2022 IEEE 9th International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA) | 978-1-6654-3445-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/CIVEMSA53371.2022.9853695

Covid-19 Lung Segmentation using U-Net CNN based on Computed Tomography Image

FX Ferdinandus
Dept. of Electrical Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia

Dept. of Informatics Engineering
Institut Sains dan Teknologi Terpadu Surabaya
Surabaya, Indonesia
ferdinandus.207022@mhs.its.ac.id

Eko Mulyanto Yuniarno
Dept. of Electrical Engineering
Dept. of Computer Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
ekomulyanto@ee.its.ac.id

I Ketut Eddy Purnama
Dept. of Electrical Engineering
Dept. of Computer Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
ketut@te.its.ac.id

Mauridhi Hery Purnomo
Dept. of Electrical Engineering
Dept. of Computer Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia

University Center of Excellence on Artificial Intelligence for Healthcare and Society (UCE AIHeS)
Surabaya, Indonesia
hery@ee.its.ac.id

Abstract ~ In medical image analysis, lung segmentation is needed as an initial step in diagnosing various diseases in the lung area, including Covid-19 infection. Deep Learning has been used for image segmentation in recent years. One of the Deep Learning-based architectures widely used in medical image segmentation is U-Net CNN. U-Net employs a semantic segmentation approach, which has the benefit of being accurate in segmenting even though the model is trained on a limited quantity of data. Our work intends to assist radiologists in providing a more detailed visualization of COVID-19 infection on CT scans, including infection categories and lung conditions. We conduct preliminary work to segment lung regions using U-Net CNN. The dataset used is relatively small, consisting of 267 CT-scan images split into 240 (90%) images for training and 27 (10%) images for testing. The model is evaluated using the K-fold cross-validation (k=10) approach, which has been believed to be appropriate for models created with limited training data. The metric used for experiments is Mean-IoU. It is commonly used in evaluating the segmentation processes. The results achieved were satisfactory, with Mean-IoU scores ranging from 90.2% to 95.3% in each test phase (k1 – k10), with an average value of 93.3%.

Keywords: Semantic segmentation, Lung segmentation, Deep Learning, U-Net, Covid-19.

I. Introduction

Segmentation is a technique used in image processing to separate the desired image area from its background. In medical image analysis, segmentation of Computed Tomography (CT) images may aid in the diagnosis of various diseases, including those of the lungs. Typically, lung segmentation is the initial stage followed by subsequent stage segmentation for diagnosing lung diseases such as inflammation (pneumonia), tumors, cancer, and covid-19.

Image segmentation may be accomplished in a variety of methods. Initially, segmentation was done using various conventional algorithms. However, with the advent of deep learning, which can acquire features automatically through the convolution process, segmentation research is mainly done using Convolutional Neural Network (CNN). One of the interesting and rapidly developing field is semantic segmentation which assigns a class label to each image pixel. The advancement of semantic segmentation has attracted the attention of several researchers to conduct reviews using both traditional methods and those based on Deep Learning or Deep Neural Networks (DNN). There are eight essential aspects of DNN-based segmentation proposed [1]. Similarly, a review was conducted by [2] on various semantic segmentation architectures for 2D images based on deep learning, such as the FCN architecture and its variants. In particular, lung segmentation using Deep Learning is also an exciting and prominent topic in recent years. Researchers [3] used the CNN Encoder-Decoder method with a four-layer encoder and four-layer decoder in segmenting the lungs. Segmentation is used to assist in the diagnosis of tuberculosis.

One widely used architecture in the medical field is U-Net, which was developed by Olaf Ronneberger et al. [4] to segment biomedical images. U-Net is a Fully Convolutional Network (FCN) architecture that uses a semantic segmentation approach. Unlike Deep Learning in general, this U-Net architecture has been proven to segment accurately and quickly even with a limited amount of training data. This factor is essential since it is rather difficult to get huge volumes of training data images in the medical field.

The use of U-Net CNN for lung segmentation from CT-scan images with the ultimate goal of detecting tumors and lung cancer has been demonstrated by [5]. The results achieved were measured using the Dice score metric of 0.95. Another researcher [6] performed lung segmentation using CNN's DeepLab Architecture with a dataset of 1736 X-Ray images. The results were measured using the Mean-IoU score metric with 94.9% for two segmentation classes.

A comparison of lung segmentation using two models, namely CNN and U-Net, was carried out by [7] using a dataset of 267 CT-scan images. The image size is resized from 128x128 to 32x32 and augmented (rotation). The results obtained show the following comparison if using CNN, it gives a Dice score of 81.34%, while using U-Net gives a Dice score of 82.61%. Unfortunately, the resulting segmentation image has a low resolution. Similarly, [8] used two architectures, namely U-Net and Fully Convolutional Neural Network (FCN), to segment Covid-19 infections in the lung area. The dataset used is in the form of CT-scan images from

Radiopaedia with 939 images of 630x630 pixels. Despite limited resources, the proposed models achieved segmentation of COVID-19 infections with good results.

Covid-19 infection is a disease that attacks the lung area. Radiological examinations revelated that the majority of COVID-19 patients had respiratory problems ranging from mild to severe. In the Journal of Radiology and Covid-19 [10], it was stated that the effects of Covid-19 infection were swift, and in a couple of days, it could change the appearance of the CT-Scan image of the patient's lung area. The effects of infection are classified up to 4 levels: Ground-glass opacity (GGO), Consolidation, Peripheral reticulation, and Crazy-paving pattern.

The diagnosis of patients infected with COVID-19 cannot be made only on the basis of ordinary observations on CT-scan images. Additional visualization is required to clarify conditions in the lung area. Meanwhile, an accurate initial phase of lung segmentation is needed since the image of lungs' form will be used in the following segmentation process to provide a visualization of the covid-19 infection in the lung area. Thus, this study aimed to segment the lungs as a first step to achieve the next goal of segmentation and visualization of covid-19 infection. The segmentation will use U-Net CNN architecture, which is accurate in segmenting even with sparse training data. The lung segmentation results will be evaluated using the k-fold cross-validation method, which is also suitable for training models with small datasets. In our contribution, we built a patient lung segmentation model based on Deep Learning through U-Net CNN architecture. We completed the first stage of our work to segment covid-19 infection in the patient's lungs.

The discussion of this paper is divided into five parts. The initial section provided an introduction, and the second part is on U-Net architecture and lung segmentation. In the third section, experiments and results were explained. The fourth part is discussion and closed with a conclusion.

II. LUNG SEGMENTATION AND METHODOLOGY

Instead of using conventional methods to simplify and speed up the segmentation process, we use an intelligent computing-based approach with Convolution Neural Network. Our motivation leads to using the U-Net CNN architecture, which has proven to be well used in various implementations of medical image analysis.

The U-Net architecture has two major parts and one specialized component. The two major sections are the contracting path section (left side) which works like an encoder, and the expanding path section, which works like a decoder (right side). These two sides form the letter "U", which is called the U-Net. Fig. 1 is a visualization of the U-Net architecture.

These two sections consist of several levels. The contracting path will accept the input image with the specified dimensions (height, weight, channels) and for each subsequent level will reduce the size (height and weight) while multiplying the number of features (channels). In contrast, the expanding path will accept the features of the contracting path and attempt to return the features to their original size. Specific components of the U-Net are depicted as green arrows in Fig.1. This component is called a skip-connection because it skips the path of the network. The operation performed is concatenate or connect the output of the contracting path as input to the expanding path. With this skip connection, it is expected that the spatial information from the original input image that may be lost when the spatial size reduction occurs during feature extraction can appear and be visible in the segmentation output at the end of the model.

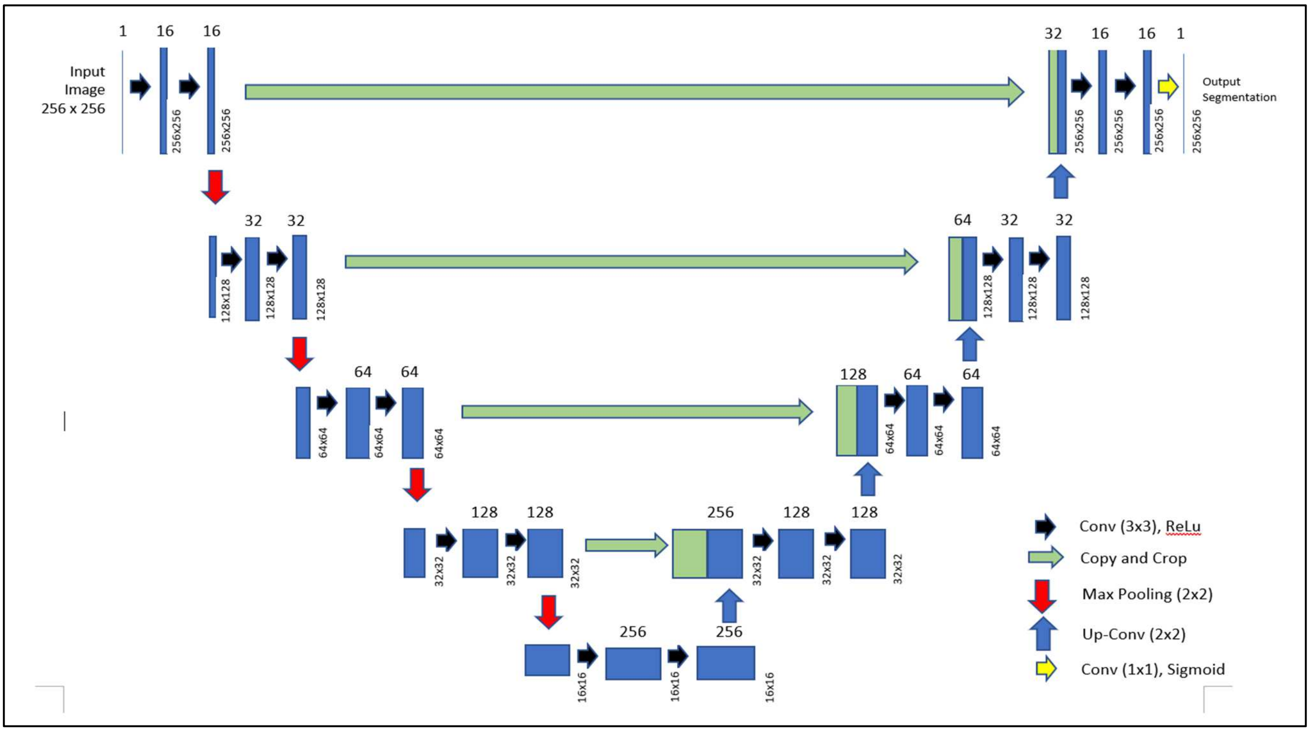


Figure-1. The U-Net architecture used for CNN model training, the left side consists of 5 levels, and the right side also consists of 5 levels.

In Fig.1, the U-Net architecture that we use has an input image with a resolution of 256x256. Each layer in the contracting path section consists of two 3x3 kernel convolution processes (black arrow) where the ReLu activation function follows each convolution (padding=same). The following layer is downsampled using the Max Pooling kernel 2x2, stride=2 (red arrow).

Each downsampling process doubles the image features from the previous (16 to 32 to 64 to 128 to 256) while reducing the image resolution by half (256 to 128 to 64 to 32 to 16).

After five layers in the contracting path section, the process will be continued by five layers in the expanding path section. Each layer in the expanding path will perform an upsampling process using a 2x2 convolution kernel (blue arrow) where the image features will be reduced by half, and the image resolution will be doubled from before. In this step, skip-connection is used (green arrow and green block). Skip connection is the process of copying the image from the contracting path to the expanding path so that it can be concatenated with the upsampling results. In the same layer, the process is continued with two times 3x3 kernel convolution followed by the ReLu activation function. In the final layer, the expanding path will use a 1x1 convolution with a Sigmoid activation function for the last 16 image features to determine the class of each pixel as segmentation output (yellow arrow).

Tabel-1 U-Net model training step

No	Process	ID	Kernel	Dim	Ch
1	Input			256x256	1
2	Conv1-1	Conv1	3x3	256x256	16
3	Dropout	D1			
4	Conv1-2	Conv2	3x3	256x256	16
5	Max Pool1	MP1	2x2 stride 2	128x128	32
6	Conv2-1	Conv3	3x3	128x128	32
7	Dropout	D2			
8	Conv2-2	Conv4	3x3	128x128	32
9	Max Pool2	MP2	2x2 stride 2	64x64	64
10	Conv3-1	Conv5	3x3	64x64	64
11	Dropout	D3			
12	Conv3-2	Conv6	3x3	64x64	64
13	Max Pool3	MP3	2x2 stride 2	32x32	128
14	Conv4-1	Conv7	3x3	32x32	128
15	Dropout	D4			
16	Conv4-2	Conv8	3x3	32x32	128
17	Max Pool4	MP4	2x2 stride 2	16x16	256
18	Conv5-1	Conv9	2x2	16x16	256
19	Dropout	D5			
20	Conv5-2	Conv10	2x2	16x16	256
21	UpSamp1 (Transpose)	Conv11	2x2	32x32	128
22	Concatenate	CNT1		32x32	256
23	Conv6-1	Conv12	2x2	32x32	128
24	Dropout	D6			
25	Conv6-2	Conv13	2x2	32x32	128
26	UpSamp2 (Transpose)	Conv14	2x2	64x64	64
27	Concatenate	CNT2		64x64	128
28	Conv7-1	Conv15	2x2	64x64	64
29	Dropout	D7			
30	Conv7-2	Conv16	2x2	64x64	64
31	UpSamp3 (Transpose)	Conv17	2x2	128x128	32
32	Concatenate	CNT3		128x128	64
33	Conv8-1	Conv18	2x2	128x128	32
34	Dropout	D8			
35	Conv8-2	Conv19	2x2	128x128	32
36	UpSamp4 (Transpose)	Conv20	2x2	256x256	16
37	Concatenate	CNT4		256x256	32
38	Conv9-1	Conv21	2x2	256x256	16
39	Dropout	D9			
40	Conv9-2	Conv22	2x2	256x256	16
41	Conv-output	Conv23	1x1	256x256	1

Since the dataset is relatively small; we added a dropout process after the first convolution in each layer to reduce overfitting and improve the generalization model. During the training, the dropout process makes several outputs ignored randomly. As a result, the training set will have slightly different data than before, which will strengthen the model in dealing with differences (generalization). Finally, our training model has 23 convolution processes (including 4 Up-Conv), 4 Max Pooling processes, and 9 Dropout processes. Table-1 shows the steps in our U-Net CNN model training process.

In evaluating the CNN model, one of the test methods that is declared suitable for limited training data is K-fold cross-validation [9]. Cross-validation is primarily used to estimate the skill of a model on previously unseen data. That is, to estimate how the model performs in general when used to make predictions on data not used during the training of the model. The procedure for performing K-fold cross-validation is in Algorithm 1.

Algorithm 1. K-Fold Cross-Validation

```
1.  Images in the dataset are scrambled
2.  The dataset is divided into k groups
3.  For each unique group: The following process is
    carried out,
    1.  One group selected as testing data set and
        the remaining groups as a training data set
    2.  Do training and testing process
    3.  Evaluate the model using the appropriate
        metrics
4.  Summarize the evaluation results and conclude
    all experimental groups.
```

In this way, each image in the dataset will have the opportunity to be once as testing data and (k-1) times as training data.

To form a visualization of the covid-19 infection, we need a mask image of the patient's lung shape. This image will be generated from the segmentation process on the CT-scan image of the patient's lungs. Segmentation will remove parts other than the lungs, such as bone, nervous tissue, and heart. The segmented image will be merged with the original CT-Scan image. This merge process will remove the background, leaving just the lungs and the infection in the lung area. Thus, the lungs must be segmented as a preliminary step. Building a U-Net CNN lung segmentation model with a model evaluation scenario using k-fold cross-validation (k=10) requires the following steps in Algorithm 2.

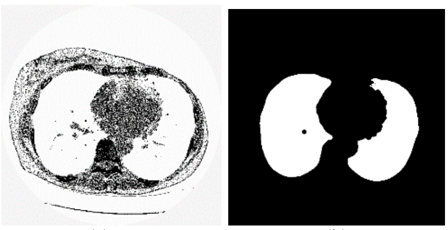
Algorithm 2. Lung Segmentation

```
Step 1: Load images from dataset
Step 2: Split the data set into 10 groups, each
group (10% dataset) becomes the testing data,
while the others (90% dataset) becomes the
training data.(k-fold cross-validation procedure,
k=10)
Step 3: Augment train dataset
Step 4: Define Input, convolutional layer and
pooling layers
Step 5: Define Up-sampling layers and
concatenation layers
Step 6: Create U-Net model
Step 7: Train the Model
Step 8: Evaluate
Step 9: Save output and display
```

III. EXPERIMENT AND RESULT

A. Dataset

The dataset used in the following experiment was obtained from Kaggle [11] in the NII/Nifty format type [12]. From the Nifty file, three-dimensional (voxel) data is sliced (axial view) into 267 two-dimensional (pixel) images with a size of 512x512. The dataset has been accompanied by Lung mask images, as shown in fig. 2. Because all CT images and Lung masks in the dataset have a clear appearance, no preprocessing of any images is done. From this dataset, 10% = 27 images are taken for testing and 90% = 240 images for training. Typically, another 10% of training data will be taken for the validation process. We reduced the image size of the dataset from 512x512 to 256x256 to speed up the training process, and the segmentation results are still good for medical image analysis purposes.

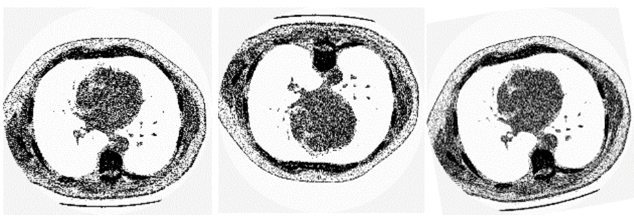


(a) (b)

Figure-2. Examples (a) CT-Scan image dan (b) Lung Mask image for the training data.

B. Augmentation

Due to the small size of training data, an offline augmentation process is carried out. It is conducted in 2 ways, image flip (vertical) and image rotation of 10 (or -10) degrees, as shown in fig. 3. After the augmentation process is complete, the training data, which previously amounted to 240 images, will become 720 images.



(a) (b) (c)

Figure-3. Examples (a) CT-Scan image (b) flip vertical augmentation, (c) rotated augmentation (-10 degree).

C. K-fold cross-validation

The dataset was randomized and divided into ten groups, according to K-fold cross-validation procedure. After being divided, each group contains 240 images for training and 27 images for testing. Each group was processed to produce a training model and then followed by testing. Fig.4 illustrates an example of five different ground-truth images.

Figure-4. Examples five Ground-truth images for testing.

D. Parameters and environment

In the model training process, the following configuration parameters are used,

1. Optimizer = Adam
2. Loss = Binary Cross Entropy
3. Metric = Keras MeanIoU
4. Batch size = 16
5. Epoch = 50
6. Early stopping = 5

The training model was built using an Intel® Core™ i7-1051OU CPU @ 1.8 GHz 2.30 GHz, 16-GB RAM without GPU. The average time required to carry out the training process for each model is 1 hour 22 minutes. Fig. 5 shows the loss and validation loss graph during the model training process (K1-K10). Table-2 shows the results of the training of each model.

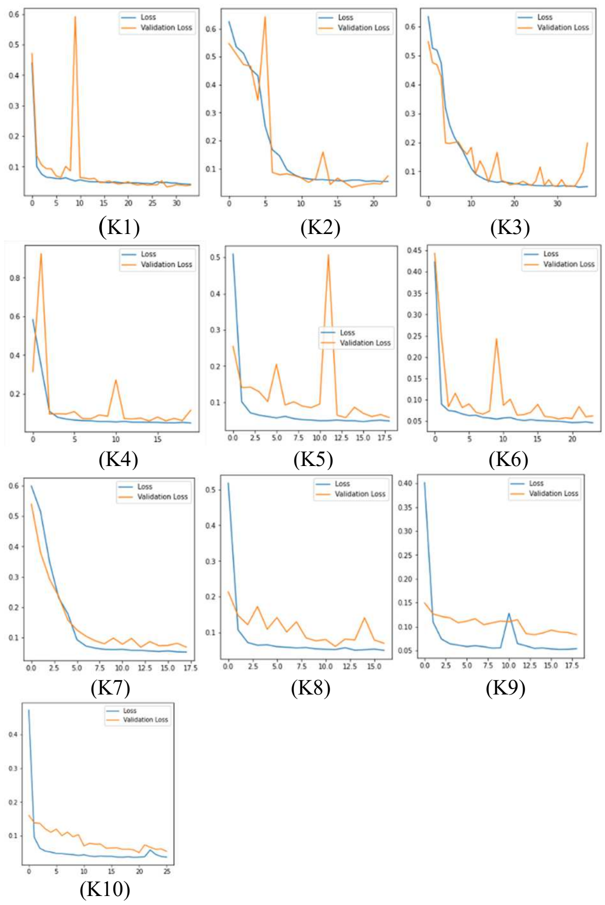


Figure-5. Loss and validation loss graph on the training model (K1-K10)

E. *Segmentation result*

The result of lung segmentation is shown in fig 6,

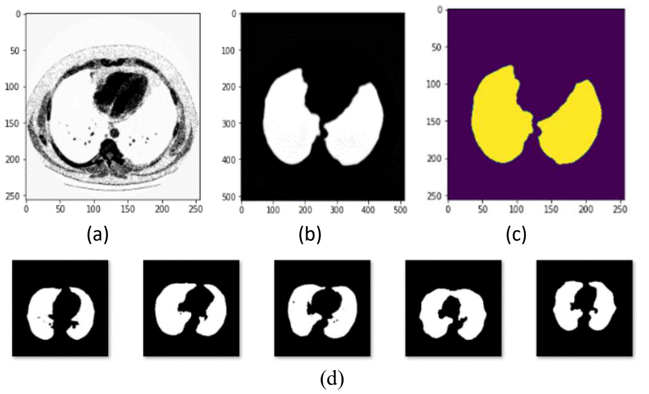


Figure-6. Examples. Lung segmentation results (a) Original image, (b) U-Net prediction result (dtype=float) (c) segmentation result after threshold (dtype=binary) (d) Five segmentation images after threshold (compared with ground truth images in Fig 4).

IV. DISCUSSION

After the testing process is complete, each test image will be compared with the ground-truth image using the IoU evaluation metric. The Mean-IoU score will be calculated by averaging the totals. The metric used in the evaluation is Intersection over Union (IoU) as in Eq. 1

$$IoU = \frac{\sum_{j=1}^{k} n_{jj}}{\sum_{j=1}^{k}(n_{ij}+n_{ji}+n_{jj})},\ i \neq j \tag{1}$$

where,

$\sum_{j=1}^{k} n_{jj}$ is the intersection between the segmented image and the ground truth image, and

$\sum_{j=1}^{k}(n_{ij} + n_{ji} + n_{jj})$ is a union between the segmented image and the ground truth image.

Mean Intersection over Union (Mean-IoU) is the average of the IoU values when a set of images is compared with the set of ground truth images as in Eq. 2

$$Mean\_IoU = \frac{1}{k}\sum_{j=1}^{k}\frac{n_{jj}}{n_{ij}+n_{ji}+n_{jj}},\ i \neq j \tag{2}$$

The IoU value ranges from 0 to 100%. This value shows the similarity between the segmentation results and the ground truth data.

The output of the U-Net model is a float number with a value between 0.0 to 1.0. The Sigmoid activation function generates this value as the final stage of the U-Net segmentation process. Up to this process, we have displayed the results of U-Net segmentation in a grayscale image. However, we need a mask for the lungs in the binary image. To generate a binary image from this value, we must perform a threshold operation. We chose several threshold values in the segmentation experiment to show the best variation of segmentation results. The relationship between the threshold value and the mean-IoU value is shown in table-3. The best results are seen using a threshold of 0.95 (yellow color) with a mean-IoU score of each k phase between 90.2% to 95.3%, and the average value of 93.3%.

In addition to using K-fold cross-validation, we also carried out evaluation by distributing 70% of the dataset for training (187 images) and 30% for testing (80 images). After the model training process is complete, the evaluation shows the results obtained are Mean-IoU of 93.8%

Additionally, we evaluated the reliability of this model with another dataset from Radiopaedia. Because the Radiopaedia dataset [13] has a slightly different appearance, we perform image preprocessing only to normalize the data, binary threshold, and binary invert operations.

Table-2: Training results in each model.
According to the procedure of K-fold cross-validation with a value of k = 10, there are ten training models resulted, K1 to K10

Model #	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10
Duration training	1:26:00	1:18:00	1.56:00	1:12:00	1:09:00	1:25:00	1:12:00	1:08:00	1:16:00	1:46:00
Epoch stopping #	34	23	38	20	19	24	18	17	19	26
Loss	0.0490	0.0402	0.0497	0.0535	0.0494	0.0493	0.0586	0.0524	0.0598	0.0363
Val_loss	0.03447	0.03351	0.04662	0.05995	0.05794	0.05469	0.06893	0.06112	0.08298	0.04964

Table-3: The relationship between the threshold value of the U-Net segmentation results with the mean-IoU score in each model.

Threshold	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	Average
0.8	93.99	91.63	92.1	92.66	92.24	92.33	93.67	93.77	90.71	90.84	92.4
0.9	94.88	92.06	92.12	93.99	93.07	93.42	94.45	94.6	92.2	90.91	93.2
0.95	95.26	92.6	90.24	94.6	93.28	94.04	93.86	94.93	93.59	90.48	93.3
0.97	94.76	92.28	87.41	94.22	92.77	94.02	92.4	94.52	94.3	89.62	92.6
0.98	93.41	91.62	84.42	93.33	91.82	93.56	90.61	93.73	94.44	88.39	91.5

Then we took 80 CT-scan images and lung mask images as testing data from the training model of 187 Kaggle dataset images. The results are excellent, reaching a Mean-IoU value of 97%. An example of the results of lung segmentation of non-Covid-19 patients is shown in fig. 7. An example of the results of lung segmentation of Covid-19 patients with GGO is seen in fig. 8. GGO is a gray-white spot seen in the lung area due to Covid-19 infection.

(a) (b) (c)

Figure-7. Example of lung segmentation results from the Radiopaedia dataset. (a) Original image of non-covid-19 patients, (b) U-Net prediction results (c) segmentation results after merging with original images (a)+(b).

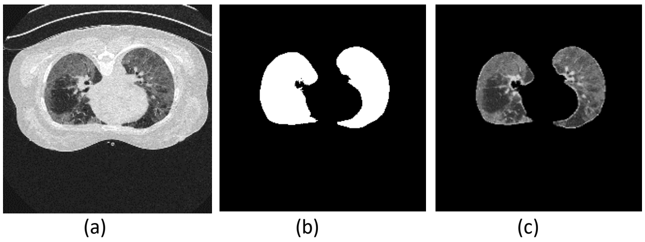


Figure-8. Examples of lung segmentation results from the Radiopaedia dataset (a) Original image of a covid-19 patient with GGO infection, (b) U-Net prediction results (c) segmentation results after merging with the original image (a)+(b)

V. CONCLUSION

Using a relatively small dataset (257 images), U-Net can be proven to provide a good segmentation model of the patient's lung shape based on the mean-IoU metrics above 90%. By using the K-fold cross-validation method (k=10), the resulting training model is believed to have a stable evaluation with a mean-IoU value of 93.3%. The second scenario similarly proves the reliability of this model by creating a model with 70% training data (187 images) and 30% testing data (80 images) which also produces a mean-IoU of 93.8%. The same model with different testing data (Radiopaedia dataset) produces mean_IoU of 97%. Based on these promising results, we will continue this work by increasing the number of datasets and their variations for better accuracy. We believe we can achieve our primary goal of segmenting and visualizing Covid-19 infection in the lungs to assist radiologists in diagnosing Covid-19 patients.

ACKNOWLEDGEMENT

The authors would like to express special thanks to The University Center of Excellence on Artificial Intelligence for Healthcare and Society (UCE AIHeS), Institut Teknologi Sepuluh Nopember for fully supporting this work.

REFERENCES

[1] Xiaolong Liu, Zhidong Deng, Yuhan Yang, "Recent Progress ini Semantic Images Segmentation", Springer, Artificial Intelligence Review. 2018, doi:10.1007/s10462-018-9641-3.

[2] Irem Ulku, Erdem Akaqunduz, "A SURVEY ON DEEP LEARNING-BASED ARCHITECTURES FOR SEMANTIC SEGMENTATION ON 2D IMAGES", arXiv:1912.10230v3 [cs.CV] 1 May 2021

[3] Kalinovsky A., Kovalev V. "Lung image segmentation using Deep Learning methods and convolutional neural networks" . In: XIII Int. Conf. on Pattern Recognition and Information Processing, 3-5 October, Minsk, Belarus State University, 2016, pp. 21-24

[4] Olaf Ronneberger, Phillip Fischer, Thoman Brox ,"U-Net : Convolutional Networks for Biomedical Image Segmentation", Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer, LNCS, Vol.9351: 234--241, 2015, available at arXiv:1505.04597 [cs.CV]

[5] Brahim AIT SKOURT, Abdelhamid EL HASSANI, Aicha MAJDA, "Lung CT Image Segmentation Using Deep Neural Networks", 2018, Procedia Computer Science 127 (2018) 109-133 Elsevier

[6] Ching-Sheng Chang, Jin-Fa Lin, Ming-Ching Lee, Christoph Palm, " Semantic Lung Segmentation Using Convolutional Neural Network", 2020, Springer Bildverarbeitung für die Medizin 2020

[7] Humera Shaziya, K, Shyamala, "Pulmory CT Images Segmentation using CNN and U-Net Models of Deep Learning", 2020 IEEE Pune Section International Conference (PuneCon).

[8] Athanasios Voulodimos, Eftychios Protopapadakis, Iason Katsamenis, Anastasios Doulamis, Nikolaos Doulamis, "Deep Learning models for Covid-19 infected area segmentation in CT images", June 2021, PETRA 2021: The 14th Pervasive Technologies Related to Assistive Environments Conference. Pages 404–411, https://doi.org/10.1145/3453892.3461322

[9] Sanjay Yadav, Sanyam Shuykla, "Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality Classification", 2016, IEEE 6th International Conference on Advanced Computing.

[10] E. Martinez Chamorro, A. Díez Tascón, L. Ibánez Sanz, S. Ossaba Vélez. S. Borruel Nacenta "Radiologic diagnosis of pasiens with COVID-19", 2021, Radiologia 56-73. Elsevier Serie: Radiology and Covid-19. www.elsevier.es/rx

[11] Lung Dataset. https://www.kaggle.com/kmader/finding-lungs-in-ct-data (accessed 10 April 2021).

[12] NIFTI Data format, http://nifty.nimh.nih.gov (accessed 10 April 2021)

[13] MedSeg Covid_Dataset Version-2, Dataset posted on 05.01.2021 by MedSeg, Håvard Bjørke Jenssen, Tomas Sakinis, "volumetric CTs from Radiopaedia", 2021, https://figshare.com/articles/dataset/Covid_Dataset_2/13521509. (accessed 23 Juli 2021).