22-24 June 2021

Zilina - Slovakia

Proceedings of

The International Conference on

Information and Digital Technologies 2021

IEEE ESRA SRDA CeBMI educational centre

2021 International Conference on Information and Digital Technologies (IDT) | 978-1-6654-3692-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/IDT52577.2021.9497550

The International Conference on

Information and Digital Technologies 2021

22-24 June 2021
Zilina, Slovakia

Co-funded by the
Erasmus+ Programme
of the European Union

ISBN 978-1-6654-3692-2

ISSN 2575-677X

IEEE Catalog Number CFP21CDT-ART

Contents

The U-Net model application for retinal vessels segmentation using the machine learning library TensorFlow

Vasyl Martsenyuk
Department of Computer Science and Automatics
University of Bielsko-Biala
Bielsko-Biala, Poland
vmartsenyuk@ath.bielsko.pl

Roksolana Milian
Department of Mathematics and Methods of its Teaching
Ternopil Volodymyr Hnatiuk
National Pedagogical University
Ternopil, Ukraine
roksolana.milian@gmail.com

Nazar Milian
Department of Cybersecurity
Ternopil National Technical University
Ternopil, Ukraine
nazar.milyan@gmail.com

***Abstract*— In this article the implementation of neural network architecture based on a dense U-Net network is proposed. It is noted that retinal blood vessels are the basis for clinical diagnosis of some diseases. A review of the convolutional networks use for classification tasks and generalizion retinal vessel segmentation algorithms is performed. The general process of the neural network is presented. The differences between the real and the obtained results were evaluated. Evaluation of the neural network is carried out on several parameters. The figure with the recognized blood vessels as a result of the model is presented.**

***Keywords* — machine learning, neural network, machine learning library, retinal vessels segmentation.**

I. INTRODUCTION

Retinal blood vessels are the basis for the clinical diagnosis of some diseases. Achieving automatic retinal vessel segmentation in fundus imaging is an important and challenging task. This paper proposes neural network architecture based on a dense U-Net network.

Convolutional networks have existed for a long time [27], their success was limited due to the size of the available training kits and the size of the considered networks. Kryzhevsky's breakthrough [1] was due to the controlled learning of a large network with 8 layers and millions of parameters of the ImageNet data set with 1 million training images. Since then, even larger and deeper networks have been taught [14].

The typical use of convolutional networks is used for classification tasks, where the source image is one class label. However, in many visual tasks, especially in biomedical image processing, the desired result should include localization, ie the class label should be assigned to each pixel. Moreover, thousands of educational images are usually inaccessible for biomedical tasks. Thus, Ciresan, Gambardella, Giusti and Schmidhuber [6] taught a network in a sliding window installation to predict the class label of each pixel by providing a local patch around that pixel. First, this network can be localized. Second, the training data in terms of patches is much larger than the number of training images.

II. MATERIAL AND METHODS

It is worth considering the network architecture shown in Figure 1, which is illustrated in [17]. It consists of a contracting path (left side) and an expansive path (right side). The laying path corresponds to the typical architecture of the convolutional network. It consists of repeated application of two 3x3 convolutions (unchanged convolutions), each of which is a rectified linear unit (ReLU) and a combination operation of max. 2x2 with step 2 to reduce the sample. At each sampling step, the number of functional channels is doubled. Each step in the expansive path consists of a resampling of the object map, followed by a 2x2 convolution (“convolution from above”), which halves the number of object channels, concatenation with a properly truncated object map from the contracting path and two 3x3 convolutions, each of which is accompanied by ReLU. Trimming is necessary due to the loss of border pixels in each gyrus. On the final layer, the 1x1 convolution is used to map each 64-component feature vector to the desired number of classes. In total, the network has 23 convolutional layers.

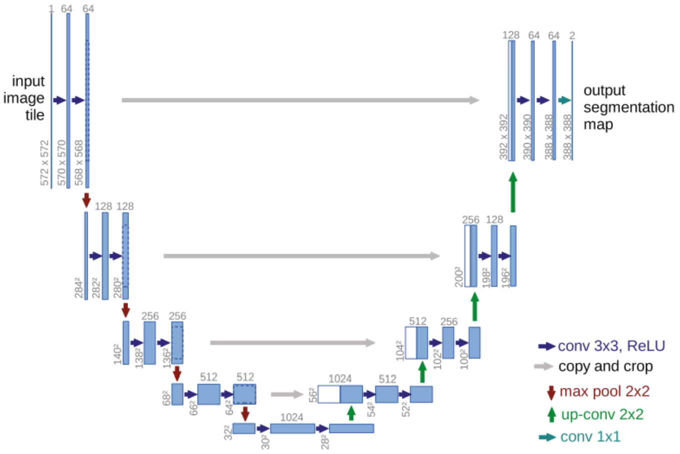


Figure 1. U-Net architecture (example for 32x32 pixels with the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is indicated at the top of the window. Size x-y is located in the lower left edge of the box. White boxes represent copied object maps. Arrows indicate various operations.

Retinal blood vessels contain significant information regarding human health. Observation of the morphological structure of retinal vessels in fundus images is used not only for screening retinal vessel diseases, but also for auxiliary diagnosis of other diseases such as stroke [5], hypertension [21], diabetes-induced retinopathy [10] and glaucoma [9]. Images of the fundus have the following characteristics: low contrast between the vessel and the background, serious interventions in the affected area and a complex vessel structure, which creates many problems to achieve segmentation of retinal vessels [3]. Currently, the main method of retinal vessel segmentation is manual annotation by professional physicians.

However, with the constant development of medical imaging technology, more and more images of the fundus require segmentation. Manual annotation of retinal vessels requires a lot of time and energy from physicians, and different physicians always adopt different standards for retinal vessel segmentation. Therefore, many algorithms for automatic retinal vessel segmentation have been proposed for Computer-Aided Diagnosis (CAD). Achieving automatic segmentation of retinal vessels can not only reduce the workload of physicians, but also avoid the subjective influence of different physicians on the results of segmentation.

According to the generalized retinal vessel segmentation algorithms [11], [23], it can be divided into two categories: supervised and unsupervised algorithms. In general, unsupervised learning algorithms do not require manual annotation data and generally use some pre-established rules to extract vessel features and achieve segmentation, such as matching algorithms based on filters [16], deforming algorithms based on models [18], and tracking-based algorithms [24]. However, fixed rules of segmentation often cannot correspond to a variety of morphological distribution of vessels.

The main idea of supervised learning algorithms is to teach a segmentation model using fundus images with segmentation annotations, which allow models to automatically extract vessel features to achieve vessel segmentation, for example, algorithms based on the Bayesian model [8], support vector machine based on algorithms [7] and deep learning algorithms [19], [4], [22], [20]. However, supervised learning algorithms require huge data with manual marking, which is difficult to obtain.

In recent years, algorithms based on deep learning have continued to be developed and worked well in the field of retinal vessel segmentation, which have gradually become the main algorithm. Long, Shelhamer and Darrell [12], proposed a Fully Collapsed Neural (FCN) network, various retinal vessel segmentation algorithms based on the FCN structure are constantly coming out. Oliveira, Pereira and Silva [2] presented a retinal vessel segmentation algorithm that combines multiscale wavelet transform and multiscale FCN. Lu, Xu, Chen and Luo [13] used a Coarse-to-Fine Fully Convolutional Neural (CF-FCN) network to extract blood vessels in fundus images. The CF-FCN aimed to use the original data information and compensate for the output of the neural network by the spatial relationship between the pixels in the fundus image. However, the results of segmentation of the FCN-based structure did not have adequate results in detail. Therefore, it usually needed some processing methods, such as Conditional Random Field (CRF) and Markov Random Field (MRF).

Among the FCN-based segmentation algorithms is the U-net structure proposed by Ronneberger, Fischer and Brox [17], changed the method of connecting the transition to the map of functions in the FCN with the addition of concatenate, which is widely used in the field of segmentation of biomedical images. In addition, a large number of advanced U-Net-based algorithms have emerged for retinal vessel segmentation. For example, Xiao, Lian and Luo [26] propose a model based on a U-network with a weighted shutter of attention for segmentation of small retinal vessels. Gao, Cai and Qiu [25] combine Gaussian and U-Net filtering to achieve retinal vessel segmentation. Alom, Hasan and Yakopcic [15] offer two advanced models based on U-Net. The first is Recurrent Convolutional Neural Network (RCNN), and the second is the Recurrent Residual Convolutional Neural Network (RRCNN), both of which have been used successfully for retinal vessel segmentation.

III. Results and Discussion

The implementation of a simple U-net model for segmentation of retinal blood vessels is presented below. The model is based on the Keras and Tensorflow libraries. The method was tested on a public DRIVE dataset. For the DRIVE data set, which consists of 40 images, we used 20 images to train the model and 20 images to test the model. Finally, complete content and organizational editing before formatting.

Contrast-constrained adaptive histogram alignment (CLAHE) is used to normalize the images. The image is divided into small blocks called “tiles” (the default tile size is 8x8 in OpenCV). Then each of these blocks is aligned. Therefore, in a small area, the histogram is limited to a small area (if there is no noise). If there is noise, it will be amplified. To avoid this, contrast restriction is applied. If the specified contrast limit is exceeded (default 40 in OpenCV), these pixels are truncated and evenly distributed in other containers before applying histogram alignment. After alignment, bilinear interpolation is used to remove artifacts within the tile. Gamma correction using the lookup table is applied.

We used the Adam optimizer with *initial_learning_rate = 0.0003, first_decay_steps = 12000, t_mul = 1000, m_mul = 0.5, alpha = 1e-5.*

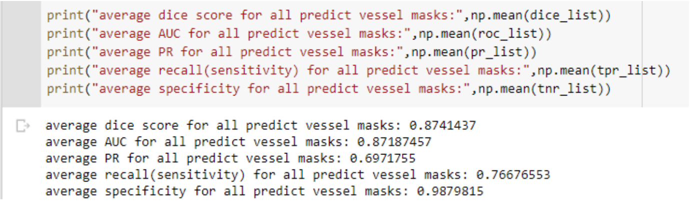
For method evaluation we use Precision, Recall, Confusion matrix, and the area under the Receiver Operating Characteristic Curve (AUC-ROC) (Figures 2 and 3).

Recall (also called True Positive Rate or Sensitivity) is defined as the ratio of truly classified vessel pixels. Precision (also called True Negative Rate or Specificity) is the ratio of truly classified non-vessel pixels.

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

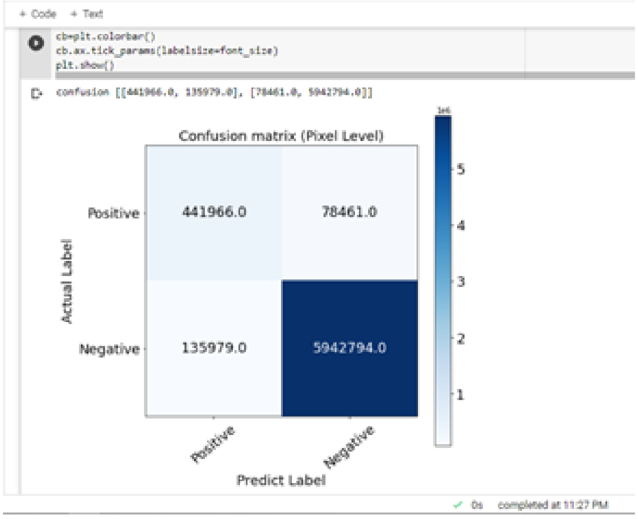
Where TP is the number of pixels that are classified as vessels in the image, which are correctly classified. FP is the number of pixels classified as vessels which are incorrectly classified. FN is the number of pixels classified as non-vessels which are incorrectly classified.



```
print("average dice score for all predict vessel masks:",np.mean(dice_list))
print("average AUC for all predict vessel masks:",np.mean(roc_list))
print("average PR for all predict vessel masks:",np.mean(pr_list))
print("average recall(sensitivity) for all predict vessel masks:",np.mean(tpr_list))
print("average specificity for all predict vessel masks:",np.mean(tnr_list))
```

```
average dice score for all predict vessel masks: 0.8741437
average AUC for all predict vessel masks: 0.87187457
average PR for all predict vessel masks: 0.6971755
average recall(sensitivity) for all predict vessel masks: 0.76676553
average specificity for all predict vessel masks: 0.9879815
```

Figure 2. Results of metrics calculation Dice Score, Precision, Recall, AUC



```
cb=plt.colorbar()
cb.ax.tick_params(labelsize=font_size)
plt.show()
```

```
confusion [[441966.0, 135979.0], [78461.0, 5942794.0]]
```

Figure 3. Confussion matrix for model evaluation

Another used metric is the AUC (Area under the curve) or ROC (Receiver Operating Characteristic Curve) (a graphical plot from (0;0) to (1;1) of the false positive rate (x-axis) versus the true positive rate (y-axis)). Figure. 4 shows the ROC curve.

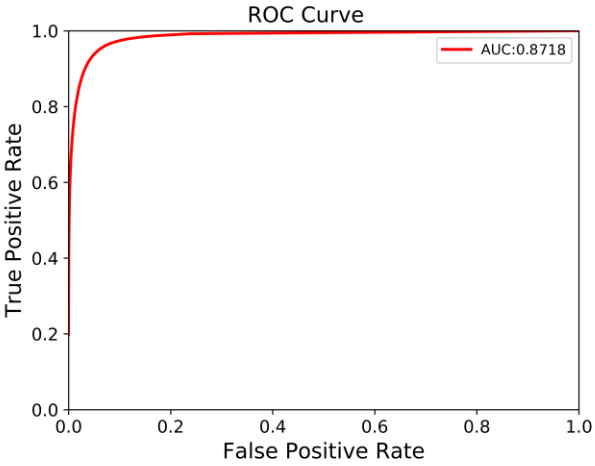


Figure 4. ROC curve

To classify the images, we exported the images in the form of 25 tiles of the same size for deep learning.

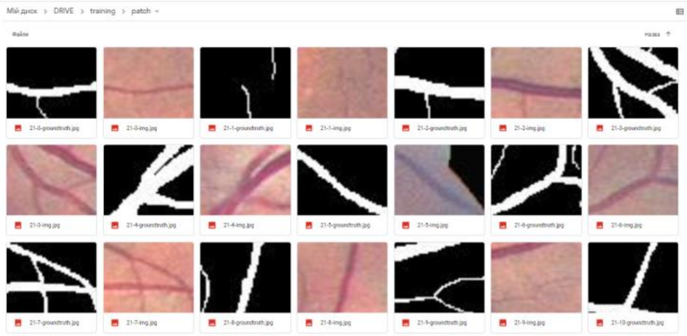
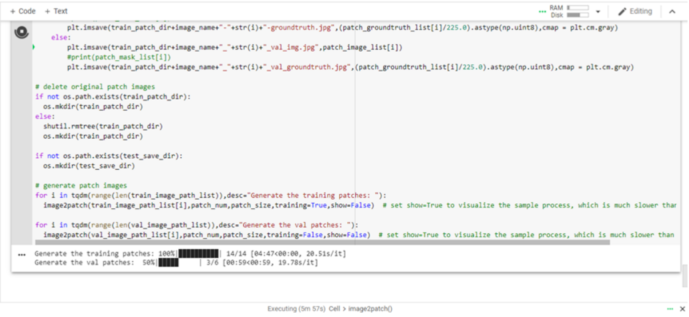


Figure 5. Exported images

The forming tiles process for the deep learning application is presented in Figure 6.



```
            plt.imsave(train_patch_dir+image_name+"-"+str(i)+"-groundtruth.jpg",(patch_groundtruth_list[i]/225.0).astype(np.uint8),cmap = plt.cm.gray)
        else:
            plt.imsave(train_patch_dir+image_name+"_"+str(i)+"_val_img.jpg",patch_image_list[i])
            #print(patch_mask_list[i])
            plt.imsave(train_patch_dir+image_name+"_"+str(i)+"_val_groundtruth.jpg",(patch_groundtruth_list[i]/225.0).astype(np.uint8),cmap = plt.cm.gray)

# delete original patch images
if not os.path.exists(train_patch_dir):
  os.mkdir(train_patch_dir)
else:
  shutil.rmtree(train_patch_dir)
  os.mkdir(train_patch_dir)

if not os.path.exists(test_save_dir):
  os.mkdir(test_save_dir)

# generate patch images
for i in tqdm(range(len(train_image_path_list)),desc="Generate the training patches: "):
  image2patch(train_image_path_list[i],patch_num,patch_size,training=True,show=False)  # set show=True to visualize the sample process, which is much slower thar

for i in tqdm(range(len(val_image_path_list)),desc="Generate the val patches: "):
  image2patch(val_image_path_list[i],patch_num,patch_size,training=False,show=False)  # set show=True to visualize the sample process, which is much slower than
```

```
Generate the training patches: 100%|██████████| 14/14 [04:47<00:00, 20.51s/it]
Generate the val patches:  50%|█████     | 3/6 [00:59<00:59, 19.78s/it]
```

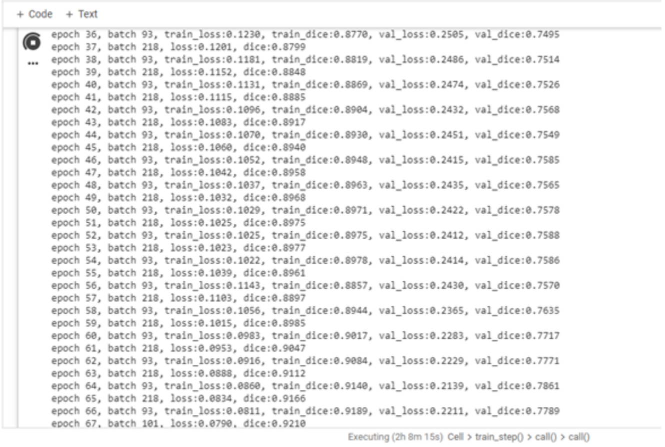
Figure 6. The forming training tiles process in the Google Colab environment

Binary cross entropy was used as a loss function to determine the error between actual and obtained results and to minimize it.

TABLE I. MINIMIZING THE LOSS FUNCTION VALUE IN THE LEARNING PROCESS

Iteration	*Loss function*	*Dice coefficient*	*Iteration*	*Loss function*	*Dice coefficient*
1	0.5218	0.4782	38	0.1181	0.8819
2	0.4937	0.5063	39	0.1152	0.8848
3	0.4742	0.5258	40	0.1131	0.8869
4	0.4552	0.5448	…	…	…
5	0.4355	0.5645	124	0.0228	0.9772
6	0.4161	0.5839	125	0.0226	0.9774
7	0.3970	0.6030	126	0.0224	0.9776
8	0.3783	0.6217	127	0.0219	0.9781
9	0.3603	0.6397	128	0.0215	0.9785
…	…	…	129	0.0214	0.9786
31	0.1410	0.8590	130	0.0214	0.9786
32	0.1368	0.8632	131	0.0207	0.9793
33	0.1334	0.8666	132	0.0206	0.9794
34	0.1293	0.8707	133	0.0205	0.9795
35	0.1262	0.8738	134	0.0201	0.9799
36	0.1230	0.8770	135	0.0200	0.9800
37	0.1201	0.8799	136	0.0198	0.9802

The calculating process of the loss function and the Dice coefficient took place directly in the software environment.



```
epoch 36, batch 93, train_loss:0.1230, train_dice:0.8770, val_loss:0.2505, val_dice:0.7495
epoch 37, batch 218, loss:0.1201, dice:0.8799
epoch 38, batch 93, train_loss:0.1181, train_dice:0.8819, val_loss:0.2486, val_dice:0.7514
epoch 39, batch 218, loss:0.1152, dice:0.8848
epoch 40, batch 93, train_loss:0.1131, train_dice:0.8869, val_loss:0.2474, val_dice:0.7526
epoch 41, batch 218, loss:0.1115, dice:0.8885
epoch 42, batch 93, train_loss:0.1096, train_dice:0.8904, val_loss:0.2432, val_dice:0.7568
epoch 43, batch 218, loss:0.1083, dice:0.8917
epoch 44, batch 93, train_loss:0.1070, train_dice:0.8930, val_loss:0.2451, val_dice:0.7549
epoch 45, batch 218, loss:0.1060, dice:0.8940
epoch 46, batch 93, train_loss:0.1052, train_dice:0.8948, val_loss:0.2415, val_dice:0.7585
epoch 47, batch 218, loss:0.1042, dice:0.8958
epoch 48, batch 93, train_loss:0.1037, train_dice:0.8963, val_loss:0.2435, val_dice:0.7565
epoch 49, batch 218, loss:0.1032, dice:0.8968
epoch 50, batch 93, train_loss:0.1029, train_dice:0.8971, val_loss:0.2422, val_dice:0.7578
epoch 51, batch 218, loss:0.1025, dice:0.8975
epoch 52, batch 93, train_loss:0.1025, train_dice:0.8975, val_loss:0.2412, val_dice:0.7588
epoch 53, batch 218, loss:0.1023, dice:0.8977
epoch 54, batch 93, train_loss:0.1022, train_dice:0.8978, val_loss:0.2414, val_dice:0.7586
epoch 55, batch 218, loss:0.1039, dice:0.8961
epoch 56, batch 93, train_loss:0.1143, train_dice:0.8857, val_loss:0.2430, val_dice:0.7570
epoch 57, batch 218, loss:0.1103, dice:0.8897
epoch 58, batch 93, train_loss:0.1056, train_dice:0.8944, val_loss:0.2365, val_dice:0.7635
epoch 59, batch 218, loss:0.1015, dice:0.8985
epoch 60, batch 93, train_loss:0.0983, train_dice:0.9017, val_loss:0.2283, val_dice:0.7717
epoch 61, batch 218, loss:0.0953, dice:0.9047
epoch 62, batch 93, train_loss:0.0916, train_dice:0.9084, val_loss:0.2229, val_dice:0.7771
epoch 63, batch 218, loss:0.0888, dice:0.9112
epoch 64, batch 93, train_loss:0.0860, train_dice:0.9140, val_loss:0.2139, val_dice:0.7861
epoch 65, batch 218, loss:0.0834, dice:0.9166
epoch 66, batch 93, train_loss:0.0811, train_dice:0.9189, val_loss:0.2211, val_dice:0.7789
epoch 67, batch 101, loss:0.0790, dice:0.9210
```

Executing (2h 8m 15s) Cell > train_step() > call() > call()

Figure 7. The calculating process of the loss function and the Dice coefficient in the Google Colab environment

The result of the neural network is well traced comparing, for example, the initial image of the fundus in the eleventh image (Figure 8) and the results of the U-Net model that presented in Figure 9 with recognized blood vessels for the eleventh image.

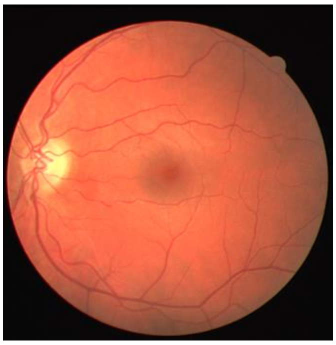
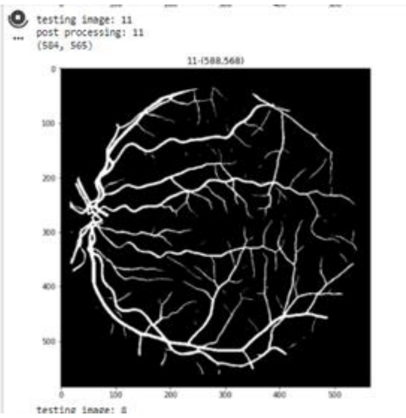


Figure 8. The initial image of the fundus in the eleventh image



```
testing image: 11
post processing: 11
(584, 565)
```

```
testing image: 8
```

Figure 9. The U-net model results for the eleventh image

IV. CONCLUSION

Automatic the retinal vessels segmentation in the fundus images is an important task, as images with detected blood vessels can diagnose some diseases. This paper proposes the implementation of a neural network architecture based on a dense U-Net network in Google Colab on a public DRIVE data set using the Tensorflow machine learning library for the retinal vessels segmentation. The general process of the neural network is presented. Binary cross-entropy is used as a loss function for calculation the error between the actual and obtained results and its minimization. The results are presented in the table. Precision, Recall, Confusion matrix, and the area under the Receiver Operating Characteristic Curve (AUC-ROC) were used evaluating the method. The results of the model, namely the recognized blood vessels are presented.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, G.E. Hinton: Imagenet classification with deep convolutional neural networks. In: NIPS, pp. 1106–1114 (2012)

[2] A. Oliveira, S. Pereira, and C. A. Silva, ‘‘Retinal vessel segmentation based on fully convolutional neural networks,’’ Expert Syst. Appl., vol. 112, pp. 229–242, Dec. 2018.

[3] C. L Srinidhi, P. Aparna, and J. Rajan, ‘‘Recent advancements in retinal vessel segmentation,’’ J. Med. Syst., vol. 41, no. 4, Mar. 2017.

[4] C. Wang, Z. Zhao, Q. Ren, Y. Xu, and Y. Yu, ''Dense U-net based on patchbased learning for retinal vessel segmentation,'' Entropy, vol. 21, no. 2, p. 168, Feb. 2019.

[5] C. Y.-L. Cheung, M. K. Ikram, C. Chen, and T. Y. Wong, ''Imaging retina to study dementia and stroke,'' Prog. Retinal Eye Res., vol. 57, pp. 89–107, Mar. 2017.

[6] D.C.Ciresan, L.M. Gambardella, A. Giusti, J. Schmidhuber: Deep neural networks segment neuronal membranes in electron microscopy images. In: NIPS, pp. 2852–2860 (2012)

[7] E. Tuba, L. Mrkela, and M. Tuba, ''Retinal blood vessel segmentation by support vector machine classification,'' present at the 27th Int. Conf. Radioelektronika (RADIOELEKTRONIKA), 2017. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7936649

[8] F. Uslu and A. A. Bharath, ''A recursive Bayesian approach to describe retinal vasculature geometry,'' Pattern Recognit., vol. 87, pp. 157–169, Mar. 2019.

[9] H. Akil, A. S. Huang, B. A. Francis, S. R. Sadda, and V. Chopra, ''Retinal vessel density from optical coherence tomography angiography to differentiate early glaucoma, pre-perimetric glaucoma and normal eyes,'' PLoS ONE, vol. 12, no. 2, Feb. 2017, Art. no. e0170476.

[10] H. Lee, M. Lee, H. Chung, and H. C. Kim, ''Quantification of retinal vessel tortuosity in diabetic retinopathy using optical coherence tomography angiography,'' Retina, vol. 38, no. 5, pp. 976–985, May 2018.

[11] J. Almotiri, K. Elleithy, and A. Elleithy, ''Retinal vessels segmentation techniques and algorithms: A survey,'' Appl. Sci., vol. 8, no. 2, p. 155, Jan. 2018.

[12] J. Long, E. Shelhamer, and T. Darrell, ''Fully convolutional networks for semantic segmentation,'' in Proc. CVPR, 2015, pp. 3431–3440. [Online]. Available: https://www.cvfoundation.org/openaccess/content_cvpr_2015/html/Long_Fully_Convolutional_Networks_2015_CVPR_paper.html

[13] J. Lu, Y. Xu, M. Chen, and Y. Luo, ''A Coarse-to-Fine fully convolutional neural network for fundus vessel segmentation,'' Symmetry, vol. 10, no. 11, p. 607, Nov. 2018.

[14] K. Simonyan, A. Zisserman: Very deep convolutional networks for large-scale image recognition (2014), arXiv:1409.1556 [cs.CV]

[15] M. Z. Alom, M. Hasan, and C. Yakopcic, ''Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation,'' in Proc. Comput. Vis. Pattern Recognit., May 2018, pp. 1–12.

[16] N. P. Singh and R. Srivastava, ''Retinal blood vessels segmentation by using Gumbel probability distribution function based matched filter,'' Comput. Methods Programs Biomed., vol. 129, pp. 45–50, Jun. 2016.

[17] O. Ronneberger, P. Fischer, and T. Brox, ''U-net: Convolutional networks for biomedical image segmentation,'' presented at the Med. Image Comput. Comput.-Assist. Intervent. (MICCAI), 2015. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28

[18] P. K. Karn, B. Biswal, and S. R. Samantaray, ''Robust retinal blood vessel segmentation using hybrid active contour model,'' IET Image Process., vol. 13, no. 3, pp. 440–450, Feb. 2019.

[19] Q. Jin, Z. Meng, T. D. Pham, Q. Chen, L. Wei, and R. Su, ''DUNet: A deformable network for retinal vessel segmentation,'' Knowl.-Based Syst., vol. 178, pp. 149–162, Aug. 2019.

[20] S. Feng, Z. Zhuo, and D. Pan, ''CcNet: A cross-connected convolutional network for segmenting retinal vessels using multi-scale features,'' Neurocomputing, vol. 392, pp. 268–2767, Jun. 2020.

[21] S. Fraser-Bell, R. Symes, and A. Vaze, ''Hypertensive eye disease: A review,'' Clin. Experim. Ophthalmol., vol. 45, no. 1, pp. 45–53, Jan. 2017.

[22] S. Guo, K. Wang, H. Kang, Y. Zhang, Y. Gao, and T. Li, ''BTSDSN: Deeply supervised neural network with short connections for retinal vessel segmentation,'' Int. J. Med. Informat., vol. 126, pp. 105–113, Jun. 2019.

[23] S. Moccia, E. De Momi, S. El Hadji, and L. S. Mattos, ''Blood vessel segmentation algorithms—Review of methods, datasets and evaluation metrics,'' Comput. Methods Programs Biomed., vol. 158, pp. 71–91, May 2018.

[24] W. Wang, J. Zhang, W. Wu, and S. Zhou, ''An automatic approach for retinal vessel segmentation by multi-scale morphology and seed point tracking,'' J. Med. Imag. Health Informat., vol. 8, no. 2, pp. 262–274, Feb. 2018.

[25] X. Gao, Y. Cai, and C. Qiu, ''Retinal blood vessel segmentation based on the Gaussian matched filter and U-net,'' presented at the CISP-BMEI, 2017. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8302199

[26] X. Xiao, S. Lian, and Z. Luo, ''Weighted Res-UNet for highquality retina vessel segmentation,'' presented at the 9th Int. Conf. Inf. Technol. Med. Educ. (ITME), 2018. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8589312

[27] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel: Backpropagation applied to handwritten zip code recognition. Neural Computation 1(4), 541–551 (1989))