RnD Project

Controlled generation of diabetic retinopathy images

Niraj Mahajan, Shalabh Gupta

{180050069, 180050095}

Advisor : Prof Suyash Awate

1 Introduction/Problem Description

Topological and geometrical properties of vessel filamentary structures provide a lot of valuable data for clinical analysis of diseases like diabetic retinopathy. Therefore presence of annotated retinal images and corresponding vessel structures is essential to diagnosis of diabetic retinopathy and necessary for further research in the field.

But bringing together annotated data is increasingly becoming difficult and costly. This is primarily because there is a huge scarcity of annotated data which is highly laborious. There are quite handful of annotated images of filamentary structures available.
In this work, we are focused upon trying to generate medical images to aid in training specific automatic tasks. While we are focused mainly on retinal images generation for diabetic retinopathy diagnosis, we also aspire to extend this work in generating automatic brain tumor segmentations. Also, we try to take into account the grade of retinal disease and condition our images on that.

1.1 Adversarial Methods

In this section, we summarise all the adversarial methods that we tried out and used in our experiments.

Generative Adversarial Networks Generative adversarial networks, or GANs are techniques to generate photo realistic images and are one of recent developments of modern deep learning. [1] In these, a discriminator and a generator play a 2 player zero sum game where the role of the discriminator is to distinguish between its two inputs and the generator tries to fool the discriminator by synthesizing artificial images very similar to original ones. [2] We are interested in a very specific variant of GANs that is cGANs.

Conditional GANs Conditional GANS or cGANs are required when we want to control the class of the data to be generated - this feature is not available in vanilla

GAN/DCGANS. To extend GANs into cGANs we can simply condition both the generator and discriminator with some extra information y which could be anything, class data or labels etc. [3]

Variational Autoecoders Variational Autoecoders, or VAEs, are autoencoders whose training is regularised to avoid overfitting and ensure that the latent space has good properties that enable generative process. Just as a standard autoencoder, a variational autoencoder is an architecture composed of both an encoder and a decoder and that is trained to minimise the reconstruction error between the encoded-decoded data and the initial data. However, in order to introduce some regularisation of the latent space, we proceed to a slight modification of the encoding-decoding process: instead of encoding an input as a single point, we encode it as a distribution over the latent space (usually gaussian). This is ensured by penalising the training on the KL Divergence loss between the generated latent space distribution and some target distribution.

Adversarial Auto Encoders AAE is a probabilistic autoencoder that uses adversarial techniques to perform variational inference by matching hidden code vector to some arbitrary distribution. The difference between this and a VAE is that while a VAE used the KL Divergence loss to regularise the latent space distribution, AAEs introduce an adversarial loss onto the latent space as shown in fig 1, forcing the generated distribution to coincide with that of the data [4]

2 Related Works

Since annotated data is scarce and costly to obtain a lot of research is ongoing in this field, particularly in the medical imaging domain. In [5], the authors propose an adversarial training strategy to generate CT images from MR images for better diagnosis. Their architecture consists of a simple fully convolutional generator and a discriminator at end to distinguish between the generated CT and training MR image. A special gradient difference loss is added to deal with blurry predictions.

The work in [2] is centred around generating retinal images directly from filamentary segmented images. They show that their method is able to generate diverse images with the same visual appearance as their original image using a mixture of GANs and neural style transfer. The neural style transfer loss in this comprises of style loss, variation loss and content loss.
Zhou et al. [6] propose a method to synthesize high-resolution fundus images which can be manipulated with arbitrary grading and lesion information. They propose a novel pipeline with various specialized modules to preserve high level minute details in the generated image.
Costa et al. [7] address the problem of synthesizing retinal color images along with vessel segmentations by applying recent techniques based on adversarial learning. As shown in Fig. 2 An adversarial autoencoder (AAE) has been implemented to learn distributions on vessel segmentations and generate new ones and a generative adversarial network is then applied to these vessel segmentations to obtain colored retinal fundus images.

The model is trained end to end to generate colored images directly from training vessel images. We wish to improve upon this method proposed by *Costa et al.* to generate controlled retinopathy images, by controlling the grade of the images generated. We also are looking to explore the prospects of inserting location controlled abnormalities in the generated images.

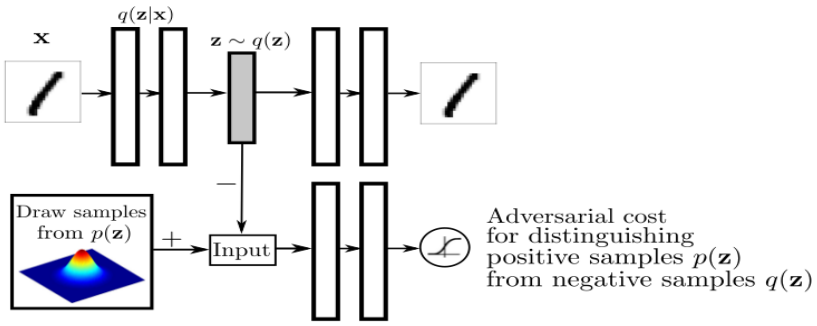


Figure 1: The standard training pipeline for an Adversarial AutoEncoder

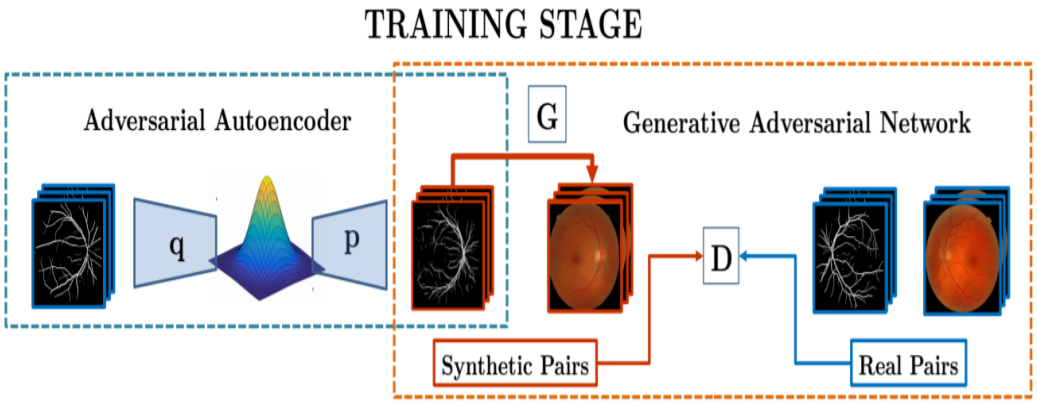


Figure 2: Architecture used by *Costa et al.* [7]
for End to End diabetic retinopathy image synthesis. The Adversarial AutoEncoder is trained to generate realistic vessel maps, while the GAN uses a pix2pix architecture [8] for developing the corresponding fundus images.

3 Our Method

In this section, we summarise the method that we used along with the details of all the relevant architectures.

3.1 Stage 1: Vessel Network Generation

In the first step, vessel images are to be taken input to generate more vessel images itself. But as pointed out in the paper, the normal VAE model learns a trivial identity mapping. The adversarial autoencoder framework is thus used 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. The autoencoder aims to minimize the reconstruction error along with trying to control the probabilistic structure of q(z) where q is encoder part and z is the input to the network, by making it match the prior distribution p(z) that can be easily sampled. The optimization process thus consists of two steps, in first of which the discriminator tries to maximise the following loss function:

$$L_{code}(D_{code}, q) = log(D_{code}(z)) + log(1 - D_{code}(q(z|v))) \tag{1}$$

The second step consists of updating encoder-decoder weights trying to minimise reconstruction error and maximise discriminator error.

$$L(D_{code}, q, p) = L_{code}(D_{code}, q) + \gamma L_{rec}(q, p) \tag{2}$$

where L_{rec} is the reconstruction error between original image and constructed image from the decoder. We used L1 loss for this and γ is a constant, set to 100 in original implementation.

3.2 Stage 2: Vessel Network to fundus translation

The second step in our method takes the generated vessel image as input, and passes it through a pix2pix architecture to produce colored retinal images as output. So basic aim of this generator G is to learn a mapping from a vessel binary segmentation to another representation r. Also this is not one to one, so G : v -¿ r1,r2,r3..rm. The role of the discriminator here is to distinguish between real pair (vessel,retina) and fake pair (vessel,retina). [7]
Loss function in this step can be written as:

$$L_{im2im}(G, D) = log(D(v, r)) + log(1 - D(v, G(v)) + \lambda[||r - G(v)||]_1 \tag{3}$$

The end to end network basically combines the two steps to obtain a model that can produce retinal images from random noise. This is done because both the steps are deeply interconnected. Better quality of vessel networks will ensure realistic retinal image and vice versa. [7] Therefore the total loss function is

$$L_{final}(G, D, D_{code}, p, q) = L_{im2im}(G, D) + L(D_{code}, p, q) \tag{4}$$

G, p and q try to minimise the loss function while D and D_{code} try to maximise it.

3.3 Details of the Architectures

Adversarial Autoencoder The AAE used in the first stage consists of an encoder q, a decoder p and a discriminator D_{code}. The encoder is a sequence of repeated convolutions and Downsampling layers to convert a $256 \times 256 \times 1$ vessel segmentation to a 512 length vector. The 512 length vector is then passed through two dense layers to output two l dimensional vectors, one of which will act as μ and other as σ, where l is the dimensions of the latent space. Now we sample a value from the standard normal distribution, let it be val. The hidden latent vector can now be obtained as follows:

$$z = \mu + \sqrt{\sigma} \times val \tag{5}$$

Note that z is l dimensional. We will be varying l in our experiments. The decoder is now a series of convolutions and Upsampling layers to convert this 32 dimensional vector z into a $256 \times 256 \times 1$ image. The code discriminator is a dense layer to convert 32 dimensional latent vector to a scalar value and apply sigmoid on it.

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		

Figure 3: Architecture of encoder and decoder in AAE

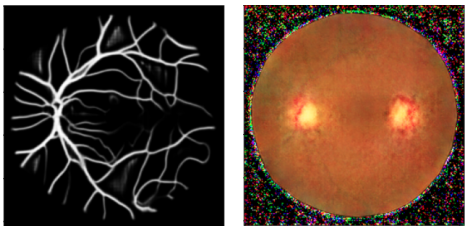
GAN For generating a fundus image from the vessel map, we use a pix2pix architecture. This consists of a discriminator and a generator. The discriminator D takes a pair of vessel segmentation($256 \times 256 \times 1$) and the corresponding generated retinal image($256 \times 256 \times 3$) concatenated over 3rd dimension, making it $256 \times 256 \times 4$ input dimensions and outputs 16×16 sigmoid values. The generator here is a UNet whose architecture is adapted from Isola *et al.* [9] taking input a vessel segmentation of dimensions $256 \times 256 \times 1$ and outputs a $256 \times 256 \times 3$ colored retinal image.

4 Experiments

Here we describe all the variations that we used for training models for the above method, and summarise all the relevant results. We have used the Messidor-1 dataset [10] for our training. Since the Messidor dataset contains only the fundus images, we used an IterNet architecture [11] to generate vessel networks from all the 1200 images in the dataset.

To replicate the results generated by *Costa et al.* [7], we trained an end to end model, with simultaneous training of the AAE for vessel map generation, and the GAN for translating the vessel maps to retinal fundus. Unfortunately, we got suboptimal results as shown in fig4 , probably due to a bug in the implementation. Thus, to stabilize the

training process and to get a better tune the parameters, we moved to implementing an pipelined architecture with sequential training which will be covered in the subsequent subsections.



(a) Generated Vessel Maps (b) Generated Fundus

Figure 4: Direct training of the end to end network led to bad results, probably due to some implementation mistake. Tu better isolate the implementation error, we subsequently reverted to two stage pipelined training

4.1 Vessel Network Generation

We trained Adversarial Autoencoders with various parameters upto 2000 epochs (some cases 5000 epochs). In the *Costa et al.* paper, it was mentioned that grades 3 and 4 segmentations were dropped because of being bad in visual appearance and were primarily not good for training for vessel maps. Our inspection of these grades also turned out to be similar and we consequently dropped these grades while training. With D_{code} loss as BCE, results were a bit better as expected.
We first trained a VAE to compare the results with different methods. Since the VAE does not have an adversarial component to it's loss, the generated images had fuzzy edges as shown in figure 5 although the underlying vessel network was somewhat sensible. The AAE, on the other hand, showed promising developments. For the AAE, we tried two adversarial loss functions for training - Binary Cross Entropy (vanilla GAN) and MSE (lsgan). The corresponding results are summarised in 6. We tried various latent spaces for both of these mathods - 16,32,64,128,256. For small latent spaces like 16, 32 the images generated really poor images as shown in figure 7, whereas images generated with higher latent spaces like 128,256 gave good results.
Apart from this, we observed that the end-to-end process generated images with well formed and continous vessel networks. Thus we also trained AAEs with a dual adversarial loss- one in the latent space, while the other on the output of the decoder (figure 8).

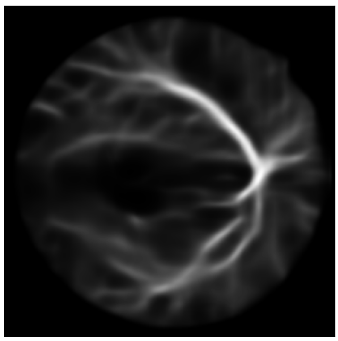
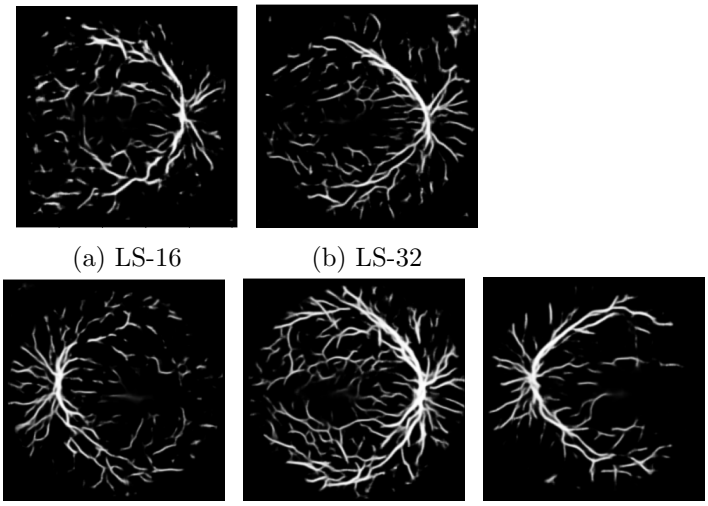


Figure 5: Generation of vessel maps using VAEs



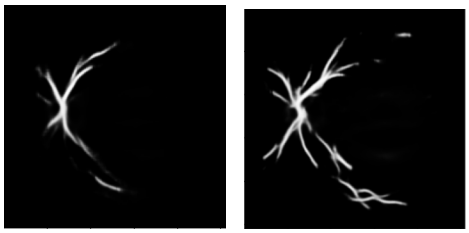
(a) LS-16 (b) LS-32

(c) LS-64 (d) LS-128 (e) LS-256

Figure 6: Training results of AAE using lsgan for adversarial loss, with varying latent space sizes

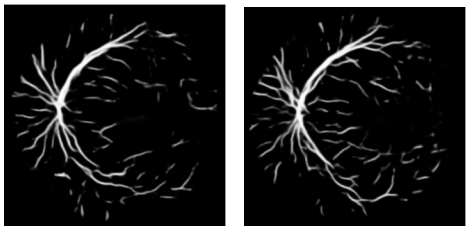
4.2 Vessel Network to Fundus Images

Now that we have vessel networks generated, we train a pix2pix network that converts the vessel network to a fundus image. We trained this over the entire dataset (all classes), hence the currently trained model does not have control over the grade of the diabetic retinopathy image produced by the model, but still, the model produces really good results as shown in figure 9. We plan to further develop this GAN by inducing conditional behaviour on the abnormality grade.



(a) LS-16 (b) LS-32

Figure 7: Training results of AAE using vanillaGAN for adversarial loss, with very low latent size. It can be concluded from these results that smaller latent sizes give really bad results.



(a) LS-256 (b) LS-512

Figure 8: Training results of AAE using a double adversarial loss - at the latent space and the output space, with varying sizes of the latent space

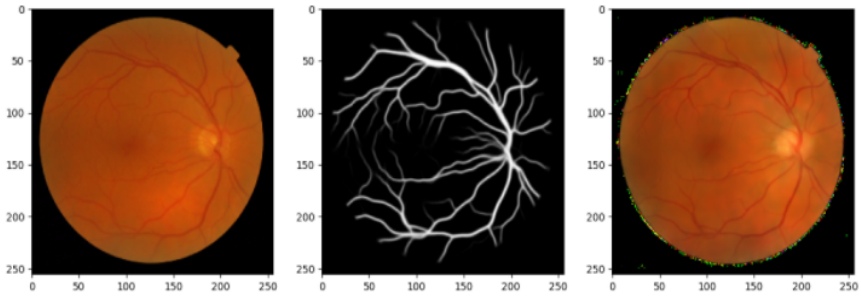


Figure 9: Results of vessel network to fundus translation using a pix2pix network. Given a vessel map, we are able to generate good and realistic fundus images.

5 Challenges/Problems faced

- One big challenge is the unavailability of segmentation data for retinal images. Because of this, we had to make our own vessel maps using a different neural

network, which had to be artificial.

- It used to take days in training due to the huge complexity of the models having a lot of parameters. Not to mention a few runs where the code was unfortunately bugged.
- Training the generators to produce images was challenging since the dataset contained a mix of retina images for both eyes, and the location of the optical disk was variable. Therefore the images produced used to have retinal spots on both sides of the image.
- Using a large batch size was unfeasible since it consumed a lot of GPU memory and so we had to keep the batch size down to around 30.
- Obtaining an optimal latent space for the AAE was a big challenge. We tried many latent space sizes but the trends were not consistent. But we observed that latent size 128 gave the best results.

6 Conclusions and Future Work

Throughout this RND project, we have learnt important concepts of various adversarial methods and GAN architectures. There are several implementation intricacies that we learnt- right from which activation function should be/should not be used to how to stabilize the GAN training, and we are confident that we will be able to apply these subtle points in the future to generate better and faster results. Although the trends of the latent space size are not quite conclusive, it is clear that latent size 128 seems to give the best results. The end to end training showed real promise while generating vessel maps, which proves that an optimal solution is attainable. Thus varying some hyperparameters as well as the training parameters should help us follow the correct path.

We had success in generating location controlled brain tumour images as a part a separate project (CS736), and are hoping to incorporate a similar method in this application. Since the diabetic retinopathy lesions are a lot more complicated than brain tumours, we propose a modification to the method proposed, where we train a special model that generates conditional DR lesions, which can then be inserted in the healthy retinopathy images. Another experiment for conditional DR image synthesis is to train a grade classifier and to use such a classifier in the adversarial loss component to bolster the conditional grade wise behaviour. Furthermore, we wish to continue tuning the parameters of the vessel map generation phase so as to achieve optimal results.

References

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[2] He Zhao, Huiqi Li, and Li Cheng. Synthesizing filamentary structured images with gans, 2017.

[3] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014.

[4] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders, 2016.

[5] Dong Nie, Roger Trullo, Caroline Petitjean, Su Ruan, and Dinggang Shen. Medical image synthesis with context-aware generative adversarial networks, 2016.

[6] Yi Zhou, Boyang Wang, Xiaodong He, Shanshan Cui, and Ling Shao. Dr-gan: Conditional generative adversarial network for fine-grained lesion synthesis on diabetic retinopathy images. *IEEE Journal of Biomedical and Health Informatics*, page 1–1, 2020.

[7] Pedro Costa, Adrian Galdran, Maria Ines Meyer, Meindert Niemeijer, Michael Abràmoff, Ana Maria Mendonça, and Aurélio Campilho. End-to-end adversarial retinal image synthesis. *IEEE Transactions on Medical Imaging*, 37(3):781–791, 2018.

[8] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2018.

[9] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2018.

[10] Etienne Decencière, Xiwei Zhang, Guy Cazuguel, Bruno Lay, Béatrice Cochener, Caroline Trone, Philippe Gain, Richard Ordonez, Pascale Massin, Ali Erginay, Béatrice Charton, and Jean-Claude Klein. Feedback on a publicly distributed database: the messidor database. *Image Analysis & Stereology*, 33(3):231–234, August 2014.

[11] Liangzhi Li, Manisha Verma, Yuta Nakashima, Hajime Nagahara, and Ryo Kawasaki. Iternet: Retinal image segmentation utilizing structural redundancy in vessel networks, 2019.