

## UNet-VGG16 with transfer learning for MRI-based brain tumor segmentation

Anindya Apriliyanti Pravitasari<sup>1</sup>, Nur Iriawan<sup>2</sup>, Mawanda Almuhayar<sup>3</sup>, Taufik Azmi<sup>4</sup>,  
Irhamah<sup>5</sup>, Kartika Fithriasari<sup>6</sup>, Santi Wulan Purnami<sup>7</sup>, Widiana Ferriastuti<sup>8</sup>

<sup>1,2,3,4,5,6,7</sup>Department of Statistics, Institut Teknologi Sepuluh Nopember, Indonesia

<sup>1</sup>Department of Statistics, Universitas Padjajaran, Indonesia

<sup>8</sup>Department of Radiology, Universitas Airlangga, Indonesia

### Article Info

#### Article history:

Received Aug 15, 2019

Revised Jan 29, 2020

Accepted Feb 26, 2020

#### Keywords:

Fully convolution network

Image segmentation

Transfer learning

U-Net

VGG16

### ABSTRACT

A brain tumor is one of a deadly disease that needs high accuracy in its medical surgery. Brain tumor detection can be done through magnetic resonance imaging (MRI). Image segmentation for the MRI brain tumor aims to separate the tumor area (as the region of interest or ROI) with a healthy brain and provide a clear boundary of the tumor. This study classifies the ROI and non-ROI using fully convolutional network with new architecture, namely UNet-VGG16. This model or architecture is a hybrid of U-Net and VGG16 with transfer Learning to simplify the U-Net architecture. This method has a high accuracy of about 96.1% in the learning dataset. The validation is done by calculating the correct classification ratio (CCR) to comparing the segmentation result with the ground truth. The CCR value shows that this UNet-VGG16 could recognize the brain tumor area with a mean of CCR value is about 95.69%.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



### Corresponding Author:

Nur Iriawan,

Department of Statistics,

Institut Teknologi Sepuluh Nopember,

Kampus ITS Sukolilo-Surabaya 60111, Indonesia.

Email: nur\_i@statistika.its.ac.id

## 1. INTRODUCTION

A brain tumor is the 15<sup>th</sup> deadly disease in Indonesia compared to all types of cancer. According to the WHO, there were 5,323 cases of brain and nervous system tumors in Indonesia with 4,229 mortality cases during 2018 [1]. Due to this reason, a brain tumor is considered to be an important topic. Detection of brain tumor could be done under the medical equipment called magnetic resonance imaging or MRI. General Hospital (RSUD) Dr. Soetomo Surabaya provides radiological examination services using MRI 1.5 Tesla and 3 Tesla. The greater the Tesla number, the better the image quality, but it's also more costly. The MRI 1.5 Tesla is a favorable service, since it has a minimum cost and also covered by the Indonesian government's social security (BPJS). However, better image quality is needed in medical treatment. The image segmentation is one of methodology that could provide better sight of brain tumors by separating the tumor area (as the region of interest or ROI) with the healthy brain and provide a clear boundary of the tumor. The clear boundary helps the medical treatment, especially in surgery, to running safely without damaging healthy parts of the brain. This study tries to segment the MRI brain tumor to give a better sight of the MRI image from a 1.5 Tesla machine.

Many types of research had been developed for image segmentation. Several methods use clustering as the basis of modeling, while others use classification. The aim is to get the best model that could recognize

the tumor area more precisely. The previous studies use the clustering as the basis of segmentation are performed by [2] which uses the genetic algorithm and [3] which employ fuzzy clustering, Otsu method and K-means cluster to segment the vehicle image. The model-based clustering is performed by [4-6] in the form of a Finite mixture model to segment the MRI brain tumor image. Different from the previous studies, this study uses the classification method with Neural Network. Neural Network is used since it can adapt the working of human neurons in recognizing images. Previous studies done under the NN approach is performed by [7-9] that used the convolutional neural network (CNN) and Deep Learning for image segmentation. This study uses the Fully Convolutional Network (FCN) since it has great performance for semantic segmentation [10, 11].

The FCN used the U-Net architecture by the recommendation of [12]. The U-Net model can achieve very good performance on very different biomedical segmentation applications. This paper uses a real dataset from General Hospital (RSUD) Dr. Soetomo. The limited number of brain tumor patients who come to Dr. Soetomo, makes the number of datasets analyzed also limited. The U-Net will be hybridized with VGG16 architecture as its encoder (contracting) layer [13] to simplify the architecture and overcome the problem of small number data classification. On the other hand, the complexity of U-Net often spends a lot of time in its execution and is greatly affected when the computer specifications are inadequate. In view of these shortcomings and to give the stage of the art of this paper we perform the transfer learning in the part of training data in the hybrid of U-Net and VGG16. The name of this proposed model or architecture is UNet-VGG16 with Transfer Learning. Moreover, we try to compare the UNet-VGG16 with several scenarios of the U-Net model and decide the best model with the value of loss and accuracy. The correct classification ratio (CCR) will be calculated to compare the segmentation results from the best model with the ground truth as the measure of Evaluation. All the data and ground truth used in this study are passed by the medical approval from Dr. Soetomo Surabaya.

## 2. RESEARCH METHOD

### 2.1. Fully convolutional network: U-net with transfer learning

Fully convolutional network (FCN) is one of the deep learning used in semantic segmentation. FCN proposed by [14] that examines pixel-to-pixel mapping and used the ground truth to determine the pixel class. The FCN is the development of the classical convolutional neural network (CNN). CNN consists of convolution, pooling, and fully-connected as the main layers, which in the FCN, the fully connected layer is replaced by the convolution layer. FCN, therefore, can classify every pixel in the image and give them the ability to make predictions on arbitrary-sized inputs. The input data is an image that consists of three layers namely height ( $h$ ), width ( $w$ ), and depth ( $d$ ). The  $h$  and  $w$  explained the spatial dimension, while  $d$  is the feature or channel dimension. The image first layer has  $h \times w$  dimension and  $d$  color channel ( $d = 1$  when it contains only grayscale intensity, or  $d = 3$  when it has red, green, and blue (RGB) intensities). Supposed we have input data vector  $x_{ij}$ , in the location  $(i, j)$  of a particular layer. We also can calculate the vector output  $y_{ij}$  with the following formula [14]:

$$y_{ij} = f_{ks}(\{x_{si+\delta_i, sj+\delta_j}\}, 0 \leq \delta_i, \delta_j \leq k) \quad (1)$$

where  $k$  is the kernel size,  $s$  is the subsampling factor, and  $f_{ks}$  specifies the type of layer used: a matrix multiplication for convolution or average pooling, a spatial max for max pooling, or other types of layers. Applying this layer can significantly reduce the number of parameters of the network. Moreover, the network can learn the correlation between neighborhood pixels [15]. Figure 1 is a structural model in the FCN method. In Figure 1, we could see that FCN can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation [14].

Several architectures are building under FCN, this study used the U-Net architecture that firstly recommended by [12], which is great for biomedical image segmentation. In Figure 2, we can see the visualization of U-Net architecture. The U-shape of architecture is the reason behind its name. The letter U contains two paths. The left path is called the encoder (contracting layer) and the right path is the decoder (expanding layer). The encoder is a network in which the output is the feature map/ vector that holds the information representing the input. The decoder which has the same structure but in the opposite way, is a network that takes feature maps from the encoder and provides a similar match of the actual input or intended output. The process in the encoder path is reducing the size input matrix by increasing the number of the feature maps, while in the decoder path is returning the matrix to its original size by minimizing the number of the feature maps. The segmentation results, therefore, can be compared with the ground truth in every pixel.

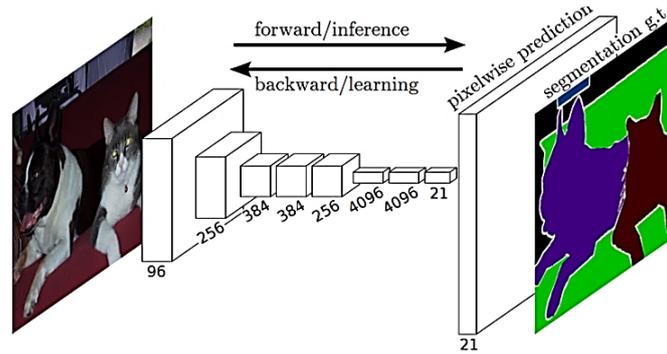


Figure 1. Structure of fully convolutional network (sourced from [14])

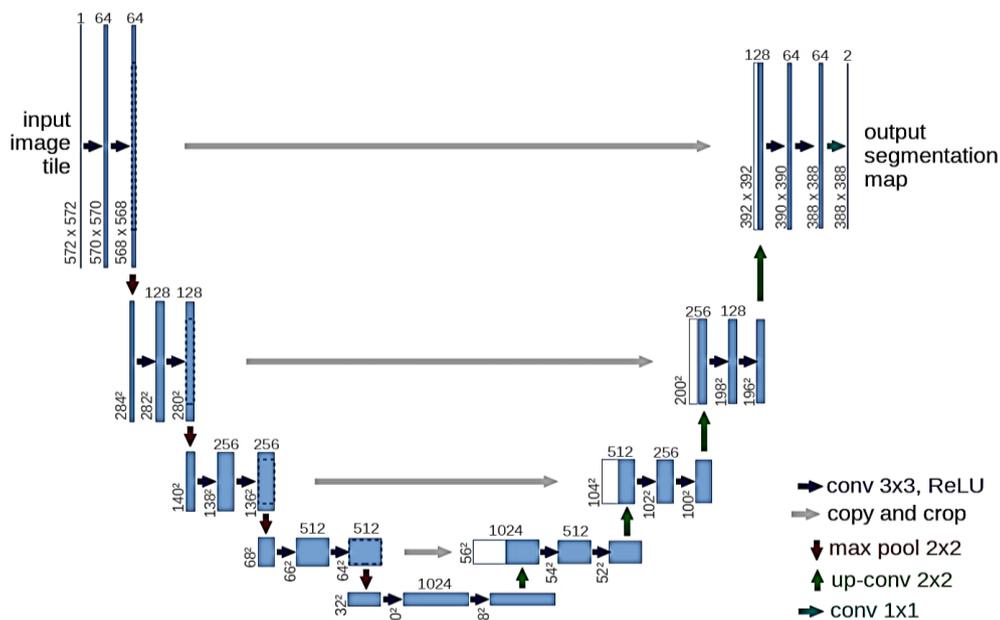


Figure 2. The U-Net architecture  
(example for 32x32 pixels in the lowest resolution, sourced from [12])

U-Net transmits the feature map of each contract path level at the analog level of the extension path so that the classifier can consider features of varying scales and complexities to make its decision. Therefore, U-Net can learn from relatively small training sets [16]. Since its complexities, the U-Net architecture often spends a lot of time on its execution. In order to deal with the problem, this study tries to hybrid the U-Net architecture with Transfer learning.

Transfer learning is an approach in which pre-trained models are used as a starting point for computer vision and language processing tasks, since the development of neural network models for these problems and due to the enormous leaps in qualifications requires extensive computing and time resources. The aim of Transfer learning is to improve learning in the target task by using the knowledge from the source task. Transfer learning is an effective technique for reducing training time [17]. This technique may be related to the development of deep learning models for image classification problems.

Moreover, to simplifying the U-Net architecture, several architectures from CNN have been considered to hybrid with the U-Net, such as LeNet [18], AlexNet [19], ZFNet [20] and VGG-Net [21]. The VGG-Net confirms that a smaller kernel size and a deep CNN can improve model performance. The architecture of VGG-Net as shown in Figure 3 is quite similar to the U-Net, therefore, this study chooses VGG-Net to replace the encoder path of U-Net as the hybrid between these two powerful architectures. The new architecture will be discussed in results and analysis.

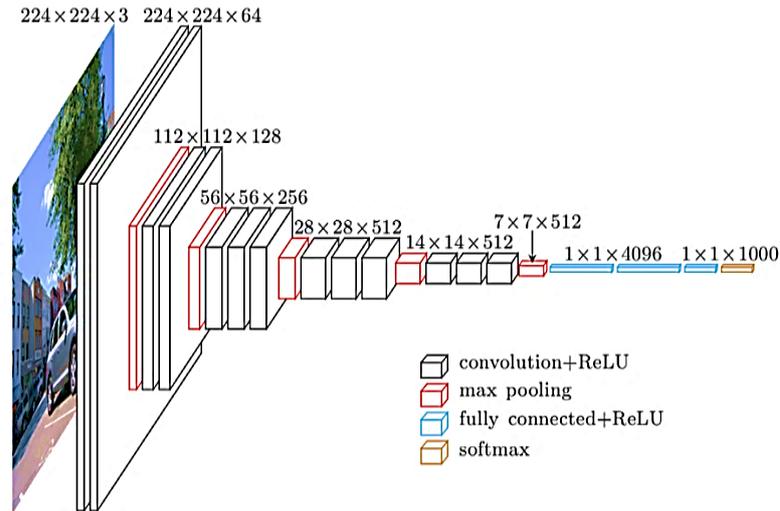


Figure 3. The VGG16 architecture (adapted from [21])

This study will use the loss and accuracy to measure model performance. Loss is the value of error between predicted and actual value. The case with two classes in machine learning uses the binary cross-entropy loss function to calculate the value of loss or error [22]. The binary cross-entropy is provided by (2).

$$H_p(q) = -\frac{1}{N} \sum_{i=1}^N z_i \log(p(z_i)) + (1 - z_i) \log(1 - p(z_i)), \quad (2)$$

where  $N$  is the number of data,  $z_i$  is the class of classification which has the value of 0 or 1, and  $p(z_i)$  is the probability of  $z_i$ . Accuracy is the closeness between predicted and actual value. In order to determine the accuracy, we use the confusion matrix as in Table 1 [22]. The formula for calculating the accuracy is shown by (3).

$$\text{accuracy} = \frac{TP+TN}{TP+FP+TN+FN}. \quad (3)$$

Table 1. Confusion matrix

Actual Value	Predicted Value	
	True	False
True	$TP$ (True Positive)	$FN$ (False Negative)
False	$FP$ (False Positive)	$TN$ (True Negative)

## 2.2. Training and optimization

The optimization used in this paper is Adaptive Moment Estimator (Adam) to estimate the parameters  $x_t$ . Adam was a combination of two method's AdaGrad and RMSProp [23]. Adam utilizes two moments' variable  $v_t$  as the first moment (the mean) and variable  $s_t$  as the second moment (the uncentered variance) of the gradients respectively. Given the hyper-parameter  $0 \leq \beta_1 < 1$  and  $0 \leq \beta_2 < 1$  and perform an exponentially-weighted moving average (EWMA), the Adam estimator can be constructed in Algorithm 1 as follows [24]

---

### Algorithm 1. Adam Estimator

---

- Initialize the value of  $\beta_1, \beta_2 \in [0,1)$ ,  $v_0, s_0$ , learning rate  $\eta$ , and stochastic objective function ( $f(x_t)$ ) with parameters  $x_t$ .
- Calculate the gradient of  $f(x_t)$  with following formula

$$g_t = \nabla_x f_t(x_{t-1}). \quad (4)$$

- Update the first and second moment with equation (5) and (6)

$$v_t = \beta_1 v_{t-1} + (1 - \beta_1) g_t, \quad (5)$$


---

$$s_t = \beta_2 s_{t-1} + (1 - \beta_2) g_t \odot g_t \quad (6)$$

d. Calculate the bias correction with (7) and (8)

$$\hat{v}_t = \frac{v_t}{1 - \beta_1^t}, \quad (7)$$

$$\hat{s}_t = \frac{s_t}{1 - \beta_2^t}. \quad (8)$$

e. Re-adjust the learning rate for each element in the model parameters using element operations with bias-corrected variables  $\hat{v}_t$  and  $\hat{s}_t$  as in following formula

$$g'_t = \frac{\eta \hat{v}_t}{\sqrt{\hat{s}_t + \varepsilon}}, \quad (9)$$

where  $\eta$  is the learning rate and  $\varepsilon$  is error criterion set to  $10^{-8}$ .

f. Update the parameter  $x_t$  as the (10)

$$x_t = x_{t-1} - g'_t, \quad (10)$$

g. Do step 2 to step 6 until the parameter  $x_t$  is convergence.

As in Algorithm 1, Adam used stochastic gradient descent to maintain a single learning rate for all weight updates. This has ensured that the learning rate does not change during the training process. The learning rate is maintained for each network weight (parameter) and adjusted separately as learning unfolds. Besides, the algorithm calculates an exponential moving average of the gradient and the squared gradient, with hyper-parameters  $\beta_1$  and  $\beta_2$  to control the decay rates. The initialization is set for learning rate  $\eta = 0.0001$ ,  $\beta_1 = 0.9$ , and  $\beta_2 = 0.999$ . The loss function used binary cross-entropy as a set up binary classification. The binary cross-entropy was a combination of sigmoid activation and cross-entropy loss. In this study, we divide data into three partitions. We do the sampling without replacement to get the randomized data for training, validation, and testing. The three sets are divided using a ratio of 80:10:10. The amount of data for each set is as Figure 4.



Figure 4. Division of the data

### 2.3. Evaluation

Correct Classification ratio ( $CCR$ ) is calculated as a measure of evaluation.  $CCR$  value is obtained to find out whether the ROI from segmentation results in accordance with the ground truth. The greater the  $CCR$  value, the better the segmentation result and vice versa. As shown in (11) is the formula of  $CCR$  [25].

$$CCR = \sum_{j=1}^2 \frac{|GT_j \cap Seg_j|}{|GT|}, \quad (11)$$

where  $GT_j$  is ground truth for non-ROI ( $j = 1$ ) and ROI ( $j = 2$ ).  $Seg_j$  for  $j = 1$  defines pixel segmented based on the model as non-ROI area, meanwhile for  $j = 2$  describes pixel segmented as ROI and  $GT = \bigcup_{j=1}^2 GT_j$ .

## 3. RESULTS AND ANALYSIS

### 3.1. The proposed architecture

U-Net is one of CNN architecture that is used specifically for image segmentation. The complexity of U-Net (this research has 31,031,685 parameters) impacts the time of execution and in several computers with the limited specification, the U-Net architecture cannot be run. To overcome this problem, we proposed

a new model that reduces the layer and parameter of U-Net by combining it with another architecture namely VGG16. The choice of VGG16 is because of its similarity to U-Net's contracted layer and its number of parameters is also less than U-Net. In addition, VGG16 already has weights from parameters that are easily accessed, so we employ these weights to this new model.

Several models used for segmentation frequently consist of the contracting layer and the expansion layer. In this study, we modified the VGG16 to resemble the U-Net architecture by adding an expansive layer consisting of several upsampling layers and convolution layers at the end of the VGG16 architecture. This is done until the overall architecture of the model is symmetrical and resembles the shape of the letter U. Therefore, in the architecture of the UNet-VGG16 model, we will have a contracting layer, which is the VGG16, and the expansion layer that will be added later. With this new architecture, the parameters will be reduced to 17,040,001 with trainable parameters is about 2,324,353.

The MRI brain tumor image will be trained using the UNet-VGG16 model with the Transfer Learning method. This method freezes the contraction layer in UNet-VGG16 so that the weighted layer is not updated when executing training data. Instead, we use the weight of the convolution layer of the VGG16 model. The goal is to reduce the computing process and speed up the training time of the model. Figure 5 shows the architecture of the proposed model, namely UNet-VGG16 with Transfer Learning, while Figure 6 is the process visualization of image segmentation under the new proposed architecture.

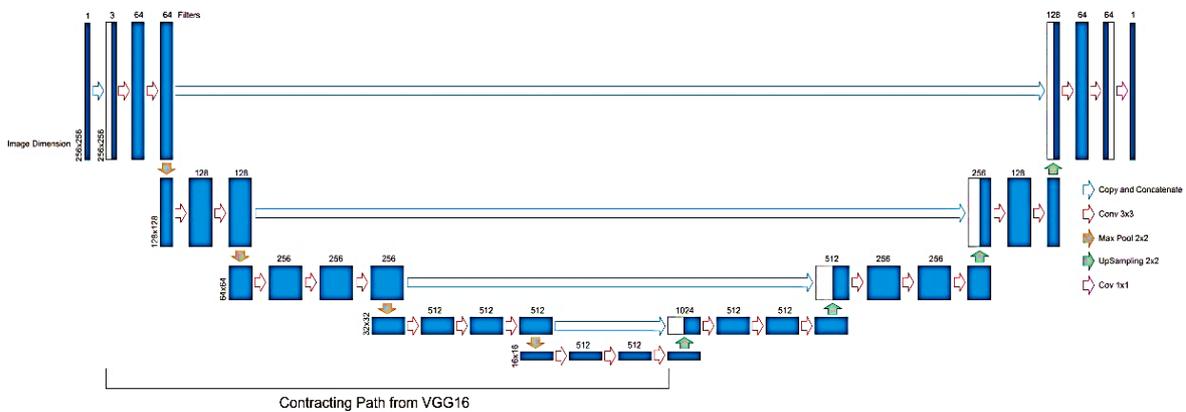


Figure 5. The architecture of UNet-VGG16 with transfer learning

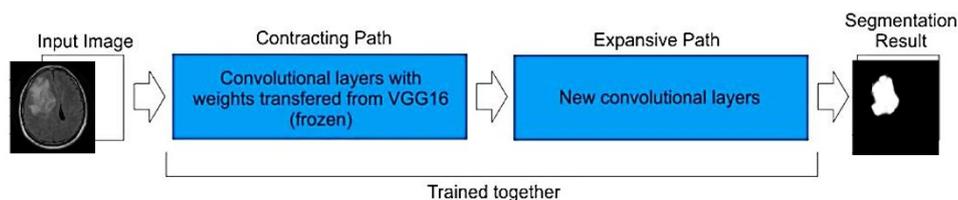


Figure 6. The segmentation process under UNet-VGG16 with transfer learning

### 3.2. Choosing the best model

In this study, we try to compare the proposed model with the previous state of the art model. U-Net architecture is developed with various scenarios so that several alternative models are obtained which will then be compared for accuracy between one another and with the proposed model. These modification scenarios of U-Net are done since the original of U-Net model cannot run in our computer with the specification of Processor Intel Core i7, 32GB RAM, 128GB SSD, and without GPU and VRAM. Table 2 is a breakdown of the number of convolution layers and convolution blocks in each U-Net scenario.

All the models build under the Python programming language with Tensorflow, Keras, and NumPy libraries. In the training process, we use 100 epochs and executed in a computer with processor Intel Core i7, 32GB RAM, 128GB SSD, and without GPU and VRAM. Each epoch of the proposed model takes 80 minutes of computer processing. It is longer than the four scenarios of U-Net since the number of parameters of the proposed model is more than the modified U-Net.

Table 2. Convolution scenario for U-Net

Scenario	Conv layer	Number of		Time/epoch
		Conv block	Parameter	
Model 1	1 Conv @ block	5 block	3,929,985	23 min
Model 2	1 Conv @ block	4 block	980,097	20 min
Model 3	2 Conv @ block	5 block	7,862,113	34 min
Model 4	2 Conv @ block	4 block	1,962,337	32 min

Figure 7 demonstrates a learning curve of four U-Net scenarios compares with the UNet-VGG16 segmentation model with Transfer Learning. Figure 7 (a) shows the comparison of the loss value of the U-Net and the proposed model, while Figure 7 (b) is a comparison of its accuracy. Based on this figure, it can be seen that during the training process with 100 epochs, the proposed model provides a minimum loss value and maximum accuracy compared to the four U-Net scenarios. Moreover, the loss and accuracy of the proposed model are faster convergent and stable with smoother movements compared to the loss and accuracy of U-Net models that are still volatile during the training process. Table 3 shows the overall performance of each model.

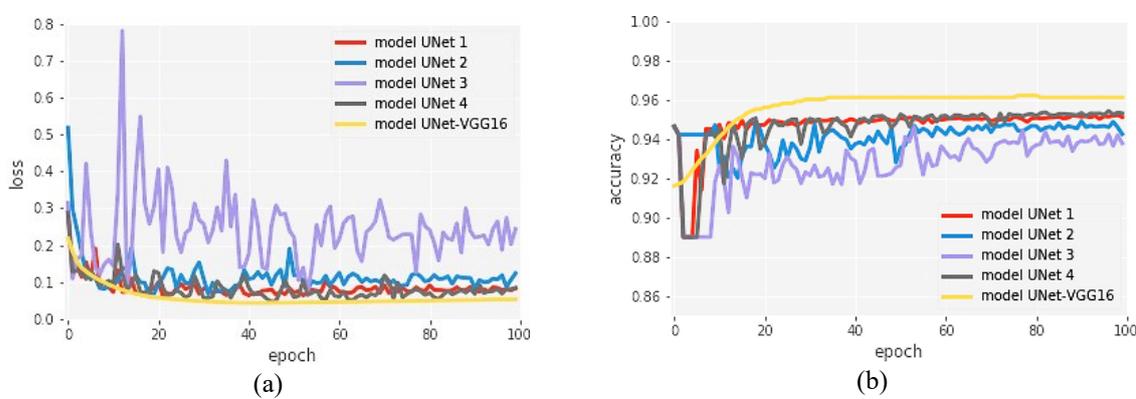


Figure 7. (a) Loss and (b) accuracy from the four U-Net scenarios compare to UNet-VGG16 with transfer learning

From Table 3 and Figure 7, the minimum loss and maximum accuracy are reached by the proposed model. Its loss value is 0.054 and the accuracy value is 0.961. Based on these results, the proposed model gives better performance than the four scenarios of U-Net. Since the best model belongs to UNet-VGG16 with Transfer Learning, for the next analysis, we will use the model as the basis of MRI brain tumor segmentation.

### 3.2. Segmentation results based on the best model

The segmentation is done to the 16-testing data under UNet-VGG16 with Transfer Learning. The visualization of segmentation results is described in Figure 8. This figure has shown that segmentation results of sample sequence could recognize the tumor area as ROI in various tumor size and location, both on the left or right of the brain. For the testing data, we calculate the *CCR* as the measure of evaluation and the summary is shown in Table 4. The segmentation results are approaching ground truth very well since the all *CCR* value above 90%. The *CCR* grand mean for all testing data is reaching 95.69%.

Table 3. The performance comparison of each model

Model comparison	Loss	Accuracy
U-Net : Model 1	0.124	0.942
Model 2	0.083	0.951
Model 3	0.085	0.953
Model 4	0.244	0.938
UNet-VGG16 with TL	0.054	0.961

Table 4. The summary of *CCR* for testing data

Testing number	CCR	Testing number	CCR
1	0.957184	9	0.957489
2	0.970047	10	0.965225
3	0.916245	11	0.961166
4	0.935806	12	0.905029
5	0.982773	13	0.933807
6	0.981598	14	0.928635
7	0.979904	15	0.985748
8	0.974136	16	0.975601

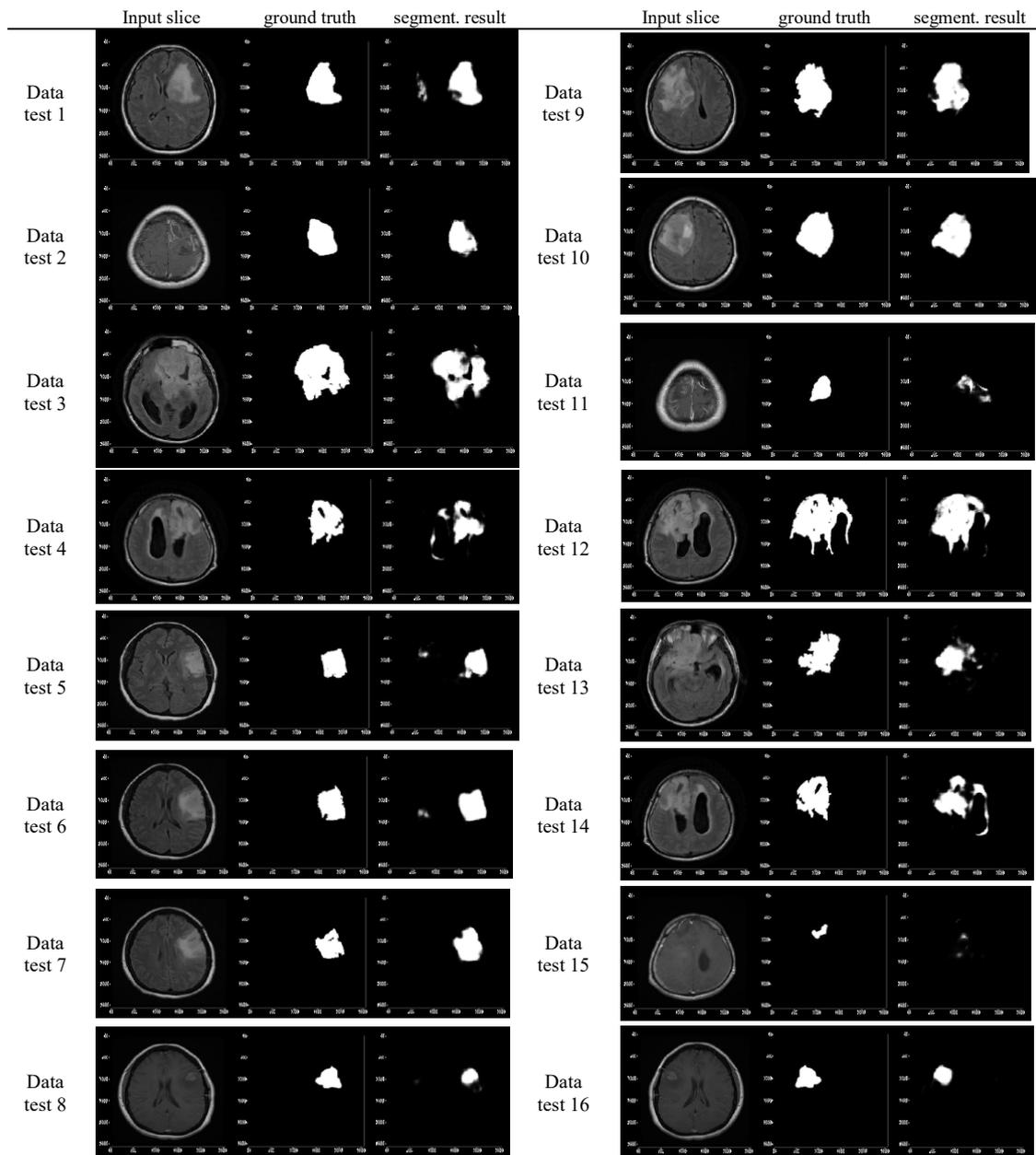


Figure 8. The segmentation results of 16 testing data

#### 4. CONCLUSION

Based on the results and analysis we can conclude that the proposed model namely UNet-VGG16 with Transfer Learning is running well on the computer with a processor of Intel Core i7, 32GB RAM, 128GB SSD, and without GPU and VRAM. The proposed model has great performance compared to the U-Net model (in four scenarios) since it has the minimum value of loss and maximum value of accuracy. The segmentation results under the proposed model tend to approach the ROI target of each brain tumor MRI image very well. The results of segmentation from testing data were obtained by CCR value of 95.69%. For future research, the different architecture or convolutional block scenario could be obtained to get more alternative models. Not to mention that the optimum epoch is still demanded to get the optimal computing time for building the training model.

## ACKNOWLEDGEMENTS

The authors are grateful to the Directorate for Research and Community Service (DRPM) Ministry of Research, Technology and Higher Education Indonesia which supports this research under PT research grant no. 944/PKS/ITS/2019.

## REFERENCES

- [1] Globocan, "Indonesia Source: Globocan 2018," 2019. [Online]. Available: <http://gco.iarc.fr/today/data/factsheets/populations/360-Indonesia-fact-sheets.pdf>.
- [2] V. Jaiswal, V. Sharma, S. Varma., "An Implementation of Novel Genetic-Based Clustering Algorithm for Color Image Segmentation," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol 17, no 3, pp 1461-1467, June 2019.
- [3] P. B. Prakoso, Y. Sari., "Vehicle Detection using Background Subtraction and Clustering Algorithms," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol 17, no 3, pp 1393-1398, June 2019.
- [4] N. Iriawan, A. A. Pravitasari, K. Fithriasari, Irhamah, S. W. Purnami, W. Ferriastuti., "Comparative Study of Brain Tumor Segmentation using Different Segmentation Techniques in Handling Noise," *2018 International Conference on Computer Engineering, Network and Intelligent Multimedia (CENIM)*, Surabaya, Indonesia, pp. 289-293, 2018.
- [5] A. A. Pravitasari, M. A. I. Safa, N. Iriawan, Irhamah, K. Fithriasari, S. W. Purnami, W. Ferriastuti, "MRI-Based Brain Tumor Segmentation using Modified Stable Student's t from Burr Mixture Model with Bayesian Approach," *Malaysian Journal of Mathematical Sciences*, vol 13, no 3, pp. 297-310, 2019.
- [6] A. A. Pravitasari, N. I. Nirmalasari, N. Iriawan, Irhamah, K. Fithriasari, S. W. Purnami, W. Ferriastuti, "Bayesian Spatially constrained Fernandez-Steel Skew Normal Mixture model for MRI-based Brain Tumor Segmentation," *In AIP Conference Proceedings*, vol. 2194, no. 1, December 2019.
- [7] M. Pathak, N. Srinivasu, V. Bairagi, "Effective Segmentation of Sclera, Iris, and Pupil in Noisy Eye Images," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol 17, no 5, pp 2346-2354, October 2019.
- [8] I. B. K. Sudiarnika, F. Rahman, Trisno, Suyoto, "Image Forgery Detection Using Error Level Analysis and Deep Learning," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol 17, no 2, pp 653-659, April 2019.
- [9] E. Gibson, W. Li, C. H. Sudre, L. Fidon, D. Shkir, G. Wang, Z. Eaton-Rosen, R. Gray, T. Doel, Y. Hu, T. Whyntie, P. Nachev, D. C. Barratt, S. Ourselin, M. J. Cardoso, T. Vercauteren., "Niftnet: a Deep-Learning Platform for Medical Imaging," *Computer Methods and Programs in Biomedicine*, vol. 158, pp. 113-122, May 2018.
- [10] P.V. Tran, "A Fully Convolutional Neural Network for Cardiac Segmentation in Short-Axis MRI," arXiv preprint arXiv:1604.00494, 2016.
- [11] Y. Li, H. Qi, J. Dai, X. Ji, Y. Wei, "Fully Convolutional Instance-Aware Semantic Segmentation," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, pp. 4438-4446, 2017.
- [12] O. Ronneberger, P. Fischer, T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2015*, vol. 9351, pp. 234-241, November 2015.
- [13] C. Balakrishna, D. Sarshar, S. Soltaninejad, "Automatic detection of lumen and media in the IVUS images using U-Net with VGG16 Encoder," *arXiv preprint arXiv:1806.07554*, June 2018.
- [14] J. Long, E. Shelhamer, T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, pp. 3431-3440, 2015.
- [15] Y. Le Cun, Y. Bengio, "Convolutional Networks for Images, Speech, and Time Series," *Hand Brain Theory Neural Netw*, pp 3361-3371, January 1995.
- [16] H. C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogue, J. Yao, D. Mollura, R. M Summers, "Deep Convolutional Neural Networks for Computer-aided Detection: CNN Architectures, Dataset Characteristics, and Transfer Learning," in *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285-1298, May 2016.
- [17] F. Chollet, "Deep Learning with Python," *Manning: Shelter Island*, 2018.
- [18] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.
- [19] A. Krizhevsky, I. Sutskever, G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," in *Advances in neural information processing systems*, pp 1097-1105, 2012.
- [20] M. D. Zeiler, R. Fergus, "Visualizing and Understanding Convolutional Networks," *European conference on computer vision*, Springer, Cham, vol. 8689, pp 818-833, 2014.
- [21] A. Kamilaris, F. A Prenafeta-Boldú, "A Review of The Use of Convolutional Neural Networks in Agriculture," *The Journal of Agricultural Science*, pp 1-11, 2018.
- [22] A. Zhang, Z. C. Lipton, M. Li, A. J. Smola, "Dive into Deep Learning," *SAGE Publications Inc*, 2019.
- [23] D. P. Kingma, J. Ba, "Adam: A Method for Stochastic Optimization," arXiv preprint arXiv:1412.6980, 2014.
- [24] S. Reddi, S. Kale, S. Kumar, "On the Convergence of Adam and Beyond," arXiv preprint arXiv:1904.09237, 2019.
- [25] C. Nikou, N. P. Galatsanos, A. C. Likas, "A Class-adaptive Spatially Variant Mixture Model for Image Segmentation," in *IEEE Transactions on Image Processing*, vol. 16, no. 4, pp. 1121-1130, April 2007.