



RV-GAN: Segmenting Retinal Vascular Structure in Fundus Photographs Using a Novel Multi-scale Generative Adversarial Network

Sharif Amit Kamran¹(✉), Khondker Fariha Hossain¹, Alireza Tavakkoli¹,
Stewart Lee Zuckerbrod², Kenton M. Sanders³, and Salah A. Baker³

¹ Department of Computer Science and Engineering, University of Nevada, Reno, NV, USA
skamran@nevada.unr.edu

² Houston Eye Associates, Houston, TX, USA

³ School of Medicine, University of Nevada, Reno, NV, USA

Abstract. High fidelity segmentation of both macro and microvascular structure of the retina plays a pivotal role in determining degenerative retinal diseases, yet it is a difficult problem. Due to successive resolution loss in the encoding phase combined with the inability to recover this lost information in the decoding phase, autoencoding based segmentation approaches are limited in their ability to extract retinal microvascular structure. We propose RV-GAN, a new multi-scale generative architecture for accurate retinal vessel segmentation to alleviate this. The proposed architecture uses two generators and two multi-scale autoencoding discriminators for better microvessel localization and segmentation. In order to avoid the loss of fidelity suffered by traditional GAN-based segmentation systems, we introduce a novel weighted feature matching loss. This new loss incorporates and prioritizes features from the discriminator's decoder over the encoder. Doing so combined with the fact that the discriminator's decoder attempts to determine real or fake images at the pixel level better preserves macro and microvascular structure. By combining reconstruction and weighted feature matching loss, the proposed architecture achieves an area under the curve (AUC) of 0.9887, 0.9914, and 0.9887 in pixel-wise segmentation of retinal vasculature from three publicly available datasets, namely DRIVE, CHASE-DB1, and STARE, respectively. Additionally, RV-GAN outperforms other architectures in two additional relevant metrics, mean intersection-over-union (Mean-IOU) and structural similarity measure (SSIM).

Keywords: Retinal vessel segmentation · Generative networks · Medical imaging · Ophthalmology · Retinal fundus

1 Introduction

The fundoscopic exam is a procedure that provides necessary information to diagnose different retinal degenerative diseases such as Diabetic Retinopathy, Macular Edema, Cytomegalovirus Retinitis [27]. A highly accurate system is required to segment retinal

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-87237-3_4) contains supplementary material, which is available to authorized users.

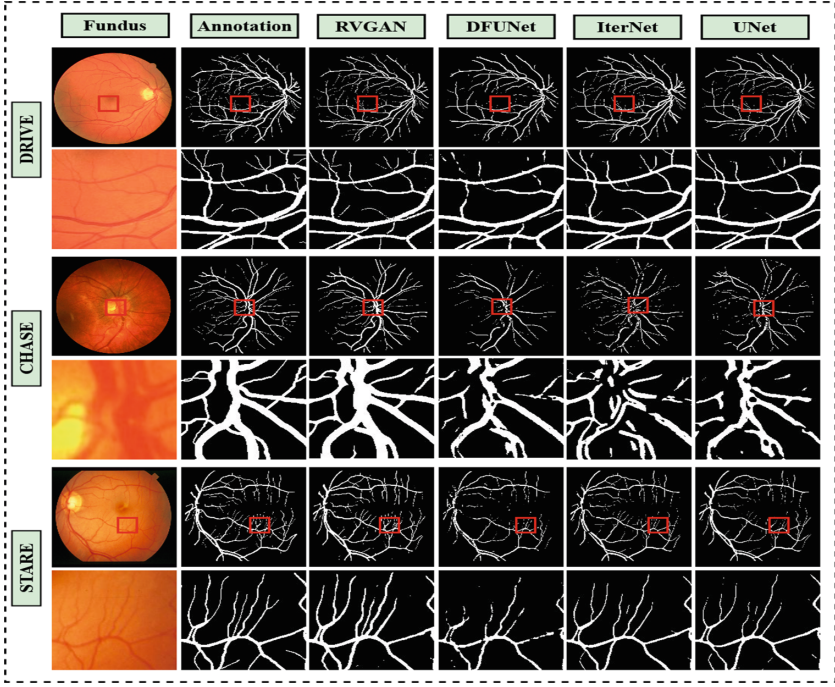


Fig. 1. RV-GAN segments vessel with better precision than other architectures. The 1st row is the whole image, while 2nd row is specific zoomed-in area of the image. The Red bounded box signifies the zoomed-in region. Here, the confidence score is, $t > 0.5$. The row contains DRIVE, CHASE-DB1 and STARE data-set. Whereas the column contains fundus images, ground-truths and segmentation maps for RV-GAN, DFUNet, IterNet and UNet. (Color figure online)

vessels and find abnormalities in the retinal subspace to diagnose these vascular diseases. Many image processing and machine learning-based approaches for retinal vessel segmentation have so far been proposed [7, 13, 23, 26]. However, such methods fail to precisely pixel-wise segment blood vessels due to insufficient illumination and periodic noises. Attributes like this present in the subspace can create false-positive segmentation [7]. In recent times, UNet based deep learning architectures have become very popular for retinal vessel segmentation. UNet consists of an encoder to capture context information and a decoder for enabling precise localization [24]. Many derivative works based on UNet have been proposed, such as Dense-UNet, Deformable UNet [10], IterNet [16] etc. These models were able to achieve quite good results for macro vessel segmentation. However, these architectures fail when segmenting microvessels with higher certainty. One reason is successive resolution loss in the encoder, and failure to capture those features in the decoder results in inferior microvessel segmentation. Recent GAN-based architecture [21, 32] tries to address this by incorporating discriminative features from adversarial examples while training. However, the discriminator being an encoder [9], only trains on patches of images rather than pixels, affecting the true-positive-rate of the model. We need an architecture that can retain discriminative manifold features

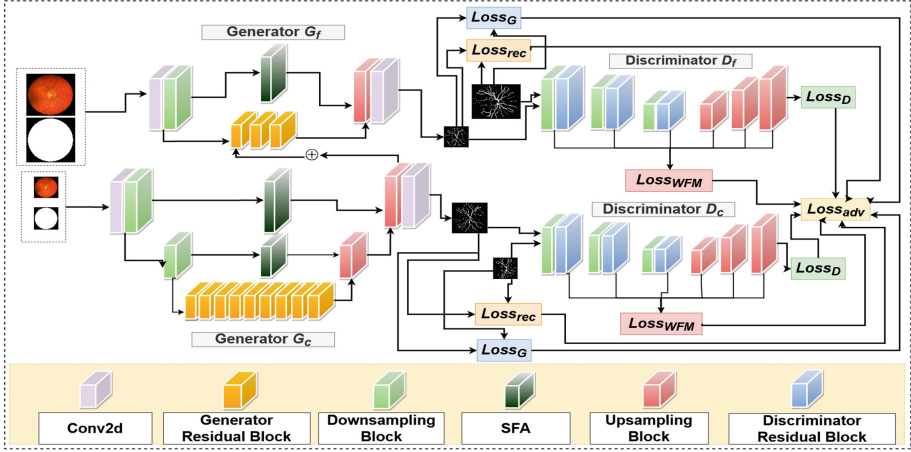


Fig. 2. RV-GAN consisting of Coarse and Fine generators G_f, G_c and discriminators D_f, D_c . The generators incorporate reconstruction loss, $Loss_{rec}$ and Hinge loss $Loss_G$. Whereas the discriminators use weighted feature matching loss, $Loss_{wfm}$ and Hinge loss $Loss_D$. All of these losses are multiplied by weight multiplier and then added in the final adversarial loss, $Loss_{adv}$. The generators consist of Downsampling, Upsampling, SFA and its distinct residual blocks. On the other hand, the discriminators consist of Downsampling, Upsampling and counterpart residual blocks.

and segment microvessels on pixel-level with higher confidence. Confidence signifies the probability distribution function of the segmented pixel falling under vessel or background. By taking all of these into account, we propose Retinal-Vessel GAN, consisting of coarse and fine generators and multi-scale autoencoder-based discriminators for producing highly accurate segmentation of blood vessel with strong confidence score. Additionally, we come up with a new weighted feature matching loss with inner and outer weights. And we combine it with reconstruction and hinge loss for adversarial training of our architecture. From Fig. 1, it is apparent that our architecture produces a segmentation map with a high confidence score.

2 Proposed Methodology

2.1 Multi-scale Generators

Pairing multi-scale coarse and fine generators produces high-quality domain-specific retinal image synthesis, as observed in recent generative networks, such as Fundus2Angio [12], V-GAN [27] and [29]. Inspired by this, we also adopt this feature in our architecture by using two generators (G_{fine} and G_{coarse}), as visualized in Fig. 2. The generator G_{fine} synthesizes fine-grained vessel segmentation images by extracting local features such as micro branches, connections, blockages, etc. In contrast, the generator G_{coarse} tries to learn and conserve global information, such as the structures of the macro branches, while producing less detailed microvessel segmentation. The detailed structure of these generators is illustrated in Fig. 2.

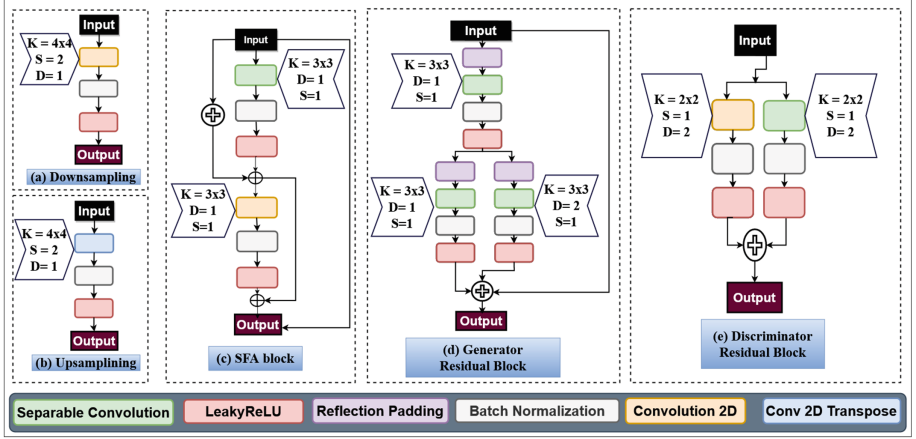


Fig. 3. Proposed Downsampling, Upsampling, Spatial Feature Aggregation block, Generator and Discriminator Residual blocks. Here, K =Kernel size, S =Stride, D =Dilation.

2.2 Residual Downsampling and Upsampling Blocks

Our generators and discriminators consist of both downsampling and upsampling blocks to get the desired feature maps and output. The downsampling block comprises of a convolution layer, a batch-norm layer and a Leaky-ReLU activation function consecutively and is illustrated in Fig. 3(a). Contrarily, the decoder block consists of a transposed convolution layer, batch-norm, and Leaky-ReLU activation layer successively and can be visualized in Fig. 3(b).

2.3 Distinct Identity Blocks for Generator & Discriminator

For spatial and depth feature propagation, residual identity blocks have become go-to building blocks for image style transfer, image inpainting, and image segmentation tasks [3, 4, 22, 25, 30]. Vanilla convolution layers are both computationally inefficient and fail to retain accurate spatial and depth information, as opposed to separable convolution [5]. Separable convolution comprises of a depth-wise convolution and a point-wise convolution successively. As a result, it extracts and preserves depth and spatial features while forward propagating the network. Recent advancement in retinal image classification has shown that combining separable convolutional layers with dilation allows for more robust feature extraction [11]. We design two unique residual identity blocks, for our generators and discriminators, as illustrated in Fig. 3(d) & Fig. 3(e).

2.4 Spatial Feature Aggregation

In this section, we discuss our proposed spatial feature aggregation (SFA) block, as illustrated in Fig. 3(c). We use spatial feature aggregation block for combining spatial and depth features from the bottom layers of the network with the top layers of the network, as illustrated in Fig. 2. The rationale behind employing the SFA block is to extract and retain spatial and depth information, that is otherwise lost in deep networks. Consequently, these features can be combined with the learned features of the deeper layers of the network to get an accurate approximation, as observed in similar GAN architectures [2, 33].

2.5 Auto-Encoder as Discriminators

For better pixel-wise segmentation, we need an architecture that can extract both global and local features from the image. To mitigate this underlying problem, we need a deep and dense architecture with lots of computable parameters. It, in turn, might lead to overfitting or vanishing gradient while training the model. To address this issue, rather than having a single dense segmentation architecture, we adopt light-weight discriminators as autoencoders. Additionally, we use multi-scale discriminators for both our coarse and fine generators, as previously proposed in [15, 30]. The arrangement consists of two discriminators with variable sized input and can help with the overall adversarial training. We define two discriminators as D_f and D_c as illustrated in Fig. 2.

2.6 Proposed Weighted Feature Matching Loss

Feature matching loss [30] was incorporated by extracting features from discriminators to do semantic segmentation. The feature-matching loss is given in Eq. 1 As the authors used Patch-GAN as a discriminator, it only contains an encoding module. By contrast, our work involves finding pixel-wise segmentation of retinal vessel and background and thus requires an additional decoder. By successive downsampling and upsampling, we lose essential spatial information and features; that is why we need to give weightage to different components in the overall architecture. We propose a new weighted feature matching loss, as given in Eq. 2 that combines elements from both encoder and decoder and prioritizes particular features to overcome this. For our case, we experiment and see that giving more weightage to decoder feature map results in better vessel segmentation.

$$\mathcal{L}_{fm}(G, D_{enc}) = \mathbb{E}_{x,y} \frac{1}{N} \sum_{i=1}^k \|D_{enc}^i(x, y) - D_{enc}^i(x, G(x))\| \quad (1)$$

$$\begin{aligned} \mathcal{L}_{wfm}(G, D_n) = \mathbb{E}_{x,y} \frac{1}{N} \sum_{i=1}^k & \lambda_{enc}^i \|D_{enc}^i(x, y) - D_{enc}^i(x, G(x))\| + \lambda_{dec}^i \|D_{dec}^i(x, y) \\ & - D_{dec}^i(x, G(x))\| \end{aligned} \quad (2)$$

Equation 1 is calculated by taking only the features from the downsampling blocks of the discriminator's encoder, D_{enc}^i . Contrarily, the proposed Eq. 2 takes features from

both the downsampling and upsampling blocks of the discriminator’s encoder, D_{enc}^i and decoder, D_{dec}^i . We insert the real x, y and the synthesized $x, G(x)$, image & segmentation map pairs consecutively. The N signifies the number of features. Here, λ_{enc} and λ_{dec} is the inner weight multiplier for each of the extracted feature maps. The weight values are between $[0, 1]$, and the total sum of the weight is 1, and we use a higher weight value for the decoder feature maps than the encoder feature maps.

2.7 Weighted Objective and Adversarial Loss

For adversarial training, we use Hinge-Loss [18, 33] as given in Eq. 3 and Eq. 4. Conclusively, all the fundus images and their corresponding segmentation map pairs are normalized, to $[-1, 1]$. As a result, it broadens the difference between the pixel intensities of the real and synthesized segmentation maps. In Eq. 5, we multiply $\mathcal{L}_{adv}(G)$ with λ_{adv} as weight multiplier. Next, we add $\mathcal{L}_{adv}(D)$ with the output of the multiplication.

$$\mathcal{L}_{adv}(D) = -\mathbb{E}_{x,y} [\min(0, -1 + D(x, y))] - \mathbb{E}_x [\min(0, -1 - D(x, G(x)))] \quad (3)$$

$$\mathcal{L}_{adv}(G) = -\mathbb{E}_{x,y} [D(G(x), y)] \quad (4)$$

$$\mathcal{L}_{adv}(G, D) = \mathcal{L}_{adv}(D) + \lambda_{adv}(\mathcal{L}_{adv}(G)) \quad (5)$$

In Eq. 4 and Eq. 5, we first train the discriminators on the real fundus, x and real segmentation map, y . After that, we train with the real fundus, x , and synthesized segmentation map, $G(x)$. We begin by batch-wise training the discriminators D_f , and D_c for a couple of iterations on the training data. Following that, we train the G_c while keeping the weights of the discriminators frozen. In a similar fashion, we train G_f on a batch training image while keeping weights of all the discriminators frozen.

The generators also incorporate the reconstruction loss (Mean Squared Error) as shown in Eq. 6. By utilizing the loss we ensure the synthesized images contain more realistic microvessel, arteries, and vascular structure.

$$\mathcal{L}_{rec}(G) = \mathbb{E}_{x,y} \|G(x) - y\|^2 \quad (6)$$

By incorporating Eqs. 2, 5 and 6, we can formulate our final objective function as given in Eq. 7.

$$\min_{G_f, G_c} \left(\max_{D_f, D_c} (\mathcal{L}_{adv}(G_f, G_c, D_f, D_c)) + \lambda_{rec} [\mathcal{L}_{rec}(G_f, G_c)] + \lambda_{wfm} [\mathcal{L}_{wfm}(G_f, G_c, D_f, D_c)] \right) \quad (7)$$

Here, λ_{adv} , λ_{rec} , and λ_{wfm} implies different weights, that is multiplied with their respective losses. The loss weighting decides which architecture to prioritize while training. For our system, more weights are given to the $\mathcal{L}_{adv}(G)$, \mathcal{L}_{rec} , \mathcal{L}_{wfm} , and thus we select bigger λ values for those.

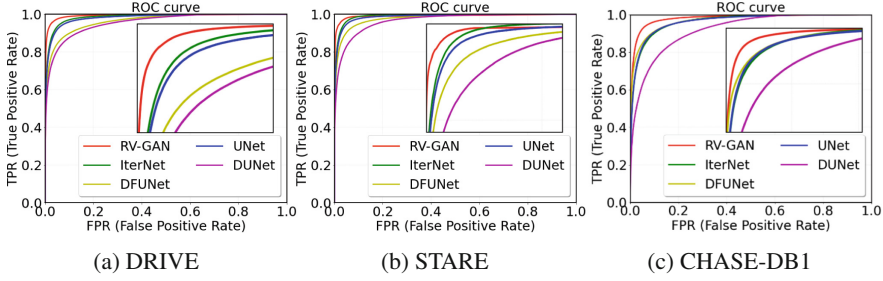


Fig. 4. ROC Curves on (a) DRIVE (b) STARE (c) CHASE-DB1.

3 Experiments

3.1 Dataset

For benchmarking, we use three retinal segmentation datasets, namely, DRIVE [28], CHASE-DB1 [20], and STARE [8]. The images are respectively in *.tif* (565×584), *.jpg* (999×960), and *.ppm* (700×605). We train three different RV-GAN networks with each of these datasets using 5-fold cross-validation. We use overlapping image patches with a stride of 32 and an image size of 128×128 for training and validation. So we end up having 4320 for STARE, 15120 for CHASE-DB1, and 4200 for DRIVE from 20, 20, and 16 images. DRIVE dataset comes with official FoV masks for the test images. For CHASE-DB1 and STARE dataset, we also generate FoV masks similar to Li et al. [16]. For testing, overlapping image patches with a stride of 3 were extracted and averaged by taking 20, 8 and 4 images from DRIVE, CHASE-DB1, and STARE.

3.2 Hyper-parameter Initialization

For adversarial training, we used hinge loss [18,33]. We picked $\lambda_{enc} = 0.4$ (Eq. 1), $\lambda_{dec} = 0.6$ (Eq. 2), $\lambda_{adv} = 10$ (Eq. 5), $\lambda_{rec} = 10$ (Eq. 6) and $\lambda_{wfm} = 10$ (Eq. 7). We used Adam optimizer [14], with learning rate $\alpha = 0.0002$, $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We train with mini-batches with batch size, $b = 24$ for 100 epochs in three stages using Tensorflow. It took between 24–48 h to train our model on NVIDIA P100 GPU depending on data-set. Because DRIVE and STARE have lower number of patches compared to CHASE-DB1, it takes less amount to train. The inference time is 0.025 second per image. The code repository is provided in this [link](#).

3.3 Threshold vs. Confidence Score

Confidence score signifies the per-pixel probability density value of the segmentation map. U-Net-derived models incorporate binary cross-entropy loss with a threshold of 0.5 to predict if the pixel is a vessel or background. So a pixel, predicted with a 0.5001 probability, will be classified as a vessel. As a result, the model suffers from Type I error or a high false-positive rate (FPR). In contrast, we use the generators to produce realistic

Table 1. Performance comparison on DRIVE [28], CHASE-DB1 [20], & STARE [8].

Dataset	Method	Year	F1 Score	Sensitivity	Specificity	Accuracy	AUC-ROC	Mean-IOU	SSIM
DRIVE	UNet [10]	2018	0.8174	0.7822	0.9808	0.9555	0.9752	0.9635	0.8868
	Residual UNet [1]	2018	0.8149	0.7726	0.9820	0.9553	0.9779	—	—
	Recurrent UNet [1]	2018	0.8155	0.7751	0.9816	0.9556	0.9782	—	—
	R2UNet [1]	2018	0.8171	0.7792	0.9813	0.9556	0.9784	—	—
	DFUNet [10]	2019	0.8190	0.7863	0.9805	0.9558	0.9778	0.9605	0.8789
	IterNet [16]	2019	0.8205	0.7735	0.9838	0.9573	0.9816	0.9692	0.9008
	SUD-GAN [32]	2020	—	0.8340	0.9820	0.9560	0.9786	—	—
	M-GAN [21]	2020	0.8324	0.8346	0.9836	0.9706	0.9868	—	—
	RV-GAN (Ours)	2021	0.8690	0.7927	0.9969	0.9790	0.9887	0.9762	0.9237
CHASE-DB1	UNet [10]	2018	0.7993	0.7841	0.9823	0.9643	0.9812	0.9536	0.9029
	DenseBlock-UNet [17]	2018	0.8006	0.8178	0.9775	0.9631	0.9826	0.9454	0.8867
	DFUNet [10]	2019	0.8001	0.7859	0.9822	0.9644	0.9834	0.9609	0.9175
	IterNet [16]	2019	0.8073	0.7970	0.9823	0.9655	0.9851	0.9584	0.9123
	M-GAN [21]	2020	0.8110	0.8234	0.9938	0.9736	0.9859	—	—
	RV-GAN (Ours)	2021	0.8957	0.8199	0.9806	0.9697	0.9914	0.9705	0.9266
STARE	UNet [10]	2018	0.7595	0.6681	0.9915	0.9639	0.9710	0.9744	0.9271
	DenseBlock-UNet [17]	2018	0.7691	0.6807	0.9916	0.9651	0.9755	0.9604	0.9034
	DFUNet [10]	2019	0.7629	0.6810	0.9903	0.9639	0.9758	0.9701	0.9169
	IterNet [16]	2019	0.8146	0.7715	0.9886	0.9701	0.9881	0.9752	0.9219
	SUD-GAN [32]	2020	—	0.8334	0.9897	0.9663	0.9734	—	—
	M-GAN [21]	2020	0.8370	0.8234	0.9938	0.9876	0.9873	—	—
	RV-GAN (Ours)	2021	0.8323	0.8356	0.9864	0.9754	0.9887	0.9754	0.9292

segmentation maps and utilize the weighted feature matching loss to combine inherent manifold information to predict real and fake pixels with higher certainty. Consequently, we can see in Fig. 4 that our model’s Receiver Operating (ROC) curves for three datasets are relatively better than other previous methods due to a high confidence score.

3.4 Quantitative Bench-Marking

We compared our architecture with some best performing ones, including UNet [10], DenseBlock-UNet [17], Deform-UNet [10] and IterNet [16] as illustrated in Fig. 1. We trained and evaluated the first three architectures using their publicly available source code by ourselves on the three datasets. For testing, we only use the generators, G_f and G_c for synthesizing retinal segmentation maps. First, we generate the feature vector from G_c by feeding the resized Fundus and FOV mask. After that, we use the original Fundus, FOV mask, and the feature vector from G_c to generate the fine segmentation map using G_f . For IterNet, the pre-trained weight was provided, so we used that to get the inference result. Next, we do a comparative analysis with existing retinal vessel segmentation architectures, which includes both UNet and GAN based models. The prediction results for DRIVE, CHASE-DB1 and STARE are provided in Table 1. We report traditional metrics such as F1-score, Sensitivity, Specificity, Accuracy, and AUC-ROC. Additionally, we use two other metrics for predicting accurate segmentation and structural similarity of the retinal vessels, namely Mean-IOU (Jaccard Similarity Coefficient) and Structural Similarity Index [31]. We chose Mean-IOU because its the gold standard

for measuring segmentation results for many Semantic Segmentation Challenges such as Pascal-VOC2012 [6], MS-COCO [19]. Contrarily, SSIM is a standard metric for evaluating GANs for image-to-image translation tasks. As illustrated in all the tables, our model outperforms both UNet derived architectures and recent GAN based models in terms of AUC-ROC, Mean-IOU, and SSIM, the three main metrics for this task. M-GAN achieves better Specificity and Accuracy in CHASE-DB1 and STARE. However, higher Specificity means better background pixel segmentation (True Negative), which is less essential than having better retinal vessel segmentation (True Positive). We want both, better Sensitivity and AUC-ROC, which equates to having a higher confidence score. In Fig. 4 we can see that our True positive Rate is always better than other architectures for all three data-set. We couldn't report SSIM and Mean-IOU for some of the architectures as source codes and pre-trained, weights weren't provided.

4 Conclusion

In this paper, we proposed a new multi-scale generative architecture called RV-GAN. By combining our novel featuring matching loss, the architecture synthesizes precise venular structure segmentation with high confidence scores for two relevant metrics. As a result, we can efficiently employ this architecture in various applications of ophthalmology. The model is best suited for analyzing retinal degenerative diseases and monitoring future prognosis. We hope to extend this work to other data modalities.

Acknowledgments. This material is based upon work supported by the National Aeronautics and Space Administration under Grant No. 80NSSC20K1831 issued through the Human Research Program (Human Exploration and Operations Mission Directorate).

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](https://arxiv.org/abs/1802.06955) (2018)
2. Chen, X., Xu, C., Yang, X., Tao, D.: Attention-GAN for object transfiguration in wild images. In: Proceedings of the European Conference on Computer Vision, pp. 164–180 (2018)
3. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: StarGAN: unified generative adversarial networks for multi-domain image-to-image translation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8789–8797 (2018)
4. Choi, Y., Uh, Y., Yoo, J., Ha, J.W.: StarGAN V2: diverse image synthesis for multiple domains. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8188–8197 (2020)
5. Chollet, F.: Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1251–1258 (2017)
6. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: a retrospective. *Int. J. Comput. Vis.* **111**(1), 98–136 (2015). <https://doi.org/10.1007/s11263-014-0733-5>
7. Fraz, M.M., Remagnino, P., Hoppe, A., Uyyanonvara, B., Rudnicka, A.R., Owen, C.G., Barman, S.A.: Blood vessel segmentation methodologies in retinal images-a survey. *Comput. Methods Progr. Biomed.* **108**(1), 407–433 (2012)

8. Hoover, A., Kouznetsova, V., Goldbaum, M.: Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. *IEEE Trans. Med. Imaging* **19**(3), 203–210 (2000)
9. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1125–1134 (2017)
10. Jin, Q., Meng, Z., Pham, T.D., Chen, Q., Wei, L., Su, R.: DUNet: a deformable network for retinal vessel segmentation. *Knowl.-Based Syst.* **178**, 149–162 (2019)
11. Kamran, S.A., Saha, S., Sabbir, A.S., Tavakkoli, A.: Optic-Net: a novel convolutional neural network for diagnosis of retinal diseases from optical tomography images. In: *2019 18th IEEE International Conference on Machine Learning And Applications (ICMLA)*, pp. 964–971 (2019)
12. Kamran, S.A., Hossain, K.F., Tavakkoli, A., Zuckerbrod, S., Baker, S.A., Sanders, K.M.: Fundus2Angio: a conditional GAN architecture for generating fluorescein angiography images from retinal fundus photography. In: *Bebis, G., et al. (eds) Advances in Visual Computing*, vol. 12510, pp. 125–138. Springer, Heidelberg (2020). https://doi.org/10.1007/978-3-300-64559-5_10
13. Kamran, S.A., Tavakkoli, A., Zuckerbrod, S.L.: Improving robustness using joint attention network for detecting retinal degeneration from optical coherence tomography images. *arXiv preprint arXiv:2005.08094* (2020)
14. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014)
15. Li, C., Wand, M.: Precomputed real-time texture synthesis with Markovian generative adversarial networks. In: *Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) Computer Vision*, vol. 9907, pp. 702–716. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-319-46487-9_43
16. Li, L., Verma, M., Nakashima, Y., Nagahara, H., Kawasaki, R.: IterNet: retinal image segmentation utilizing structural redundancy in vessel networks. In: *The IEEE Winter Conference on Applications of Computer Vision*, pp. 3656–3665 (2020)
17. Li, X., Chen, H., Qi, X., Dou, Q., Fu, C.W., Heng, P.A.: H-DenseUNet: hybrid densely connected UNet for liver and tumor segmentation from CT volumes. *IEEE Trans. Med. Imaging* **37**(12), 2663–2674 (2018)
18. Lim, J.H., Ye, J.C.: Geometric GAN. *arXiv preprint arXiv:1705.02894* (2017)
19. Lin, T.Y., et al.: Microsoft COCO: common objects in context. In: *Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds) Computer Vision*, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
20. Owen, C.G., et al.: Measuring retinal vessel tortuosity in 10-year-old children: validation of the computer-assisted image analysis of the retina (CAIAR) program. *Investigat. Ophthalmol. Vis. Sci.* **50**(5), 2004–2010 (2009)
21. Park, K.B., Choi, S.H., Lee, J.Y.: M-GAN: retinal blood vessel segmentation by balancing losses through stacked deep fully convolutional networks. *IEEE Access* **8**, 146308–146322 (2020)
22. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2337–2346 (2019)
23. Ricci, E., Perfetti, R.: Retinal blood vessel segmentation using line operators and support vector classification. *IEEE Trans. Med. Imaging* **26**(10), 1357–1365 (2007)
24. Ronneberger, O., Fischer, P., Brox, T.: 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*, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28

25. Shaham, T.R., Dekel, T., Michaeli, T.: SinGAN: learning a generative model from a single natural image. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4570–4580 (2019)
26. Soares, J.V., Leandro, J.J., Cesar, R.M., Jelinek, H.F., Cree, M.J.: Retinal vessel segmentation using the 2-D Gabor wavelet and supervised classification. *IEEE Trans. Med. Imaging* **25**(9), 1214–1222 (2006)
27. Son, J., Park, S.J., Jung, K.H.: Retinal vessel segmentation in fundoscopic images with generative adversarial networks. arXiv preprint [arXiv:1706.09318](https://arxiv.org/abs/1706.09318) (2017)
28. Staal, J., Abràmoff, M.D., Niemeijer, M., Viergever, M.A., Van Ginneken, B.: Ridge-based vessel segmentation in color images of the retina. *IEEE Trans. Med. Imaging* **23**(4), 501–509 (2004)
29. Tavakkoli, A., Kamran, S.A., Hossain, K.F., Zuckerbrod, S.L.: A novel deep learning conditional generative adversarial network for producing angiography images from retinal fundus photographs. *Sci. Rep.* **10**(1), 1–15 (2020)
30. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional GANs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8798–8807 (2018)
31. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
32. Yang, T., Wu, T., Li, L., Zhu, C.: SUD-GAN: deep convolution generative adversarial network combined with short connection and dense block for retinal vessel segmentation. *J. Digit. Imaging* 1–12 (2020). <https://doi.org/10.1007/s10278-020-00339-9>
33. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-attention generative adversarial networks. In: International Conference on Machine Learning, pp. 7354–7363 (2019)