Sci. Rep. Kanazawa Univ.
Vol. 63, pp. 67–78, 2019

ARTICLES

Convolutional Neural Network to Classify Capillaries of Images in Human Fingertips

Mochammad HAFIIZH[1,2,*]

[1]Graduate School of Natural Science and Technology, Kanazawa University
Kakuma, Kanazawa, 920-1192, Japan
[2]Department of Mathematics, Universitas Negeri Malang
Malang, 65145, Indonesia

(Received June 26, 2019, and accepted in revised form August 28, 2019)

Abstract In this paper, we report on training an image classifier implemented as a convolutional neural network (CNN) to judge the state of human capillaries from microscope photographs. This work is motivated by medical research that suggests that their shape and other features are linked to other health measures. Instead of medical doctor's diagnosis and therefore to possibly lower the cost and consistency of the diagnosis, we propose a CNN classifier. To overcome the difficulty of obtaining a large number of training images, we utilize the transfer learning approach. We base our model on a pre-trained VGG16 model for the lower layers of the network and our implementation uses Python and the Torch library. Abnormal types of capillaries are divided into several patterns and appropriate pre-processing is performed including a partial differential equation approach. The trained CNN has been applied to actual capillary patterns to verify its performance.

Keywords. Image, capillaries, CNN, transfer learning

1 Introduction

Prevention is important in medicine. According to private discussion with medical experts, there is a medical research that suggests that the shape and other features of human capillaries are linked to other health measures. Ogawa [10] outlined that the health of a person affects the shape of capillaries at the root of a fingernail. The reference [15] also reported that the structure of capillaries is strongly related to serious diseases: connective tissue disease, diabetic Mellitus, and cancer. Hence, examining the capillaries status is important as a diagnostic tool.

Capillaries are the smallest-diameter blood vessels which connect arteries to veins. The size of capillaries is about 5 to 10 micrometers in diameter and their diameter averages 8 micrometers [3]. In general, checking capillaries directly is not easy, for example, when examining the capillaries at eye's fundus, the dilation of the pupil is required. However, by assuming the capillary is the track of red blood cells, then capturing the image of capillaries around the root part of a fingernail can be conducted just by a microscope with magnification around 700-900. Thus, it is possible even to

*Email: moch.hafiish.fmipa@um.ac.id

Figure 1: The first two columns are the images of actual capillaries on human fingertip taken by a microscope. The last two columns are the images of artificial capillaries drawn by a hand.

conduct the observations with a cheap microscope and record the changes of patterns easily. The first two columns in Figure 1 show examples of capillaries on human fingertips taken by 700-900 magnification of a microscope. The lens of microscope were located on the skin near the root of a fingernail.

However, diagnosis based on these images is not an easy task because it requires the skill of medical experts (doctors). This skill is unique and not every doctor can perform this diagnosis. Different doctors might give different diagnosis results. The problem is how to learn diagnosis from the skilled doctors and how to develop a consistent diagnosis. To overcome these problems, we propose a CNN classifier. We believe this approach can also possibly lower the cost.

CNN is a mapping from a set of inputs to a set of outputs determined by a set of adjustable parameters. It is the most powerful tool used for a large-scale image recognition [8], [12], and [13]. It is also applied to other classification problems, for example, in [9] and [14]. In addition, it is applied to medical images [2], [6], [11], and [16]. These are the main reasons for choosing CNN.

Generally, CNN requires a large number of images in training data. All these images determine the adjusted parameters during training, which is the process of updating parameters. However, this can be overcome by utilizing the transfer learning approach. The transfer learning means we copy the model and its optimum parameters. This model is a so-called pre-trained. We base our model on a pre-trained VGG16 model for the lower layers of the network. Hence, the input of the images is feeding up to the pre-trained model first, and then the output becomes input to our network. We use Python and the Torch library to implement our CNN model.

The purpose of this research is to train an image classifier implemented as a CNN to judge the state of human capillaries from microscope photographs according to medical motivation. We present two models of CNN. The first model has two categories. The second model has nine categories.

The first CNN model has two categories because the starting point of this paper is the research work by Hung, Nakane, and Hashimoto [5] which developed a judgment category by measuring curve oscillation. Their trial was the following: (i) detecting the edge of the object of capillaries by image processing, (ii) measuring the curvature of the edge, and (iii) calculating the oscillation density. Edge refers to the boundary of the object in the image. Thus, they defined meander category. They presented the images of capillaries related to either healthy or unhealthy persons. According to those images, the simple structure of capillaries which looks like the U-configuration is related to the healthy person. If it has a complicated structure, such as meaningless meandering, it is related to the unhealthy person. The dissertation [4] implemented the partial differential equation approach by applying the Fourier transform on the signed curvature of the edge to classify the status of images of the capillaries into straight and wiggly categories. The straight category is of the simple structure while the wiggly category is of the complicated structure.

Unfortunately, the meaning of complicated structures is very difficult to transfer the infor-

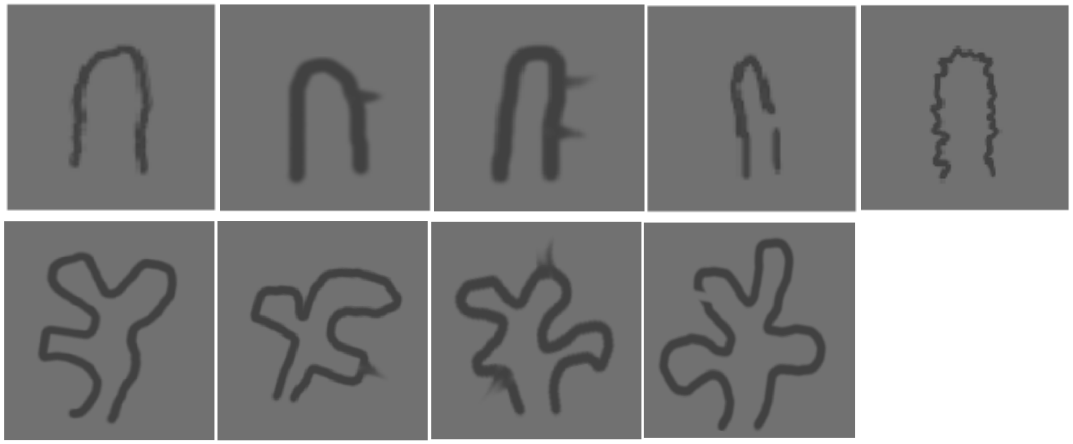

Figure 2: The images of artificial capillaries of each category in the second CNN model. From left to right: the first row is the straight fine, straight 1 branch, straight 2 branches, straight jump, and straight sharp, while the second row is the wiggly fine, wiggly 1 branch, and wiggly 2 branches, and wiggly jump.

mation or technique to other doctors. The first model of CNN can distinguish the structure of capillaries into two categories, but no further information can be added by this model. We think it is insufficient because (i) skill cannot be transferred to other doctors, (ii) the causes and effects are not clear, which is not good for further research, and (iii) new detail explanation cannot be found. To overcome these problems, the first is that we divide the abnormal types of capillaries into several patterns such as meandering, unexpected separation pattern, and branch appearing. The second is by defining a mathematical category for each pattern. The third is by creating artificial data for training. The last is to train the CNN model.

For each pattern, the stage number is given. We give the meandering category by performing the pre-processing which is the partial differential equation approach as in [4]. For the case of unexpected separation pattern and branch appearing, we count the number of 'separation' and 'branch' points and give the indicator. Therefore, we present the second CNN model which has nine categories: straight fine, straight 1 branch, straight 2 branches, straight jump, straight sharp, wiggly fine, wiggly 1 branch, wiggly 2 branches, and wiggly jump. The first category is the simple structure without any abnormal type. The second category is the simple structure with a branch point. The third category is the simple structure with two branch points. The fourth category is the simple structure with a jump. The fifth category is the simple structure with sharp oscillation. The sixth category is the meandering pattern without any separation and branch. The seventh to ninth category is similar to the second to fourth category respectively except the pattern which is meandering type.

We create artificial data and give indicators to each category. These artificial data created by hand-drawing or computer software are mainly used for training. We perform the appropriate pre-processing including a partial differential equation approach to obtain the indicators, especially for dividing the artificial data into straight and wiggly category. We use artificial data to train the first and second CNN model. The last two columns in Figure 1 show artificial data for each category in the first CNN model. Figure 2 shows the artificial data of nine categories in the second CNN model. Thus, both trained CNN have been applied to actual capillary patterns to verify its performance.

The benefit of this approach is the possibility to train CNN by a small number of test data. The other merit is not only the capillaries are classified into straight and wiggly, but the damage patterns of capillaries are also classified. The output is expressed by numbers (score) for each category. This score means how close the input image to the training data of each category. We consider this score when applying the trained model to the images of actual capillary patterns.

Finally, this paper is organized as follows. The first section is introduction. The CNN approach is briefly described in the second section of this paper. The last section shows the results.

2 Convolutional Neural Network

The basis of CNN is a neural network model. Generally, this model is a mapping from a set of inputs to a set of outputs determined by a set of adjustable parameters. This mapping consists of several composed functions known as the layers. Each layer contains several nodes, whereas each node has some parameters. For more detailed explanation of neural networks and CNN, refer to [1].

Training, validation, and test data in this research is based on hand-drawn images. The quantity is increased by stretching, rotating, and adding noise to each image. The angles of rotation are limited because we assume that the input images are not flipped up and down. Each image is labeled by the category where the image belongs to.

All images as the input of VGG16 model are pre-processed, including the partial differential equation approach. These images are resized such that they satisfy the requirement of VGG16. The average of pixel values of each image is subtracted such that it is around zero due to the ReLU activation function. The pixel values are multiplied by $\frac{5}{255}$ to make its pixel values become dense. The denominator is 255 because the original range value of the pixel is integer between 0 and 255. The numerator can be specified by the other number, five is not the only option.

Transfer learning means that the model and its optimum parameters after being trained by the others on a huge amount of training data is being copied. That model is a so-called pre-trained model. The pre-trained model in this research is VGG16. More layers are then added at the end of that model.

VGG is the abbreviation for Visual Geometry Group. This VGG16 model was developed by Simonyan and Zisserman from Department of Engineering Science, University of Oxford. It is a very deep CNN model. It was the basis of their ImageNet Challenge 2014 submission, where their team secured the first and second places in the localization and classification tracks respectively [13]. VGG16 classifies images into 1000 categories. Among these categories, some examples are snakes, spoons, microphones, and paintbrushes. We assume that the complicated structures of capillaries are similar to snakes and that the simple structures are similar to spoons, microphones, or paintbrushes. The input images of VGG16-model are in RGB format of fixed 224×224-size. We use the first 37 layers of this pre-trained model, as shown in Table 1. The output of this pre-trained model becomes the input of our proposed model. Figure 3 shows a summary of the CNN model in this paper.

We propose a model of fully connected layer consisting of five layers. The first layer consists of 256 nodes. Consequently, this layer requires 1,048,832 parameters because the output from the pre-trained model has 4,096 nodes. Let the output of pre-trained model be $\mathbf{y}^{(37)} \in \mathbb{R}^{4096}$. Thus, the output of layer p_1 is

$$\mathbf{y}^{(p_1)} = w^{(p_1)}\mathbf{y}^{(37)} + w_{(0)}^{(p_1)}$$

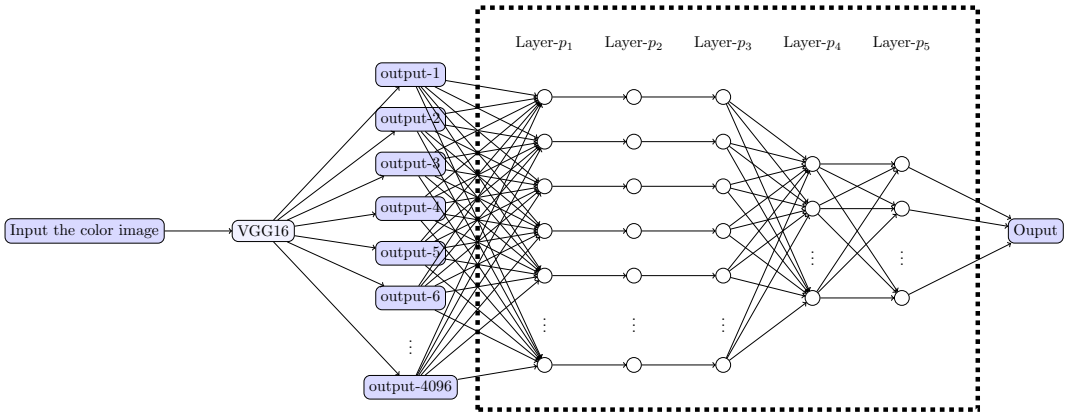

Figure 3: The flow chart of CNN model. The proposed model is in the dashed rectangle.

where $\mathbf{y}^{(p_1)}, w_{(0)}^{(p_1)} \in \mathbb{R}^{256}$ and $w^{(p_1)} \in \mathbb{M}^{256\times 4096}$. The output of layer p_2 is

$$y_{(i)}^{(p_2)} = \max\left(y_{(i)}^{(p_1)}, 0\right)$$

where $i = 1, 2, \ldots, 256$. The Dropout in layer p_3 has factor 0.2 which means 20% of the output of this layer in each channel is deactivated randomly. The output of layer p_3 is

$$y_{(i)}^{(p_3)} = \begin{cases} y_{(i)}^{(p_2)} & \text{if node-}i\text{ is activated} \\ 0 & \text{if node-}i\text{ is deactivated} \end{cases}$$

where $i = 1, 2, \ldots, 256$. Let n be the number of categories. Layer p_4 has output as follows,

$$\mathbf{y}^{(p_4)} = w^{(p_4)}\mathbf{y}^{(p_3)} + w_{(0)}^{(p_4)}$$

where $\mathbf{y}^{(p_3)} \in \mathbb{R}^{256}$ is the output of layer p_3, $\mathbf{y}^{(p_4)}, w_{(0)}^{(p_4)} \in \mathbb{R}^n$, and $w^{(p_1)} \in \mathbb{M}^{n\times 256}$.

The last layer is LogSoftmax, an activation function, where the output of this layer p_5 is

$$y_{(i)}^{(p_5)} = \log\left(\frac{\exp\left(y_{(i)}^{(p_4)}\right)}{\sum_{j=1}^{n}\exp\left(y_{(j)}^{(p_4)}\right)}\right)$$

for $i = 1, 2, \ldots, n$. We propose two CNN models. The first model is of $n = 2$ and the second model is of $n = 9$.

The loss function, in this case, is Average Negative Log Likelihood. Since the final layer score is already LogSoftmax-layer, we need to multiply the output from the final layer by negative one. We define $\mathbf{T}$ as the set of targeted output. Let $M \in \mathbb{Z}^+$ be the number of images in training, $\mathbf{y}^{(p_5,m)} \in \mathbb{R}^n$ be the output of the layer p_5 for m^{th}-image, and $\mathbf{t}_m \in \mathbf{T}$ be the targeted output of m^{th}-image. Since the final layer is LogSoftmax, then the negative log-likelihood is defined as

$$\mathbf{L}(\mathbf{y}^{(p_5)}, \mathbf{t}, \mathbf{w}, M) = \frac{1}{M}\sum_{m=1}^{M} -y_{(i)}^{(p_5,m)}\Big|_{i=\text{argmax }\mathbf{t}_m}$$

where $\mathbf{y}^{(p_5)}$ is the set of the output of layer p_5 for all M images, $\mathbf{t}$ is the set of the targeted output for all M images, and $\mathbf{w}$ is the set of trainable parameters.

Table 1: The model of VGG16 up to layer 37.

Layer (type)	Output shape	# Parameters
Conv2d-1	[64, 224, 224]	1,792
ReLU-2	[64, 224, 224]	0
Conv2d-3	[64, 224, 224]	36,928
ReLU-4	[64, 224, 224]	0
MaxPool2d-5	[64, 112, 112]	0
Conv2d-6	[128, 112, 112]	73,856
ReLU-7	[128, 112, 112]	0
Conv2d-8	[128, 112, 112]	147,584
ReLU-9	[128, 112, 112]	0
MaxPool2d-10	[128, 56, 56]	0
Conv2d-11	[256, 56, 56]	295,168
ReLU-12	[256, 56, 56]	0
Conv2d-13	[256, 56, 56]	590,080
ReLU-14	[256, 56, 56]	0
Conv2d-15	[256, 56, 56]	590,080
ReLU-16	[256, 56, 56]	0
MaxPool2d-17	[256, 28, 28]	0
Conv2d-18	[512, 28, 28]	1,180,160
ReLU-19	[512, 28, 28]	0
Conv2d-20	[512, 28, 28]	2,359,808
ReLU-21	[512, 28, 28]	0
Conv2d-22	[512, 28, 28]	2,359,808
ReLU-23	[512, 28, 28]	0
MaxPool2d-24	[512, 14, 14]	0
Conv2d-25	[512, 14, 14]	2,359,808
ReLU-26	[512, 14, 14]	0
Conv2d-27	[512, 14, 14]	2,359,808
ReLU-28	[512, 14, 14]	0
Conv2d-29	[512, 14, 14]	2,359,808
ReLU-30	[512, 14, 14]	0
MaxPool2d-31	[512, 7, 7]	0
Linear-32	[4096]	102,764,544
ReLU-33	[4096]	0
Dropout-34	[4096]	0
Linear-35	[4096]	16,781,312
ReLU-36	[4096]	0
Dropout-37	[4096]	0
Total params: 134,260,544		
Trainable params: 0		
Non-trainable params: 134,260,544		

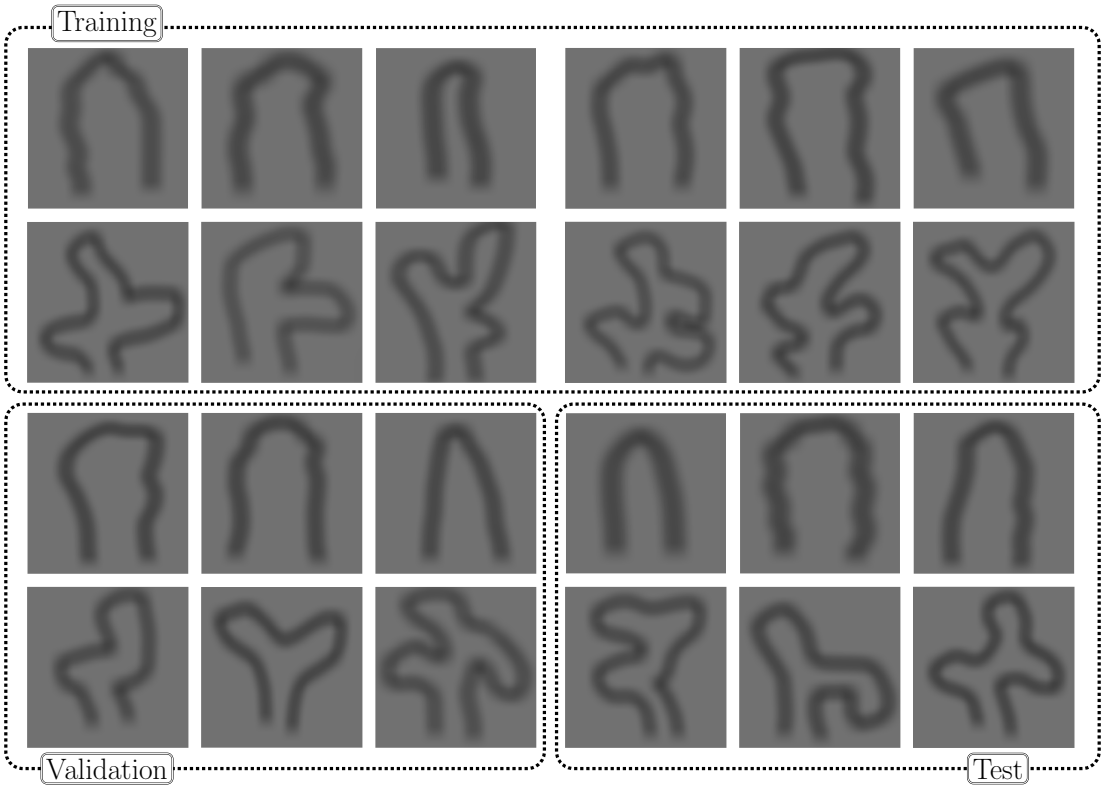

Figure 4: All the hand-drawn images grouped into three blocks: Training, Validation, and Test. The first row of each block is labeled by straight while the second row of each block is labeled by wiggly.

For the first CNN model, the number of categories are two. Thus, the set of the targeted output is $\mathbf{T} = \{(1,0),(0,1)\}$ where $(1,0)$ belongs to the first category and $(0,1)$ belongs to the second category. The second CNN model has nine categories. Thus, the targeted output for the category-i is $\mathbf{t}^{(i)} \in \mathbb{R}^9$ where the i^{th}-component is 1 and the other components are 0.

Adam method is implemented to find the minimizer of $\mathbf{L}$ with respect to the trainable parameters. The name of this method is derived from the adaptive moment estimation. This method computes individual adaptive learning rates for different parameters from the estimation of the first and second moments of the gradients [7].

The output of CNN model is the score of each category. This score means how close the input image to the training data of each category. Increasing the number of training images might get better results. From this score, we determine that the input image is classified to the highest score category. We also state that the image is classified correctly if the highest score category is the same as the label of that image. We define average accuracy by

$$\frac{m_c}{M} \times 100\%,$$

where m_c is the number of images classified correctly and M is the number of input images.

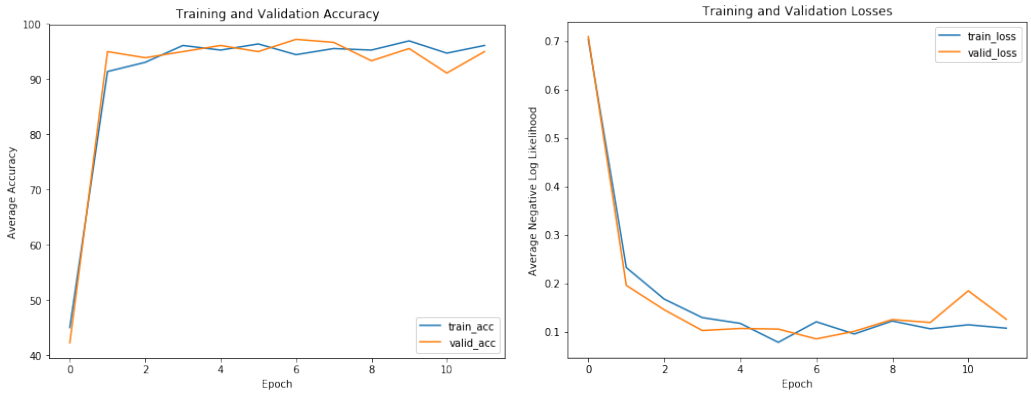

Figure 5: The result of the process of updating parameters of the first CNN model.

3 Results

For the first CNN model, we created 12 hand-drawn images for each category, as shown in Figure 4. Six of them were applied during training, the other three were applied as validation data, while the rests were applied as test data. The number of images was increased by stretching, rotating, and adding noise. The stretch factors were 1.2, 1, and 0.7. These stretch factors are restricted to avoid a wiggly capillary to look like a straight capillary. The angles of rotation were -18, -9, 0, 9, and 18. By this small number of angles, we expect to lower the computational time keeping higher accuracy. We added noise to the copy of each image. Hence, the total number of images in the training data were 360, while the total images in both validation and test data are the same, 180 images.

Figure 5 shows the accuracy and the loss function of training and validation in each time step (epoch) of the first CNN model. The initial parameters are given randomly such that the accuracy is the lowest, whereas the loss is the highest. The accuracy at epoch 2 increases while the loss decreases. After epoch 2, the accuracies are stable above 90%, and the loss is less than 0.2. The parameters at epoch 11 are called the optimum parameters. The accuracy of the training at last epoch is around 95%. It means around 18 of 360 images are classified incorrectly.

The accuracy of the first trained CNN model implemented to the artificial test data is 97%. Out of 180 images, five are classified incorrectly. All these incorrect classifications are from the straight category, as shown in the second row in Figure 6. The first row of Figure 6 shows some examples of the correct classification of the test data from both categories.

From this result, we show that the first CNN model to classify the capillaries into straight and wiggly patterns works successfully. The implementation of this model to the validation data also works similarly to the training results. In addition, when we apply this model to the test data which are different with the training data, the accuracy is similar to the training result. Therefore, it is acceptable to train the large number of parameters by the small number of images in training. The other merit is that the computational cost during the training is reduced.

The first trained CNN model is applied to images of real capillaries. We present eight images of the capillaries in Figure 7. Instead of showing the accuracy, we present the score of each category. The straight category at the top images in Figure 7 has higher score than the wiggly category. The first three columns of the bottom images in Figure 7 have higher scores for the wiggly category. The higher score for the last column of this below image is the straight category, although the capillary has more than two turning points.

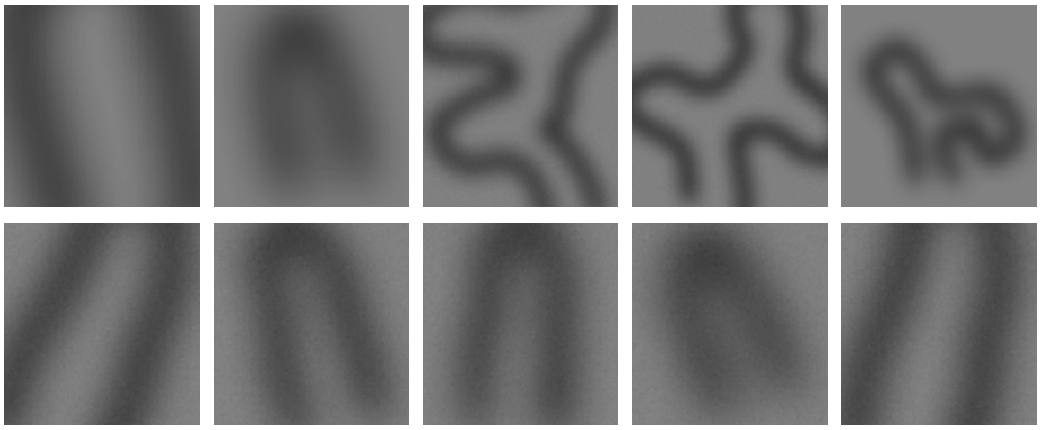

Figure 6: Some images in the test data which the incorrect classification is shown in the second row and the correct classification is shown in the first row.

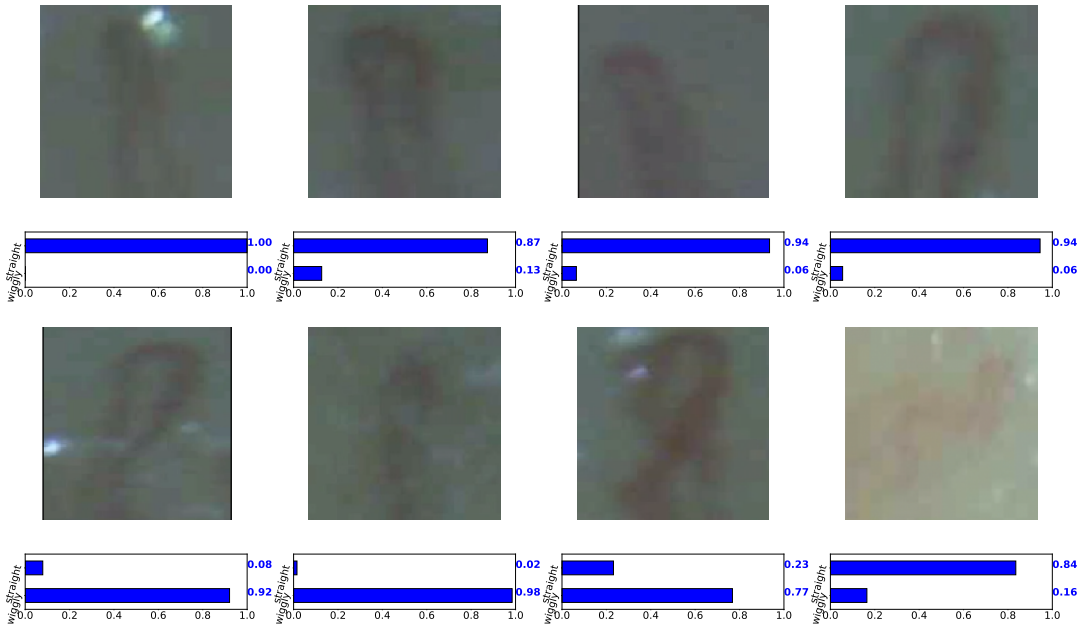

Figure 7: The results of first CNN model applied to the images of capillaries taken by a microscope. The score in each category is presented by the horizontal bar below each image.

For the second CNN model, we create more images for each category. Each image is stretched, rotated, and noise is added. The total is 7,200 images in the training set and each category has 800 images. The total number of images for validation is 2,250.

We perform 24 iterations to train the second CNN model. The accuracies and losses can be seen in Figure 8. The graph of accuracy increases and the loss decreases, but after 3 epochs, both change slowly. The trend of the graph of validation follows the training.

The training accuracy trend of the last six iterations seems flat at around 70%. There is also a decreasing trend of the validation accuracy from the nineteenth to twenty-first iteration. Therefore, the training is stopped after 24 iterations.

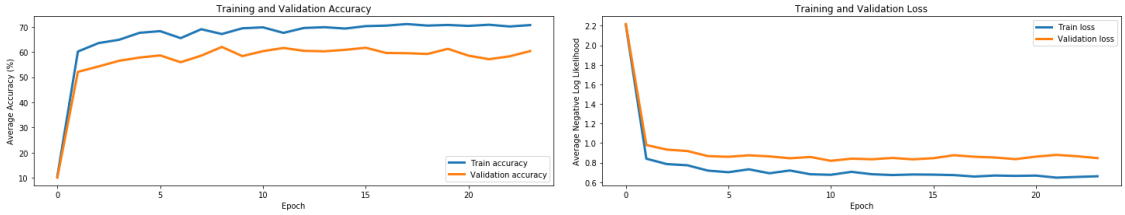

Figure 8: The result of the process of updating parameters of the second CNN model.

The highest training accuracy is 70.8% at the nineteenth, twenty-first, and twenty-third iterations. Among these iterations, the highest validation accuracy occurs at the nineteenth iteration. Hence, we choose the result of the nineteenth iteration as the optimum parameters.

The test data contain $9 \times 750 = 6{,}750$ images. The accuracy when we applied the model to the test data is 63.98%. Table 2 shows the details of the accuracy and loss of each category in the second CNN model. The accuracies of the four categories are more than 87%, but the other four categories have less than 50% accuracy. The lowest accuracy is the category of the wiggly jump, that is 18.93%.

Figure 9 shows the results when the second trained CNN model is applied to the images of actual capillaries. The horizontal bars in the second column show the score of each category. The right vertical axis shows the precise number of each bar. The left vertical axis shows the category number related to Table 2.

The images in the first three rows of Figure 9 show that the second category is the highest score. The fifth image has the highest score for the wiggly fine category. The category of the straight 2 branches has the highest score for the rest images.

Table 2: The accuracy and loss of each category from the second CNN model.

No.	Category	Accuracy	Loss
1	Straight fine	38.13	1.040
2	Straight 1 Branch	95.87	0.161
3	Straight 2 Branches	87.73	0.284
4	Straight Jump	96.53	0.130
5	Straight Sharp	99.87	0.003
6	Wiggly fine	65.47	1.097
7	Wiggly 1 Branch	27.33	1.415
8	Wiggly 2 Branches	46.00	1.215
9	Wiggly Jump	18.93	1.483
Average		63.98	0.759

Conclusions According to the results, we successfully train the first CNN model such that it can classify the images of capillaries into two categories: straight and wiggly. Transfer learning can support the limitation of the number of images in training. For the second CNN model, although

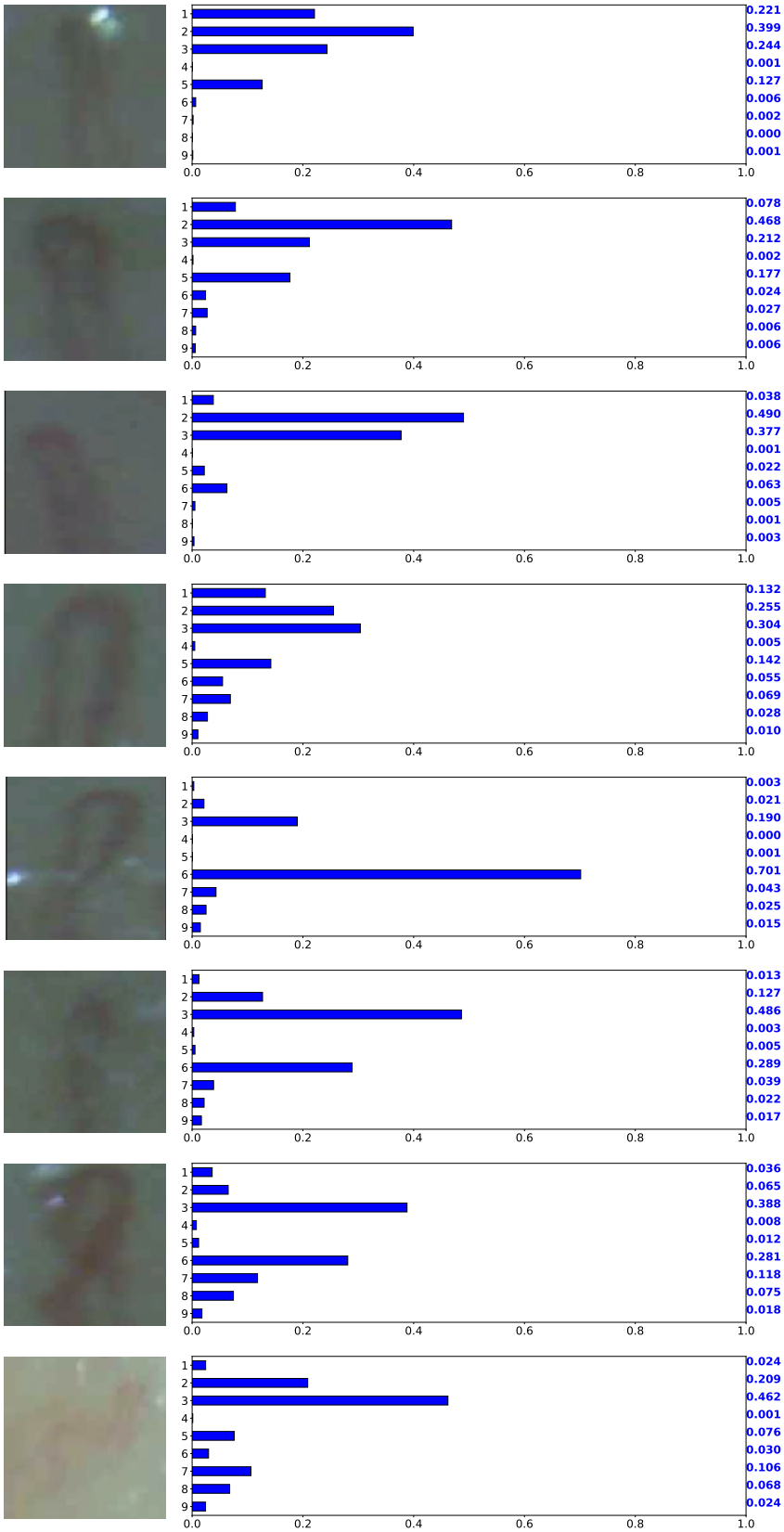

Figure 9: The results of the second CNN model applied to the images of capillaries taken by a microscope. The score in each category is presented by the horizontal bar next to each image.

the average accuracy of nine categories on the artificial test data is 63%, the accuracies of the four categories are less than 50%. We also present the score of each category from both trained CNN models to the actual capillaries. For future research, a large number of values for the angles of rotation would be implemented, and the other pre-trained models would be tested and compared.

Acknowledgements

This research is financially supported by LPDP (Lembaga Pengelola Dana Pendidikan) which is the Indonesia Endowment Fund for Education established by the Ministry of Finance, Republic of Indonesia. The author would like to thank Prof. Seiro Omata as the author's Ph.D. supervisor for the great advice and discussion during the process of completing this paper.

References

[1] Bishop, C. : Pattern recognition and machine learning. Springer. 2006.

[2] Filho, A. O., Silva, A. C., Paiva, A. C., and Nunes, R. A., and Gattass, M. : Classification of patterns of benignity and malignancy based on CT using topology-based phylogenetic diversity index and convolutional neural network, Pattern Recognition, volume: 81, pages 200-212. 2018.

[3] Freitas, R. : Nanomedicine, Volume I: Basic Capabilities, Landes Bioscience: Georgetown, TX. 1999.

[4] Hafiizh, M. : The partial differential approach and the convolutional neural network approach to image processing and its application, Thesis Dissertation. 2019.

[5] Hung, C. N. D., Nakane, K., Ito, T., and Hashimoto, I. : Classification of capillary images based on the average curvature estimation, The Science Reports of Kanazawa University, Volume: 56, page 35 - 44. 2012.

[6] Javia, P., Rana, A., Saphiro, N., and Shah, P. : Machine learning algorithms for classification of microcirculation images from septic and non-septic patients, 17th IEEE International Conference on Machine Learning and Applications. 2018.

[7] Kingma, D., and Ba, J. : Adam: A method for stochastic optimization, In ICLR. 2014.

[8] Krizhevsky, A., Sutskever, I., and Hinton, G. E. : ImageNet classification with deep convolutional neural networks, In NIPS. pp. 1106 - 1114. 2012.

[9] Meijer, D., Scholten, L., Clemens, F., and Knobbe, A. : A defect classification methodology for sewer image sets with convolutional neural networks, Automation in Construction, Volume: 104, page 281 - 289. 2019.

[10] Ogawa, S. : Mousaikekkanzou to rinnshou, Toriumi Shobo. 1994.

[11] Rasti, R., Teshnehlab, M., and Phung, S. L. : Breast cancer diagnosis in DCE-MRI using mixture ensemble of convolutional neural networks, Pattern Recognition, volume: 72, pages 381-390. 2017.

[12] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. : Integrated recognition, localization and detection using convolutional networks. In Proc. ICLR. 2014.

[13] Simonyan, K., and Zisserman, A. : Very deep convolutional networks for large-scale image recognition, Published as a conference paper at ICLR. 2015.

[14] Ting, F. F., Tan, Y. J., and Sim, K. S. : Convolutional neural network improvement for breast cancer classification, Expert System with Application, Volume: 120, page 103-115. 2019.

[15] www.news.yahoo.co.jp. : Wakakute mo yudan dekinai mousaikekkan ga kieteiku gousuto kekkan ' no risuku. 2018.

[16] Xie, H., Yang, D., Sun, N., Chen, Z., and Zhang, Y. : Automated pulmonary nodule detection in CT images using deep convolutional neural networks, Pattern Recognition, pages 109-119. 2019.