End-to-End Adversarial Retinal Image Synthesis

Pedro Costa, Adrian Galdran, Maria Ines Meyer, Meindert Niemeijer, Michael Abràmoff, Ana Maria Mendonça, and Aurélio Campilho

***Abstract*— In medical image analysis applications, the availability of the large amounts of annotated data is becoming increasingly critical. However, annotated medical data is often scarce and costly to obtain. In this paper, we address the problem of synthesizing retinal color images by applying recent techniques based on adversarial learning. In this setting, a generative model is trained to maximize a loss function provided by a second model attempting to classify its output into real or synthetic. In particular, we propose to implement an adversarial autoencoder for the task of retinal vessel network synthesis. We use the generated vessel trees as an intermediate stage for the generation of color retinal images, which is accomplished with a generative adversarial network. Both models require the optimization of almost everywhere differentiable loss functions, which allows us to train them jointly. The resulting model offers an end-to-end retinal image synthesis system capable of generating as many retinal images as the user requires, with their corresponding vessel networks, by sampling from a simple probability distribution that we impose to the associated latent space. We show that the learned latent space contains a well-defined semantic structure, implying that we can perform calculations in the space of retinal images, e.g., smoothly interpolating new data points between two retinal images. Visual and quantitative results demonstrate that the synthesized images are substantially different from those in the training set, while being also anatomically consistent and displaying a reasonable visual quality.**

Manuscript received July 18, 2017; revised September 22, 2017; accepted September 22, 2017. Date of publication October 2, 2017; date of current version March 1, 2018. This work was supported in part by the ERDF European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme, in part by the National Funds through the FCT Fundação para a Ciência e a Tecnologia Portuguese Foundation for Science and Technology under Grant CMUP-ERI/TIC/0028/2014, and in part by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement within the project NanoSTIMA: Macro-to-Nano Human Sensing: Towards Integrated Multimodal Health Monitoring and Analytics under Grant NORTE-01-0145-FEDER-000016. *(Pedro Costa and Adrian Galdran contributed equally to this work.) (Corresponding authors: Pedro Costa; Adrian Galdran.)*

P. Costa, A. Galdran, and M. I. Meyer are with the Institute for Systems and Computer Engineering, Technology and Science, 4200-465 Porto, Portugal (e-mail: pvcosta@inesctec.pt; adrian.galdran@inesctec.pt; maria.i.meyer@inesctec.pt).

M. Niemeijer is with IDx LLC, Iowa City, IA 52246 USA (e-mail: niemeijer@eyediagnosis.net).

M. Abramoff is with the Stephen A. Wynn Institute for Vision Research, University of Iowa, Iowa City, IA 52242 USA (e-mail: michael-abramoff@uiowa.edu).

A. M. Mendonça and A. Campilho are with the Institute for Systems and Computer Engineering, Technology and Science, 4200-465 Porto, Portugal, and also with the Faculdade de Engenharia, Universidade do Porto, 4200-465 Porto, Portugal (e-mail: amendon@fe.up.pt; campilho@fe.up.pt).

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.2017.2759102

***Index Terms*— Retinal image synthesis, retinal image analysis, generative adversarial networks, adversarial autoencoders.**

I. Introduction

THE ability to generate meaningful synthetic information is highly desirable for many computer-aided medical applications, where annotated data is often scarce and costly to obtain. A wide availability of such data may allow researchers to develop and validate more sophisticated computational techniques. This pressing need for annotated data, particularly images, has largely increased with the advent of deep neural networks, which are progressively becoming the standard approach in most machine learning tasks [1]. However, these techniques require large amounts of data to be trained. Therefore, the problem of medical data generation is of great interest, and as such, it has been deeply studied in recent years [2]. Nevertheless, the realistic synthesis of high-quality medical data still remains a widely unsolved challenge.

Most medical image generation methods follow two main strategies. The most conventional approach endeavors to formulate a mathematical model of the observed data. These models can range from simple digital phantoms [3] to more complex methodologies attempting to mimic anatomical and physiological medical knowledge [4]. In combination with the modeling of relevant characteristics of the different acquisition devices, these techniques can generate new high-quality images by sampling an appropriate parameter space. This approach is often referred to as image simulation.

In recent years the data-driven approach of image synthesis has started gaining popularity. In this context, the intrinsic variability within a large pool of training images is extracted by means of machine learning techniques. Ideally, the model is able to learn the underlying probability distribution that defines the manifold of real images. Once trained, the same system can be sampled to output new images that are likely to lie on that manifold, *i.e.* realistic synthetic images. This approach has recently been successfully applied to improve classification of multi-sequence MRI with missing/corrupted sequences [5], to estimate cross-modality transformations [6], or to perform knowledge transfer by learning features invariant to the MR scanning protocol [7].

In the retinal image analysis field, in [8] the authors propose an algorithm for the generation of the retinal background and the fovea, and a separate technique for the generation of the optical disk. For the former, the method relies on the construction of a large dictionary of small vessel-free image patches. These patches are extracted from a dataset of co-registered real images and clustered together,

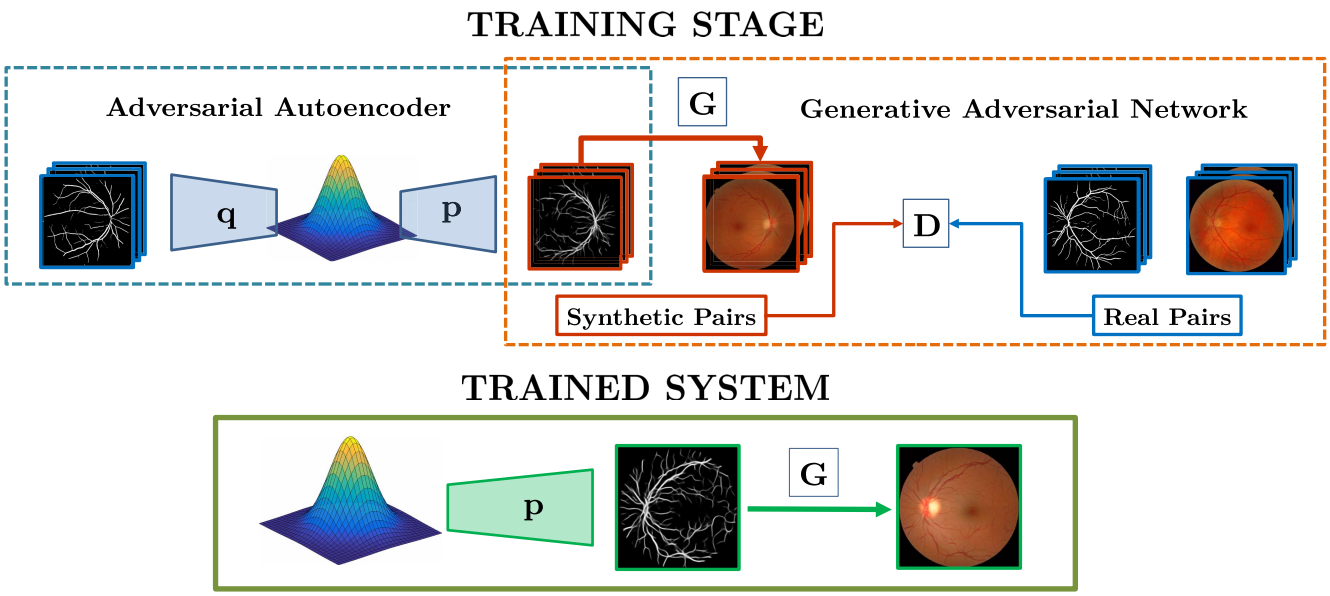

Fig. 1. Overview of our approach. The pair (p, q) is an adversarial autoencoder trained to reconstruct retinal vessel maps. The pair (G, D) is a Generative Adversarial Network trained to generate color retinal images out of vessel maps. Once the model is trained, the system can generate a new retinal image and an associated vessel map. The only required input is sampling a distribution p, which is enforced to follow a simple multi-dimensional Gaussian distribution during training by means of an adversarial loss.

before tiling them in a consistent manner. For the latter, a parametric intensity model is proposed, with the parameters being estimated over a dataset of real images.

The work in [2] is complementary to [8], since it focuses on the generation of the vascular network only. The authors propose a method to generate realistic retinal vessel trees. The parameters controlling the geometry are learned from real vessel trees. The method also enforces meaningful vessel orientation and calibers by following a physical bifurcation law describing the correct oxygenation of the retinal surface [9]. The output of both approaches can then be superimposed, allowing for the generation of high-quality large-resolution images. However, concatenation of both techniques results in a considerably complex computational pipeline, relying on sensitive sub-processes such as image registration, patch-to-image stitching or image blending.

Recently, a purely data-driven approach has been proposed in [10]. It consists of a simple application of adversarial learning methods [11], in which a model is trained on pairs of real vessel networks and their corresponding retinal fundus images. The goal is to learn a transformation between them, and once trained, this technique can generate a plausible retinal image out of a pre-existing binary vessel tree. Unfortunately, this approach has been shown to have a relevant drawback: the model is dependent on the availability of a pre-existing vessel network in order to generate a new retinal image. The vessel networks employed for generating images were obtained by application of an independent vessel segmentation method to real retinal images. If the original image is defocused, the retrieved vessel tree will be undercomplete, and the obtained synthetic image will contain visual artifacts [10].

In this work, we substantially improve upon [10] by removing the dependence of the model on the previous existence of a retinal vessel tree. This is achieved by building an autoencoder that can learn to generate realistic retinal vessel trees. Moreover, by minimizing an adversarial loss, the autoencoder allows to generate vessel networks by simply sampling a multi-dimensional Normal distribution. A schematic representation of our approach is depicted in Fig. 1.

It is worth noting that it is theoretically possible to perform a separate training of the retinal vessel synthesis module and the vessel network to retinal image mapping. However, since both tasks are closely related, it is more natural to train both systems jointly. We achieve this by combining the loss functions associated to each task in a more general framework. The resulting method presents several advantages over previously proposed approaches:

1) The adversarial learning framework allows us to model the underlying distribution of plausible retinal images only from training data, without manually interacting with parameters controlling complex mathematical models of the retinal anatomy.
2) Once trained, the model improves upon [10] by allowing to generate any amount of realistic retinal images, with associated vessel trees, in an efficient manner.
3) Unlike [2], [8], we generate separate parts of the retinal anatomy through the same process, avoiding the combination of complex image processing tasks.

The proposed framework provides an effective end-to-end retinal image synthesis tool, capable of producing realistic eye fundus images and associated vessel networks with a simple sampling procedure. We provide objective evaluation of both the quality and the applicability of our synthetic images. Even if the generated images and associated vessel maps are of low resolution, suffer from small inconsistencies, and may still not be used to train more complex retinal image analysis

algorithms, we show them to be useful for learning a retinal vessel segmentation model with reasonable performance. This represents a promising first step towards achieving synthetic data that can be used in more complex automatic retinal image analysis applications.

II. Adversarial Image Generation

A. Vessel Network to Retinal Image Translation

The research herein reported considers retinal color image generation out of an existing vessel network as an image-to-image translation problem, learning a mapping G from a binary vessel map v into another representation r [12]. Since many retinal images could share a similar binary vessel network due to variations in color, texture, illumination, etc., in our case G is a multi-valued mapping $G : v \rightarrow \{r_1, \ldots, r_m\}$. As such, learning G is an ill-posed problem and some uncertainty is present.

Connected to this is the choice of the objective function to be minimized while learning G. Training a model to minimize the L_2 distance between $G(v_i)$ and r_i for a collection of training pairs given by $\{(r_1, v_1), \ldots, (r_n, v_n)\}$ will produce low-quality results with lack of detail [13], due to the model selecting an average of many potentially valid representations.

Recent ideas based on Generative Adversarial Networks (GANs) [11] are able to overcome this problem by learning a more suitable loss function directly from data [12]. The underlying strategy of adversarial methods consists of emulating a competition, in which the mapping G, called Generator, attempts to produce realistic images, while a second player, the Discriminator D, is trained to distinguish the output generated by G from real examples. Here, both G and D are neural networks, and act as adversaries, since the goal of G is to maximize the misclassification error of D, while D's objective is to beat G by learning to identify generated images. As in [11], the adversarial loss, driving the learning of G and D, is:

$$\mathcal{L}_{adv}(G, D) = \mathbb{E}_{v,r\sim p_{data}(v,r)}[log(D(v, r))] + \mathbb{E}_{v\sim p_{data}(v)}[log(1 - D(v, G(v)))], \quad (1)$$

where $\mathbb{E}_{v,r\sim p_{data}(v,r)}$ is the expectation over the pairs (v, r), sampled from the joint data distribution of real pairs $p_{data}(v, r)$ and $p_{data}(v)$ is the real vessel trees distribution. The Discriminator's objective is to maximize (1), while the Generator's goal is to minimize it. Therefore, it is D that provides the training signal to G, replacing more conventional loss functions.

Although minimizing the above loss function induces G to produce visually sharp results, recent work in [12] and [14] has shown that combining Eq. (1) with a global L_1 loss provides more consistent results. Thus, the loss function to optimize becomes:

$$\mathcal{L}_{im2im}(G, D) = \mathcal{L}_{adv}(G, D) + \lambda\mathbb{E}_{v,r\sim p_{data}(v,r)}[||r - G(v)||_1], \quad (2)$$

where λ balances the contribution of the two losses. The discriminator's objective is in this case local, *i.e.*, it attempts to discriminate $N \times N$ image regions as real or generated, but the goal of G is supplemented with a requirement not only to generate realistically looking images but also images that preserve a global regularity. Since the L_1 loss guarantees that the output of G is globally consistent, D can concentrate on modeling only high frequency structures. Thus, while D penalizes locally over-smooth image regions, the L_1 loss promotes the consistency of global visual features, such as the presence of a single optical disk and macula in the image. An overview of this model is shown in Figure 2.

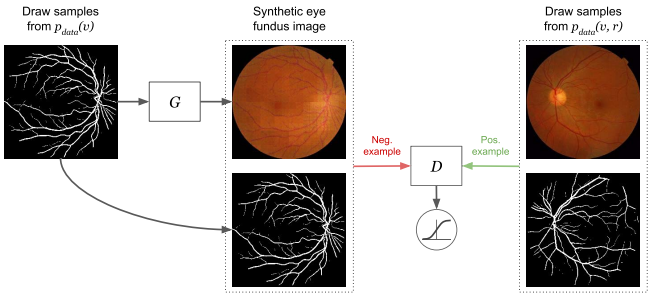

Fig. 2. The discriminator D learns to distinguish between real pairs of vessel networks and eye fundus images (v, r) and synthetic pairs. The generator G maps an input vessel network v to a color eye fundus image r.

B. Adversarial Autoencoders for Vessel Trees Generation

Ideally, an end-to-end retinal image synthesis system should also generate realistic vessel networks. Such a model would also learn from data and generate as many vessel networks as the user requires, with a high degree of variability, while remaining anatomically plausible. In this work, we propose to achieve this goal by means of an adversarial autoencoder.

Autoencoders are models trained to reconstruct their input. They are composed of two submodels: 1) an encoder Q, that maps a training example v to a latent (hidden) representation $z = Q(v)$, and 2) a decoder P, mapping z to an output that aims to be a replica of the input. An autoencoder can thus be trained on a training set of vessel trees v, in order to minimize a reconstruction objective $\mathcal{L}_{rec}(Q, P)$.

Modern autoencoders feature deep neural networks both for the encoder and the decoder, and introduce stochasticity by considering probability distributions instead of deterministic mappings Q, P. Here we define both the decoder and the autoencoder to be conditional probability distributions, $q(z|v)$ and $p(v|z)$.

Autoencoders can be employed to learn useful abstractions of the data through their latent representations. These can then be applied in other contexts, *e.g.* data compression or semi-supervised learning. However, in the above form, the trivial mapping that associates each vessel tree example v in the training set to itself can succeed in minimizing the reconstruction loss while failing to learn any valuable abstraction. To avoid this, several types of regularization can be added to the loss, *e.g.* minimizing $\mathcal{L}_{rec}(q, p)$ while requiring the latent representation to be sparse [15].

However, even when properly regularized, an autoencoder still has no ability to fulfill the goal of generating new

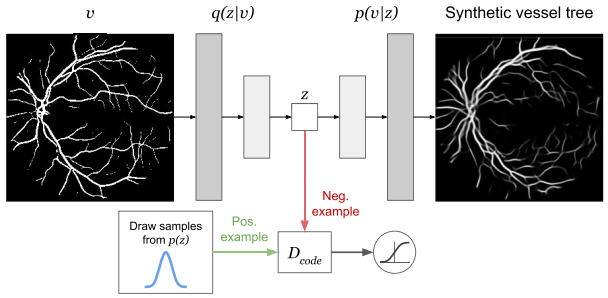

Fig. 3. At first, the discriminator D_{code} is trained to distinguish between samples from the given prior $p(z)$ and latent representations of training vessel networks v from the encoder $q(z|v)$. Then, the autoencoder is trained to minimize the reconstruction loss between its output and v and, at the same time, maximize the misclassification of D_{code}.

elements close to the true data manifold, since we do not have knowledge of the underlying probability distribution $q(z)$ governing the space of latent representations. This prevents us from sampling it in order to obtain a new code z that can then be mapped by p to a retinal image.

To achieve the twofold goal of turning the autoencoder into a generative model while regularizing it in such a way that it can learn interesting representations of retinal vessel trees, we apply the adversarial autoencoder framework, proposed in [16]. In this case, the autoencoder learning process is embedded in an adversarial competition, similar to the one described in the previous section. The goal of the autoencoder is to minimize the reconstruction error, but at the same time, we attempt to gain control on the probabilistic structure of $q(z)$ by matching it to a prior distribution $p(z)$ that can be easily sampled (*e.g.* a multi-dimensional unit normal distribution). The encoding distribution $q(z|v)$ in the autoencoder is the generator component of the adversarial game. This consists of a neural network enforced to produce latent representations z following the pre-specified prior distribution $p(z)$. This is achieved via the maximization of the classification error of the discriminator module D_{code}, which is trained to classify codes z sampled from $q(z)$ according to whether they come from the true prior distribution $p(z)$ or not. Figure 3 depicts a schematic representation of this process.

The autoencoder training is performed by gradient descent, with the gradients computed by standard backpropagation. The optimization process consists of two alternate stages. In the first step, the discriminator is updated to distinguish samples generated by q from those coming from the prior distribution $p(z)$. This is achieved by maximizing the following loss:

$$\mathcal{L}_{code}(D_{code}, q) = \mathbb{E}_{z\sim p(z)}[log(D_{code}(z))] + \mathbb{E}_{v\sim p_{data}(v)}[log(1 - D_{code}(q(z|v)))]. \quad (3)$$

In addition, both the encoder and the decoder weights are updated to minimize the reconstruction error and, at the same time, to maximize the classification error of the discriminator. In this way, the complete loss function that drives the learning of the adversarial autoencoder is a combination of both losses:

$$\mathcal{L}_{AAE}(D_{code}, q, p) = \mathcal{L}_{code}(D_{code}, q) + \gamma\, \mathcal{L}_{rec}(q, p), \quad (4)$$

where γ weights the importance of the two losses. The goal of q and p is to minimize $\mathcal{L}_{AAE}$, while D_{code} attempts to maximize it. When the optimization process reaches an equilibrium point of Eq. (4), the decoder p defines a generative model than can be employed to generate new vessel trees starting from a sample of the imposed prior $p(z)$ on the latent distribution.

C. From Random Samples to Retinal Images

The vessel-to-retinal image model presented in section II-A can map a vessel tree v to a realistic eye fundus image r, while the adversarial autoencoder defined in the previous section generates a vessel network v from a random sample z coming from a simple probability distribution. When both models are combined, we obtain a single system capable of generating a vessel map and a retinal image r from a random sample z.

However, both sub-tasks are deeply interconnected. The generation of vessel networks of better quality will lead to a more realistic retinal image r. Conversely, if the generated image r is able to deceive the discriminator in such a way that it classifies it as plausible, it means that the vessel network v contained in it also needs to be plausible.

Following this argument, we build a single joint model, in which both sub-systems are trained at the same time, instead of independently. In our case, the loss functions defining both models are differentiable almost everywhere. Accordingly, to build a joint loss function we can directly combine them by simple addition. Nonetheless, we need to redefine the image-to-image losses in Eqs. (1) and (2), so that they take the output of the adversarial autoencoder as the input to G:

$$\widetilde{\mathcal{L}}_{adv}(G, D) = \mathbb{E}_{v,r\sim p_{data}(v,r)}[log(D(v, r))] + \mathbb{E}_{v\sim p_{data}(v)}[log(1 - D(\tilde{v}, G(\tilde{v})))], \quad (5)$$

$$\widetilde{\mathcal{L}}_{im2im}(G, D) = \widetilde{\mathcal{L}}_{adv}(G, D) + \lambda \mathbb{E}_{v,r\sim p_{data}(v,r)}[||r - G(\tilde{v})||_1], \quad (6)$$

where $\tilde{v} = p(q(v))$ is the vessel tree generated by the adversarial autoencoder. With this modification, both loss functions can be linearly combined into a global one:

$$\mathcal{L}(G, D, D_{code}, q, p) = \widetilde{\mathcal{L}}_{im2im}(G, D) + \mathcal{L}_{AAE}(D_{code}, q, p). \quad (7)$$

In this formulation, the goal of G, q and p is to minimize the loss function in Eq. (7), while D and D_{code} attempt to maximize it. The main advantage of this joint training scheme is that the discriminator D also provides with a better loss function for the adversarial autoencoder. The decoder p needs to produce realistic looking vessels in order to maximize the misclassification of D. Also, part of the training signal that arrives to p flows through G. As a consequence, the adversarial autoencoder also benefits when the generator produces realistic eye fundus images. A schematic representation of the whole model is shown in Figure 4.

D. Understanding the Latent Space

After training the model as described above, it is possible to sample from $p(z)$ in order to produce a synthetic pair of

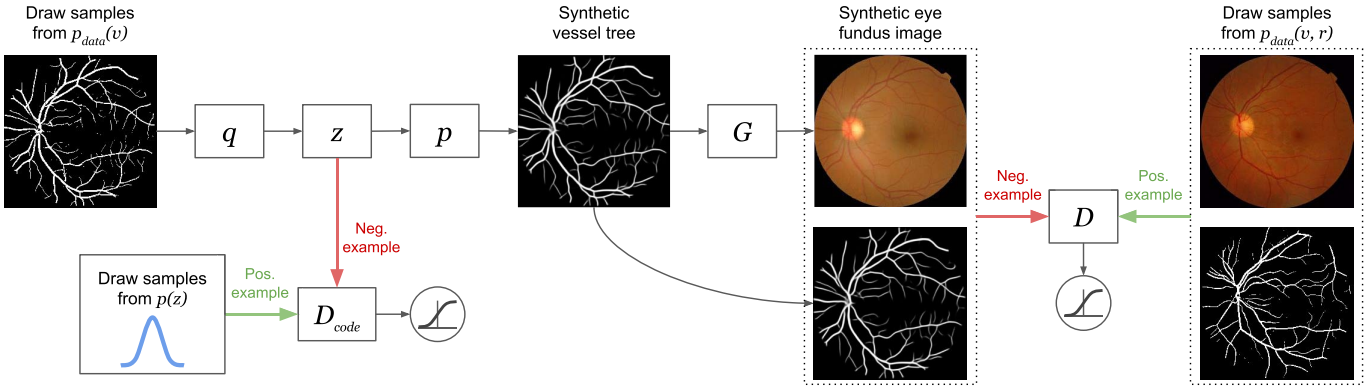

Fig. 4. The model consists of an adversarial autoencoder followed by a conditional Generative Adversarial Network. The adversarial autoencoder and the conditional GAN are trained to minimize the distance between their output and the training pair $\triangleright v, r \triangleleft$ and, at the same time, maximize the misclassification of D and D_{code}. Simultaneously, D learns to distinguish between real pairs (v, r) and synthetic pairs and D_{code} learns to distinguish between latent representations produced by the encoder q and samples from the given prior $p(z)$.

vessel network and eye fundus images. Nonetheless, the latent space might contain zones that are not on the manifold learned during training. This implies that points sampled from $p(z)$ that are far from the latent representations of all the training examples might produce pairs that are not plausible (*e.g.* an eye fundus image with two optical disks).

Fortunately, there are techniques that allow to sample from generative models in order to avoid these cases. For instance, given two real vessel network images v_1, v_n, we may apply the encoder q to obtain their latent representations z_1, z_n, and interpolate between these two known locations in the latent space to obtain a smooth transition between two images, $\{z_2, \ldots, z_{n-1}\}$. If the model did not overfit the training data, the vessel trees obtained when decoding these intermediate representations, *i.e.*, $\{q(z_2), \ldots, q(z_{n-1})\}$, will be plausible vessel networks that are not present on the set of real vessel networks in which the model was trained with.

To find a correct path linking z_1 to z_n, typically, linear interpolation is applied. However, this is not recommendable when a Gaussian prior is used [17], as is our case. Linearly interpolated latent representations traverse points that are indeed unlikely given this prior. Instead, it has been shown that the application of a spherical interpolation (*slerp*) [17] produces better results. This is defined by the following equation:

$$slerp(z_1, z_n, t) = \frac{sin((1-t)\theta)}{sin(\theta)} z_1 + \frac{sin(t\theta)}{sin(\theta)} z_n, \qquad (8)$$

where θ is the angle between z_1 and z_n and t is a value ranging from 0 to 1. For $t = 0$ the result of *slerp* is z_1, whereas for $t = 1$, it takes the value of z_n. On every intermediate value, the *slerp* interpolation outputs a point in a great arc from a sphere containing z_1 and z_n as shown in Fig. 5.

It is also well known that the latent space learned by an autoencoder contains a semantic structure, which implies that it allows us to perform meaningful vector space arithmetic. As an example, in this vector space we are able to solve visual analogies [18]. An analogy is defined as a 4-tuple:

$$z_1 : z_2 :: z_3 : z_4, \qquad (9)$$

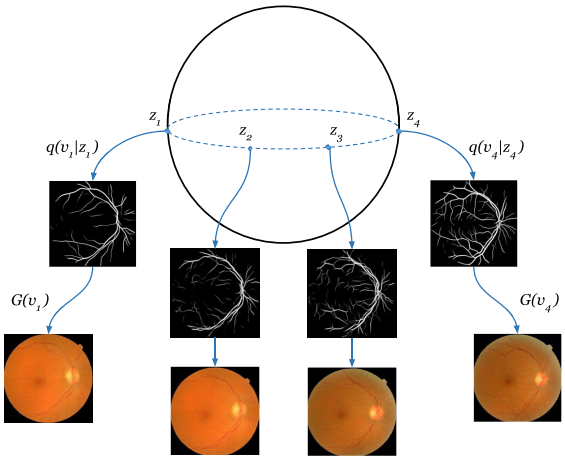

Fig. 5. An example of a spherical interpolation between two points z_1 and z_4 from the latent space.

which symbolizes that the relationship between z_1 and z_2 is the same as the relationship between z_3 and z_4.

For instance, we can analyze the result of applying the same transformation between z_1 and z_2 to z_3, which would be written in analogies terminology as $z_1 : z_2 :: z_3 : ?$. If the points z_i lie in a space supporting vector arithmetic, this analogy can be resolved by vector addition, simply computing:

$$z_4 = z_1 - z_2 + z_3. \qquad (10)$$

For instance, given two images encoded by the latent factors z_1, z_2, we can compute a transformation mapping one image to the other by simply obtaining the vector given by $\overrightarrow{z_1 z_2}$. After this, we can apply that same transformation to a third image by encoding it into a latent representation z_3, and computing $z_4 = z_3 + \overrightarrow{z_1 z_2}$.

In the case of the retinal images synthesized by our model, the latent space is embedded in an N–dimensional vector space, where N is a hyperparameter of the model. This provides a finer degree of control on the high-level properties of the generated images. Applying the above technique, we can isolate factors of variation in the associated space of vessel

TABLE I
ARCHITECTURE OF ENCODER AND DECODER NETWORKS

Encoder	D64	C64	D128	C128	D256	C256	D512	C512	D512	C512	D512	C512	D512	C512	D512	F32 μ F32 σ
Decoder	F512	U512	U512	U512	U512	U256	C256	U128	C128	U64	C64	U64	C64	C1		

trees defined by $p_{data}(v)$. In this case, we gain control on global visual properties such as the position of the optical disk or the amount of vessels. Visual examples of these concepts are demonstrated in the Evaluation section below.

E. Implementation and Training

To be trained, the proposed model requires a dataset of vessel trees and associated eye fundus image pairs. In order to have enough training data, automatic retinal vessel segmentations of the Messidor-1 dataset [19] were used. As this dataset does not include manual segmentations, the vessel tree was extracted using a U-Net model trained on the DRIVE dataset [20]. This model achieved a 0.9755 AUC on the DRIVE test set, a result aligned with state-of-the-art methods for retinal vessel segmentation [21]–[24]. Further details about the implementation are described in [10]. Then, the model as trained on DRIVE was used to segment images from the Messidor-1 dataset [19]. The obtained segmentations were validated visually by the fifth author (MDA), who is a fellowship-trained retinal specialist. This strategy allowed us to use a larger set of training examples, since the Messidor-1 dataset contains 1200 images, while DRIVE contains only 40. In addition, employing Messidor-1 images increases the variability of the data in training time, since they contain more diverse colors and texture.

However, Messidor-1 also contains images with different grades of diabetic retinopathy (grade 0 to 4), while in DRIVE only 7 images display signs of mild diabetic retinopahty (grade 1). This led to a poor generalization of our vessel segmentation technique, which produced incorrect segmentations for images in a later stage of diabetic retinopathy. For this reason, only images from Messidor-1 with grades 0, 1 and 2 were used in this work, reducing the number of example pairs to 946. This dataset was randomly divided into training (614 pairs), validation (155 pairs) and test (177 pairs) sets, which were downscaled to 256 × 256 before training the model.

For the encoder network q, our model assumes that the posterior probability of the encoder $q(z|v)$ follows a normal distribution on a N-dimensional space, with mean $\mu(v)$ and standard deviation $\sigma(v)$, *i.e.* $z \sim \mathcal{N}(\mu(v), \sigma(v))$. The dimension N of the latent space was set to 32. For back-propagating the gradient through the encoder network, the re-parameterization trick proposed by Kingma and Welling [25] was used.

Table I shows the architectures for both the encoder and decoder, where Cn stands for a 3 × 3 Convolutional layer with n filters followed by a Batch-Normalization layer [26], Dn is the same but the Convolutional layer is applied with stride 2 to downsample the activation map, Un doubles the size of the activation map before applying a Cn block, and Fn is a Fully-Connected layer with n units. All blocks are followed by a Leaky ReLU [27] activation function except for the last layers. The outputs of the encoder $(\mu(v), \sigma(v))$ are followed by a linear activation function while the output of the decoder is followed by a sigmoid activation function. Finally, D_{code} is a F64 followed by a F1 with a sigmoid as the activation function.

The discriminator D in Eq. (2) classifies 16 × 16 patches of 31 × 31 pixels and the generator G in Eq. (2) is a U-Net. Both D and G have the same architecture as in the work of Isola *et al.*, see [12] for further details. After a hyper-parameter search, both α and γ in Eqs. (2) and (4) were set to 100. The model was implemented with Keras [28].[1]

Regarding the training process, we monitored D's loss on the validation set and stopped training when the loss stopped increasing. The accuracy achieved by the Discriminator at distinguishing real and synthetic pairs was approximately 0.5. After training, the generator took approximately 17.35s. to generate a set of 100 images.

III. EXPERIMENTAL EVALUATION

The proposed technique can be employed to generate as many synthetic retinal images as the user requires. To simplify the evaluation of our results, we generated a fixed dataset containing the same amount of image pairs (vessel network/retinal image) as in our initial training dataset (614 pairs), and performed experimental qualitative and quantitative comparisons on them. This dataset will be denoted as Synthetic Dataset (*SD*). For quality comparison with real image pairs, we considered two different datasets: 1) The set of images used during training, containing 614 real retinal images and corresponding vessel trees extracted from the Messidor-1 database, denoted Training Real Dataset (*TrainRD*); 2) The held-out test set, that was not used during training, and contains 177 real retinal images and associated vessel trees, denoted Test Real Dataset (*TestRD*).

As the model outputs pairs of images with 256 × 256 resolution, every real image was downscaled to the same size in order to perform a meaningful comparison. Although these resolutions are lower than those of currently acquired retinal images, Gulshan *et al.* [29] showed that it is possible to obtain state-of-the-art results in diabetic retinopathy classification with eye fundus images of size 299 × 299, which is close to the output resolution of our model.

[1] **Code to reproduce our experiments available at https://github.com/costapt/adversarial_retinal_synthesis.**

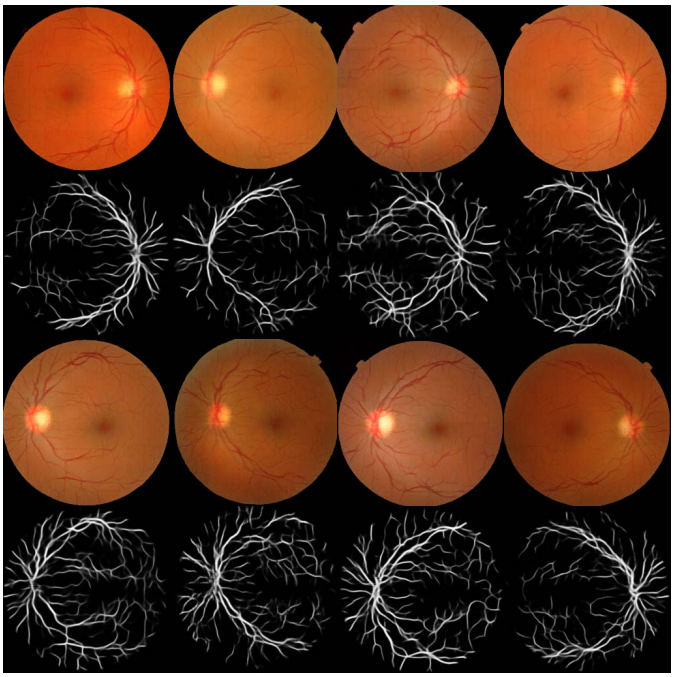

Fig. 6. Random samples of eye fundus images and corresponding vessel networks generated by our model.

A. Subjective Visual Quality Evaluation

For visual evaluation, in Figure 6 we show some of our synthetically generated retinal images, together with the associated vessel networks. The global consistency of the image is correct, since the model learned to introduce visual content only in the circular Field of View. Also, the optical disk and the macula appear correctly located. The vessel network[2] also shows high plausibility, with the two main arcades displaying a thicker caliber than the rest of the vascular segments. It is worth noting that our model also learned to insert the optical disk at the confluence of these arcades. Likewise, color and illumination were generated in a consistent manner.

In the case of machine learning-based generative methods, it is useful to verify that the model has not simply memorized the training data. This can be accomplished by analyzing the distance between the real images used for training and the synthetic ones. If the method did not memorize, it is expected that synthetic images will display visual differences with respect to the training set.

We proceed by extracting a synthetic vessel network $\tilde{v}$ from *SD* and finding the vessel network v in *TrainRD* that is closest to $\tilde{v}$. To perform this matching, we apply the Mutual Information (MI) measure, widely used in medical image registration [30]. This metric allows us to quantify the amount of information overlap between v and $\tilde{v}$ by computing the following:

$$MI(v,\tilde{v}) = \sum_{v_i \in v} \sum_{\tilde{v}_i \in \tilde{v}} p(v_i,\tilde{v}_i) log\Big(\frac{p(v_i,\tilde{v}_i)}{p(v_i)p(\tilde{v}_i)}\Big), \qquad (11)$$

[2]Note that the synthetic vessel trees contain continuous values in [0,1], due to our model minimizing the cross-entropy loss. These vessel networks can be considered as probability maps, and thresholded appropriately if a binary vessel network is needed for some further application.

TABLE II
ISC QUALITY MEASURE ON REAL/SYNTHETIC IMAGES

	Mean ISC score	Std. dev.
Real Images	0.9832	0.1117
Synthetic Images	0.9671	0.0307

where $v_i, \tilde{v}_i$ are pixel intensities in the vessel tree images. Finally, we visually compare the real retinal image associated to v and the synthetic image related to $\tilde{v}$. An example of this experiment is shown in Fig. 7, where it can be appreciated that the generated retinal images, although sharing a similar vessel network, were markedly different in terms of global appearance. This verifies the assumption that our model generalizes properly and did not trivially memorize the examples in the training set.

B. Quantitative Quality Evaluation

Objectively verifying the degree of realism of synthetically generated images is known to be a challenging task when no reference is available [31]. In the case of generative models, it is also well known that a specific quality measure should be used for each application [32]. Accordingly, to report a quantitative image quality analysis, we employ the Image Structure Clustering (ISC) metric proposed in [33]. This is a no-reference quality metric that is trained on an independent dataset of retinal images, previously annotated by retinal specialists who indicated whether the quality of the images was good enough to evaluate the image for the presence of diabetic retinopathy.

The ISC score estimates if there is a correct proportion of pixel intensities corresponding to the relevant retinal anatomical structures, *i.e.* the vessel tree, the optical disk, the macula and the background. It achieves this goal by decomposing a retinal image, assigning each pixel to one out of 5 clusters in the space of responses to Gaussian derivatives of several orders. Responses are aggregated into histograms, and a classifier is trained on these histograms' counts in order to decide if a retinal image contains a reasonable visible proportion of such structures, under the assumption that the lack of presence of one of these clusters is an indicator of low quality, see [33] for the technical details.

The ISC score was computed on the retinal images from the entire *SD* and on *TrainRD*, which contained the same amount of images. Both quality score distributions were normal according to the Kolmogorov-Smirnov test. The resulting data was therefore expressed as mean $\pm$ standard deviation, and compared with the unpaired Student's t-test. All p-values were two-tailed and $p < 0.05$ was considered significant. Statistical analyses were performed using GraphPad Prism 7 (Graphpad Software Inc.) software. The results are reported in Table II, and show that, even if the synthetic images obtained a statistically significantly lower quality score ($p < 0.00063$), a large fraction of the image quality present in the training dataset was preserved while generating new retinal images.

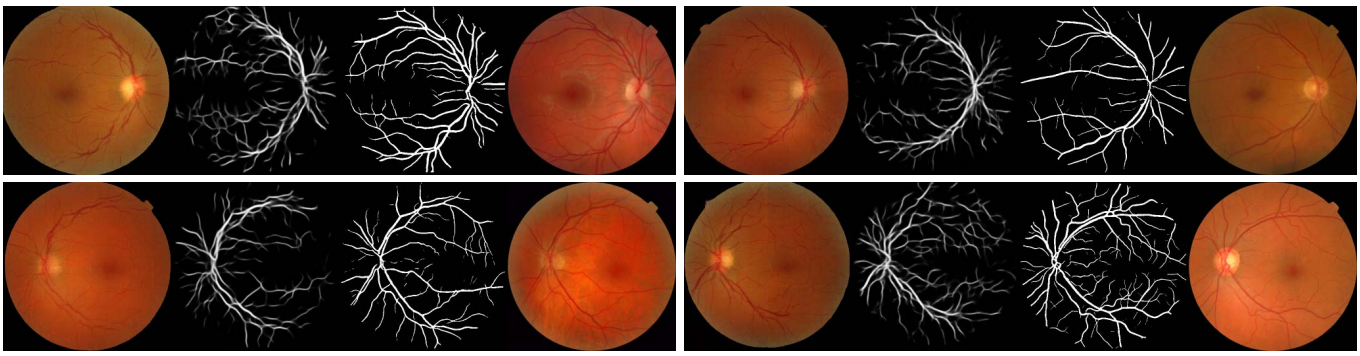

Fig. 7. Comparison of randomly selected synthetic images with the training pair that is closest with respect to the vessel network. For each set of images, from the leftmost column to the last: synthetic eye fundus image; corresponding synthetic vessel network $\tilde{v}$; training vessel network closest to $\tilde{v}$ with respect to the Mutual Information distance; and corresponding training eye fundus image. The generated vessel networks are clearly different from the closest ones on the training set, indicating that the model did not simply memorize the training examples.

C. Retinal Vessel Segmentation Using Synthetic Training Data

One of the main motivations for the present work is the growing need for annotated data in the automated medical image analysis area. The technique introduced in this paper provides pairs of synthetic images and corresponding vessel trees. It is thereby meaningful to evaluate if the generated images could potentially be applied for segmenting the vessel tree from eye fundus images.

To verify how the synthetically generated data performs in this task, the segmentation model described in section II-E was trained first using real data, and afterwards using only synthetic data for performance comparison. The considered dataset was the publicly available DRIVE dataset [20]. For a fair comparison, and due to the low resolution of the produced synthetic images, DRIVE images were downsampled to a resolution of 256×256.

We trained both on real and synthetic images. In the experiment with real images, the remaining 20 images in DRIVE were used to train. In the experiment with synthetic images, 20 images and corresponding vessel maps were extracted from *SD*. The selected pairs where those achieving highest ISC scores. In both cases, 20 images were used for testing the model. Since the training process of this model is not deterministic due to the stochastic gradient descent, for a robust evaluation in both cases 11 different models were trained separately, and the resulting performance figures were averaged. The resulting average ROC curves, built from varying the decision threshold, are displayed in Fig. 8. The models trained with real images obtained an average AUC of 0.887 ± 0.004, while when using only synthetic images, the average AUC was 0.841 ± 0.009.

The results from these experiments are encouraging. The performance of the vessel segmentation model when trained with synthetic images is well above a baseline random model, and when allowed a fraction of false positives approximately greater than 0.35, the resulting system shows greater sensitivity than the same segmentation model trained with real images. However, results should be interpreted with caution. First, the obtained vessel segmentation performances are well below state-of-the-art results on this dataset, something probably due to the considered reduced resolution. Second, we observed a decrease in performance when combining synthetic and real images for training. This seems to hint at a possible lack of quality in the resulting images, especially the vessel maps, which sometimes show certain inconsistencies, *e.g.* as vessel interruptions.

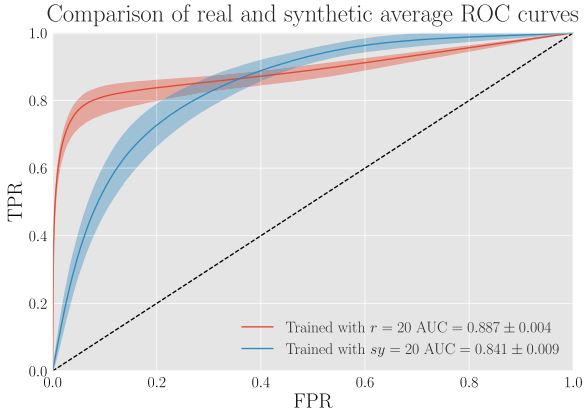

Fig. 8. ROC curves for models trained using 20 synthetic (*sy*) images (blue) and 20 real (*r*) images (red). FPR refers to False Positive Rate and TPR to True Positive Rate, where a False Positive is the incorrect declaration of a background pixel as vessel pixel, and a True Positive represents the correct declaration of a vessel pixel as belonging to a vessel. Bands represent 3 $\times$ standard deviation from the mean performance at a given decision threshold.

D. Exploring the Latent Space

Generative models produce latent spaces that allow us to better understand the underlying structure of our training data, as well as how good our model is in generalizing to new inputs. In this section, we provide an exploratory analysis of the latent space structure our model learned, in terms of the techniques presented in Section II-D. Every real image employed in this section was extracted from the *TestRD* set, which was not used during training.

An example of interpolation between two points in the latent space is shown in Fig. 9. In this case, we selected two vessel tree images v_1 and v_2 from *TestRD*, corresponding to images with the optical disk on the left and on the

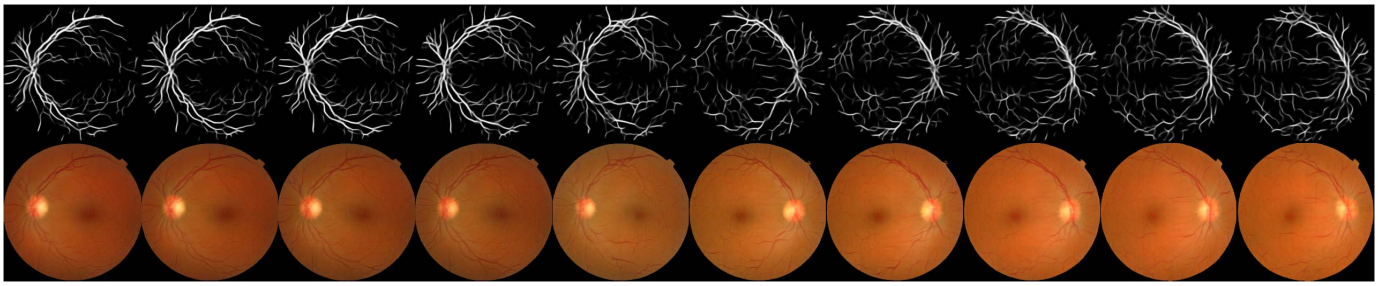

Fig. 9. Interpolation between the latent representation of a vessel tree with the optical disk on the left and another with the optical disk on the right.

right respectively. These images were mapped by the encoder into their corresponding latent representations $p(z_1|v_1)$ and $p(z_2|v_2)$. A set of intermediate points was computed following Eq. (8). Those intermediate representations were then used to generate synthetic vessel networks and corresponding retinal images, displayed in Fig. 9. We can appreciate how the vessel networks clearly change from one point to another, which is yet another indicator that the model did not simply memorize the training examples. Moreover, the transition between left and right optical disk is sharp, indicating that the model successfully captured the knowledge that valid vessel networks only contain one optical disk. Also, the color and texture of the eye fundus images varies smoothly, even on the sharp transition between left and right-located optical disk.

Next, in order to analyze if the latent space captures the semantic properties of the vessel networks, we performed two visual analogies. In the first case, three retinal network images v_{11}, v_{12}, and v_{21} were chosen such that: 1) v_{11} contains relatively few visible vasculature and the optical disk to the left; 2) v_{12} contains relatively much visible vasculature and the optical disk also to the left; 3) v_{21} contains relatively less visible vasculature and the optical disk on the right. We then apply the encoder q to obtain their associated latent representations $q(z_{11}|v_{11})$, $q(z_{12}|v_{12})$, and $q(z_{21}|v_{21})$. Finally, we compute a fourth latent representation z_{22} associated to z_{21} in the following sense: z_{22} should have the same relationship with respect to z_{21} as z_{12} has with respect to z_{11}. This means that the decoded vessel network image should maintain the optical disk on the right, while showing a larger amount of visible vessels. We thus apply Eq. (10) to obtain z_{22}, and synthesize the corresponding vessel network and associated retinal image. The results of this experiment are shown in Fig. (10a). As expected, the generated images contained a more visible vasculature, while the optical disk's position was preserved.

In our last experiment, we tested if we were able to disentangle the latent factors related to the position of the optical disk. For that, we selected three vessel tree images od_{11}, od_{12}, and od_{21} such that: 1) od_{11} contains the optical disk to the right; 2) od_{12} contains the optical disk to the left; 3) od_{21} contains the optical disk on the left. After application of the same strategy as before, we should expect to synthesize a retinal image that preserves the amount of vasculature, but translates the optical disk to the right. As shown in Fig. (10b), our model successfully displaced the optical disk while keeping the amount of visible vessels, implying that it correctly disentangled the latent space direction related to the optical disk's location.

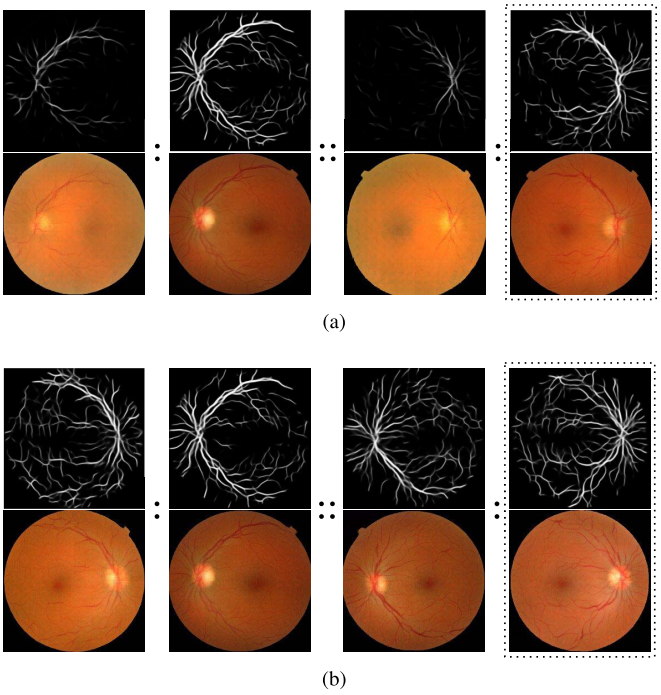

Fig. 10. Vessel networks from the first 3 columns (v_1, v_2 and v_3) of both Figs. (10a) a (10b) were encoded from test set vessel networks. The vessel network from the last column (v_4) is the result of applying the same transformation between v_1 and v_2 to v_3. In Fig. (10a), the relationship that was successfully captured by the model was the increase in the amount of visible vessels. In Fig. (10b), the true relationship was the change in the position of the optical disk.

IV. Limitations and Future Work

Generative models are always limited by the information contained within the training set, and how it captures the variability of the underlying real world data distribution. In this sense, the proposed technique was only trained on 614 healthy macula-centered retinal images, extracted from a single database (Messidor-1). Even with such a relatively small training set, our technique shows a remarkable capability of generating realistic synthetic images that substantially differ from the examples the system observed during training. Nevertheless, this reduced training set limits its capability to generate, *e.g.*, optical disk centered images, or pathological instances. Overcoming this obstacle is the first natural extension of our work. A first alternative could be to train a one-class classifier

with synthetic healthy images, and treat pathology as an anomaly discovery problem. A more general approach would involve addressing the diagnosing problem by implementing Class-Conditional Adversarial Models, such as [34] or [35], in which the training data comes with annotations. These kind of models can generate points in the data manifold corresponding to a particular label. In this way, not only diagnosing systems but many applications can be enhanced by newly generated images, annotated with information of interest. For instance, a compelling problem to investigate would be to employ generated retinal images to replace missing or corrupted images within longitudinal studies. This could be achieved by means of a model that learns to interpolate between different time points within a large dataset.

There are other limitations of the proposed approach that should be object of future research. First, the size of the synthetic images (256×256) is far from the resolution provided by images produced by current retinal fundus image acquisition systems. Also, although the generated images and associated vessel networks have an overall consistent appearance, and they seem to be reasonably useful to train a vessel segmentation model without manual vessel annotations, the realism of the synthetic vessel maps still does not reach that of real vasculatures. The generated synthetic vessel networks often exhibit abnormal interruptions, unusual width variation along the same vessel, and there does not seem to be a clear distinction between veins and arteries.

Most of the above drawbacks can be attributed to the amount of available data and computational resource restrictions, and not to a limitation intrinsic to the proposed technique. Therefore, in the future, the introduction of clinical labels or annotations in the context of a large scale high-resolution data collection will be the first natural extension of our model, as a part of the more general goal of producing realistic and interesting synthetic images that can be employed to train models to solve more complex retinal image analysis tasks. These may involve locating different areas of the retinal anatomy, or performing diabetic retinopathy diagnosis, to name a few.

In general, the availability of an additional set of training examples that can be efficiently generated on-demand could greatly impact the size and capacity of the models the retinal image analysis community train. These new annotated examples can be applied to validate novel retinal image understanding techniques, or to supplement existing datasets by expanding them with meaningful data. In addition, the proposed approach is not limited to retinal imaging. In our case, we employed the vessel tree as a proxy that serves as a guide for the model to learn to locate all parts of the anatomy consistently while generating plausible texture. The same methodology could be applied to different medical image analysis problems in which there exists such an intermediate structure.

V. Conclusion

In this work, a generative model capable of synthesizing new vessel networks and corresponding eye fundus images was presented. This model learns the underlying structure of the manifold of plausible retinal images from examples of pairs of vessel networks and eye fundus images. Once trained, it can generate both synthesized vessel networks and retinal images, that are shown to contain rich visual information and to be different from the training examples. The method is capable of generating realistic vessel geometries and retinal image texture, while keeping the global structure consistent.

Notably, the user is only required to sample from an N-dimensional predefined prior Gaussian distribution $p(z)$ to generate a new pair of images. Additionally, we provided visual experiments demonstrating that the latent space associated with our generative model contained a well-defined semantic structure. Furthermore, our results show that it is possible to exploit that structure in order to gain more control over its output.

Acknowledgments

MDA is The Robert C. Watzke Professor of Ophthalmology and Visual Sciences. MDA is founder, President, Director and shareholder of IDx, LLC, Iowa City, and has patents and patent applications that may complement or compete with the technology that is the subject of this study. IDx, LLC is not associated with the presented study and has no interest in the presented methods.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[2] E. Menti, L. Bonaldi, L. Ballerini, A. Ruggeri, and E. Trucco, "Automatic generation of synthetic retinal fundus images: Vascular network," in *Proc. Int. Workshop Simulation Synth. Med. Imag. (SASHIMI)*, 2016, pp. 167–176.

[3] D. L. Collins *et al.*, "Design and construction of a realistic digital brain phantom," *IEEE Trans. Med. Imag.*, vol. 17, no. 3, pp. 463–468, Jun. 1998.

[4] E. Hodneland, E. Hanson, A. Z. Munthe-Kaas, A. Lundervold, and J. M. Nordbotten, "Physical models for simulation and reconstruction of human tissue deformation fields in dynamic MRI," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 10, pp. 2200–2210, Oct. 2016.

[5] G. van Tulder and M. de Bruijne, "Why does synthesized data improve multi-sequence classification?" in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, Oct. 2015, pp. 531–538.

[6] D. Nie, R. Trullo, C. Petitjean, S. Ruan, and D. Shen. (Dec. 2016). "Medical image synthesis with context-aware generative adversarial networks." [Online]. Available: http://arxiv.org/abs/1612.05362

[7] K. Kamnitsas *et al.* (Dec. 2016). "Unsupervised domain adaptation in brain lesion segmentation with adversarial networks." [Online]. Available: http://arxiv.org/abs/1612.08894

[8] S. Fiorini, L. Ballerini, E. Trucco, and A. Ruggeri, "Automatic generation of synthetic retinal fundus images," in *Proc. Eur. Italian Chapter Conf.*, 2014, pp. 41–44.

[9] C. D. Murray, "The physiological principle of minimum work applied to the angle of branching of arteries," *J. General Physiol.*, vol. 9, no. 6, pp. 835–841, Jul. 1926.

[10] P. Costa *et al.* (Jan. 2017). "Towards adversarial retinal image synthesis." [Online]. Available: https://128.84.21.199/abs/1701.08974

[11] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[12] 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

[13] W. Lotter, G. Kreiman, and D. Cox. (2015). "Unsupervised learning of visual structure using predictive generative networks." [Online]. Available: https://arxiv.org/abs/1511.06380

[14] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. (2016). "Learning from simulated and unsupervised images through adversarial training." [Online]. Available: https://arxiv.org/abs/1612.07828

[15] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun, "Efficient learning of sparse representations with an energy-based model," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2006, pp. 1137–1144.
[16] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. (2015). "Adversarial autoencoders." [Online]. Available: https://arxiv.org/abs/1511.05644
[17] T. White. (2016). "Sampling generative networks." [Online]. Available: https://arxiv.org/abs/1609.04468
[18] S. E. Reed, Y. Zhang, Y. Zhang, and H. Lee, "Deep visual analogy-making," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1252–1260.
[19] E. Decencière *et al.*, "Feedback on a publicly distributed image database: The Messidor database," *Image Anal. Stereol.*, vol. 33, no. 3, pp. 231–234, Aug. 2014.
[20] J. 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.
[21] K.-K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. Van Gool, "Deep retinal image understanding," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI*. Cham, Switzerland: Springer, Oct. 2016, pp. 140–148.
[22] 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.
[23] H. Fu, Y. Xu, S. Lin, D. W. K. Wong, and J. Liu, "DeepVessel: Retinal vessel segmentation via deep learning and conditional random field," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, Oct. 2016, pp. 132–139.
[24] P. Liskowski and K. Krawiec, "Segmenting retinal blood vessels with deep neural networks," *IEEE Trans. Med. Imag.*, vol. 35, no. 11, pp. 2369–2380, Nov. 2016.
[25] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Represent.*, 2013.
[26] S. Ioffe and C. Szegedy. (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift." [Online]. Available: https://arxiv.org/abs/1502.03167
[27] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. 30th ICML*, 2013, pp. 1–6.
[28] F. Chollet. (2015). *Keras*. [Online]. Available: https://github.com/costapt/adversarial_retinal_synthesis
[29] V. Gulshan *et al.*, "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *JAMA*, vol. 316, no. 22, pp. 2402–2410, 2016.
[30] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever, "Mutual-information-based registration of medical images: A survey," *IEEE Trans. Med. Imag.*, vol. 22, no. 8, pp. 986–1004, Aug. 2003.
[31] Z. Wang, A. C. Bovik, and L. Lu, "Why is image quality assessment so difficult?" in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 4. May 2002, pp. IV-3313–IV-3316.
[32] L. Theis, A. van den Oord, and M. Bethge, "A note on the evaluation of generative models," in *Proc. Int. Conf. Learn. Represent.*, 2016.
[33] M. Niemeijer, M. D. Abràmoff, and B. van Ginneken, "Image structure clustering for image quality verification of color retina images in diabetic retinopathy screening," *Med. Image Anal.*, vol. 10, no. 6, pp. 888–898, Dec. 2006.
[34] K. Sohn, X. Yan, and H. Lee, "Learning structured output representation using deep conditional generative models," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2015, pp. 3483–3491.
[35] M. Mirza and S. Osindero. (Nov. 2014). "Conditional generative adversarial nets." [Online]. Available: http://arxiv.org/abs/1411.1784