

Malaria parasite segmentation using U-Net: Comparative study of loss functions

Julisa Bana Abraham*

Department of Electrical Engineering and Information Technology, Faculty of Engineering, Universitas Gadjah Mada, Yogyakarta 55281, Indonesia

Article history:

Received: 15 November 2019 / Received in revised form: 29 November 2019 / Accepted: 1 December 2019

Abstract

The convolutional neural network is commonly used for classification. However, convolutional networks can also be used for semantic segmentation using the fully convolutional network approach. U-Net is one example of a fully convolutional network architecture capable of producing accurate segmentation on biomedical images. This paper proposes to use U-Net for Plasmodium segmentation on thin blood smear images. The evaluation shows that U-Net can accurately perform Plasmodium segmentation on thin blood smear images, besides this study also compares the three loss functions, namely mean-squared error, binary cross-entropy, and Huber loss. The results show that Huber loss has the best testing metrics: 0.9297, 0.9715, 0.8957, 0.9096 for F1 score, positive predictive value (PPV), sensitivity (SE), and relative segmentation accuracy (RSA), respectively.

Keywords: Deep learning, Fully convolutional network; Malaria Parasite; *Plasmodium*; Segmentation

1. Introduction

Deep learning is a compelling and versatile method. This method has been widely used to solve various problems in various fields [1]. Deep convolutional networks are one of deep learning architectures designed for the areas of computer vision and image processing, convolutional networks first proposed by [2] and began to receive attention in the world after Alexnet [3] won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [4] which is an image classification competition using 8 layers of convolutional neural networks (CNN) that are trained using graphics processing units (GPU).

Although CNN is commonly used for classification, convolutional networks can also be used for semantic segmentation using the fully convolutional network [5] approach. U-Net [6] is one example of fully convolutional network architecture capable of producing accurate segmentation on biomedical images, U-Net is designed explicitly for biomedical image segmentation and gets the highest intersection over union (IoU) for ISBI cell tracking challenge 2015. U-Net is trained using data augmentation because the amount of data is very minimal. Due to the excellent U-Net performance for biomedical segmentation, this paper tries to implement U-Net to segment Malaria parasite or *Plasmodium* on thin blood smears.

There have been many studies focusing on the segmentation of *Plasmodium*. Gonzalez-Betancourt et al. [7] propose a system to determine markers for watershed segmentation based

on Radon transform and mathematical operators, this study uses a morphological filter to reduce noise and ensure the preservation of cell edges. Dave et al [8] use adaptive thresholding methods and morphology operations such as erosion and dilation on thin blood smear images. Rosado et al [9] also applied adaptive segmentation and erosion, Somasekar et al [10] also applied thresholding and morphology operators such as dilation and closing to close segmentation holes. Devi et al [11] proposed a *Plasmodium* segmentation system using a controlled watershed marker with a minimum internal marker. Oliveira et al. [12] combine artificial intelligence [13] with mathematical morphology, the search space for the classification operations are reduced using preprocessing by removing the background using erosion to train and test the classifier model. Nugroho et al [14] proposed multilevel Otsu's thresholding combined with closing operator, the model proposed by this paper obtained 96.74 ± 0.7075 %, 76.77 ± 2.1441 %, 99.74 ± 0.1397 %, 97.84 ± 1.2514 % and 96.61 ± 0.8021 % of accuracy, sensitivity, specificity, prediction value positive and prediction value negative, respectively.

From the review that has been done, it can be concluded that the method using mathematical morphology is the most common approach for *Plasmodium* segmentation, but these methods have a weakness that is the need to determine the optimum parameters to produce accurate segmentation, this paper proposes different approach by using U-Net for *Plasmodium* segmentation on thin blood smear images. There are two contributions of this paper, namely, evaluate the performance of U-Net for *Plasmodium* segmentation on thin blood smear images and determining the best loss function for

* Corresponding author. Tel.: +6281515611084; .
Email: julisa.bana.abraham@mail.ugm.ac.id.

Plasmodium segmentation using U-Net.

The remainder of this paper is organized as follows, Section 2 discusses materials and methods which contain detailed datasets, U-Net architecture, and evaluation methods, section 3 is the result and discussion, and section 4 contains conclusions.

2. Materials and Methods

2.1. Dataset

The dataset used in this study was 30 images of *Plasmodium* in thin blood smears with three channels in PNG format used by [14], along with the corresponding ideal segmentation images. The images have a size of 200×200 pixels. Dataset will be divided into two parts: training and testing with the proportion of 21 as training images and nine as testing images. An example of *Plasmodium* images used in this study can be seen in Fig. 1.

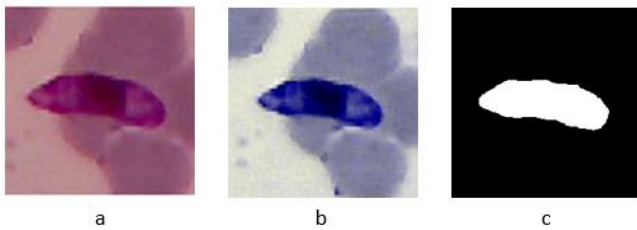


Fig. 1 Example of (a) Original thin blood smear image, (b) Preprocessed by [14], (c) Ideal Segmentation

2.2. Preprocessing and Augmentation

Plasmodium has a small size and is difficult to distinguish from noise such as white blood cells, thus requiring an enhancement method designed to improve segmentation accuracy. Dataset used in this study has been preprocessed before by [14], which has been proven to be optimal for conducting *Plasmodium* segmentation, hence preprocessing used in this study only resizes the images to 224×224 pixels and convert them to grayscale images using (1).

$$Y = 0.299 R + 0.587 G + 0.114 B \quad (1)$$

The reason why images are resized to 224×224 is that U-Net requires some particular input sizes and 224×224 chosen because it is the closest to the original image size of 200x200 pixels. All pixels in the image are normalized to 0-1 to be used to train the U-Net model by dividing each pixel by 255. Due to the training images are very few, data augmentation needs to be done so that the model that has been trained is robust to variance [6]. The data augmentation method in this study uses a commonly used method that is by rotating, shifting, and zooming training images so that the augmented dataset will produce far more images than the original data. In this study, augmentation was carried out on both the thin blood smear images and its corresponding ideal segmentation only on the training dataset until it amounts to 500 images, which are deemed enough to train the U-Net without getting stuck on the local optimum.

2.3. U-Net

There are two parts in the U-Net architecture, the contractive part, and expansive part, this architecture reminds us of the

convolutional autoencoder [15] architecture which also has two parts, encoder, and decoder, but what distinguishes the U-Net architecture from the autoencoder is the U-Net has no fully connected layer, so the U-Net is classified as a fully convolutional network [16], but with a symmetrical contractive and expansive layer

The contractive part is similar to the convolutional network architecture used for classification. By stacking two 3×3 unpadded convolutional layers followed by the ReLu activation function and 2×2 max pooling with 2 strides for downsampling. The expansion part has upsampling followed by a 2×2 convolutional layer or the so-called up-convolutional layer which halves the feature layer, up-convolutional is connected by two 3×3 convolutional layers and the ReLu activation function. Also, the dropout layer is placed between the contractive and expansive layer as regularization with a rate of 0.5 that aims to prevent the model from overfitting. On the last layer, there is a 1x1 convolutional layer that is used to map 64 component feature vectors to the desired output shape. U-Net architecture has a total of 23 convolutional layers [6]. U-Net architecture can be seen in Fig. 3.

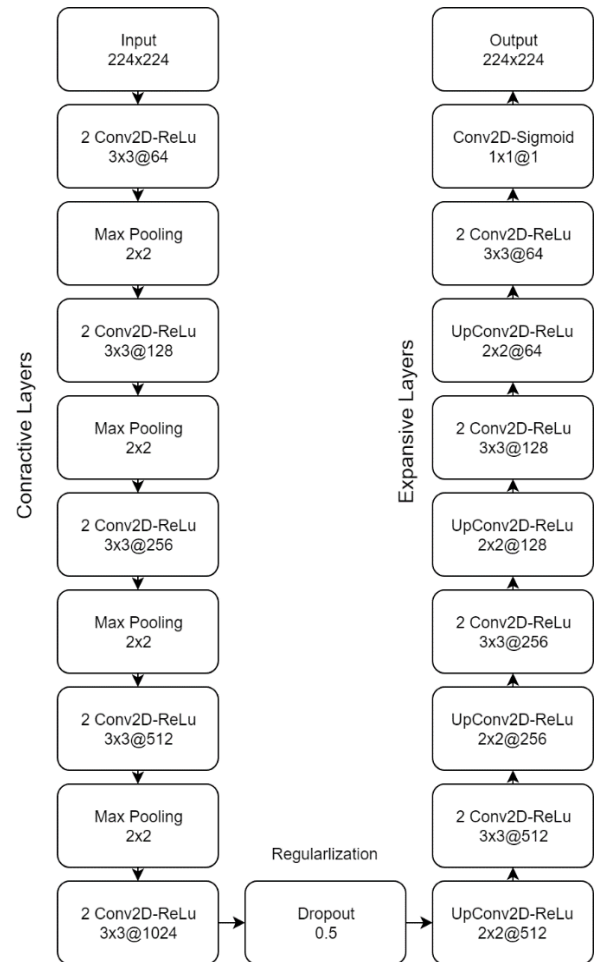


Fig. 2 U-Net architecture

2.4. Training

In this study, *Plasmodium* images and their corresponding ideal segmentation were used to train the U-Net. This study compares several loss functions that are used to train U-Net, namely binary cross-entropy, mean-squared error, and Huber [17]. Unlike the [6], which uses stochastic gradient descent

(SGD), in this paper, Adam [18] with a learning rate of $1e-5$ is used as an optimizer for all loss functions used with the number of epochs equal to 100. Same as [6], the amount of training batch used in this study is 1. The training processes were done on Google Colaboratory with GPU accelerator and take approximately 30 minutes for 100 epochs.

2.5. Binary cross-entropy

Binary cross-entropy (BCE) is a loss function that is often used to train a neural network model for binary classification. The weighted version of BCE can also be used in [6] to train neural network models used for semantic segmentation. BCE equation can be seen in (2).

$$BCE(t, y) = -(t \log(y) + (1 - t) \log(1 - y)) \quad (2)$$

where t is the target and y is the output of the model. In this paper, BCE is implemented using the `binary_crossentropy` function from Keras [19]. Weighted BCE was also tested in this study, but the performance of weighted BCE to reduce false-negative is still lower than the unweighted BCE.

2.6. Mean-squared Error (MSE)

Mean-squared error (MSE) is commonly used to measure the error rate in the curve-fitting model for time-series data, but MSE can also be implemented for image processing such as [20] which uses MSE as a loss function to train the deep convolutional generative adversarial network model. The MSE equation can be seen in (3).

$$MSE(t, y) = \frac{1}{m} \sum_{i=1}^m (t_i - y_i)^2 \quad (3)$$

where t is a vector with dimension m , which is the target and y is a vector with dimension m which is the output of the model. In this case, t and y are the values of each pixel in the image, while m is the dimension of the image. In this study, MSE is implemented using the `mean_squared_error` function found in Keras.

2.7. Huber

Huber loss [17] is, same as MSE, commonly used for regression and curve-fitting problems, but Huber loss is more insensitive to outliers compared to MSE so that the resulting model will be more robust. This study tries to implement the Huber loss as a loss function to train the U-Net model. The Huber loss equation can be seen in (4).

$$H(t, y) = \begin{cases} \frac{1}{2}(t - y)^2 & \text{if } |t - y| \leq \delta, \\ \delta|t - y| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases} \quad (4)$$

where δ is a constant and is set to 0.5 in this study. Huber loss is implemented using the `huber_loss` function on Keras.

2.8. Experiment

The purpose of this study is to evaluate the performance of U-Net for *Plasmodium* segmentation on thin blood smears and determine the best loss function out of the three different loss functions tested in this study. This study applies data

augmentation to overcome the limitations of data to train the U-Net. Data augmentation is only done on 21 training data, while the remaining 9 data used for testing are not augmented.

This study uses four performance metrics: F1, Positive Predictive Value (PPV), Sensitivity, and Relative Segmentation Accuracy (RSA) [21]. Accuracy was not used in this study because it was judged not suitable for measuring performance for segmentation. In this paper, accuracy is substituted by RSA.

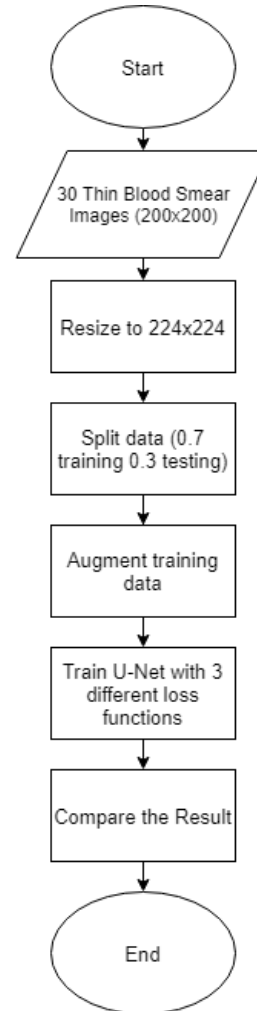


Fig. 3 Experiment Flow Chart

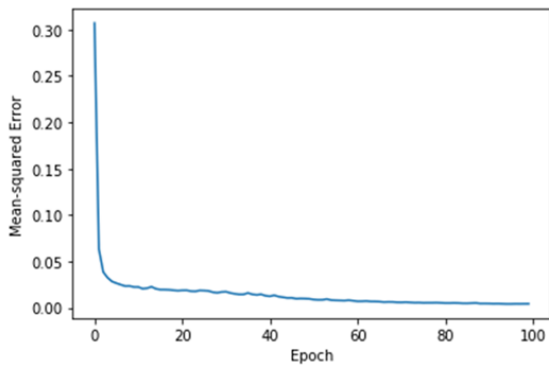
3. Results and Discussion

This section shows the performance of U-Net in segmenting *Plasmodium* on thin blood smears and determining the loss function that is most suitable for this task. This section is divided into two: training results and testing results. It aims to show whether the model being trained experiences overfitting or not.

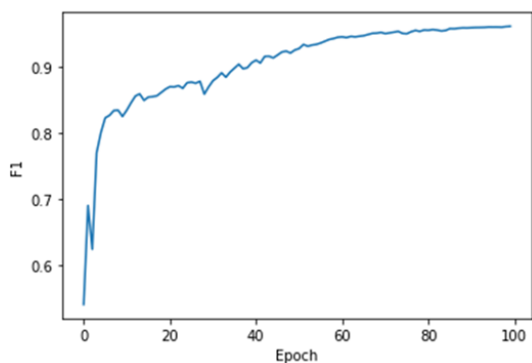
3.1. Training Result

Training results are shown to determine the performance of the model to segment the training data that has gone through the augmentation process. Before discussing the training results metrics, the following graph shows the number of epochs with each loss function tested. Fig. 4-6 shows the graphs of epoch vs loss functions and epoch vs F1 in the training process for MSE,

BCE, and Huber, respectively.

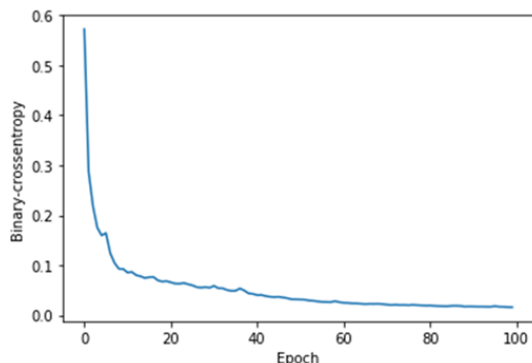


(a)

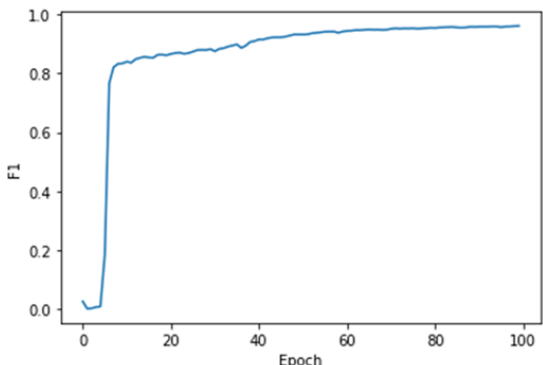


(b)

Fig. 4 (a) MSE vs epoch, (b) Training F1 vs epoch using MSE



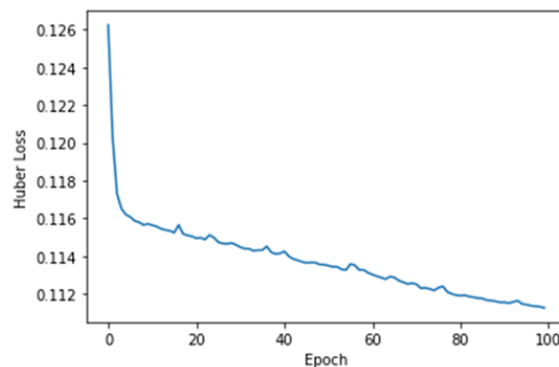
(a)



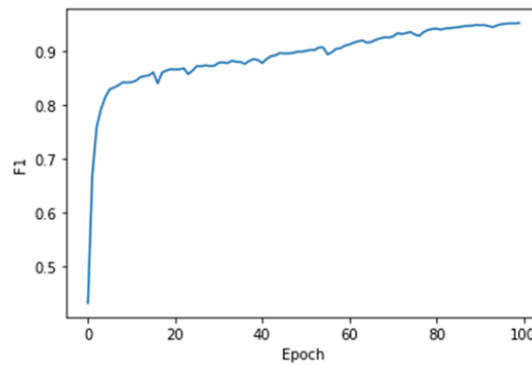
(b)

Fig. 5 (a) BCE vs epoch, (b) Training F1 vs epoch using BCE

To clarify the comparison of training results, Table 1 shows the comparison of performance metrics for three different loss functions.



(a)



(b)

Fig. 6 (a) Huber loss vs epoch, (b) Training F1 vs epoch using Huber loss

Table 1. Training result comparison

Loss Function	F1	PPV	Sensitivity	RSA
MSE	0.9749	0.9934	0.9571	0.9634
BCE	0.9613	0.9639	0.9588	0.9671
Huber	0.9558	0.9935	0.9209	0.9270

The model trained using MSE as a loss function can provide the highest F1, but BCE has the highest sensitivity and RSA, and Huber has the highest PPV.

Another interesting thing to observe is the RSA values, which were originally used to measure the relative accuracy of segmentation for welding defect segmentation, are similar to sensitivity because RSA measures and compares the number of pixels that have a defect or value of 1 between ground truth and segmentation results, which is similar to the calculation of recall or sensitivity, which measures the ratio between true positive relative to the number of a true-positive and false-negative.

3.2. Testing Result

Although the model can have high metrics during the training process, the model may not necessarily have the same metrics if tested using data that has never been seen before. Table 2 shows a comparison of performance metrics from models that have been trained using three different loss functions.

The model trained using Huber loss is able to produce the highest metrics F1, sensitivity, and RSA value compared to MSE and BCE, even though the F1 and RSA metrics during training is lower than MSE and BCE, this shows that the models trained using MSE and BCE may experience overfitting and this does not happen to the model that is trained using

Huber loss. The most observable evidence that the model trained using MSE and BCE experienced overfitting is the difference of the Sensitivity metrics of training and testing that is considerably far about 8% and 12% for MSE and BCE respectively, while the model trained using Huber loss is differ by only 3% on Sensitivity metrics.

Table 2. Testing result comparison

Loss Function	F1	PPV	Sensitivity	RSA
MSE	0.9103	0.9586	0.8722	0.8981
BCE	0.9077	0.9979	0.8324	0.8341
Huber	0.9297	0.9715	0.8957	0.9096

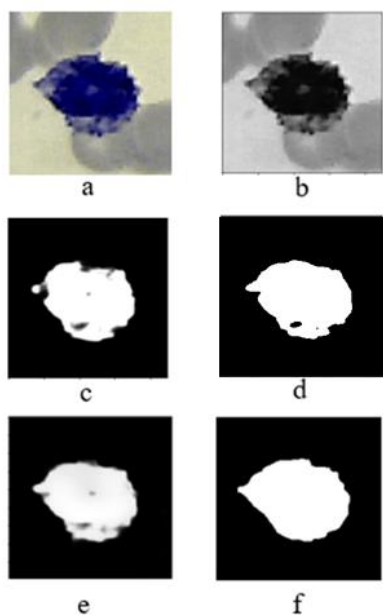


Fig. 7 Example of testing result (a) thin blood smear image, (b) grayscale image. (c) MSE, (d) BCE, (e) Huber, (f) ideal segmentation

Although BCE has the lowest testing F1 value among the three loss functions, BCE has the highest PPV of 99%. This is interesting because for other applications such as localization, a model that has high PPV values can be used as an ROI proposal method, which afterward can be linked to the classifier for type classification for *Plasmodium* or other objects. That is because a high PPV value means it has a low false-positive rate, so it is suitable as an ROI proposal scheme for classification.

The testing results cannot be compared directly with [14] because this paper has a different evaluation scheme and different images size due to the resizing, but it can be said that the U-Net may be able to surpass the sensitivity and F1 values of [14].

4. Conclusion

U-Net can accurately perform *Plasmodium* segmentation on thin blood smear images, besides this study also compares the three different loss functions to train the network, namely mean-squared error, binary cross-entropy, and Huber loss. The results show that Huber loss has the best overall testing metrics: 0.9297, 0.9715, 0.8957, 0.9096 for F1, PPV, Sensitivity, and RSA, respectively. There are several suggestions for further research, which are doing further preprocessing such as different colorspace, doing contrast stretching, noise filtering,

or combining the results of U-Net segmentation with morphology operators so that segmentation results are more accurate than just using the raw output from U-Net. This study has also tried to implement other loss functions such as Dice loss [22], Jaccard distance [23], focal loss [24], and Wasserstein distance [25] but U-Net trained using those loss functions cannot successfully learn *Plasmodium* segmentation on the thin blood smear images, as the loss keeps decreasing, but the metrics do not show any improvement due to the lack of datasets, so further study with more datasets for training should be considering to implement those loss functions.

Acknowledgments

The author would like to thank Dr. Hanung Adi Nugroho from the Department of Electrical and Information Engineering, Universitas Gadjah Mada; Department of Parasitology, Faculty of Medicine, Public Health, and Nursing, Universitas Gadjah Mada, and Eijkman Institute of Molecular Biology for providing the dataset used in this paper.

References

1. A. Shrestha and A. Mahmood, *Review of deep learning algorithms and architectures*, IEEE Access. 7 (2019) 53040–53065.
2. Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 1998, pp. 2278–2324.
3. A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks*, Commun. ACM. 60 (2017) 84–90.
4. O. Russakovsky et al, *Imagenet large scale visual recognition challenge*, Int. J. Comput. Vis. 115 (2015) 211–252.
5. J. Long, E. Shelhamer, and T. Darrell., *Fully convolutional networks for semantic segmentation*, in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 39 (2015) 3431–3440.
6. O. Ronneberger, P. Fischer, and T. Brox, *U-net: convolutional networks for biomedical image segmentation*, In: Navab N., Hornegger J., Wells W., Frangi A. (eds) Medical Image Computing and Computer-Assisted Intervention. MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham.
7. A. González-Betancourt, P. Rodríguez-Ribalta, A. Meneses-Marcel, S. Sifontes-Rodríguez, J. V. Lorenzo-Ginori, and R. Orozco-Morales, *Automated marker identification using the Radon transform for watershed segmentation*, IET Image Process. 11 (2017) 183–189.
8. I. Dave and K. Upla., *Computer aided diagnosis of malaria disease for thin and thick blood smear microscopic images*. In Proceedings of the 2017 4th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2017, pp. 561.
9. L. Rosado, J.M.C. da Costa, D. Elias and J.S. Cardoso. *Mobile-based analysis of malaria-infected thin bloodsmears: automated species and life cycle stage determination*. Sensors 17 (2017) 2167.
10. S. Jalari and B. E. Reddy, *A novel two-stage thresholding method for segmentation of malaria parasites in microscopic blood images*, J. Biomed. Eng. Med. Imaging 4 (2017) 31–31.
11. S. S. Devi, J. Singha, M. Sharma, and R. H. Laskar, *Erythrocyte segmentation for quantification in microscopic images of thin blood smears*, J. Intell. Fuzzy Systems 32 (2017) 2847–2856.
12. A. D. Oliveira, et al., *The malaria system microapp: a new, mobile device-based tool for malaria diagnosis*, JMIR Res. Protoc. 6 (2017) 70.

13. P. Viola and M. Jones, *Rapid object detection using a boosted cascade of simple features*, in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2001, pp. I-I.
14. H. A. Nugroho, et al., *Multithresholding approach for segmenting plasmodium parasite*, 2019 11th International Conference on Information Technology and Electrical Engineering (ICITEE), 2019.
15. J. Yu, D. Huang, and Z. Wei, *Unsupervised image segmentation via stacked denoising auto-encoder and hierarchical patch indexing*, Signal Process. 143 (2018) 346–353.
16. J. Long, E. Shelhamer, and T. Darrell, *Fully convolutional networks for semantic segmentation*, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3431–3440.
17. P. J. Huber, *Robust Estimation of a Location Parameter*, Ann. Math. Statist. 35 (1964) 73–101.
18. D. P. Kingma and J. Ba, *Adam: a method for stochastic optimization*, 3rd International Conference for Learning Representations, San Diego, 2015.
19. F. Chollet, *Deep learning with python*, 1st ed. Greenwich, CT, USA: Manning Publications Co, 2017.
20. A. Radford, L. Metz, and S. Chintala, *Unsupervised representation learning with deep convolutional generative adversarial networks*, (international conference on learning representations , 2016.
21. C. Dang, J. Gao, Z. Wang, F. Chen, and Y. Xiao, *Multi-step radiographic image enhancement conforming to weld defect segmentation*, IET Image Process. 9 (2015) 943–950.
22. L. R. Dice, *Measures of the amount of ecologic association between species*, Ecology 26 (1945) 297–302.
23. M. Levandowsky and D. Winter, *Distance between sets*, Nature 234 (1971) 34–35.
24. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, *Focal loss for dense object detection*, IEEE Trans. Pattern Analysis Mach. Intell. 74 (2018) 2980-2988.
25. I. Olkin and F. Pukelsheim, *The distance between two random vectors with given dispersion matrices*, Lin. Algebra Appl. 48 (1982) 257–263.