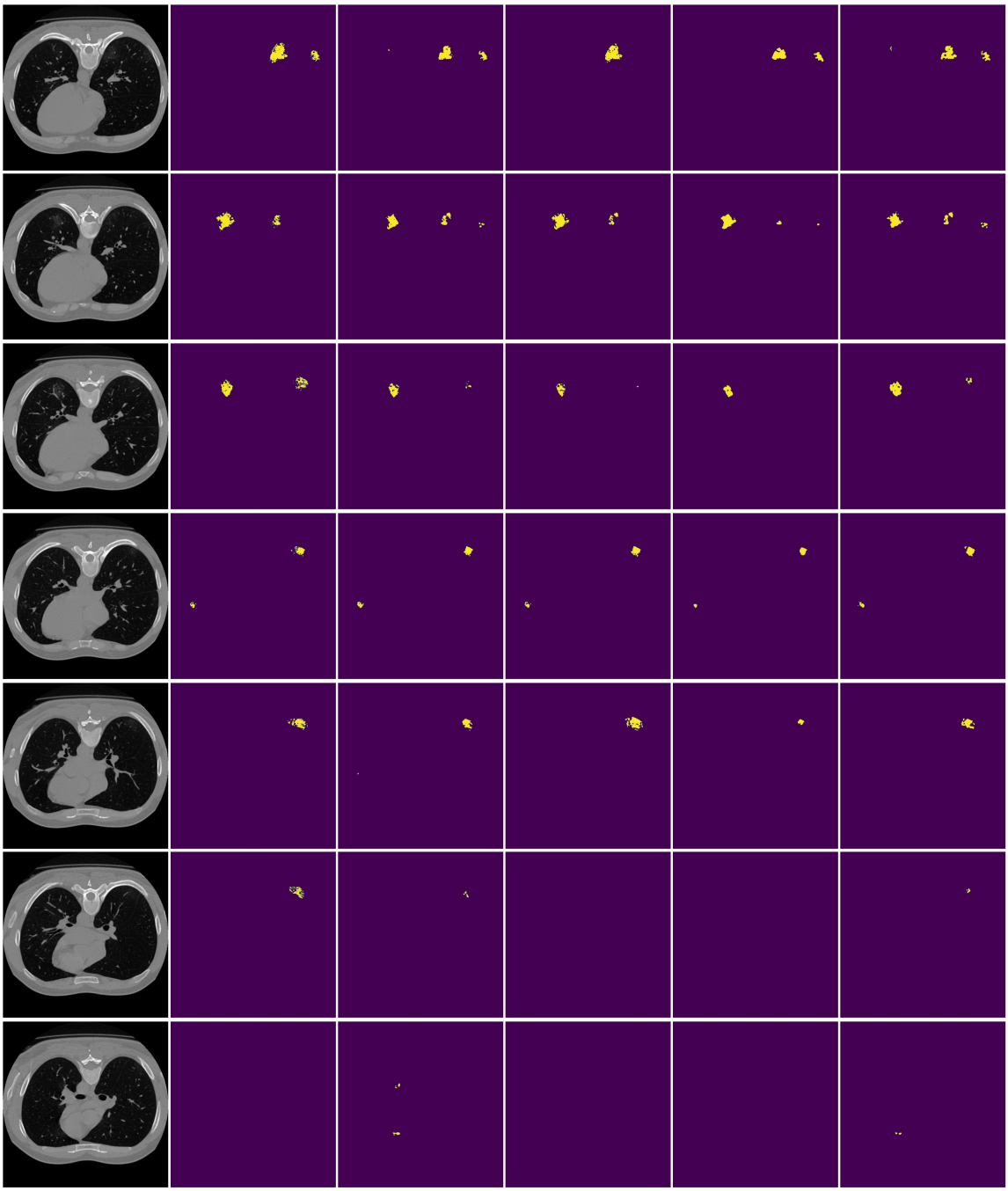


Figure 10.2: The slices numbered 16, 17, 18, 19, 21, 22, 24 has been shown. The images in each row from left to right are CT scan, ground truth, U-Net prediction, U-Net with VGG19 prediction, U-Net with Resnet34 prediction, U-Net with MobileNetV2 prediction.

Bibliography

[1] Xiaocong Chen, Lina Yao, and Yu Zhang. *Residual Attention U-Net for Automated Multi-Class Segmentation of COVID-19 Chest CT Images*. 2020. arXiv: 2004.05645 [eess.IV].

[2] Deng-Ping Fan 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. DOI: 10.1109/TMI.2020.2996645.

[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*. JMLR Workshop and Conference Proceedings. 2010, pp. 249–256.

[4] Mikhail Goncharov et al. ?CT-Based COVID-19 triage: Deep multitask learning improves joint identification and severity quantification? In: *Medical Image Analysis* 71 (2021), p. 102054. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2021.102054. URL: https://www.sciencedirect.com/science/article/pii/S1361841521001006.

[5] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[6] Andrew G. Howard et al. *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. 2017. arXiv: 1704.04861 [cs.CV].

[7] Sergey Ioffe and Christian Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015. arXiv: 1502.03167 [cs.LG].

[8] Cheng Jin et al. *Development and evaluation of an artificial intelligence system for COVID-19 diagnosis*. Oct. 2020. DOI: 10.1038/s41467-020-18685-1. URL: https://doi.org/10.1038/s41467-020-18685-1.

[9] Yann Lecun et al. *Efficient BackProp*.

[10] Jonathan Long, Evan Shelhamer, and Trevor Darrell. ?Fully Convolutional Networks for Semantic Segmentation? In: *CoRR* abs/1411.4038 (2014). arXiv: 1411.4038. URL: http://arxiv.org/abs/1411.4038.

[11] Jun Ma et al. **?**Toward data-efficient learning: A benchmark for COVID-19 CT lung and infection segmentation**?** In: *Medical Physics* 48.3 (2021), pp. 1197–1210. DOI: https://doi.org/10.1002/mp.14676. eprint: https://aapm.onlinelibrary.wiley.com/doi/pdf/10.1002/mp.14676. URL: https://aapm.onlinelibrary.wiley.com/doi/abs/10.1002/mp.14676.

[12] S. P. Morozov et al. *MosMedData: Chest CT Scans With COVID-19 Related Findings Dataset*. 2020. arXiv: 2005.06465 [cs.CY].

[13] Adel Oulefki et al. **?**Automatic COVID-19 lung infected region segmentation and measurement using CT-scans images**?** In: *Pattern Recognition* 114 (2021), p. 107747. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2020.107747. URL: https://www.sciencedirect.com/science/article/pii/S0031320320305501.

[14] Yu Qiu et al. *MiniSeg: An Extremely Minimum Network for Efficient COVID-19 Segmentation*. 2021. arXiv: 2004.09750 [cs.CV].

[15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. **?**U-Net: Convolutional Networks for Biomedical Image Segmentation**?** In: *CoRR* abs/1505.04597 (2015). arXiv: 1505.04597. URL: http://arxiv.org/abs/1505.04597.

[16] Mark Sandler et al. *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. 2019. arXiv: 1801.04381 [cs.CV].

[17] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV].

[18] Lucas O. Teixeira et al. *Impact of lung segmentation on the diagnosis and explanation of COVID-19 in chest X-ray images*. 2021. arXiv: 2009.09780 [eess.IV].

[19] Pengyi Zhang et al. *CoSinGAN: Learning COVID-19 Infection Segmentation from a Single Radiological Image*. 2020.