Supervised Segmentation of Un-Annotated Retinal Fundus Images by Synthesis

He Zhao, Huiqi Li, Sebastian Maurer-Stroh, Yuhong Guo, Qiuju Deng, and Li Cheng

***Abstract*— We focus on the practical challenge of segmenting new retinal fundus images that are dissimilar to existing well-annotated data sets. It is addressed in this paper by a supervised learning pipeline, with its core being the construction of a synthetic fundus image data set using the proposed R-sGAN technique. The resulting synthetic images are realistic-looking in terms of the query images while maintaining the annotated vessel structures from the existing data set. This helps to bridge the mismatch between the query images and the existing well-annotated data set. As a consequence, any known supervised fundus segmentation technique can be directly utilized on the query images, after training on this synthetic data set. Extensive experiments on different fundus image data sets demonstrate the competitiveness of the proposed approach in dealing with a diverse range of mismatch settings.**

***Index Terms*— Biomedical optical imaging, image segmentation, phantoms.**

I. Introduction

RETINAL fundus image segmentation is a fundamental step in retinal image analysis and the follow-up ophthalmic diagnostics [1], [2]. Due to the laborious nature of manual annotation by domain experts, only a small set of annotated vessel structures in fundus images is available. Notable examples include the set of 10 or 20 training images in the STARE [3] or DRIVE [4] fundus image benchmarks, respectively. Moreover, there is no any vessel structure annotation for many fundus image datasets (e.g. Kaggle [5]). To further complicate the matter, different fundus image datasets often exhibit distinct textural appearances. This is illustrated in Fig. 1, where exemplar fundus images from DRIVE, STARE, HRF [6], Kaggle, a mobile fundus imaging dataset (Mobile [7]), and our clinical dataset (Anzhen) are showcased. Visual discrepancies among these datasets are large enough so that direct usage of segmentation models learned on existing annotated dataset (e.g. DRIVE/STARE) to a new dataset does not perform well. This phenomenon is in fact commonly presented in everyday clinical practice. Even for the same type of device, the obtained fundus images may vary significantly due to the variabilities in clinical settings, subjects, and acquisition protocols. It is a more challenging problem of performing fundus image segmentation on a new and distinct fundus image dataset in the absence of manual annotations.

To address this problem, we present in this paper a supervised learning pipeline. The key point in our approach is the construction of a synthetic fundus image dataset that is capable of bridging the gap between an existing reference dataset (where annotated vessel structures are available) and the new query dataset (where no annotation is available). In this way, existing supervised/deep learning methods for fundus image segmentation can be engaged to learn a model dedicated to the set of query images. The contribution of our approach can be summarized into two aspects: first, to the best of our knowledge our work is the first to address such a practical fundus segmentation problem by leveraging the existing labeled dataset; second, the proposed R-sGAN technique is capable of synthesizing fundus images that are realistic-looking in terms of the query images, while preserving the annotated vessel structures of the reference dataset. Our approach has been tested on a wide range of fundus image datasets and superior performance is obtained. The implementation of our approach and the results are also made publicly accessible [1].

II. Related Work

Due to its clinical importance, fundus image segmentation has received ample attention [1], [8] over the years. Existing methods can be roughly divided into two categories based on whether an annotated training set is required: unsupervised and supervised methods. Supervised methods learn their models

Manuscript received May 31, 2018; revised July 4, 2018; accepted July 8, 2018. Date of publication July 24, 2018; date of current version December 28, 2018. This work was supported by A*STAR JCO Grants. The work of H. Zhao was supported by the CSC Chinese Government Scholarship. *(Corresponding authors: Huiqi Li; Li Cheng.)*

H. Zhao is with the Beijing Institute of Technology, Beijing 100081, China, and also with the Bioinformatics Institute, A*STAR, Singapore 138671.

H. Li is with the Beijing Institute of Technology, Beijing 100081, China (e-mail: huiqili@bit.edu.cn).

S. Maurer-Stroh is with the Bioinformatics Institute, A*STAR, Singapore 138671.

Y. Guo is with the School of Computer Science, Carleton University, Ottawa, ON K1S 5B6, Canada.

Q. Deng is with the Beijing Anzhen Hospital, Beijing Institute of Heart, Lung, and Blood Vessel Diseases, Beijing 100029, China.

Li Cheng is with the Bioinformatics Institute, A*STAR, Singapore 138671, and also with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2R3, Canada (e-mail: chengli@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMI.2018.2854886

[1] Results and source code are available after acceptance at https://web.bii.a-star.edu.sg/archive/machine_learning/Projects/filaStructObjs/Segmentation/filaSegBySyn/

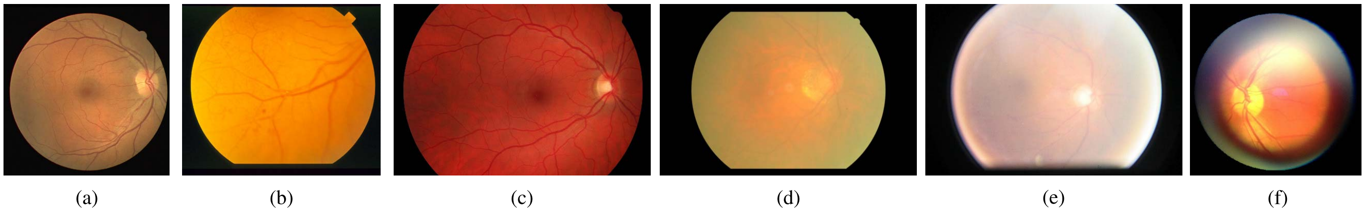


Fig. 1. An illustration of the observation that different fundus image datasets often exhibit distinct visual textural appearances. Here exemplar images from DRIVE, STARE, HRF, Anzhen, Kaggle and Mobile datasets are displayed.

based on a set of training examples, while unsupervised methods do not require a training set.

Regarding unsupervised methods in retinal vessel segmentation, Hessian-based techniques have been proposed to utilize the second order derivatives to characterize the foreground boundaries [9], or to incorporate the eigenvalues [10] to facilitate the delineation of vascular structures. Nonetheless they could be awkward when dealing with the irregular-shaped vessels. Alternatively, morphological information can be incorporated as prior knowledge in unsupervised methods. For example, Zana and Klein [11] propose a method based on mathematical morphology and curvature evaluation to segment the vessel structures from background. The recent work of Multi-scale line detector (MSLD) [12] is arguably one of the most powerful unsupervised fundus segmentation methods, which is based on multi-scale line detection. Different from the above methods, supervised methods [13]–[16] have been very popular recently, where excellent segmentation results can often be obtained by exploiting the available training dataset. Becker *et al.* [14] propose a kernel boosting framework to learn the filters. In [16], the structured and contextual features are extracted to train gradient boosted tree as classifier in identifying vessel foregrounds. In particular, noticeable progress has been made by the deep learning based methods [17], [18] in the past few years, with performance surpassing even human annotators on established datasets. In [15], Ronneberger *et al.* proposed a U-net structure with skip-connection to produce image to image segmentation. Further improvement is obtained in the deep retinal image understanding (DRIU) method [18] by fine-tuning the pretrained visual geometry group (VGG) networks and extracting specialized layers. This inspires us to consider the exploration of supervised deep learning techniques in our context.

Segmentation of unlabeled images can be treated as the issue of unsupervised domain adaptation [19], which provides a different perspective of our work. Unsupervised domain adaptation concerns on transferring feature representations from an annotated source dataset to a relevant target domain where there are few or none annotations. In this view, our problem can be recast as a domain adaptation with none target labels. Recent work of domain adaptation has focused on deep learning of the feature representations by either directly minimizing the discrepancy between source and target domains [20]–[22], or via a latent embedding space [23]. The work of [22], [23] is particularly close, as they also engage the generative adversarial networks (GANs) [24]. Meanwhile, these methods [22], [23] primarily focus on other applications such as city scenes and handwritten digits, where biomedical applications are not considered. Moreover, these methods are much more complicated, where losses of transferring from source to target, and backward from target to source are involved, among other complicated losses [23]. As a result, dedicated deep learning models are trained with to achieve the goal.

We would also like to mention the related efforts in fundus image synthesis, style transfer, and recurrent neural networks. One of the earliest efforts in fundus image synthesis is perhaps for surgical simulations [25]. Later efforts [26], [27] are primarily based on prior knowledge of the underlying physics law and statistical modeling. More recently, the work of [28], [29] focus on *data-driven* fundus image synthesis, that is, the synthesized phantoms bear the same textural characteristics of the set of training fundus images. One major underpinning technique of them is the GANs introduced by Goodfellow *et al.* [24] and its improved variant [30]. Image style transfer has been studied in textural analysis community under various names such as image analogies [31], [32]. Recent work of Gatys *et al.* [33] successfully demonstrates the applications in artistic drawing. On the other hand, the so called recurrent neural network or RNN has emerged in a wide range of applications such as speech recognition [34], language modeling [35], imaging captioning [36], and image generation [37]. Long-Short Term Memory (LSTM) [38] and Gated Recurrent Unit (GRU) [39] are the recent developments to overcome the vanishing gradient phenomenon when training RNNs. In our context, a variant of GRU is proposed to obtain a more compact form of deep learning representation, where the key essence of GANs has been incorporated as the generator gate, and the style transfer losses have also been utilized to enforce faithful representation of the style and content from inputs.

III. Our Approach

Let us start by stating the problem of interest: given an existing well-annotated retinal image dataset and a testing set without annotation (images are usually dissimilar to the existing annotated dataset), we would like to train a supervised segmentation model that can best segment the testing image set. Using the domain adaptation terminology, the known annotated fundus images dataset can be regarded as the source dataset $\mathcal{D}^{(s)} = \left\{\left(\boldsymbol{x}_i^{(s)}, \boldsymbol{y}_i^{(s)}\right)\right\}_{i=1}^{n_s}$, where $\left(\boldsymbol{x}_i^{(s)}, \boldsymbol{y}_i^{(s)}\right)$ denotes a pair of raw fundus image and the corresponding pixelwise annotation of its vessel structure, and n_s denotes the number

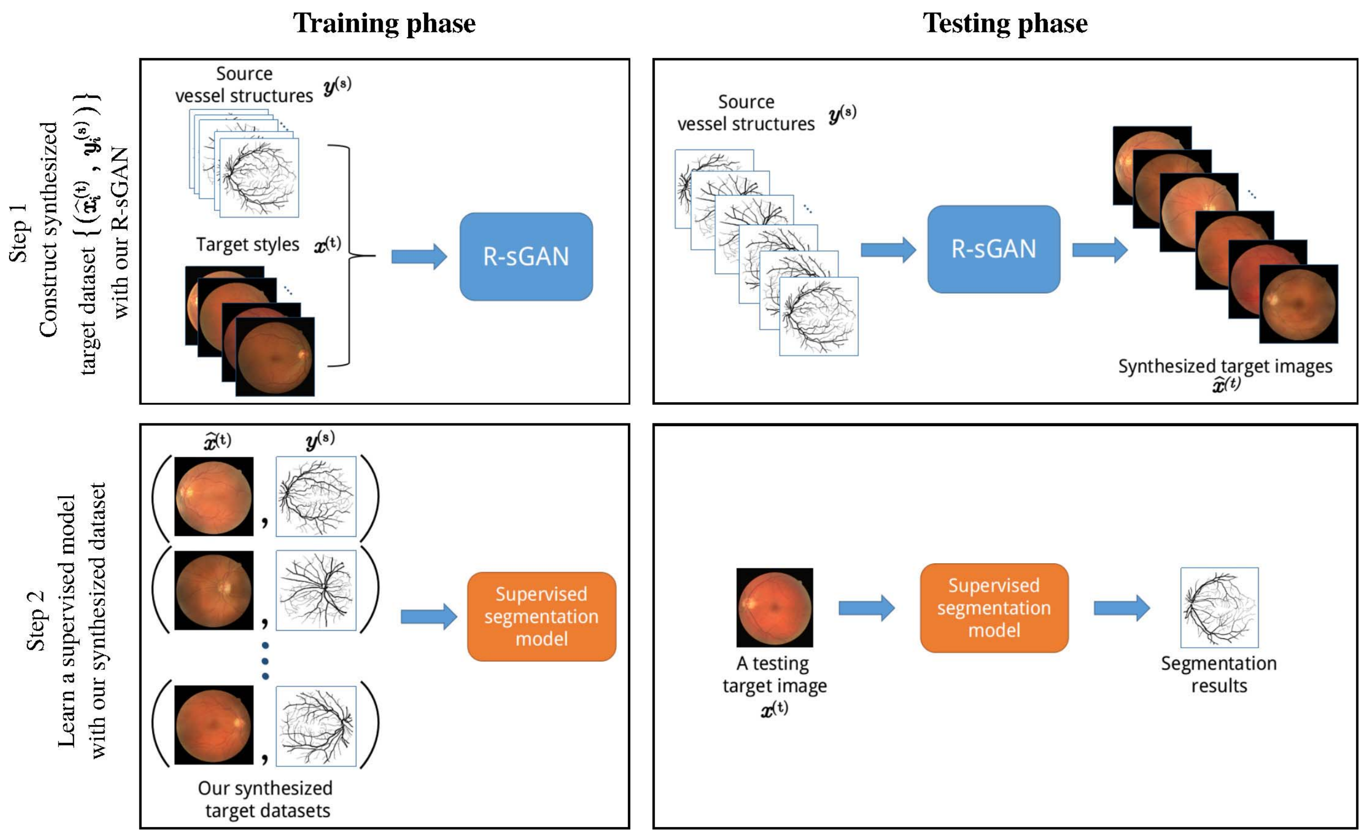

Training phase
Testing phase
Step 1
Construct synthesized target dataset {(x̂_i^(t), y_i^(s))} with our R-sGAN
Source vessel structures y^(s)
Target styles x^(t)
R-sGAN
Source vessel structures y^(s)
R-sGAN
Synthesized target images x̂^(t)
Step 2
Learn a supervised model with our synthesized dataset
x̂^(t)
y^(s)
Supervised segmentation model
Our synthesized target datasets
A testing target image x^(t)
Supervised segmentation model
Segmentation results

Fig. 2. The flowchart of our approach that contains two steps. Step 1 involves the construction of synthesized target training set with the vessel structures of the reference training set and the textural appearance of the target query images. Step 2 is a supervised segmentation method to train a dedicated target model.

of such pairs. The goal is to automatically segment the new testing set images $\left\{\boldsymbol{x}_j^{(t)}\right\}_{j=1}^{n_t}$ by engaging a state-of-the-art supervised segmentation method. Note there is no label information on the set of testing or target images.

To tackle this problem, a two-step pipeline is proposed in our work. As illustrated in Fig. 2, step 1 focuses on the construction of a synthesized target dataset, $\hat{\mathcal{D}}^{(t)} = \left\{\left(\hat{\boldsymbol{x}}_i^{(t)}, \boldsymbol{y}_i^{(s)}\right)\right\}_{i=1}^{n_t'}$. The label $\boldsymbol{y}_i^{(s)}$ of each training example is precisely a segmentation annotation from the existing training set, while its corresponding instance $\hat{\boldsymbol{x}}_i^{(t)}$ is a synthesized fundus image with the style (i.e. textures) of target image set. After such a synthesized dataset is available, step 2 proceeds to learn a supervised segmentation model dedicated to the query images. Generally speaking, any existing supervised segmentation method can be engaged here.

The main technical innovation of this paper lies on the proposed R-sGAN technique to generate synthesized images. Our synthesis approach can generate realistic-looking training images containing the content (i.e. vessel structures) of the source dataset with the same textural style of the target images as depicted in Fig. 2. Different from [29], a recurrent network is considered in our approach. The generator is embedded into the cell structure of recurrent neural network as a gate to generate target style images based on current input and previous cell state. This combination of GAN with recurrent fashion enables us to learn a more compressed model. With the advantage of recurrence, multiple styles of images can be generated at the test phase once the model is trained. We will focus on the proposed R-sGAN in what follows.

A. R-sGAN

Our R-sGAN is a non-linear variant of the GRU [39], a form of recurrent neural networks. In particular, non-linear neural net functions are used in place of the matrix multiplications of a typical GRU, in a way similar to the convolutional LSTM idea in [40]. Moreover, as illustrated in Fig. 3(b-c), the generator of GAN [24] is incorporated in the R-sGAN cell as a generator gate, while the discriminator of GAN is included as part of the loss function. These adaptations enable our R-sGAN model to synthesize realistic-looking fundus images efficiently and effectively, and in particular, it enables to synthesize images of multiple styles from one model.

More specifically, as presented in Fig. 3(a), our R-sGAN model is composed of a sequence of cells connected by the cell states, $\boldsymbol{h}_0, \boldsymbol{h}_1, \ldots, \boldsymbol{h}_T$, where the first cell state $\boldsymbol{h}_0$ is initialized with zero values. Each cell at time τ acts as a local conditional model, which consists of the following three components: reset gate $\boldsymbol{r}_\tau$, generator gate $\widetilde{\boldsymbol{x}}_\tau$, and update gate $\boldsymbol{u}_\tau$. Similar to the original GRU, the reset and update gates $\boldsymbol{r}_\tau$ and $\boldsymbol{u}_\tau$ are engaged for guiding the current information flow, while the generator gate $\widetilde{\boldsymbol{x}}_\tau$ combines the current vessel structure, a noise code, and current cell state to produce synthesized images, which is illustrated in Fig. 3(b).

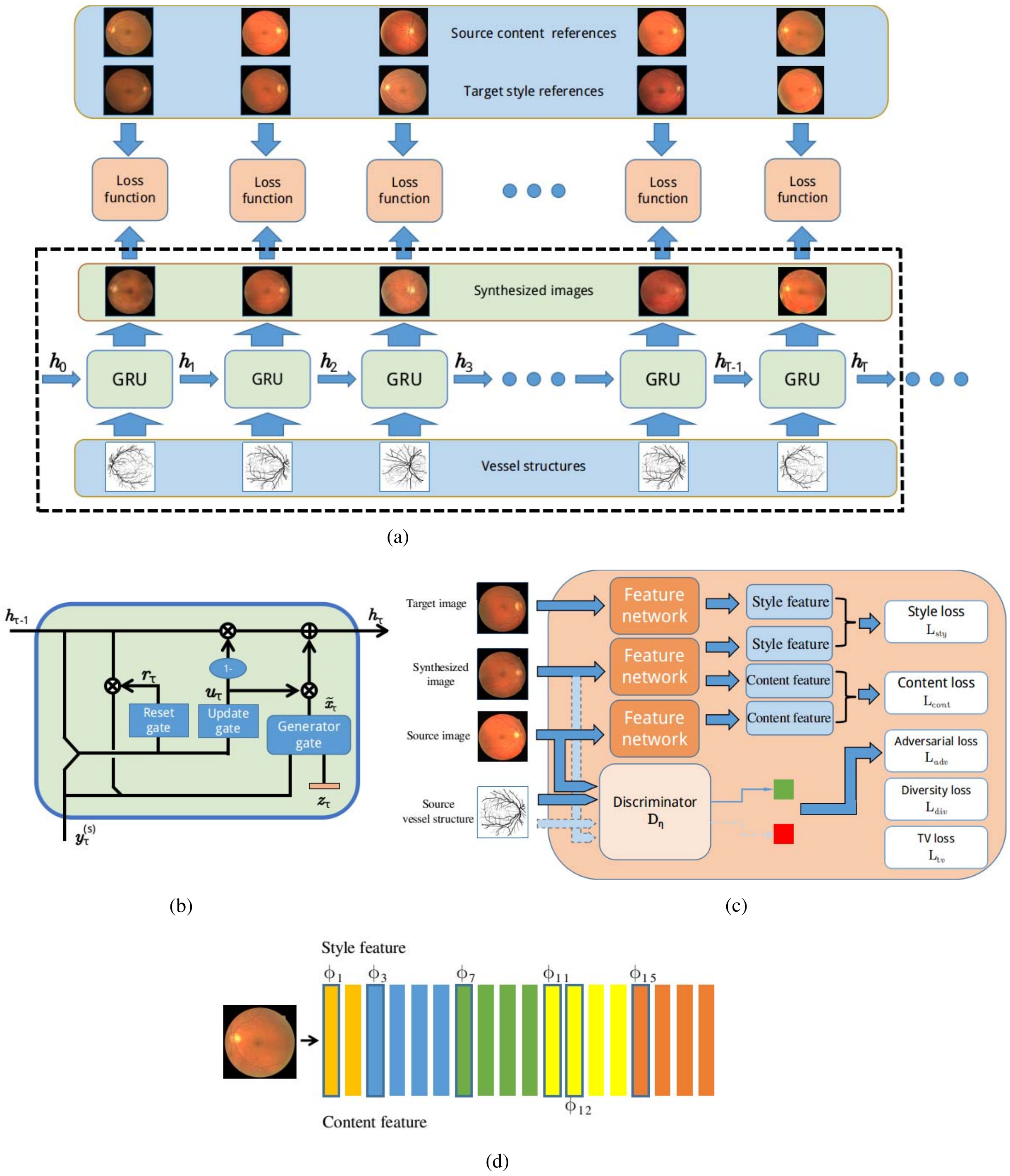


Fig. 3. (a) The proposed R-sGAN synthesis process, where the GRU component is utilized and depicted in (b). The involved loss functions are summarized in (c). The generator gate adopts the U-net structure [15], while inspired by the GAN, a corresponding discriminator is considered in (c) that also induces the adversarial loss and the diversity loss. The style and content features considered in our approach are directly obtained from the deep learning representation. Without loss of generality, the VGG network is considered in our work to extract these features, which is illustrated in (d).

1) Reset Gate: The reset gate $\boldsymbol{r}_\tau$ dictates how much information of the previous cell state can enter to the current cell. This allows the temporal network to drop any information that is irrelevant in future, while keeping a compact representation. In our work, the reset gate corresponds to a fully convolutional network (FCN) $\boldsymbol{f}_\gamma$, followed by the $sigmoid$ activation function:

$$\boldsymbol{r}_\tau = \sigma\left(\boldsymbol{f}_\gamma\left(\boldsymbol{y}_\tau^{(s)}, \boldsymbol{h}_{\tau-1}\right)\right), \tag{1}$$

where $\boldsymbol{y}_\tau^{(s)}$ is the source vessel structure input at time step τ, $\boldsymbol{h}_{\tau-1}$ refers to the previous cell state. Details of the FCN function $\boldsymbol{f}_\gamma$ is relegated to Section II of supplementary file (available in the supplementary file /multimedia tab.) due to space constraints. After the reset gate, the cell state is changed to

$$\widetilde{\boldsymbol{h}}_{\tau-1} = \boldsymbol{r}_\tau \otimes \boldsymbol{h}_{\tau-1}, \tag{2}$$

with $\otimes$ denoting element-wise multiplication.

2) Generator Gate: The U-net structure of [15], [29] is employed here to generate synthetic images. The θ parameterized generator G with the input of the source vessel structure $\boldsymbol{y}_\tau^{(s)}$ and the cell state from the previous time step can produce a synthetic image that is similar to its target style fundus image $\boldsymbol{x}_\tau^{(t)}$. This amounts to learn a mapping

$$\widetilde{\boldsymbol{x}}_\tau = G_\theta\left(\boldsymbol{y}_\tau^{(s)}, \widetilde{\boldsymbol{h}}_{\tau-1}, z_\tau\right). \tag{3}$$

Here $\boldsymbol{y}_\tau$ is the source vessel structure which the generator is conditioned on, z_τ is the noise code to diversify the generated images.

3) Update Gate: Similar to reset gate, the update gate $\boldsymbol{u}_\tau$ also consists of a FCN $\boldsymbol{f}_\mu$ which is followed by a sigmoid activation function. Different from other gates, the update gate controls how much information in the previous state can be updated to a new cell state. This is realized by considering a momentum term $\boldsymbol{u}_\tau$ which is inspired by the stochastic meta descent in non-linear optimization [41]. Concretely, $\boldsymbol{u}_\tau$ is computed by

$$\boldsymbol{u}_\tau = \sigma\left(\boldsymbol{f}_\mu\left(\boldsymbol{y}_\tau^{(s)}, \boldsymbol{h}_{\tau-1}\right)\right). \tag{4}$$

The network architectural details of $\boldsymbol{f}_\mu$ are elaborated in Section II of the supplementary file (available in the supplementary file /multimedia tab). After the update gate, the signal obtained from the generator G becomes

$$\hat{\boldsymbol{x}}_\tau^{(t)} = (1 - \boldsymbol{u}_\tau) \otimes \boldsymbol{h}_{\tau-1} \oplus \boldsymbol{u}_\tau \otimes \widetilde{\boldsymbol{x}}_\tau, \tag{5}$$

where $\oplus$ refers to element-wise addition. With the help of the update and reset gates, our GRU cell learns to capture dependencies over the sequence of inputs. On the other hand, the generator gate concentrates on synthesis with GAN.

B. Loss Function

Our R-sGAN is inspired by the adversarial learning methodology of GAN [24]. That is, in addition to the generator G, a discriminator D_η or short-hand D is also considered, as $d_\tau = D_\eta(\boldsymbol{x}_\tau, \boldsymbol{y}_\tau)$ with $d_\tau \in [0, 1]$. When the input $\boldsymbol{x}_\tau$ is a phantom $\hat{\boldsymbol{x}}^{(t)}$, ideally $d_\tau \to 0$; if the input is a real retinal image $\boldsymbol{x}^{(t)}$, ideally $d_\tau \to 1$.

Following [24], the discriminator D is learned by maximizing

$$L_D = \sum_i \left(\sum_\tau \log D\left(\boldsymbol{x}_{i,\tau}^{(s)}, \boldsymbol{y}_{i,\tau}^{(s)}\right) + \log\left(1 - D(\hat{\boldsymbol{x}}_{i,\tau}^{(t)}, \boldsymbol{y}_{i,\tau}^{(s)})\right)\right), \tag{6}$$

where i presents the i-th training sequence, the index pair (i, τ) indicates the τ-th time step in the i-th sequence. On the other hand, the other player (i.e. the generator) G is updated by minimizing an objective function that can be decomposed into five losses as follows.

1) Adversarial Loss: During training, the goal of generator G in each cell is to synthesize realistic-looking images that can fool the discriminator D. Following [24], the objective function for this purpose is the adversarial loss

$$L_{\text{adv}} = -\log D\left(\hat{\boldsymbol{x}}^{(t)}, \boldsymbol{y}^{(s)}\right). \tag{7}$$

In order to simplify our notation, the indices i and τ are not imposed, which will be straightforward to deploy when necessary.

2) Style and Content Losses: To evaluate how faithful our synthesized phantom is with respect to a specific target fundus image, the style and content losses of [33] are utilized here. For style loss, based on a deep learning feature representation, the Gram matrix is considered to characterize pairwise textural correlations between feature responses. Experimentally, the convolutional neural network of VGG-19 [42] is utilized here for feature representation. The Gram matrix $G^l = (g_{mn}^l)$ is defined as the inner product between the m-th and n-th vectorized feature maps in the l-th layer:

$$g_{mn} = \sum_k \phi'_{mk}\phi'_{nk}. \tag{8}$$

ϕ' is the vectorized version of feature map ϕ in VGG-19 networks, as is also illustrated in Fig. 3(d). Now, With the style representation G^l of the target style $\boldsymbol{x}_\tau^{(t)}$ and the phantom $\hat{\boldsymbol{x}}^{(t)}$, its style loss can be defined as

$$L_{\text{sty}} = \sum_l \sum_{m,n} \frac{\omega_l}{W_l H_l} \left\| g_{mn}^l\left(\boldsymbol{x}^{(t)}\right) - g_{mn}^l\left(\hat{\boldsymbol{x}}^{(t)}\right)\right\|^2. \tag{9}$$

Here ω_l depicts the weight of the l-th layer Gram matrix and is set to 0.2 in practice. W_l and H_l are the width and height of the feature map ϕ_l.

The content loss is also engaged to enforce the generated image retaining the prescribed vessel structure, which is defined as

$$L_{\text{cont}} = \sum_l \frac{1}{W_l H_l} \left\| \phi^l\left(\boldsymbol{x}^{(s)}\right) - \phi^l\left(\hat{\boldsymbol{x}}^{(t)}\right)\right\|_F^2. \tag{10}$$

3) Diversity Loss: One main drawback of existing GAN techniques such as [30] is that they tend to generate fixed output for a given input. To encourage diversity in the generated images, we proposed a diversity loss with a simple idea: if the input noises z and z' bear noticeable difference, the resulting phantoms $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{x}}'$ should also be different. It leads to the following loss function

$$L_{\text{div}} = -\left\|z - z'\right\|_2 \cdot \left\|\hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}'\right\|_1. \tag{11}$$

4) Total Variation Loss: In support of spatial smoothness in the generated phantom $\hat{\boldsymbol{x}}$, the following total variation loss is also considered, as

$$L_{\text{tv}} = \sum_{w,h}\left(\left\|\hat{x}_{w,h+1} - \hat{x}_{w,h}\right\|_2^2 + \left\|\hat{x}_{w+1,h} - \hat{x}_{w,h}\right\|_2^2\right). \tag{12}$$

where $w \in 1, \ldots, W$, $h \in 1, \ldots, H$, and $\hat{x}_{w,h}$ denotes the pixel value of a given location in phantom image $\hat{\boldsymbol{x}}$ with width W and height H.

All above loss functions together give rise to

$$L_G = \omega_{\text{adv}} L_{\text{adv}} + \omega_{\text{sty}} L_{\text{sty}} + \omega_{\text{cont}} L_{\text{cont}} + \omega_{\text{div}} L_{\text{div}} + \omega_{\text{tv}} L_{\text{tv}}. \quad (13)$$

In practice, the values of ω_{adv}, ω_{div}, and ω_{tv} are empirically fixed to 1, 0.6, 100 respectively. The values of ω_{sty}, ω_{cont} are dataset dependent, which will be stated later in the experimental section. As stated earlier, during training our generator is updated by minimizing L_G mentioned above, while the discriminator is to maximize Eq. (6).

IV. Experimental Set-Up

A. Datasets

The benchmark datasets we used in our paper are DRIVE [4], STARE [3], high-resolution fundus or HRF [6], IOSTAR [43]. DRIVE contains 20/20 (train/test) fundus images with size of 584 × 565 pixels. Meanwhile STARE contains 10/10 fundus images of size 605 × 700 pixels. IOSTAR is a dataset of 24 images acquired from scanning laser ophthalmoscope with a resolution of 1024 × 1024 pixels, with the first half for training set. HRF is a high resolution fundus image dataset with image size of $3,304 \times 2,336$ pixels, with a train/test split of 22/23 being adopted here. The vessel structure annotations are all available for these benchmarks.

We also consider recent retinal fundus datasets that come without vessel structure annotations as the aforementioned benchmarks, including Kaggle [5], Mobile [7], and our Anzhen datasets. For the Kaggle dataset [5], here we focus on the subset images of the most severe rating category (i.e. Proliferative diabetic retinopathy) as these images are with a diverse range of textures, thus are challenging to handle. The Mobile dataset contains relatively low-quality fundus images captured by a smart-phone based ophthalmoscope, a product of manufacturer Welch Allyn. Ten images obtained from Xu *et al.* [7] are used in our experiment. Finally, the Anzhen dataset is collected in Beijing Anzhen Hospital during the past 7 years from cataract patients aged between 35-64. The images are captured using a Canon CR2 non-mydriatic camera with a 45° angle of view at a resolution of $2,544 \times 1,696$. The cataract is a common pathological disease where the main branches of retinal vein or artery are dimly visible in retinal image, as a result of progressively clouding of the eye lens that leads to a decrease in vision.

B. Pre-/Post-Process for Synthesized Dataset Construction

In this paper, our R-sGAN is trained with a standard image size of 512 × 512. Consider for example the DRIVE dataset as the source dataset: as the images in DRIVE are of size 584 × 565, we first crop the images to 565 × 565, then resize them to 512 × 512 via bicubic interpolation. Besides the vessel structures, the optical disk is also considered when training the R-sGAN. In other words, when constructing the synthesized target image dataset, both DRIVE vessel structures and the corresponding optical disk annotations are used as the source inputs to G_θ, which are collectively referred to as the vessel structures when there is no confusion. The generated images are then resized to be the same size of the source fundus images. Similar pre- & post-processes are carried out when working with other source and target datasets.

C. Model Architecture of the Proposed R-sGAN

Our R-sGAN has been introduced in previous sections, as well as illustrated in Fig. 3. Here we emphasize on its practical aspects. The stem of our R-sGAN is Gated Recurrent Unit, whose structure is described in Fig. 3(c). The two fully convolutional networks, $\boldsymbol{f}_\gamma$ and $\boldsymbol{f}_\mu$, are used in the reset gate and update gates, respectively. Both of them consist of two convolutional and deconvolutional layers with weights γ and μ learned by minimizing the loss function L_G. The network G_θ of the generator gate consists of 6 convolutional layers and 5 transposed convolutional layers with a kernel size of $4 \times 4 \times l_f$, where l_f is self-manifested by the third dimension of its consecutive layer. The skip connection structure of U-net [15] is also adopted in G_θ. The discriminator D_η is built by 5 convolutional layers with the same kernel size just described, as well as a fully connected layer to produce the final prediction. The length of noise code z is 400. More detailed information of the neural networks are provided in Section II of the supplementary file due to space constraints. In addition, the layers in VGG-19 to extract style and content features are at 1st, 3rd, 5th, 9th, 15th / 10th layers, respectively. Regarding the values of the style and content losses, ω_{sty}, ω_{cont}: the values are set to 10 and 1 respectively for both DRIVE and HRF. Their values are further fixed to 15 and 3 respectively for STARE, and to 5 and 1 respectively for IOSTAR dataset empirically. The discussion of parameter settings is provided in Section XI of supplementary file.

D. Computation Time

Our experiments are carried out on a standard desktop with an Intel iCore 7 CPU and a Titan-X GPU of 12GB memory. During experiments, we fix $T = 5$ for sequence length due to the GPU memory limitation. The training time of our R-sGAN using Python implementation amounts to 347 minutes. At test run of step one, the average time of synthesizing a DRIVE-size fundus image is 0.4471 second.

V. Experiments

A. Examples of the Synthesized Images

The experiments here emphasize on qualitative examination of the synthesized images. Fig. 4 presents several exemplary synthesized images: in each pair, the left image presents the vessel structure from the source dataset, the middle one is the query image of the target dataset, and the right one is the synthesized image. More specifically, the source dataset is STARE for (a)(c), and DRIVE for (b)(d). The target dataset is DRIVE for (a)(c), STARE for (b), and HRF for (d). These visual results demonstrate that the generated images are capable of capturing the textual style from the target images as well as the vessel

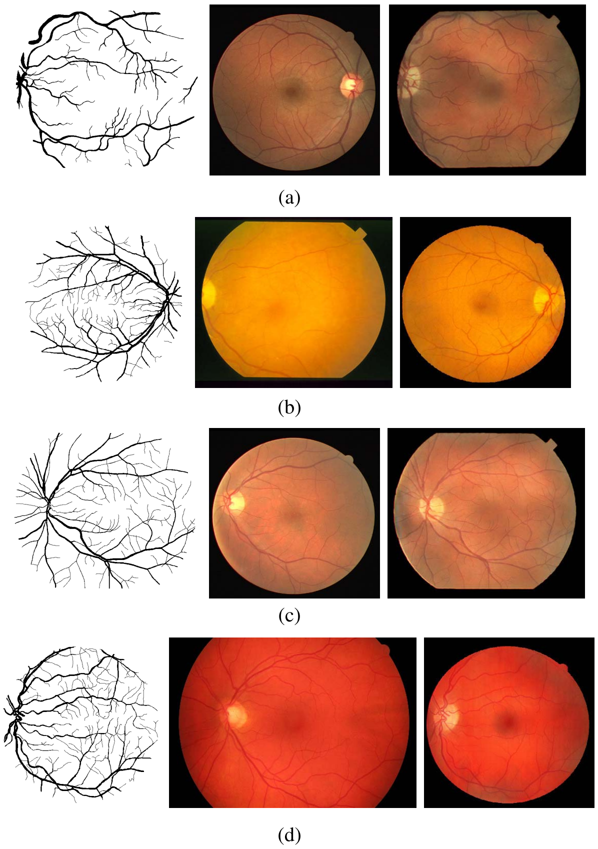


Fig. 4. A gallery of synthesized images with different target styles. Each subplot displays a triplet of source vessel structure, the target style and one of the synthesized images: on the left is an exemplar source vessel structure, in the middle is an exemplar target style image; On the right is an exemplar generated phantom having the same style of the target dataset while maintaining the vessel structure of the source dataset. See text for details. (a) Source: STARE, target: DRIVE, (b) Source: DRIVE, target: STARE, (c) Source: STARE, target: DRIVE, (d) Source: DRIVE, target: HRF.

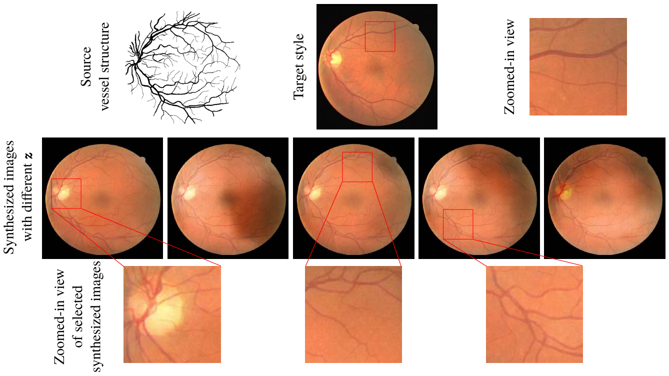

Source vessel structure
Target style
Zoomed-in view
Synthesized images with different **z**
Zoomed-in view of selected synthesized images

Fig. 5. An illustration that distinct phantom outputs are generated from the same source vessel structure but with different noise code z. Top row displays the source vessel and target style. Second row shows the diverse synthesized images. The last row presents the zoomed-in view of selected parts.

structures from the source training set. More results of the synthesized images could be found in the supplementary file.

Moreover, Fig. 5 displays exemplar phantom images generated with the same source vessel structure and target style, but with different noise code z. To focus on the promising capacity of our approach in generating multiple distinct phantoms from the same input images, here the source vessel structure and the target style are from the training set and the testing set of the same DRIVE dataset, respectively. From Fig. 5, it is observed that these images are with different textural patterns and local luminance, while all maintain the same vessel structure. It showcases the effect of our diversity loss in the R-sGAN training phase. We believe this diversity loss helps in addressing the mode collapse phenomenon in GANs that is well-known to plague the existing GAN related research [28], [30], where significant changes in noise vector values do not result in noticeable appearance change in the generated images.

B. Quantitative Evaluation of Segmentation Results on the Annotated Datasets

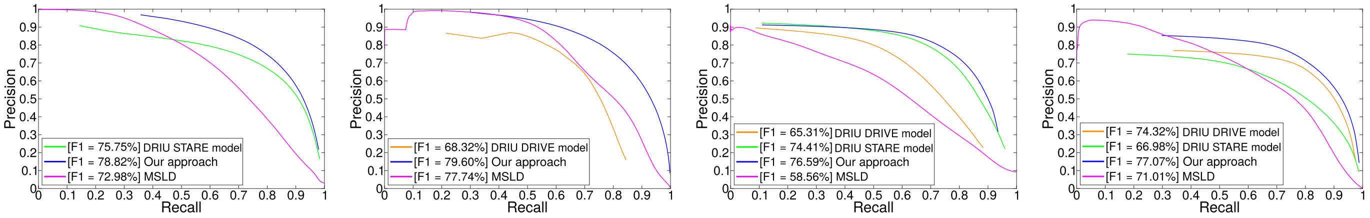
To evaluate the performance gain of utilizing the synthesized target datasets on segmentation task, we consider two distinct deep learning segmentation methods, and one typical supervised learning method. One is direct image-to-image translation, i.e. taken an entire fundus image as input, and predict the whole segmentation map as output. The second one is pixel-wise prediction. A sliding window is used to predict on the central pixel being vessel or not based on the current image patch. The third one is based on feature extraction and classification. More specifically, for image-to-image segmentation methods, the state-of-the-art DRIU [18] is engaged; for pixel-wise segmentation methods, a fully convolutional neural network method is utilized with the basic components being the residual blocks of the ResNet [44], which also achieves competitive performance and is later referred to as Pixel-CNN. The third one is the Kernel Boost method of [14]. For the purpose of a fair comparison, the same training protocol is applied in training these different models on each of the target datasets. It is worth mentioning that as a sanity check, these two segmentation methods have also been trained and tested on the same well-annotated datasets, where their reported performance may serve as an upper bound in our problem context. More concretely, in terms of DRIU [18], its segmentation F1 score on DRIVE and STARE test sets are 81.62% and 82.43%, respectively. For Pixel-CNN, its respective performance drops to 80.33% and 79.02%. For Kernel Boost, its respective performance is 78.10% and 77.30%. Note that throughout the experiments, by default DRIVE is used as the reference or source dataset. The only exception is when the target datasets is DRIVE itself, in this case STARE is used as its source dataset. It is also worth noting that the FOVs of these datasets are relatively similar, and empirically it does not have a major impact in the final results. Further discussion about FOVs could be found in the supplementary file.

Our approach is evaluated over the major fundus benchmarks that come with vessel structure annotations, namely, the DRIVE, STARE, HRF, and IOSTAR datasets. Table I summarizes the quantitative results over an unsupervised as well as several supervised baselines. Here each column provides the segmentation results of a specific model.

TABLE I

Performance Comparison of State-of-the-Art Methods Over a Diverse Range of Target Benchmarks. Segmentation Methods Include the Unsupervised MSLD Method [12] and IUWT Method [9], as Well as the Pixel-CNN, the DRIU [18] Models and Kernel Boost [14] Models Trained on DRIVE or STARE Datasets Respectively, as Well as Our Approach. Results Are Reported in F1-Score (%), Sensitivity (%) and Specificity (%)

		IUWT [9]	*MSLD* [12]	**DRIU**			**Pixel-CNN**			**Kernel Boost**		
				DRIVE model	*STARE model*	*Our approach*	*DRIVE model*	*STARE model*	*Our approach*	*DRIVE model*	*STARE model*	*Our approach*
DRIVE	F1-score	68.81	72.98	/	75.75	78.82	/	69.30	76.02	/	76.09	77.00
	Sensitivity	68.38	66.11	/	78.90	79.01	/	68.57	76.41	/	74.86	74.81
	Specificity	97.12	98.58	/	97.20	97.95	/	97.20	97.66	/	97.92	98.14
STARE	F1-score	73.07	77.74	68.32	/	79.60	71.87	/	76.55	72.73	/	75.36
	Sensitivity	71.93	74.15	67.12	/	79.49	71.29	/	76.31	75.96	/	74.67
	Specificity	97.96	98.63	98.03	/	78.36	97.85	/	98.10	97.34	/	98.10
HRF	F1-score	68.87	58.56	65.31	74.41	76.59	53.90	41.23	58.63	70.63	68.57	71.45
	Sensitivity	66.31	53.50	62.97	73.02	76.08	53.66	46.69	54.71	70.69	66.26	70.37
	Specificity	97.80	97.53	97.57	98.07	98.13	96.07	92.86	97.64	97.54	97.74	97.78
IOSTAR	F1-score	62.70	71.01	74.32	66.98	77.07	66.71	53.13	68.15	65.43	68.61	70.33
	Sensitivity	64.68	68.87	79.55	72.06	79.15	71.13	57.28	68.42	69.36	68.91	69.40
	Specificity	96.67	98.12	97.34	96.57	97.92	96.64	95.27	97.38	96.59	97.44	97.77



(a) (b) (c) (d)

Fig. 6. Precision recall curves of the DRIVE, STARE, HRF, and IOSTAR datasets as the target. By default, the DRIVE dataset is the source dataset, with the exception that when DRIVE is the target dataset, STARE is used instead as the source. DRIU [18] is the supervised segmentation method engaged here in our approach. (a) DRIVE. (b) STARE. (c) HRF. (d) IOSTAR.

Results from the DRIU models are shown in 5th–7th columns, while those of the Pixel-CNN models are presented in 8th–10th columns, and those of the Kernel Boost models are displayed in 11th–13th columns. For example, under the DRIU-tagged columns, DRIVE model refers to the DRIU model trained on the DRIVE training set. Meanwhile, each row displays the results of applying the respective model to a specific target testing set. Besides the supervised model, the widely used unsupervised methods, Multi-scale line detector (MSLD) [12] and IUWT [11] are also considered as reference methods. From Table I, it is observed that MSLD consistently produces comparable results across different datasets. On the other hand, the performance of the supervised models varies significantly when applied to different datasets: it usually outperforms that of MSLD when it works well; when it does not work well, the performance may degrade noticeably and sometimes may be much worse than that of MSLD. Overall, our approach engaged with the DRIU segmentation method of [18] consistently outperforms these supervised and unsupervised methods in comparison. It is also interesting to mention that within the DRIU-tagged columns, our approach achieves better results than other DRIU models that are trained on other datasets, and the same phenomenon can be observed within the Pixel-CNN-tagged columns. Take HRF for example, the DRIU DRIVE model achieves a F1-score of 65.31%, vs. 74.41% from the DRIU STARE model. Our approach using DRIU method obtains the best performance of 77.07%. Statistical significance test (i.e. Wilcoxon signed-rank test) also indicates that our approach outperforms the comparison methods with a statistical significance. Interested readers may refer to Section III of the supplementary file for details.

In what follows, we will focus on the DRIU segmentation method of [18] by default. We also note that same conclusion can be drawn if Pixel-CNN is instead considered. Figure 6 reveals more detailed information by precision-recall curves over each of the datasets as the test set. In most cases, our approach outperforms the rest methods throughout the PR curve plots, often by a noticeable margin, which is in strong agreement to our initial hypothesis that training with such synthesized dataset of target style and source content does help in bridging the gap, thus leads to better segmentation performance. Representative results are also illustrated in Fig. 7, where red color in the resulting error maps indicates the false negative pixels, green color refers to the background pixels wrongly classified as vessel pixels. It can be seen that our approach produces less false negative pixels than both supervised and unsupervised peer methods, which indicates that more vessel pixels can be captured. We believe the reason is that our segmentation model has the advantage of learning the proper set of features from our synthesized target dataset to detect these vessel pixels out of cluttered backgrounds. Meanwhile, there are less green pixels in (c) than in (f), which indicates our approach results in less false alarms. This high precision can also be reflected in Fig. 6 where the precision is the highest under the same recall rate.

C. Qualitative Evaluation of Segmentation Results on Un-Annotated Datasets: Anzhen, Kaggle, and Mobile

It has been demonstrated that our approach improves the segmentation performance on these well-known benchmarks with annotations. Moreover, it is of interest to see how it

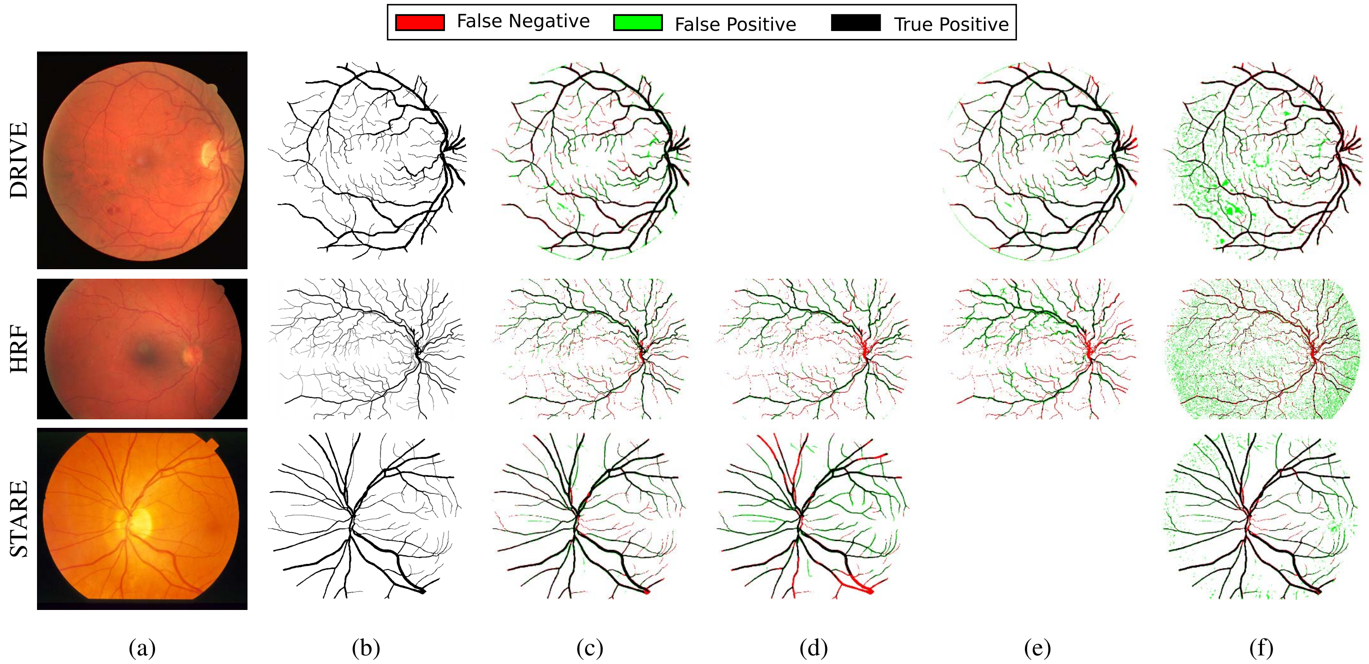


Fig. 7. A visual comparison of segmentation results over different datasets. (a) Input fundus images from various datasets: DRIVE, HRF and STARE; (b) Ground truth vessel structures; (c-f) are the error maps of the comparison methods. Note (c) is our approach with DRIVE being used as source dataset, except for the first row where STARE is used as source instead. (d-f) are the supervised/unsupervised comparison methods trained on DRIVE and STARE datasets, respectively. DRIU [18] is the supervised segmentation method engaged here in our approach.

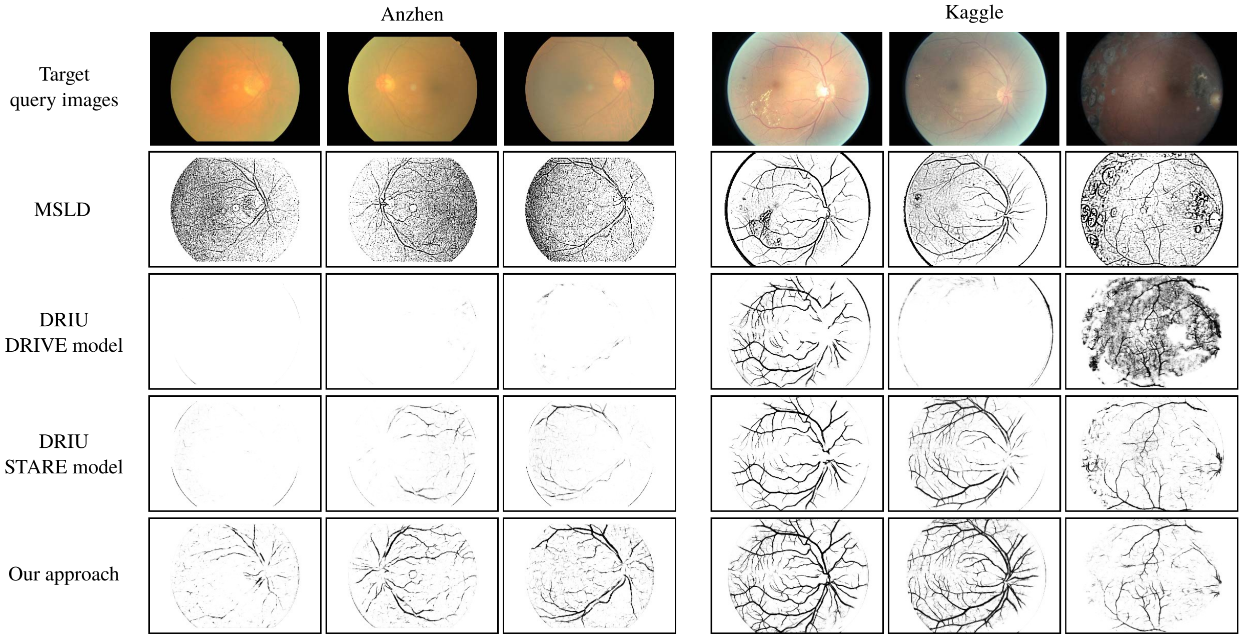


Fig. 8. Qualitative comparisons of our approach vs. direct application of DRIU models and MSLD on Anzhen and Kaggle datasets. First row displays several query Anzhen or Kaggle images. The 2nd to 4th rows present segmentation results of applying the MSLD method, as well as the DRIU DRIVE and STARE models, respectively. The last row displays results from our approach when accessing to a source DRIVE dataset.

performs on new query images without vessel annotations. This motivates us to consider the following three datasets: Anzhen, Kaggle, and Mobile.

In our in-door Anzhen dataset, most images are from cataract patients. Cataract is a common pathological disease among seniors. The retinal images of cataract patient are often blurred and vessels are much less visible when comparing with the normal retinal images. As presented in the left panel of Fig. 8, the unsupervised MSLD method as well as the supervised DRIU DRIVE and STARE models encounter considerable difficulties here in adapting to this context and properly detecting the main vessel trunks or any vessel at all. In contrast, main vessels and many thin and tiny branches can be picked up by our approach. On the other hand for the

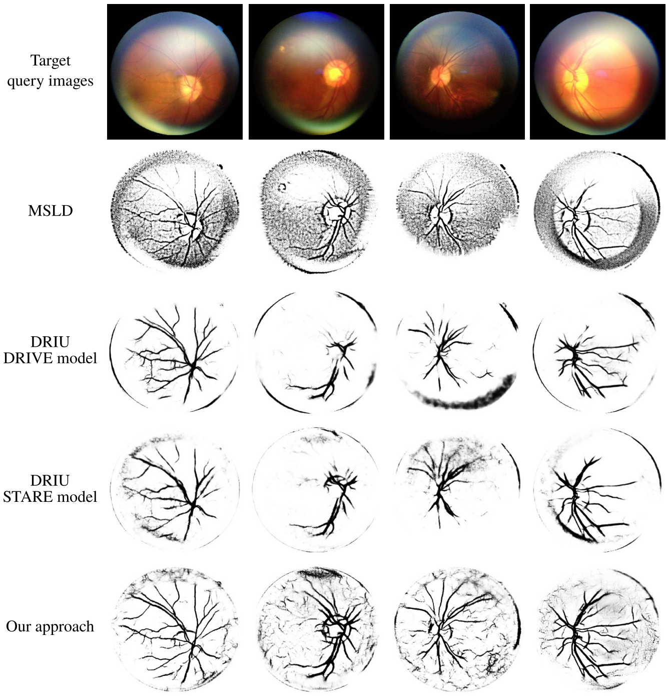


Fig. 9. Qualitative comparisons of our approach vs. direct application of DRIU models and the MSLD method on the Mobile dataset. First row displays several query Mobile images. The 2nd to 4th rows show the segmentation results of applying the MSLD method, as well as the DRIU DRIVE and STARE models, respectively. The last row presents the results of our approach when accessing to a source DRIVE dataset.

Kaggle dataset, there are noticeable variability in contrast and luminance, mostly due to the presence of diabetic retinopathy. Some exemplar images are displayed in the right panel of Fig. 8. Similar to the Anzhen dataset, the DRIU DRIVE model is easily affected by these textural deviations from the typical DRIVE images, which results in poor segmentation results with either almost no vessels or lots of false positive noises. MSLD is also noticeably affected by the pathological patterns with a lot of false alarms. The DRIU STARE model performs better at retaining the main vessel trunks. Finally, our approach produces the most satisfactory results visually: it is capable of retrieving the main parts while tiny vessels can also be better captured from noisy backgrounds.

Retinal fundus images captured by the mobile fundus devices could be even comparable to those from the standard but more expensive fundus cameras. However the imaging quality may deteriorate a lot for a less experienced user. As a result, we often end up with low quality and poorly illuminated fundus images as shown in Fig. 9. These images become challenging for DRIU DRIVE or STARE models, where although some vessel trunks are found, many are left out erroneously as backgrounds. For the unsupervised MSLD method, it is again negatively influenced by wrongly recognizing massive amount of background pixels as part of the foreground tubular structure. Our approach, on the other hand, are much better in picking up the main vessel trunks as well as the detailed branches without introducing much of these false alarms. Note there are some false positives in our results, where most are from the peripheral dark regions, and many can be easily removed in a postprocessing step by applying proper morphological image operators.

VI. Conclusion and Outlook

This paper is to address the challenging problem of reliably segmenting a new set of un-annotated query fundus images with the help of a reference dataset, which is well-annotated but the images are dissimilar to the target query images. It leads us to consider a two-step approach that relies on the construction of a synthetic dataset by a recurrent generative model, R-sGAN, to bridge the dissimilarity gap. Experimental evaluations on a diverse range of fundus image datasets demonstrate the effectiveness of the proposed approach. For future work, we plan to further investigate its application to the downstream clinical goal of ophthalmic diagnostics.

References

[1] C. Kirbas and F. Quek, “A review of vessel extraction techniques and algorithms,” *ACM Comput. Surv.*, vol. 36, no. 2, pp. 81–121, 2004.

[2] K. Jordan, M. Menolotto, N. M. Bolster, I. A. Livingstone, and M. E. Giardini, “A review of feature-based retinal image analysis,” *Expert Rev. Ophthalmol.*, vol. 12, no. 3, pp. 207–220, 2017.

[3] A. D. Hoover, V. Kouznetsova, and M. Goldbaum, “Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response,” *IEEE Trans. Med. Imag.*, vol. 19, no. 3, pp. 203–210, Mar. 2000.

[4] J. Staal, M. D. Abramoff, M. Niemeijer, M. A. Viergever, and B. van Ginneken, “Ridge-based vessel segmentation in color images of the retina,” *IEEE Trans. Med. Imag.*, vol. 23, no. 4, pp. 501–509, Apr. 2004.

[5] (2015). *Diabetic Retinopathy Detection*. [Online]. Available: http://www.kaggle.com/c/diabetic-retinopathy-detection

[6] T. Köhler, A. Budai, M. F. Kraus, J. Odstrčilik, G. Michelson, and J. Hornegger, “Automatic no-reference quality assessment for retinal fundus images using vessel segmentation,” in *Proc. IEEE Int. Symp. Comput.-Based Med. Syst.*, Jun. 2013, pp. 95–100.

[7] X. Xu *et al.*, “Smartphone-based accurate analysis of retinal vasculature towards point-of-care diagnostics,” *Sci. Rep.*, vol. 6, Oct. 2016, Art. no. 34603.

[8] M. M. Fraz *et al.*, “Blood vessel segmentation methodologies in retinal images—A survey,” *Comput. Methods Programs Biomed.*, vol. 108, no. 1, pp. 407–433, 2012.

[9] P. Bankhead, C. N. Scholfield, J. G. McGeown, and T. M. Curtis, “Fast retinal vessel detection and measurement using wavelets and edge location refinement,” *PLoS ONE*, vol. 7, no. 3, pp. e32435-1–e32435-12, 2012.

[10] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, “Multiscale vessel enhancement filtering,” in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 1998, pp. 130–137.

[11] F. Zana and J. C. Klein, “Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation,” *IEEE Trans. Image Process.*, vol. 10, no. 7, pp. 1010–1019, Jul. 2001.

[12] U. T. V. Nguyen, A. Bhuiyan, L. A. F. Park, and K. Ramamohanarao, “An effective retinal blood vessel segmentation method using multi-scale line detection,” *Pattern Recognit.*, vol. 46, no. 3, pp. 703–715, 2013.

[13] J. V. B. Soares, J. J. G. Leandro, R. M. Cesar, H. F. Jelinek, and M. J. Cree, “Retinal vessel segmentation using the 2-D Gabor wavelet and supervised classification,” *IEEE Trans. Med. Imag.*, vol. 25, no. 9, pp. 1214–1222, Sep. 2006.

[14] C. Becker, R. Rigamonti, V. Lepetit, and P. Fua, “Supervised feature learning for curvilinear structure segmentation,” in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2013, pp. 526–533.

[15] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Proc. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.

[16] L. Gu, X. Zhang, H. Zhao, H. Li, and L. Cheng, “Segment 2D and 3D filaments by learning structured and contextual features,” *IEEE Trans. Med. Imag.*, vol. 36, no. 2, pp. 596–606, Feb. 2017.

[17] Q. Li, B. Feng, L. Xie, P. Liang, H. Zhang, and T. Wang, "A cross-modality learning approach for vessel segmentation in retinal images," *IEEE Trans. Med. Imag.*, vol. 35, no. 1, pp. 109–118, Jan. 2016.

[18] K.-K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. Van Gool, "Deep retinal image understanding," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2016, pp. 140–148.

[19] G. Csurka, "A comprehensive survey on domain adaptation for visual applications," in *Domain Adaptation in Computer Vision Applications*. Cham, Switzerland: Springer, 2017, pp. 1–35.

[20] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. ICML*, 2015, pp. 1180–1189.

[21] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. CVPR*, 2017, pp. 2962–2971.

[22] J. Hoffman *et al.* (2017). "CyCADA: Cycle-consistent adversarial domain adaptation." [Online]. Available: https://arxiv.org/abs/1711.03213

[23] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim. (2017). "Image to image translation for domain adaptation." [Online]. Available: https://arxiv.org/abs/1712.00479

[24] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 2672–2680.

[25] M. A. Sagar, D. Bullivant, G. D. Mallinson, and P. Hunter, "A virtual environment and model of the eye for surgical simulation," in *Proc. ACCIT*, 1994, pp. 205–212.

[26] S. Fiorini, M. Biasi, L. Ballerini, E. Trucco, and A. Ruggeri, "Automatic generation of synthetic retinal fundus images," in *Proc. Eurograph Workshop*, 2014, pp. 41–44.

[27] E. Menti, L. Bonald, L. Ballerini, F. Rugger, and E. Trucco, "Automatic generation of synthetic retinal fundus images: Vascular network," in *Proc. Int. Workshop Simulation Synth. Med. Imag.*, 2016, pp. 167–176.

[28] P. Costa *et al.*, "End-to-end adversarial retinal image synthesis," *IEEE Trans. Med. Imag.*, vol. 37, no. 3, pp. 781–791, Mar. 2018.

[29] H. Zhao, H. Li, S. Maurer-Stroh, and L. Cheng, "Synthesizing retinal and neuronal images with generative adversarial nets," *Med. Image Anal.*, vol. 49, pp. 14–26, Jul. 2018.

[30] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. (2016). "Image-to-image translation with conditional adversarial networks." [Online]. Available: https://arxiv.org/abs/1611.07004

[31] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *Proc. SIGGRAPH*, 2001, pp. 327–340.

[32] L. Cheng, S. V. N. Vishwanathan, and X. Zhang, "Consistent image analogies using semi-supervised learning," in *Proc. CVPR*, Jun. 2008, pp. 1–8.

[33] L. A. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," in *Proc. NIPS*, 2015, pp. 262–270.

[34] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. NIPS*, 2015, pp. 577–585.

[35] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. Interspeech*, 2015, pp. 1045–1048.

[36] J. Krause, J. Johnson, R. Krishna, and L. Fei-Fei, "A hierarchical approach for generating descriptive image paragraphs," in *Proc. CVPR*, 2017, pp. 3337–3345.

[37] K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra, "DRAW: A recurrent neural network for image generation," in *Proc. ICML*, 2015, pp. 1462–1471.

[38] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[39] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proc. NIPS Workshop Deep Learn.*, 2014, pp. 1–9.

[40] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. NIPS*, 2015, pp. 802–810.

[41] N. N. Schraudolph, "Local gain adaptation in stochastic gradient descent," in *Proc. ICANN*, Sep. 1999, pp. 569–574.

[42] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[43] S. Abbasi-Sureshjani, I. Smit-Ockeloen, J. Zhang, and B. T. H. Romeny, "Biologically-inspired supervised vasculature segmentation in SLO retinal fundus images," in *Proc. Int. Conf. Image Anal. Recognit.*, 2015, pp. 325–334.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Jun. 2016, pp. 770–778.