International Journal of Engineering & Technology, 7 (4) (2018) 4062-4067

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet. v7i4.18031
Research paper

Vessels semantic segmentation with gradient descent optimization

Wahyudi Setiawan [1] *, Moh. Imam Utoyo [2], Riries Rulaningtyas [3]

[1] Informatics Department, University of Trunojoyo Madura, PO BOX 2 Kamal, Bangkalan, Indonesia
[1,2] Mathematics Department, University of Airlangga, Kampus C Mulyorejo, Surabaya, Indonesia
[3] Physics Department, University of Airlangga, Kampus C Mulyorejo, Surabaya, Indonesia
**Corresponding author E-mail: riries-r@fst.unair.ac.id*

Abstract

Semantic segmentation is a segmentation based on the accuracy of the pixel. Segmentation of retinal vessels aims to differentiate between vessels and non-vessels. There are many conventional methods of retinal vessels segmentation. Conventional methods should perform feature extraction manually. Currently, there has been widely researching on deep learning. Deep learning performs feature extraction automatically during the training phase. One of the methods of deep learning is the Convolutional Neural Network (CNN). The research used three steps i.e creating network layers, training the network, and evaluation of accuracy. We use ten layers of CNN. The CNN architecture consists of Convolution layers, Rectified Linear Unit layers, MaxPooling Layers, Transpose Convolution Layers and Softmax Layer. Furthermore, training the network include the optimization of the training phase. There are three optimization algorithms used for the training phase. The optimization algorithms are Stochastic Gradient Descent with Momentum (SGDM), Root Means Square Propagation (RMSProp) and Adaptive Moment Optimization (Adam). The dataset used for the experiment is Digital Retinal Image for Vessel Extraction (DRIVE). Computation process using the Graphical Processing Unit. Variation scenarios are performed to get the optimal accuracy. The vessels semantic segmentation average accuracy achieved 91.11% - 98.08%.

Keywords: *Convolutional Neural Network; Deep Learning; Gradient Descent; Retinal Vessels; Semantic Segmentation.*

1. Introduction

Semantic Segmentation is a technique for clustering objects in an image. Clustering is based on differences in pixel color intensity [1], [2], [3], [4]. Research on semantic segmentation has grown extensively for autonomous driving [5], robotic vision [6], and medical purposes [7]. One of semantic segmentation research for medical purposes is the segmentation of retinal vessels.

The normal retina contains blood vessels, optic disc, fovea, and macula. The structure of the retinal vessels has a complicated shape. The features of the retinal vessels include length, width, tortuosity or branching pattern and angle direction [8], [9]. Fig.1 is a visualization of the retinal fundus and retinal vessels ground truth.

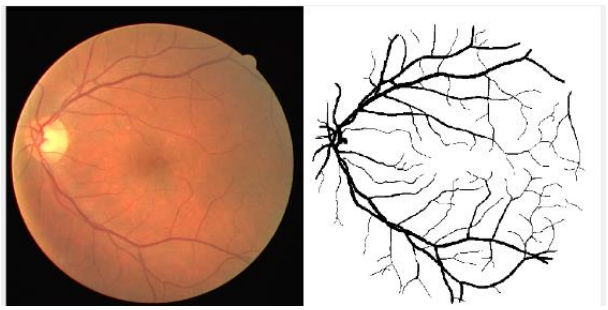


Fig. 1: Visualize Of Fundus Image and Vessels Ground Truth from DRIVE.

Retinal vessel segmentation have been done with various methods such as a matched filter [10], mathematical morphology [11], vessel tracking [12], and machine learning [13], [9]. Currently, research using machine learning especially deep learning has grown widely. Deep learning does not require manual feature extraction. It is automatically performed in the training phase. Deep learning can do classification and segmentation with a lot of data and classes. There are studies about retinal segmentation using deep learning. Melinscak et al. have researched retinal vessel segmentation using Convolutional Neural Network (CNN). This research uses Caffe Architecture. Layers consist of convolution, max-pooling, ReLU, stride, padding and fully connected. The specifications of the device that used for the test are Graphical Processing Unit (GPU), 2 Intel Xeon processors, 64 MB RAM, and Tesla K20C graphics card. The testing dataset using DRIVE. The research achieved 94.66% for accuracy [14].

Maji et al. using a CNN architecture that consists of 11 layers. The layers are convolution, ReLU, and Softmax. The algorithm for training phase optimization uses RMSProp. The test results using the DRIVE dataset achieved an accuracy up to 94.7% [15].

Maninis et al. performing blood vessel segmentation and retinal optic disc using CNN. The CNN architecture used is GoogleNet. The experiment using a single GPU. Dataset using public data include DRIVE, STARE, DRIONSDB, and RIM-ONE. Epoch was very large. It was up to 20,000. Learning rate was very small 10^{-8}. Resulted show Precision Call Region (PCR) is 82.22 % for DRIVE dataset and 83.1% for STARE dataset [16].

Liskowski & Krawiec perform retinal vessel segmentation. Experiment using patches obtained from DRIVE, STARE and CHASEDB datasets. Preprocessing is done to enhance image

Copyright © 2018 Wahyudi Setiawan et. al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

quality. Tests were conducted using a single GPU. The CNN architecture consists of nine layers. The layers consist of four convolution layers, two max pool layers, and three fully connected layers. The accuracy shows 94.73% for DRIVE dataset, 93.49% for STARE dataset and 95.38% for CHASEDB dataset [17].
Pratt et al. present retinal vessels segmentation. It is used the patches of the DRIVE dataset. Size of patches is 21×21 pixels. The number of patches used is 101.416. Architecture using RES18. GPU based experiments. The highest accuracy results achieved 80.27% [18].
The studies have not achieved maximum accuracy and there is no documentation of the time required for the computation process. The novelty of this research is the CNN model consists of 10 layers. Furthermore, an optimization algorithm at the training phase. The optimization method using three different algorithms include Stochastic Gradient Descent with Momentum (SGDM), Root Mean Square Propagation (RMSProp) and Adaptive Moment Optimization (ADAM). The result of the semantic segmentation is compared with ground truth. The dataset experiment was taken from DRIVE [19]. The different parameter is used to get the best accuracy. The parameter is the learning rate and minibatch size. Learning rate are 0.001, 0.05 and 0.01. Minibatch size are 4,8,16,32, and 64. Image input used 128×128 pixel.

2. Research method

2.1. Deep learning

Deep learning is part of machine learning. Deep Learning can do learning itself. It depends on the algorithm used. Deep learning develops the concept of a neural network with more layers inside. Deep learning can be done with supervised and unsupervised learning. Supervised learning uses for classification. Otherwise, unsupervised learning uses for clustering [20].

2.2. Convolutional neural network

Convolutional Neural Network (CNN) is one of the methods in deep learning. CNN is a development of the neural network. The neural network structure consists of the input layer, hidden layer, and output layer. CNN modified hidden layer structure consists of several different layers. The layers are convolution, pooling, ReLU, transposed convolution, softmax and fully connected layer (FCL). Padding operation follows convolution layer. The pooling layer followed by stride operation [21].
CNN has several architectural models. The architecture includes LeNet, AlexNet, ZF Net, GoogleNet, VGGNet, and ResNet. Le Net is used to read zip codes and digits [22]. AlexNet is an architectural model that is widely used to build CNN. AlexNet has a resemblance to LeNet, but the architecture is deeper and larger. Convolution layer feature followed by pooling layer at the beginning of the CNN layer [21].
Furthermore, ZFNet is an architecture development model from AlexNet. ZFNet modifies the architecture hyperparameters, extends the size of the convolutional middle layer, making the stride size and filter at the beginning of the layer smaller [23].
GoogleNet develops an inception module that reduces the number of parameters on the network. The AlexNet 60M can be reduced to 4M. Pooling using average pooling. There is an architectural model development from GoogleNet. It is Inception-v4 [24].
Furthermore, VGGNet contributes to the depth of the network is a significant component to produce better output. The best system generated by VGGNet consists of sixteen convolution layers and fully connected layer. Pretrained models can be used directly at Caffe. VGGNet requires more memory and 140M parameters. Most parameters are in the initial FCL [25]. ResNet uses skip connections as well as batch normalization. The ResNet architecture also eliminates FCL at the end of the network [26].
Convolution layer is a layer that performs convolution operation between input matrix image with a filter matrix. Filter sizes are generally much smaller than the image size. Convolution operation shows in Fig. 2.

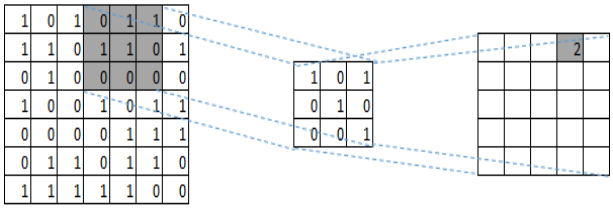


Fig. 2: Convolution Operation.

There is a padding operation in the convolution layer. Padding occurs when the matrix does not match the filter matrix. Padding operation is done at the edge matrix value. There are two options for padding that is using zero padding or valid padding. Zero padding means changing the value of the edge matrix to zero. Valid padding means not changing any value on the edge matrix [27]. Fig. 3 shows zero padding.

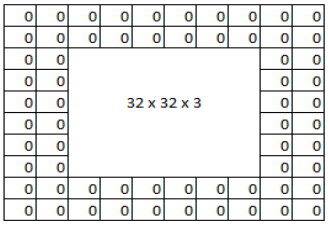


Fig. 3: Zero Padding.

Pooling layer is a layer that reduces the matrix dimension to the quarter size. There are three types of pooling: max pooling, average pooling, and sum-pooling. Max pooling is the most widely used pooling type. Max pooling means taking the maximum value among the existing pixel values. Max-pooling shows in Fig.4. The pooling layer, there is a stride operation. Stride is a sliding windows operation, which means moving the filter windows horizontally or vertically in accordance with the input value [28].

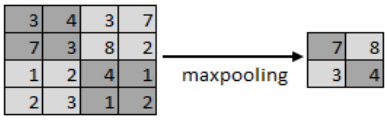


Fig. 4: Max Pooling.

Rectified Linear Unit (ReLU) layer consists of zero and positive value. If the pixel is negative then changed to the zero. The positive value is a fixed value of the matrix. Fig.5 shows the ReLU operation.

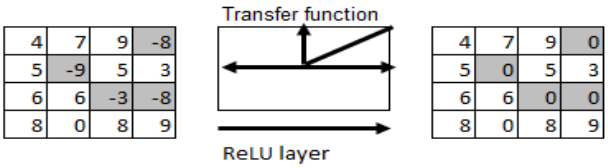


Fig. 5: RELU Operation.

Transposed convolution layer is a layer that performs transposed convolution operation between the input matrix with the filter. Transpose operation exchanged for column and row dimensions.
Furthermore, FCL is a layer that works similar to the neural network. An activation function generally follows the FCL. The neural network has an activation function include linear activation, sigmoid, hard limit and bipolar. CNN activation function uses the softmax [29], [30].

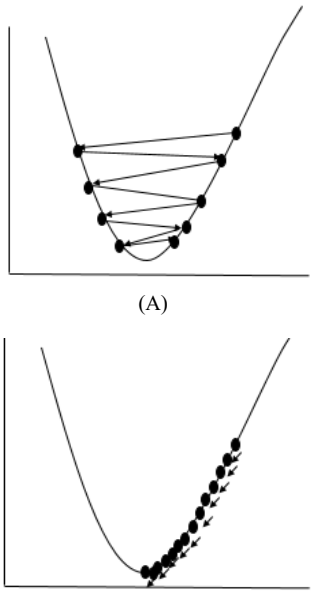
2.3. Training optimization using gradient descent

Gradient descent is an optimization neural network method. Gradient Descent performs the objective function optimization. The

objective function is a set of parameters. Parameters such as learning rate, max epoch, and batch size. Gradient descent works by giving the initial values of the parameters. Moving forward moves iteratively to the number of specific parameters. The goal is to minimize the function. Gradient descent can find the optimal weight value with small error. Time to see the optimal weight value automatically determines the training time [31].
The higher learning rate will affect the greater of the training step. If the learning rate is too large, the gradient descent algorithm becomes unstable. Vice versa, if the learning rate is set too small, it will cause the gradient descent algorithm to converge for a long time. The gradient descent equation is found in equation 1. Fig. 5 shows the movement of the gradient descent function [31].

$$\theta_{t+1} = \theta_t - \alpha\, \delta L\, (\theta_t) \quad (1)$$

With θ_t = weight, θ_{t+1}= next weight, , α = learning rate, $L\ (\theta_t)$ = gradient loss function to θ_t . The gradient descent updates iterative the parameter θ.



(A)

(B)

Fig. 5: Movement of Gradient Descent Function A) Large Learning Rate, B) Small Learning Rate.

Three gradient descent algorithms used in this research included Stochastic Gradient Descent with Momentum (SGDM), Root Means Square Propagation (RMSProp) and Adaptive Moment Optimization (Adam). The purpose of the three algorithms determines the optimal value of each parameter weight. But it has a different way of assessing the value of the learning rate. Learning rate is used to determine the speed of training by minimizing the divergent risk by passing the minimum amount.

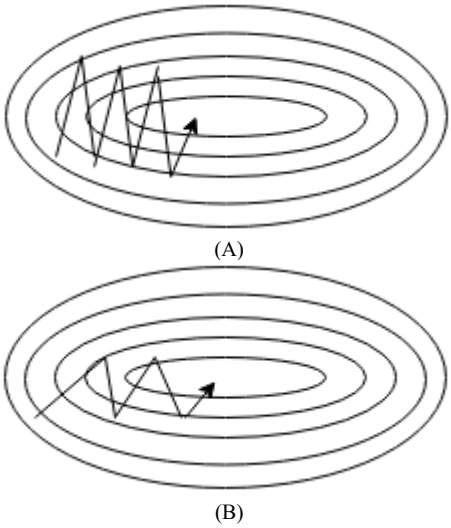
2.3.1. Stochastic gradient descent with momentum (SGDM)

Gradient Descent uses all n samples of the data. There is a faster way to do it. Stochastic Gradient Descent (SGD) use one by one example of the data. SGDM is a development algorithm of SGD with additional momentum. Momentum accelerates the relevant direction and reduces the oscillations. The difference in the movement of SGD without momentum and SGD with momentum is shown in Fig. 6. The SGDM equation is adding the m variable to the vector update.

$$v_{t+1} = \mu\, v_t - \alpha\, \delta L(\theta_t) \quad (2)$$

$$\theta_{t+1} = \theta_t + v_{t+1} \quad (3)$$

with μ = momentum
The μ variable is a constant that governs the amount of the previous gradient. The μ variable is generally set to 0.9



(A)

(B)

Fig. 6: Movement of A) SGD B) SGDM (31).

2.3.2. Root means square propagation (RMSProp)

RMSProp provides different update rates on each dimension of the vector parameter θ_t and can adapt based on specific indicators. One of the symbols used is the change value of the vector gradient in specific dimensions. This change is known as the Adaptive Subgradient Descent (AdaGrad) algorithm. Adagrad has the problem of slow execution time. For that modification is done by adding γ and $1-\gamma$ constant to set the magnitude of g_t^2 and n_t [32].

$$n_t = (1-\gamma)\, g_t^2 + \gamma n_{t-1} \quad (4)$$

$$\theta_t = \theta_{t-1} - \alpha \frac{g_t}{\sqrt{n_t + \varepsilon}} \quad (5)$$

With n_t = dimension , g_t = gradient

2.3.3. Adaptive moment optimization (Adam)

Adam is a first-order gradient-based optimization algorithm. Computationally efficient requires low memory. Adam combines two approaches: momentum and adaptive subgradient [33].

$$m_t = (1-\beta)\, g_t + \beta\, m_{t-1} \quad (6)$$

$$\hat{m}_t = \frac{m_t}{1-\beta^t} \quad (7)$$

$$n_t = (1-\gamma)\, g_t^2 + \gamma\, n_{t-1} \quad (8)$$

$$\hat{n}_t = \frac{n_t}{1-\gamma^t} \quad (9)$$

$$\theta_t = \theta_{t-1} - \alpha \frac{m_t}{\sqrt{n_{t+\varepsilon}}} \quad (10)$$

With m_t = momentum, n_t = adaptive subgradient

2.4. Minibatch size

Minibatch size is a variation of the gradient descent algorithm that divides training datasets into specific batch sizes. Minibatch size can speed up computing time. The minibatch will process according to the size given in parallel. A large of minibatch size will also require considerable memory. Generally, a large minibatch size will converge more slowly when compared to a smaller minibatch

size. However, low values will affect the appearance of noise during the training process.
The minibatch size for the GPU can be adjusted with memory progression. The size used is a power of 2 includes 4,8, 16, 32, and 64

2.5. Cross-entropy loss

Cross-Entropy Loss (CEL) is a measurement of loss using a comparison of probabilities and results of probability predictions. CEL actually for binary probability (0 and 1) [34].

$$Loss(p,q) = -\sum_x p\,(x) \log q\,(x) \tag{11}$$

Herewith p is the probability (0 or 1), q is predicted probability.

2.6 Vessels semantic segmentation

Vessels segmentation based on the difference in pixel intensity. Segmentation is classifying two classes, vessels, and non-vessels. The vessel pixel value is 255 while non-vessel is 0.
A structure of CNN is built. CNN uses 10 layers model. Layer sequence used is convolution layer with padding size 1, ReLU layer, max-pooling layer with stride size 2, convolution layer with padding size 1, ReLU layer, transposed convolution layer with stride size 2, convolution layer, softmax layer, and pixel classification layer output. Fig. 7 shows the configuration of ten layers CNN.

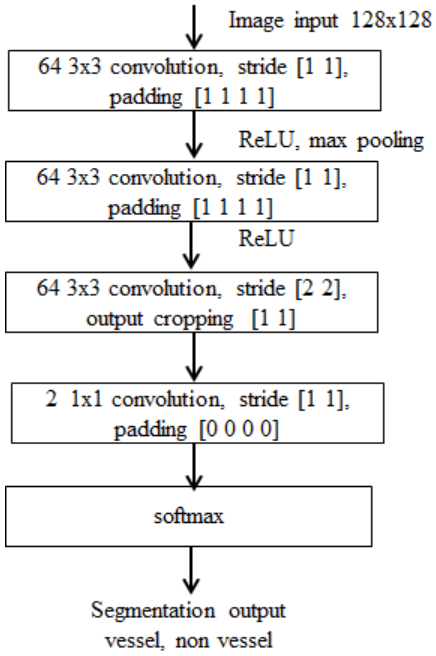


Fig. 7: Sequence of CNN 10 Layers Model.

The next process is to optimize the training data using gradient descent based algorithm included SGDM, RMSProp, and ADAM. The algorithm of vessels semantic segmentation is shown in Fig. 8.

3. Result and discussion

The data used for testing is data DRIVE in the test section. Data consists of 20 RGB images and 20 ground truth images. The ground truth image used is the second expert. The experiment was done on a single Graphical Processing Unit (GPU) with specifications of Core i7 7700, MSI Z270 a pro, GTX 1060 6 Gb D5 amp, DDR4 16 Gb, PSU 800w, SSD 60 Gb.

INPUT : Load Images, Groundtruths

PROCESS :

1. Create datastore for Images
2. Create datastore for Groundtruths
3. Create CNN architecture
   (numfilters, filterSize, numClasses, Layers)
4. Setup training with optimization algorithms
   (trainingOptimizationsAlgoritms, initialLearningRate, maxEpochs, minibatchSize)
5. Create trainingData that contains datastore
   for Images and Groundtruths
6. Train the Network
   (trainingData, Layers, Setup Training Optimizations)

OUTPUT :

Accuracy, Computation time

Fig. 8: Algorithms of Vessels Semantic Segmentations.

The experiment has been carried out on 32 × 32, 64 × 64, 128 × 128, 256 × 256 and 512 × 512 pixels. The best results are obtained at 128 × 128. For this reason, this article only reports the experiment with 128 × 128 image size.
The experiment scenario consists of three parts, include:

a. Test using SGDM optimization algorithm with learning rate value of 0.001, 0.05, and 0.01.
b. Test using the RMSProp optimization algorithm with learning rate value of 0.001, 0.05, and 0.01.
c. Test using Adam's optimization algorithm with learning rate value of 0.001, 0.05, and 0.01.

The results of the first scenario are shown in Table 1. The table explains the performance measures of accuracy, cross-entropy loss, iterations and time (in seconds). The results show the best accuracy up to 100% in learning rates of 0.001 and 0.05. Minibatch sizes 4 and 8. Time to computed the network between 17 and 24 seconds. While a cross-entropy loss is between 0.0017 and 0.0025.
The second scenario results are shown in table 2. The table shows the best accuracy of up to 99.97% at a learning rate of 0.01. Minibatch size 4. The time computed the network is between 17 and 24 seconds. While a cross-entropy loss is between 5e-5 and 0.0132.
Table 3 is the result of the third scenario. The results show the best accuracy up to 99.78% at the learning rate of 0.05. Minibatch size 8 and 16. For the last, the time needed for the network is between 17 and 25. A cross-entropy loss is between 4e-8 and 0.0023.
Furthermore, Table 4 presents the average accuracy, cross-entropy loss and the time to computed the network. The average accuracy for the SGDM, RMSProp and Adam algorithms is 98.8%, 92.09%. and 91.11%. The average cross-entropy loss is 0.0021, 0.0023, and 0.0006. while the average time to compute the network is 20.6, 20.27, and 20.53 seconds. The scenarios are used epoch 100. For minibatch size 16, 32, and 64, it is needs 100 iterations. While for minibatch size 8 requires iterations up to 200. Minibatch size 4 requires iterations up to 500.
Eventually, Table 5 shows the comparison of the accuracy of the previous research with the accuracy in this article. The results show that the ten layers of the CNN network can achieve good accuracy above 90%.

Table 1: The Result with SGDM Algorithm

Mb size	Iterations	Learning rate = 0.001			Learning rate = 0.05			Learning rate = 0.01		
		Accuracy	Loss	Time	Acc.	Loss	Time	Acc.	Loss	Time
64	100	92.74	0.0024	22	99.96	0.0022	21	99.96	0.0023	22
32	100	94.82	0.0024	20	99.96	0.0022	21	99.96	0.0023	21
16	100	95.82	0.0025	17	99.95	0.0025	17	99.95	0.0025	18
8	200	88.29	0.0025	19	100	0.0017	21	100	0.0023	20
4	500	99.79	0.0022	23	100	0.0004	23	100	0.0017	24

Table 2: The Result with RMSProp algorithm

Mb size	Iterations	Learning rate = 0.001			Learning rate = 0.05			Learning rate = 0.01		
		Accuracy	Loss	Time	Acc.	Loss	Time	Acc.	Loss	Time
64	100	93.03	0.0002	21	95.1	0.0028	20	99.66	0.0006	20
32	100	89.74	0.0003	20	90.08	0.0007	20	92.85	0.0006	20
16	100	91.28	0.0003	17	96.41	0.0071	17	99.95	0.0007	17
8	200	82.09	0.0006	20	99.9	0.0067	20	98.04	6e-5	20
4	500	87.57	5e-5	24	65.98	0.132	24	99.97	0.0027	24

Table 3: The Result with Adam Algorithm

Mb size	Iterations	Learning rate = 0.001			Learning rate = 0.05			Learning rate = 0.01		
		Accuracy	Loss	Time	Acc.	Loss	Time	Acc.	Loss	Time
64	100	93.14	0.0002	21	98.93	0.001	20	91.5	0.0003	20
32	100	93.2	0.0002	20	99.71	0.0009	20	68.87	0.0002	20
16	100	91.29	0.0002	18	99.78	0.0009	17	94.96	0.0003	17
8	200	98.3	9e-6	22	99.78	0.0015	20	71.89	5e-7	20
4	500	97.41	4e-8	25	100	0.0007	24	68.3	2e-5	24

Table 4: Average Accuracy, Loss, and Time Processed

Algorithm	Accuracy	Loss	Time (second)
SGDM	98.08	0.0021	20.6
RMSProp	92.09	0.0023	20.27
Adam	91.11	0.0006	20.53

Table 5: Performance Comparison Accuracy

Author	Dataset	Accuration
Melinscak et al. [14]	DRIVE	94.66 %
Maji et al. [15]	DRIVE	94.7 %
Maninis et al. [16]	DRIVE	82.2 %
Liskowski et al. [17]	DRIVE, STARE, CHASEDB	94.73 %, 93.49%, 95.38 %
Pratt et al. [18]	DRIVE	80.27 %
Proposed method	DRIVE	91.11 % - 98.08%

4. Conclusion

Semantic segmentation of vessels has been built. It is used CNN architecture with ten layers. Variables that affect the level of accuracy and time consumed include optimization algorithm, learning rate, and minibatch size.

For further research, CNN architecture can be modified. In addition, the development of gradient descent optimization algorithms is needed to improve performance.

Acknowledgment

We would like to acknowledge the Ministry of Research, Technology and Higher Education, Indonesia for funding this research. We also thanks to DRIVE dataset as experimental data.

References

[1] Thoma M. A Survey of Semantic Segmentation. 2016; 1–16. Available from: http://arxiv.org/abs/1602.06541

[2] Hong S, Noh H, Han B. Decoupled Deep Neural Network for Semi-supervised Semantic Segmentation. 2015 1–9. Available from: http://arxiv.org/abs/1506.04924

[3] Shelhamer E, Long J, Darrell T. Fully Convolutional Networks for Semantic Segmentation.2016; 1–12. Available from: http://arxiv.org/abs/1605.06211

[4] Lempitsky V, Vedaldi A, Zisserman A. A Pylon Model for Semantic Segmentation.2011 (228180):1–9. Available from: http://eprints.pascal-network.org/archive/00008426/

[5] Chen C, Seff A, Kornhauser A, Xiao J. DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving. Princet Vison&Robotics [Internet].2018 ;(Figure 1):1. Available from: http://deepdriving.cs.princeton.edu/

[6] Lenz I, Lee H, Saxena A. Deep learning for detecting robotic grasps. Int J Rob Res. 2015; 34(4–5):705–24. https://doi.org/10.1177/0278364914549607.

[7] Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, et al. A survey on deep learning in medical image analysis. Med Image Anal. 2017; 42(December 2012):60–88.

[8] Fraz MM, Barman SA, Remagnino P, Hoppe A, Basit A, Uyyanonvara B, et al. An approach to localize the retinal blood vessels using bit planes and centerline detection. Comput Methods Programs Biomed [Internet].2012; 108(2):600–16. Available from: https://doi.org/10.1016/j.cmpb.2011.08.009.

[9] Welikala RA. Automated Detection of Proliferative Diabetic Retinopathy from Retinal Images. 2014 ;(September).

[10] Li Q, You J, Zhang D. Vessel segmentation and width estimation in retinal images using multiscale production of matched filter responses. Expert Syst Appl [Internet]. 2012; 39(9):7600–10. Available from: https://doi.org/10.1016/j.eswa.2011.12.046.

[11] Fraz MM, Remagnino P, Hoppe A, Uyyanonvara B, Owen CG, Rudnicka AR, et al. Retinal vessel extraction using first-order derivative of gaussian and morphological processing. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics). 2011; 6938 LNCS (PART 1):410–20. https://doi.org/10.1007/978-3-642-24028-7_38.

[12] Morales S, Naranjo V, Fernando L, Alca M, Paristech M. Determination Of Retinal Network Skeleton Through Mathematical Morphology. In: Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European. 2014. p. 1–5.

[13] Hameed IS, Ocbagabir H, Barkana BD, Yildirim B. Blood vessel segmentation in retinal images by morphological operations and by a novel pixel-tracking algorithm. Int J Innov Comput Inf Control. 2015; 11(1):189–201.

[14] Melinscak M, Prentasic P, Loncaric S. Retinal Vessel Segmentation Using Deep Neural Networks. 2012; 1–6.

[15] Maji D, Santara A, Mitra P, Sheet D. Ensemble of Deep Convolutional Neural Networks for Learning to Detect Retinal Vessels in Fundus Images.
[16] Maninis K, Pont-tuset J, Arbel P, Gool L Van. Deep Retinal Image Understanding.
[17] Liskowski P, Krawiec K. Segmenting Retinal Blood Vessels with Deep Neural Networks. IEEE Trans Med Imaging. 2016; 0062(c):1–12. https://doi.org/10.1109/TMI.2016.2546227.
[18] Pratt H, Williams BM, Ku JY, Vas C, Mccann E, Al-bander B, et al. Automatic Detection and Distinction of Retinal Vessel Bifurcations and Crossings in Colour Fundus Photography †. 2018;1–14.
[19] Staal J, Member A, Abràmoff MD, Niemeijer M, Viergever MA, Ginneken B Van, et al. Ridge-Based Vessel Segmentation in Color Images of the Retina. 2004; 23(4):501–9.
[20] Deng L, Yu D. Deep L Earning : 2014. 1-134 p.
[21] Krizhevsky A, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. : 1–9.
[22] Lechun. Gradient-based learning applied to document recognition. proc.of IEEE. 1998 ;(november):1–46.
[23] Zeiler MD, Fergus R. Visualizing and Understanding Convolutional Networks arXiv: 1311.2901v3 [cs.CV] 28 Nov 2013. Comput Vision–ECCV 2014 [Internet]. 2014; 8689:818–33. Available from: http://link.springer.com/10.1007/978-3-319-10590-1_53%5Cnhttp://arxiv.org/abs/1311.2901%5Cnpapers3://publication/uuid/44feb4b1-873a-4443-8baa-1730ecd16291
[24] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit. 2015; 07–12–June: 1–9.
[25] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition.2014; 1–14. Available from: http://arxiv.org/abs/1409.1556
[26] He K, ZHang X, Ren S, Sun J. Deep residual learning for image steganalysis. Multimed Tools Appl. 2015; 1–17.
[27] Dumoulin V, Visin F. A guide to convolution arithmetic for deep learning.2018; 1–31.
[28] Nagi J, Ducatelle F. Max-pooling convolutional neural networks for vision-based hand gesture recognition. 2011 IEEE Int Conf Signal Image Process Appl [Internet]. 2011; 342–7. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6144164
[29] Wang C, Xi Y. Convolutional Neural Network for Image Classification. 1997 ;(1969). Available from: http://www.cs.jhu.edu/~cwang107/files/cnn.pdf
[30] Sainath T, Vinyals O, Senior A, Sak H. Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks. ICASSP, IEEE Int Conf Acoust Speech Signal Process - Proc. 2015; 2015–Augus: 4580–4.
[31] Ruder S. An overview of gradient descent optimization. 2016; 1–14.
[32] Hinton GE, Srivastava N, Swersky K. Lecture 6a- overview of mini-batch gradient descent. COURSERA Neural Networks Mach Learn [Internet]. 2012; 31. Available from: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf
[33] Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. 2014; 1–15. Available from: http://arxiv.org/abs/1412.6980
[34] Janocha K, Czarnecki WM. On Loss Functions for Deep Neural Networks in Classification. 2017 1–10. Available from: http://arxiv.org/abs/1702.05659.