Cell image segmentation by using feedback and convolutional LSTM

Eisuke Shibuya¹ · Kazuhiro Hotta¹

Accepted: 18 June 2021 / Published online: 4 July 2021
© The Author(s) 2021

Abstract

Human brain is known to have a layered structure and perform not only feedforward process from lower layer to upper layer, but also feedback process from upper layer to lower layer. Neural network is a mathematical model of the function of neurons, and several models are proposed until now. Although neural network imitates the human brain, everyone uses only feedforward process and direct feedback process from upper layer to lower layer is not used in prediction process. Therefore, in this paper, we propose Feedback U-Net using convolutional LSTM. Our model is a segmentation model using convolutional LSTM and feedback process. The output of U-Net at the first round is fed back to the input, and our method re-considers the segmentation result at the second round. By using convolutional LSTM, the features are extracted well based on the features extracted at the first round. On both of the Drosophila cell image and Mouse cell image datasets, our model outperformed conventional U-Net which uses only feedforward process.

Keywords Feedback · Convolutional neural network · Semantic segmentation · Cell image · Convolutional LSTM

1 Introduction

Human brain is a layered structure, and each layer is the collection of nerve cells called neurons. In addition, feedforward process from lower layer to upper layer and feedback process from upper layer to lower layer are performed in human brain. The lower layer handles low-level information, and the upper layer handles high-level information. Neurons are good at information processing and propagation. A mathematical model of neurons is called a neural network, and a complex function approximation is possible by connecting many layers. Therefore, convolutional neural network (CNN) [19] with convolution layers and pooling layers is effective for image recognition.

Recently, the development of CNN has been successful in various tasks such as image classification [18,28], segmentation [5,33], object detection [10,24] and object tracking [3,20], and image generation [11,36]. Since the accuracy of network is influenced by network's depth, many researchers have been focusing on deepening the network [12,27]. In addition, attention mechanisms [29] that focus on important parts in feature maps can also be used for improving the per-

formance. Squeeze-and-Excitation Networks [14], a kind of attention mechanism, is very useful because it can be used in various models. Collaborative learning that uses multiple networks is also used to improve the accuracy [34].

In recent years, various modes have been proposed for CNN that imitates the human brain, but feedback processing from the upper layer to the lower layer is not used well. Since feedback is used in the visual cortex, we consider that the accuracy will be improved by incorporating it into CNN. In this paper, we proposed Feedback U-Net using convolutional LSTM [26]. This model is further improvement on Feedback U-Net.

Bottom row in Fig. 1 shows our method. Our approach is the only one method which feeds back the output obtained once to the input layer of the network again. In detail, we prepare one network for training. Raw image is fed into the network, and the output at the first round is generated. Loss is not calculated yet. The output at the first round is fed back to the input of the same network, and final segmentation result is obtained. Loss is calculated by using both the final output and ground truth, and network weight is updated by backpropagation. Since the same layers are used twice, we use convolutional LSTM [32] which deals with sequential data. We maintain the features extracted in the first round and extract features in the second round based on the features in the first round. We evaluated our method on both of the

✉ Eisuke Shibuya
160442066@ccalumni.meijo-u.ac.jp

¹ Meijo University, 1-501 Shiogamaguchi, Tempaku-ku,
468-8502 Nagoya, Japan

Fig. 1 Top row shows the structure of human brain. Middle row shows the structure of neural network, and bottom row shows the structure of our method

Drosophila cell image and Mouse cell image datasets. Our proposed Feedback U-Net outperformed conventional U-Net which uses only feedforward process.

This paper organized as follows. Section 2 describes the related works. The architecture of the proposed Feedback U-Net using convolutional LSTM is presented in Sect. 3. Section 4 shows the experimental results on two kinds of cell image datasets. Finally, conclusion is described in Sect. 5.

2 Related works

2.1 Semantic segmentation

Semantic segmentation is a task for assigning class labels to each pixel in an image. Segmentation is used in various fields such as in-vehicle cameras [7] and medical image processing [16,30]. The recent semantic segmentation methods using deep learning are based on fully convolutional network (FCN) [22]. FCN does not use fully connected layers and allows segmentation on images of any size. Further, encoder–decoder structure is also used in semantic segmentation [2].

One of the most famous segmentation models is U-Net [25]. U-Net was proposed for medical image segmentation. An architecture of U-Net is shown in Fig. 2. The most important characteristic of U-Net is skip connection between encoder and decoder. The feature map with the position information in the encoder is concatenated to the restored feature map in the decoder. Therefore, the position information is complemented, and class label can be more accurately assigned to each pixel. In addition,

an improved model has been proposed. U-Net++ [35] integrates multi-scale features. Attention U-Net [23] used the attention mechanism in skip connection. Bridged U-Net [6] used two U-Nets. In addition, Bridged U-Net introduces skip connection and bridging method between two networks. Thus, it is easy to reach convergence and dive into a optimal solution. However, Bridged U-Net used only feedforward processing from lower layers to the upper layer. Furthermore, the number of parameters which has to be set increases because they used two networks.

2.2 Conventional methods using feedback

There is no model that feeds back the output of network to input. For example, multi-stage refinement network like mFCN-PI [4] and stage-wise refinement network [31] used multiple networks. They do not use feedback process, and multi-stage networks are just used. But there are several approaches to feed back the layer's output. RU-Net [1] is a medical image segmentation model which is composed of U-Net and recurrent neural network. RU-Net replaces each convolutional layer with recurrent convolutional layer [21]. Recurrent convolutional layer is a model that the concept of recurrent neural network is adopted to convolutional layer. Figure 3a shows recurrent convolutional layer. In recurrent convolutional layer, the value of state is fed back, and the value is added to the next state. RU-Net repeatedly performs convolution at each scale in recurrent convolutional layer and accumulates feature information. Therefore, feature representation is better than standard convolution. However, since RU-Net repeatedly performs convolution with the same input as shown in Fig. 3, we see that it is not feedback but deepening of network. Furthermore, even if the output of network is fed back in this model, convolution of the first and second rounds is performed independently.

Our approach uses convolutional LSTM instead of recurrent convolutional layer. Convolutional LSTM is convolutional version of LSTM [13], and it deals sequential data. Convolutional LSTM consists of input gate, output gate, forget gate [9], and cell as shown in Fig. 3b. By adding the gate that controls input and output to the conventional recurrent neural network, long-term dependency has been solved; especially, forget gate [9] has the ability to forget unnecessary information from the features maintained in the cell. Convolutional LSTM is also used for predicting the movement of rain clouds [17].

In this paper, the sequential information of the first and second rounds is used. The features extracted in the first round are maintained in the cell, and the features in the second round are extracted based on the maintained features.

Fig. 2 U-Net architecture. Skip connection between encoder and decoder efficiently complements the feature maps

Fig. 3 Recurrent convolutional layers and convolutional LSTM. **a** Recurrent convolutional layer. **b** Convolutional LSTM which consists of input gate, forget gate, output gate and cell

3 Feedback U-net with convolutional LSTM

We made three major changes to U-Net. The first change is to do feedback the output of U-Net to input layer. The second change is addition or concatenation of the outputs at the first round to the outputs at the second round. The third change is the usage of convolutional LSTM. Figure 4 illustrates our method, and the details are explained as follows.

The input for the U-Net is a grayscale image, and we acquire the probability map of each class by a softmax function at the final layer. Segmentation results are obtained from those probability maps. However, in our model, image and

probability maps are the input of network, and probability maps of each class are obtained at the final layer. After that, the input image and probability maps obtained at the first round are fed into the network again, and the probability maps obtained at the second round are used as the final segmentation result.

For example, in the case of segmentation of 4 classes, the input of network is a grayscale image and the probability maps for 4 channels. As the first input, all probability maps are set to 0.25. This means that all pixels have equal probability for all classes. The output of network are the probability maps of 4 classes. Next, we feed back the probability maps obtained at the first round to the input layer.

Fig. 4 Feedback U-Net with convolutional LSTM. In our model, the output is fed back to input layer once. Further, our model replaces convolution layer with convolutional LSTM. We use convolutional LSTM at the locations where local and global features are available

Finally, the output obtained at the second round is used as the final segmentation result. Note that we use the same convolution (LSTM) layers at each round. However, we use different batch normalization for each round as shown in Fig. 5.

Our approach devices to make effective use of the first and second rounds. In the second round, the features before softmax layer in the first round are added or concatenated to the features before softmax layer in the second round. Figure 6 shows the details. In the case of addition, the number of channels does not change before and after the addition of feature maps at the second. Thus, we can perform convolution after the addition at the second round as shown in Fig. 6. However, in the case of concatenation, the number of channels is doubled after concatenation of feature maps. Thus, we used convolutional layer to adjust the number of channels. The output of final convolution at each round has 8 channels. So, the number of channels becomes 16 by the concatenation of the output at first and second rounds. Then, we perform convolution to the feature maps with 16 channels, and we create feature maps with 8 channels. Next, we perform convolution to get the feature map with 4 channels. Finally, we perform softmax function to the feature map with 4 channels, and we obtained the final output.

In ablation study shown in Sect. 4, we try to evaluate the case that we compute subtraction between the probability

Fig. 5 Batch Normalization at each round. Red line shows the first round and green line shows the second round. We use different batch normalization at each round

maps and 0.25. In other words, the input of network at the second round is the difference between probability maps and 0.25.

When feedback is performed in normal convolution layer, only weights are shared. Thus, the features extracted at the first round are unrelated to the features extracted at the second round. In contrast, our approach replaces convolution layer with convolutional LSTM. Since convolutional LSTM has the function that maintains the features extracted before, it is possible to perform convolution based on the features extracted at the first round. Thus, when we extract features at the second round, the features extracted at the

Fig. 6 **a** Case of addition. **b** Case of concatenation. Red lines mean the processing at the first round and green lines mean that at the second round. In concatenation, we used convolution layer to adjust the number of channels

first round are also used so that more useful features can be obtained. To explain in detail, in the first round, convolutional LSTM stores the features from the input image in cell. In the second round, convolutional LSTM takes two inputs: the output of convolutional LSTM at the first round and the input at the second round using feedback information. From these two inputs, convolutional LSTM obtains values between 0 and 1 with input gate, output gate, forget gate. Next, the important information is selected in the cell. Finally, only necessary features from the contents of cell are used as output. In other words, sequential data of the first and second round are used to extract the useful features. In this paper, we put convolutional LSTM at the locations where local and global features are available. Figure 4a–e shows the locations. It is common for two kinds of cell image datasets used in experiments. In location (a), (b), (d), and (e), resolution is the highest and they have local features with location information. Thus, the locations (a), (b), (d) and (e) attempts to complement classes with small area.

Our model is trained with softmax cross entropy loss defined as.

$$L = - \sum_i \sum_x \sum_y \sum_c p_c^{i_{xy}} \log q_c^{i_{xy}} \quad (1)$$

where i means the i -th sample in dataset, x, y mean coordinates, c means the c -th class, $q_c^{i_{xy}}$ is the probability of class c at the coordinate (x, y) of the i -th sample.

4 Experiments

4.1 Datasets and metrics

We use Drosophila cell image dataset [8] as shown in left two columns of Fig. 7. The dataset consists of 4 classes; cytoplasm, cell membrane, mitochondria and synapses. Since the original size is 1024×1024 pixels, we cropped a region of 256×256 pixels from original images due to the size of GPU memory. There is no overlap for cropping areas, and the total number of crops is 320. We used 192 regions for training, 48 for validation and 80 for test. We fixed test images as same as competitions, while changing training and validation sets 5 times.

We also use mouse cell image dataset [15] as shown in two right columns of Fig. 7. The dataset consists of 3 classes: cytoplasm, cell membrane and cell nucleus. We did data augmentation which includes 90 degrees rotations and left–right flip. By the augmentation, we have 400 images from 50 original images. We used 280 for training, 40 for validation and 80 for test. Similar to the Drosophila cell image dataset, we fixed test images while changing training and validation sets 8 times.

In semantic segmentation, Intersection over Union (IoU) is used as evaluation measure. IoU is the overlap ratio between prediction and ground truth labels. In this paper, we use IoU of each class and mean IoU which is the average IoU of all classes.

Fig. 7 Examples of datasets. Left shows Drosophila cell image dataset which consists of cytoplasm, cell membrane, mitochondria, and synapses. Right shows mouse cell image dataset which consists of cytoplasm, cell membrane and cell nucleus

Table 1 Comparison result on the Drosophila cell image dataset

Method	Cytoplasm (%)	Cell membrane (%)	Mitochondria (%)	Synapse (%)	Mean IoU (%)
U-Net	92.2	75.7	73.6	37.1	69.6
Bridged U-Net	92.2	75.2	74.2	43.8	71.4
RU-Net	92.2	75.8	74.3	43.2	71.4
Feedback U-Net without convolutional LSTM					
Addition	91.8	75.1	72.7	38.7	69.6
Concatenation	91.9	75.0	71.8	35.9	68.7
Feedback U-Net with convolutional LSTM					
Addition	92.4	75.7	74.5	39.9	70.6
Concatenation	92.3	75.7	74.5	44.3	71.7

4.2 Implementation details

In this paper, we use Keras library and train our network using Adam for 1500 epochs with a learning rate of 0.0001. Batch size is set to 16 for the Drosophila cell image dataset and 10 for the Mouse cell image dataset. Furthermore, class weight is used to solve class imbalance problem. The number of filters at convolution and convolutional LSTM layers is set to 8, 16, 32, 64 and 128 from the top to bottom of the U-Net.

We compare our method with 4 other methods; conventional U-Net, Bridged U-Net, RU-Net, Feedback U-Net without convolutional LSTM. Bridged U-Net uses two U-Nets. Concatenation is used for bridging and addition is used for skip connection. RU-Net is only the conventional method using recurrent convolutional layer and U-Net. Time-step for RU-Net is set to 2. This is the same with original paper [1]. For Feedback U-Net with/without convolutional LSTM, we compare addition with concatenation described in Sect. 3.

4.3 Comparison with another method

In Table 1, we compare U-Net with our proposed Feedback U-Net with/without convolutional LSTM on Drosophila cell image dataset. Our method with concatenation achieved the best accuracy which is 71.7% on mean IoU. Bridged U-Net and RU-Net provided 71.4%. In contrast, for Feedback U-Net without convolutional LSTM, there is no improvement in accuracy over the baseline. There was a decrease in accuracy in almost classes. We consider that high-level features are lost without convolutional LSTM and IoU decreased.

In Table 2, we also evaluate our method on mouse cell image dataset. The proposed method with concatenation achieved the best accuracy 59.3% on mean IoU. Other conventional methods do not improve the accuracy from U-Net. The images of this dataset have dark and unclear parts. Bridged U-Net and RU-Net are influenced by them.

Table 2 Comparison result on the Mouse cell image dataset

Method	Cytoplasm (%)	Cell membrane (%)	Cell nucleus (%)	Mean IoU (%)
U-Net	77.0	33.3	64.6	58.3
Bridged U-Net	77.2	32.0	63.6	57.6
RU-Net	77.0	32.2	64.0	57.7
Feedback U-Net without convolutional LSTM				
Addition	77.0	33.8	64.5	58.5
Concatenation	76.6	33.9	62.7	57.7
Feedback U-Net with convolutional LSTM				
Addition	77.2	34.7	65.2	59.0
Concatenation	76.8	35.9	65.1	59.3

Fig. 8 Qualitative Results. From left to right, input image, ground truth image, the result by U-Net, Feedback U-Net without convolutional LSTM and our proposed method

Experimental results show that our approach has higher generalization ability than Bridged U-Net and RU-Net. From the results on two kinds of cell image datasets, the effectiveness of our method is demonstrated. Furthermore, we see that concatenation is better to combine the features at each round than the addition.

4.4 Qualitative results

Figure 8 shows the segmentation results by each method. From left to right, input image, ground true image, the results by U-Net, Feedback U-Net without convolutional LSTM and Feedback U-Net with convolutional LSTM are shown. In the case of the Drosophila cell image dataset, Feedback U-Net without convolutional LSTM is worse distinction between cell membrane and mitochondria than U-Net. In contrast, our approach gave good segmentation result for all classes. In the case of the

Mouse cell image dataset, there is no noticeable difference between U-Net and Feedback U-Net without convolutional LSTM, and cell membrane is severely broken. However, our approach improved the accuracy of cell membrane and detects more connected membrane especially above center.

Figure 9 shows the sum of the outputs of the first convolutional layer in Feedback U-Net without convolutional LSTM at the second round. We also show the sum of outputs of the first convolutional LSTM layer in our method at the second round. ReLU function is used after convolution. From left to right, ground truth image, the output of Feedback U-Net without convolutional LSTM and the output of our method are shown. White means 0, and dark red means 255. Thus, it turns out that the feature maps of Feedback U-Net without convolutional LSTM have evenly information all classes. In contrast, the feature maps of ours are highlighted with cell membrane, cell nucleus, mitochondria

Fig. 9 Sum of outputs in the first convolutional layer or convolutional LSTM layer at the second round. From left to right, ground truth, Feedback U-Net without convolutional LSTM and our method are shown

and synapses. In other words, there is more information of these classes than background. According to these results, we consider that our approach complements for the features of object class not background in the second round. This is because our proposed method outperformed conventional methods.

4.5 Ablation study

In Tables 3 and 4, we conduct an ablation study about the locations of convolutional LSTM. Note that locations are shown in Fig. 4a–e. The bottom shows the accuracy of our full model. Both tables show that our method achieved the best accuracy and all convolutional LSTM layers are effective. On the Drosophila cell image dataset, the maximum difference in accuracy was 2.5%. On the Mouse cell image dataset, there are not the results exceeded 59%.

In control engineering, we compute the difference between feedback signal and target value. Thus, we also evaluate our method with/without the difference between feedback information (output probabilities) and input probabilities. Tables 5 and 6 show the evaluation results with/without taking the subtraction in feedback. ○ means that we compute the difference between the probability at the first round and 0.25. On both datasets, there is no significant difference with/without subtraction. When we use the concatenation without subtraction, our model achieved the best accuracy.

4.6 Verification of the third round

In above experiments, we have used the feedback process only once. In this subsection, we conduct an experiment using feedback processing twice. In the case of twice feedback, we add or concatenate three feature maps for the first, the second and the third round before the final layer.

Tables 7 and 8 show the results with feedback without subtraction on both datasets. On the Drosophila cell image dataset, the accuracy of almost all classes was improved from once feedback regardless of addition or concatenation. When we use concatenation, our model achieved the best accuracy which is 72.3% on mean IoU. In contrast, on the Mouse cell image dataset, the accuracy improvement is not observed. We consider that this phenomenon is occurred by the difference of brightness of the datasets. Since Drosophila dataset is clearer than the Mouse cell dataset, useful information is obtained by performing feedback process twice. However, mouse dataset is darker and it may be difficult to obtain new effective features by conducting the third round. Through-out experiments, we demonstrated the effectiveness of our Feedback U-Net with convolutional LSTM.

In addition, we evaluated the proposed method with feedback three and fourth times. Table 9 shows the evaluation results while changing the number of feedback on the Drosophila cell image dataset. In the case of three and four times, the mean IoU does not exceed 71%. We consider that

Table 3 Ablation study on the Drosophila cell image dataset. “a”, “b”, “c”, “d” and “e” are locations where convolutional LSTM layers are placed

Ablation	Cytoplasm (%)	Cell membrane (%)	Mitochondria (%)	Synapse (%)	Mean IoU (%)
a	92.2	75.3	73.3	44.1	71.2
b	91.8	74.9	70.6	39.7	69.2
c	92.0	75.1	73.3	43.7	71.0
d	92.2	75.7	74.5	42.8	71.3
e	92.2	75.9	73.1	39.4	70.2
Ours	92.3	75.7	74.5	44.3	71.7

Table 4 Ablation study on the Mouse cell image dataset

Ablation	Cytoplasm (%)	Cell membrane (%)	Cell nucleus (%)	Mean IoU (%)
a	76.7	35.5	64.2	58.8
b	77.1	34.1	64.0	58.4
c	76.7	35.6	64.4	58.9
d	77.0	35.0	64.8	58.9
e	77.0	34.4	64.2	58.5
Ours	76.8	35.9	65.1	59.3

Table 5 Evaluation with/without the subtraction in feedback on the Drosophila cell image dataset

Subtraction	Add or concat	Cytoplasm (%)	Cell membrane (%)	Mitochondria (%)	Synapse (%)	Mean IoU (%)
○	Add	92.4	76.5	75.1	39.0	70.8
○	Concat	92.3	76.3	74.7	42.2	71.4
×	Add	92.4	75.7	74.5	39.9	70.6
×	Concat	92.3	75.7	74.5	44.3	71.7

Table 6 Evaluation with/without the subtraction in feedback on the Mouse cell image dataset

Subtraction	Add or concat	Cytoplasm (%)	Cell membrane (%)	Cell nucleus (%)	Mean IoU (%)
○	Add	77.0	35.8	64.7	59.2
○	Concat	76.9	36.0	64.7	59.2
×	Add	77.2	34.7	65.2	59.0
×	Concat	76.8	35.9	65.1	59.3

Table 7 Comparison with once feedback on Drosophila cell image dataset

Number of the feedbacks	Add or concat	Cytoplasm (%)	Cell membrane (%)	Mitochondria (%)	Synapse (%)	Mean IoU (%)
Once	Add	92.4	75.7	74.5	39.9	70.6
	Concat	92.3	75.7	74.5	44.3	71.7
Twice	Add	92.3	76.0	75.6	41.0	71.2
	Concat	92.3	76.4	74.9	46.1	72.3

Table 8 Comparison with once feedback on Mouse cell image dataset

Number of the feedbacks	Add or concat	Cytoplasm (%)	Cell membrane (%)	Cell nucleus (%)	Mean IoU (%)
Once	Add	77.2	34.7	65.2	59.0
	Concat	76.8	35.9	65.1	59.3
Twice	Add	77.0	35.4	64.6	59.0
	Concat	76.6	35.2	64.4	58.7

Table 9 Evaluation result while changing the number of feedbacks

Number of the feedbacks	Cytoplasm (%)	Cell membrane (%)	Mitochondria (%)	Synapse (%)	Mean IoU (%)
Once	92.3	75.7	74.5	44.3	71.7
Twice	92.3	76.4	74.9	46.1	72.3
Three times	91.9	75.5	72.6	43.7	70.9
Four times	91.7	75.2	69.3	45.4	70.4

this is due to the rapid decrease in the number of feature maps. For example, in the case of feedback three times, the feature maps at the first, second, third and fourth rounds are concatenated in the fourth round. The feature map has 32 channels (= 8 channels $\times$ 4 rounds). After that, convolution is adopted to the feature map, and the feature map becomes 8 channels. Furthermore, when the number of feedback increases, more processing time is required for both training and test. From those results, we consider that the proposed method with twice feedbacks is the best.

5 Conclusion

In this paper, we proposed Feedback U-Net with convolutional LSTM which used feedback process like human brain. Our results demonstrated the effectiveness of the combination of feedback process from output layer to input layer and convolutional LSTM layer which handles sequential data. Convolutional LSTM makes it possible to extract feature maps of object classes (e.g. cell membrane, cell nucleus, mitochondria and synapses) not background. Furthermore, by conducting feedback process, the accuracy improvement was observed on both datasets. Although we used global feedback from output layer to input layer, local feedback at each resolution block may be effective to improve the accuracy. This is a subject for future works.

Acknowledgements This work is partially supported by MEXT/JSPS KAKENHI Grand Number 18H04746 “Resonance Bio” and 18K11382.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material

in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Alom, M.Z., Hasan, M., Yakopcic, C., Taha, T.M., Asari, V.K.: Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation. *arXiv preprint arXiv:1802.06955* (2018)
2. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: a deep convolutional encoder–decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(12), 2481–2495 (2017)
3. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: *European Conference on Computer Vision*, pp. 850–865. Springer (2016)
4. Bi, L., Kim, J., Ahn, E., Kumar, A., Fulham, M., Feng, D.: Dermoscopic image segmentation via multistage fully convolutional networks. *IEEE Trans. Biomed. Eng.* **64**(9), 2065–2074 (2017)
5. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder–decoder with atrous separable convolution for semantic image segmentation. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 801–818 (2018)
6. Chen, W., Zhang, Y., He, J., Qiao, Y., Chen, Y., Shi, H., Wu, E.X., Tang, X.: Prostate segmentation using 2d bridged u-net. In: *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7. IEEE (2019)
7. Fu, C., Hu, P., Dong, C., Mertz, C., Dolan, J.M.: Camera-based semantic enhanced vehicle segmentation for planar lidar. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 3805–3810. IEEE (2018)
8. Gerhard, S., Funke, J., Martel, J., Cardona, A., Fetter, R.: Segmented anisotropic ssTEM dataset of neural tissue. *figshare* (2013)
9. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: continual prediction with lstm. *Neural Comput.* **12**(10), 2451–2471 (2000)
10. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation.

- In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
11. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (2014)
 12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
 13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
 14. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
 15. Imanishi, A., Murata, T., Sato, M., Hotta, K., Imayoshi, I., Matsuda, M., Terai, K.: A novel morphological marker for the analysis of molecular activities at the single-cell level. In: Cell Structure and Function, p. 18013 (2018)
 16. Khosravian, A., Rahmanianesh, M., Keshavarzi, P., Mozaffari, S.: Fuzzy local intensity clustering (FLIC) model for automatic medical image segmentation. *Vis. Comput.* **37**, 1–22 (2020)
 17. Kim, S., Hong, S., Joh, M., Song, S.K.: DeepRain: Convlstm network for precipitation prediction using multichannel radar data. arXiv preprint [arXiv:1711.02316](https://arxiv.org/abs/1711.02316) (2017)
 18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Commun. ACM* **60**(6), 84–90 (2017)
 19. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proc. IEEE* **86**(11), 2278–2324 (1998)
 20. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High performance visual tracking with siamese region proposal network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8971–8980 (2018)
 21. Liang, M., Hu, X.: Recurrent convolutional neural network for object recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3367–3375 (2015)
 22. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
 23. Oktay, O., Schlemper, J., Folgoc, L.L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N.Y., Kainz, B., et al.: Attention u-net: learning where to look for the pancreas. arXiv preprint [arXiv:1804.03999](https://arxiv.org/abs/1804.03999) (2018)
 24. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(6), 1137–1149 (2016)
 25. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 234–241. Springer (2015)
 26. Shibuya, E., Hotta, K.: Feedback u-net for cell image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 974–975 (2020)
 27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
 28. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
 29. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
 30. Wang, D., Hu, G., Lyu, C.: FRNet: an end-to-end feature refinement neural network for medical image segmentation. *Vis. Comput.* **37**, 1–12 (2020)
 31. Wang, T., Borji, A., Zhang, L., Zhang, P., Lu, H.: A stagewise refinement model for detecting salient objects in images. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4019–4028 (2017)
 32. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: a machine learning approach for precipitation nowcasting. In: Advances in Neural Information Processing Systems, vol. 1, pp. 802–810 (2015)
 33. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2881–2890 (2017)
 34. Zheng, H., Xie, L., Ni, T., Zhang, Y., Wang, Y.F., Tian, Q., Fishman, E.K., Yuille, A.L.: Phase collaborative network for two-phase medical image segmentation. arXiv preprint [arXiv:1811.11814](https://arxiv.org/abs/1811.11814) (2018)
 35. Zhou, Z., Siddiquee, M.M.R., Tajbakhsh, N., Liang, J.: Unet++: a nested u-net architecture for medical image segmentation. In: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, pp. 3–11. Springer (2018)
 36. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2223–2232 (2017)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Eisuke Shibuya received the B.Eng. degree from Meijo University in 2020. Since 2020, he is a student in masters course at the Graduate School of Science and Engineering, Meijo University. His current research interest is deep learning.

Kazuhiro Hotta received the B.Eng., M.Eng., and Dr.Eng. degrees from Saitama University in 1997, 1999, and 2002, respectively. From 1999 to 2002, he was a JSPS research fellow (DC1). From 2002 to 2010, he was an Assistant Professor at the University of Electro-Communications. From 2010 to 2018, he was an Associate Professor at Meijo University. Since 2018, he has been a Professor at Meijo University. In 2012, he was a visiting scholar at the University of Maryland. His research interests include pattern recognition, computer vision and machine learning. He is a member of the IEEE, IEICE, IPSJ, and JSAP.