Paired GAN: Image-Mask Pair Generation for Medical Image Segmentation

Yiheng Zhou (yz996), Eva Gao (eyg2), Olina Zhu (qz258)

May 9, 2023

Abstract

Data scarcity remains a challenge for training deep learning applications in biomedical image segmentation. We propose a new data augmentation method to efficiently multiply mask and image pair volumes, achieved by a Generative Adversarial Network (GAN)-powered, paired image and mask synthesis process - VessGAN and Vess2Image. The VessGAN model, represented by a vanilla GAN, takes ground-truth masks as input and outputs synthetic masks; whereas the Vess2Image model, comprised of a U-Net generator and a patchGAN discriminator, reconstructs images based on synthetic masks generated by VessGAN, thereby pipelining the paired synthesis of biomedical images and masks for further training. Architecture exploration and experimentation results show that the proposed framework is potentially valid, but fine-tuning is required to mitigate the issues of mode collapse and convergence failure, thereby generating high quality synthetic images and masks for real world applications.

1 Introduction

1.1 Background and Motivation

Deep neural networks have demonstrated impressive performance in the field of medical image segmentation. By accurately delineating the anatomical structures and pathological features in the images, deep learning-based segmentation methods enable quantitative analysis of disease manifestations which further support healthcare professionals and patients from diagnosis to treatment. However, such robustness is typically powered by large-scale well-annotated training data sets which are rarely available and can be costly to obtain, especially in the cases of emerging diseases or diseases that have intrusive imaging procedures. Data scarcity therefore becomes one of the major challenges in deep learning applications, leading to inadequate feature representation and lack of generalization.

1.2 Related Works

To mitigate the problem of data scarcity, a series of data augmentation techniques have been proposed, as a data-space solution, to artificially enhance the size and diversity of the training datasets. Traditional techniques commonly perform geometric and color transformations on existing images and masks identically, such as rotation, shearing, contrast stretching and histogram equalization, but have shown limited ability to capture anatomical variations in real-world settings and are highly sensitive to parameter selection (Zhao *et al.*, 2019). Alternatively, the state-of-the-art techniques, particularly generative adversarial networks (GANs), can unlock additional information from the original dataset by creating synthetic instances (Shorten & Khoshgoftaar, 2019).

The GAN consists of two networks: a generator learns to produce synthetic samples, and a discriminator learns to distinguish fake data from real ones. Multiple GAN architectures have been introduced for high-quality image synthesis, but their effectiveness and applicability are largely limited in supervised image segmentation tasks which require the generation of synthetic image-mask pairs. A straightforward solution is to train with the concatenated fusion of the mask and the image along the channel axis, rather than inputting the image or the mask independently (Pandey *et al.*, 2020). However, it is considerably challenging to optimize hyperparameters for such GAN frameworks which then suffer from training instability and non-convergence issues. Another promising approach is to leverage two separate GAN models such that the first model outputs new synthetic masks, whereas the second one translates masks into images through style transfer. Compared to the conventional augmentation techniques and the single GAN candidate, the collaborative two-GAN method can achieve superior performance in terms of accuracy, efficiency, and robustness (Pandey *et al.*, 2020).

This study aims to adapt the two-GAN approach to enrich the existing retinal datasets and aid in early detection and management of various retinal diseases. Specifically, we employ a two-step approach containing a paired-model architecture, where the output from the GAN model in the first step is fed successively into the Pix2Pix model in the second step as input.

2 Model

Two models were developed in this project: VessGAN and Vess2Image, as described below.

2.1 VessGAN

VessGAN is an adaptation of the vanilla GAN proposed by Goodfellow *et al.* (2015), composed of a generator and a discriminator. The generator consists of four hidden blocks (a linear layer followed by either a leaky ReLU or Tanh activation function), and the discriminator also has four hidden blocks (linear layer followed by dropout and activation function). The last discriminator layer has a Sigmoid as activation. The discriminator and the generator were trained for 200 epochs after fine-tuning of learning rates, and the discriminator was trained twice for each iteration of the generator. The VessGAN generator takes in a noise vector and learn to generate a mask. The discriminator takes in both real masks and fake masks outputted by the generator. VessGAN also uses the Binary Cross Entropy loss function for calculating the discriminator and the generator losses.

2.2 Vess2Image

The Vess2Image is a modified variation of the Pix2Pix conditional GAN model (Isola *et al.*, 2018), which uses a U-Net based architecture as a generator and a convolutional PatchGAN classifier as a discriminator. The model was adapted to run using PyTorch Lightning, which has been shown to effectively reduce the runtime of relevant deep learning models (Maurya, 2021). The U-Net consists of eight encoder blocks followed by seven decoder blocks. Each block in the encoder consisted of a sequence of the following: 2D convolutional operation and an activation function. All except the first and last encoder block had batch normalization. Similarly, each block in the decoder consisted of a sequence of 2D convolution operation and batch normalization, and only the first three decoder block had dropout. Skip connection was added between the encoder and the decoder states to prevent against the vanishing gradient problem. PatchGAN is a variation of the CNN, which classifies each patch of the input as real or fake. The PatchGAN consists of four successive down-sampling blocks of the same structure as the encoder and decoder blocks, followed by a convolutional layer. During each epoch, the discriminator was trained before training the generator. The generator loss combines

adversarial loss with L1 loss to ensure pixel accuracy. The hyperparameter λ penalizes the generator loss if the new generated image is not close to the ground truth: (Isola *et al.*, 2018). Through trial and error, the working λ value was identified to be 160. The Binary Cross Entropy Loss was used as the discriminator loss function. Fig. 7 displays the model architecture as well as inputs and expected outputs. The pseudo-code for the training step is shown below.

Algorithm 1: Training a Vess2Image

```
Result: Generator outputs a fake image when given mask
Initialize generator, discriminator;
Set X be training data;
for epochs in 1....NUMEPOCHS do
    for l in 1...NUMBATCHS do
        mask, image = X[l]
        Loss_disc = trainDiscriminator(mask, image)
        Loss_gen = trainGenerator(mask, image)
        Calculate gradients for both losses;
        Update both generator and discriminator weights;
    end
end
```

2.3 Generalization Testing

The generalizability of both models were tested using the COVID chest slic CT dataset with binary masks that differentiate between abnormal and healthy regions. The Vess2Image model was trained again using the new dataset.

3 Results and Contributions

Three sets of output masks for VessGAN are shown in Fig. 1 in the **Figures** section below. Each group A, B, and C represent masks generated by a separate VessGAN model. Left columns in each group display model input, and the right displays output. The top row shows inputs and outputs for one mask in the training batch at epoch 152, middle row corresponds to epoch 180, and bottom row represents epoch 200. The first epoch selected for display is 152 because previous epochs do not show apparent structure of interest against a background of noise. The output masks in Fig. 1 do not achieve the same caliber of quality compared to the input, and only one retinal structure was generated for every instance of the VessGAN model, i.e., the only difference seen in output at different epochs is the quality of image but not variation in vessel structure. The discriminator and generator losses for VessGAN trained on 200 epochs are shown in Fig. 2. There is evident decrease in generator loss in the first 20 epochs and the loss begins to oscillate until the last epochs. The discriminator loss oscillates at high magnitudes without stabilizing. The results from Fig. 1 and Fig. 2 both suggest mode collapse and poor performance of the VessGAN model.

The output images for the second step in the project, Vess2Image, are shown below in Fig. 3. The rows represent output at epochs 1, 6, 11, 16, and 21 respectively from top to bottom. From left to right are the input masks, the target output style, and the generated output images. The generated images demonstrate a significant level of vessel structure variance between epochs trained on shuffled batches, and therefore all output in Fig. 3 were generated through training a single Vess2Image model instance, in contrast to VessGAN which required a separate model instance for generating each vessel

structure. Despite such, output in Fig. 3 still demonstrates low image quality that is incomparable to the original images, shown in the middle column. The losses for discriminator and generator in the Vess2Image model are shown in Fig. 4. The low quality images in Fig. 3 and discriminator loss that approximates to zero beginning at the very first epochs and continually increasing generator loss indicate convergence failure. However, it is difficult to observe trends in the discriminator loss shown in Fig. 4 as the scale of the generator loss is tens of thousands times that of the discriminator loss. This could potentially be attributed to the λ value of 160 which amplified the generator loss.

The generalizability of the VessGAN and Vess2Image models was tested using a chest CT dataset (Figs. 5 and 6). Fig. 5 shows the output of VessGAN when ran on the new dataset for 200 epochs, where the left is a selected original mask from the dataset and right is the model output. The image quality is still low and mode collapse was evident. Fig. 6 shows the output of Vess2Image trained on the CT dataset for 5 epochs, where the left is the input image, the middle is the target output style, and the right is the model output.

Eva, Yiheng, and Olina were all responsible for solidifying the research topic, completing background research, and report finalization. All three participated in the coding and debugging processes of each model, and provided aid to one another in their specific tasks. Specifically, Eva was responsible for data preprocessing and implementing the U-Net generator for Vess2Image, Yiheng was responsible for implementing the discriminator, training, and visualization steps in Vess2Image, and Olina was responsible for implementing the VessGAN model as well as its training and visualization.

4 Discussion and Conclusion

The VessGAN model performed poorly, demonstrating evident mode collapse through the generation of a single repeated image in any model instance and oscillation of losses (Figs. 1 and 2). This is likely due to the hyperparameter choice of batch size of 1, which restricts the latent dimension, thereby making the model susceptible to mode collapse (Brownlee, 2019). However, the root problem stems not from poor hyperparameter choice, but rather hardware limitations. The code for this project ran exclusively on Google Colaboratory, which provides limited memory and GPU despite purchasing hardware package upgrades. Running with any larger batch sizes through great numbers of epochs resulted in out of memory errors. Moreover, the limited input dataset size of 20 masks may also be a contributor, as limited input restricts the mode diversity available for learning.

In addition to low output diversity, the VessGAN also exhibited low output image quality. This project experimented with different hyperparameters including learning rate, number of epochs, training a varying number of discriminators for each iteration of the generator, yet no experiment results were able to improve image quality significantly. The image quality in the first 90 epochs were limited to generating a noisy background eyeball shape, but no vessel structures were yet generated on a consistent basis. Apparent vessel structures only began to show consistently since epoch 110, but lack resolution until around epoch 150. Epoch 180 generally produces the best quality images, whereas the model appears to stop learning useful information in epochs 190-200 and the output again blurs. Given the low resolution and diversity, VessGAN output masks were not fed into the Vess2Image model, and instead masks from the original dataset were used in substitution in order to model the performance of Vess2Image in the case that VessGAN performs well.

The Vess2Image model captures vessel structures well, but does not show appropriate resolution nor contrast against noisy backgrounds (Fig. 3). The model seemed to only learn the features marked explicitly by the masks, such as eyeball shape and vessels, but none of the other features to generate a realistic image, such as highlights and shadowing to indicate depth. It did show some consideration of eyeball structure at its edges, but this is only demonstrated by the sharpened contrast at the

eyeball outlines, but none exist to reflect the 3-dimensional shape of the eyeball interiors (Fig. 3). This, along with losses shown in Fig. 4, reflects apparent convergence failure of the model. This could potentially be due to insufficient number of nodes implemented, or that training was stuck in a local minima, which is suggested by the decrease in performance in higher epochs (Fig. 3).

In addition, the generalization testing of both models on the chest slice CT dataset show that the same model and hyperparameters perform differently across datasets (Figs. 5 and 6), especially the Vess2Image model which shows significantly higher quality and more life-like images on the chest CT dataset (Fig. 6). This potentially indicates that hyperparameters require further fine-tuning when used on different datasets, or that the models perform better on images that require less depth definition and more attention to details on a 2-dimensional space.

A possible future direction to improve performance of the first step, VessGAN, could be combining the design of variational autoencoders (VAEs) and GAN (Khan *et al.*, 2018) in substitution of a traditional GAN. Through using variational autoencoder as the generator and an autoencoder as the discriminator in a GAN architecture, the model could compare latent representations and generate a similarity metric. Existing work using such an architecture has seen high-quality outputs (Khan *et al.*, 2018), and future work utilizing our two-step model for paired image generation could also experiment with pairing VAE-based GAN as the first step with Pix2Pix as the second step. Convergence failure in the Vess2Image model could potentially be countered by implementing more nodes to allow better learning, increase the number of training data provided, or experimenting with more suitable activation functions.

In summary, this project explores the possibility of utilizing a two-step framework of generating paired image-masks for deep learning. The VessGAN model was challenged by mode collapse due to hardware and potential implementation limitations, and the Vess2Image model showed convergence failure. Despite such, this project demonstrated that the paired network design is an appropriate method to use for matched image-mask generation in limited data conditions, and with appropriate fine-tuning of model performance, it could bring significant value to real-world situations such as timely generation of large paired-training datasets targeting emerging diseases. Future efforts could focus on improving model performances in the proposed paired-model architecture by substituting VessGAN with adversarial VAEs or increasing the number of nodes in the Vess2Image model.

5 Literature Cited

[1] Pandey, S., Singh, P.R., & Tian, J. (2020). An image augmentation approach using two-stage generative adversarial network for nuclei image segmentation. *Biomedical Signal Processing and Control*, 57, 101782, accessed at: https://doi.org/https://doi.org/10.1016/j.bspc.2019.101782
[2] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), accessed at https://doi.org/10.1186/s40537-019-0197-0
[3] Zhao, A., Balakrishnan, G., Durand, F., Guttag, J.V., & Dalca, A.V. (2019). Data augmentation using learned transformations for one-shot medical image segmentation. *arXiv*, accessed at: https://arxiv.org/abs/1902.09383
[4] Brownlee, J. (2019). How to identify and diagnose gan failure modes. *Generative Adversarial Networks*, accessed at: https://machinelearningmastery.com/practical-guide-to-gan-failure-modes/
[5] Khan, H.S., Munawar, H., & Nick Barnes. (2018). Adversarial training of variational auto-encoders for high fidelity image generation. *arXiv*, accessed at: https://arxiv.org/abs/1804.10323
[6] Isola, P., Zhu, J., Zhou ,T., & Efros, A.A. (2018). Image-to-image translation with conditional adversarial networks. *arXiv*, accessed at: https://arxiv.org/abs/1611.07004
[7] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., & Bengio, Y. (2014). Generative adversarial networks. *Communications of the ACM 63(11), 139-144.*, accessed at : https://dl.acm.org/doi/pdf/10.1145/3422622
[8] Maurya, A. (2021). Pix2Pix-Image to image translation with conditional Adversarial Networks, accessed at: https://librecv.github.io/blog/gans/pytorch/2021/02/13/Pix2Pix-explained-with-code.html
[9] Dodge, S.F., & Karam, L. (2016). Understanding how image quality affects deep neural networks. *QoMEX, 1-6.*
[10] Saood, A., & Hatem, I. (2021). COVID-19 lung CT image segmentation using deep learning methods: U-Net versus SegNet. *BMC Med Imaging 21, 19*, accessed at: https://doi.org/10.1186/s12880-020-00529-5

6 Figures

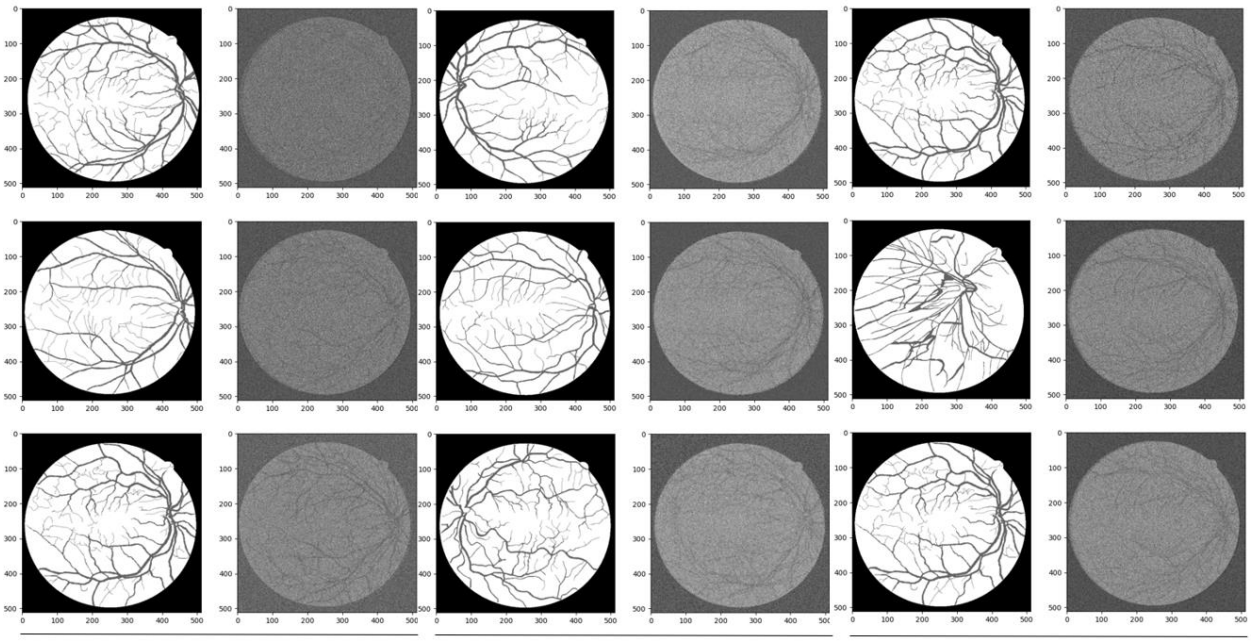

Figure 1. Example images generated by VessGAN. Each group A, B, and C represent images generated by a separate VessGAN model. Left images in each group display model input, and right images display output. The top row shows inputs and outputs for one image in the training batch at epoch 152, middle row corresponds to epoch 180, and bottom row represents epoch 200.

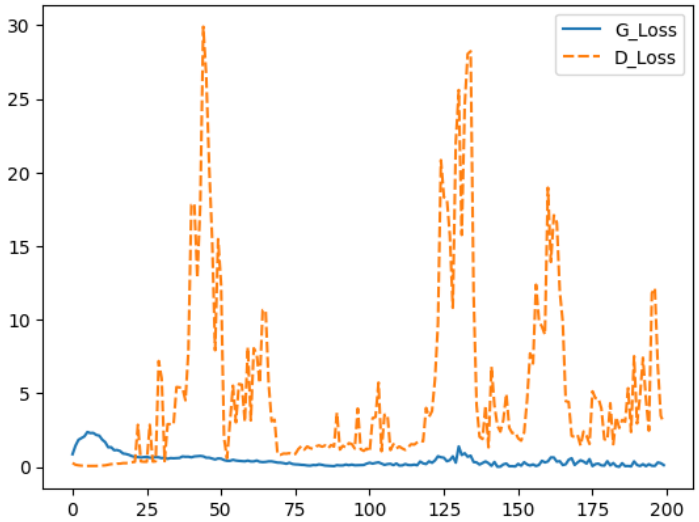

Figure 2. Line chart of generator loss and discriminator losses for 200 epochs on VessGAN.

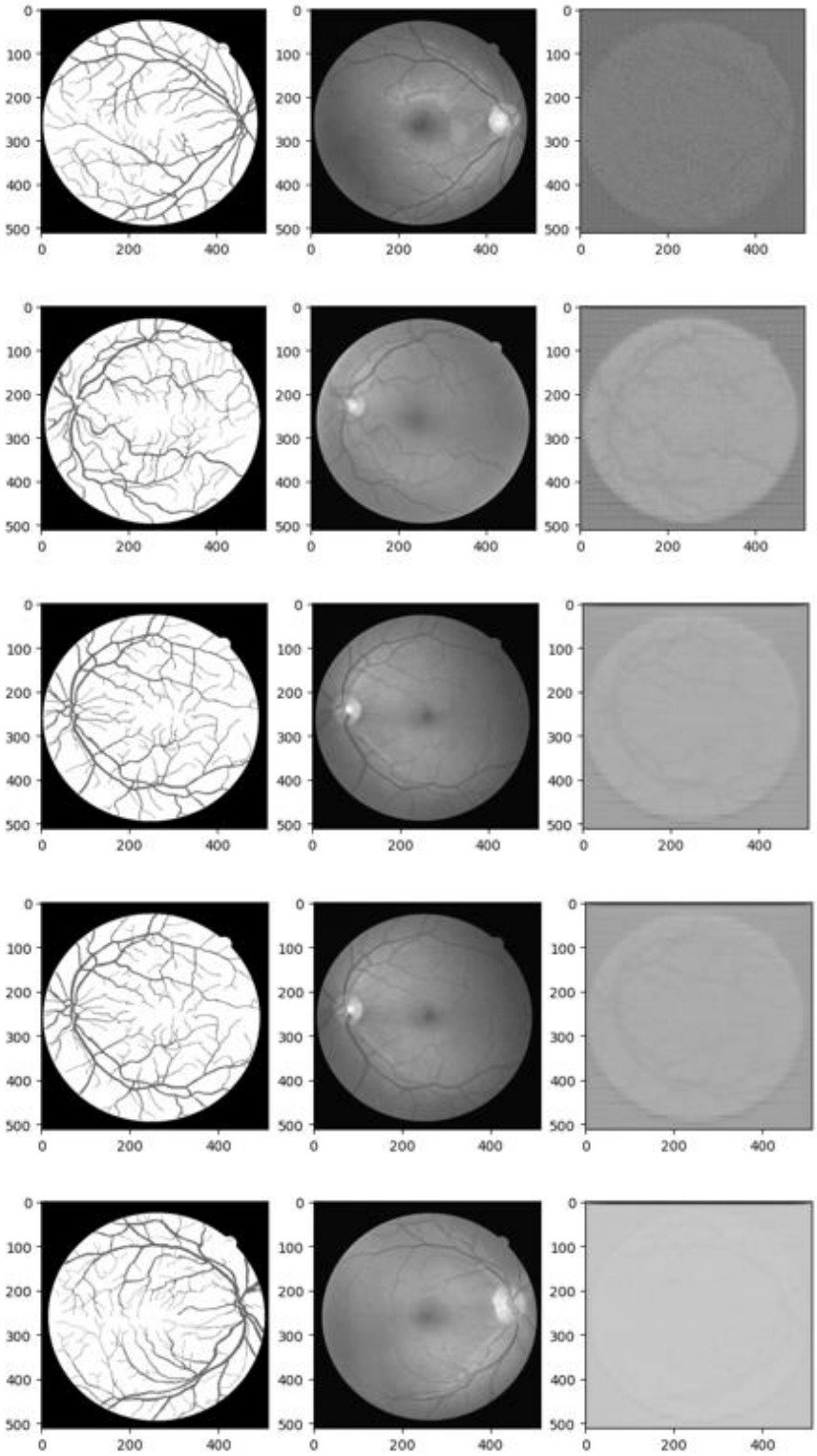

Figure 3. Example images generated by Vess2Image at epochs 1, 6, 11, 16, and 21 respectively from top to bottom. From left to right are the input images, expected image, and output.

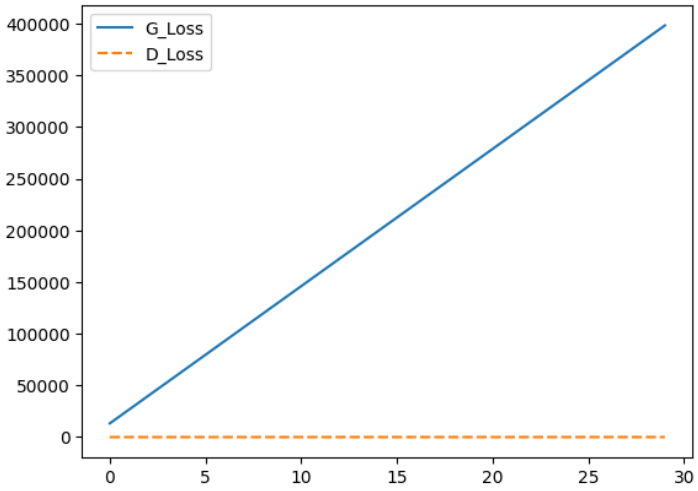

Figure 4. Figure 2. Line chart of generator loss and discriminator losses for 30 epochs on Vess2Image.

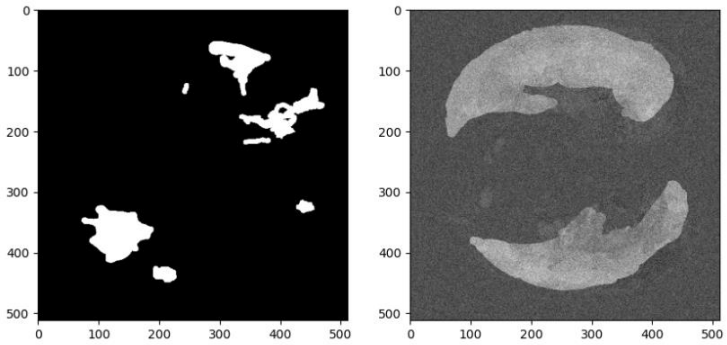

Figure 5. Example output of generated fake mask by VessGAN model trained on chest CT dataset. The left is a selected original mask from the dataset and right is the model output.

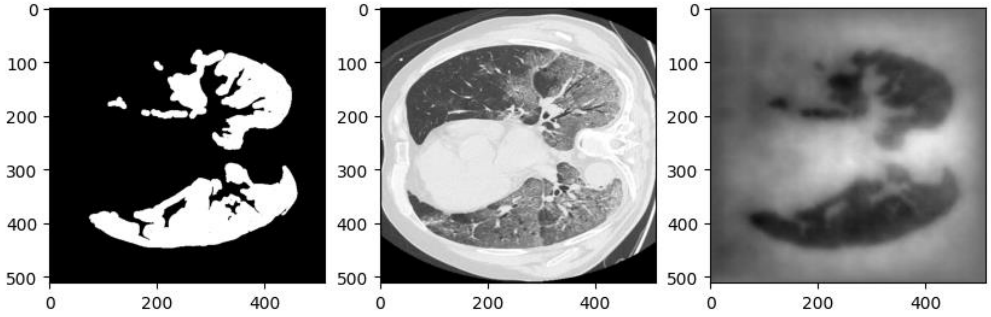

Figure 6. Example output of generated mask by Vess2Image model trained on chest CT dataset. The left is the input image, the middle is the expected output, and the right is the model output.

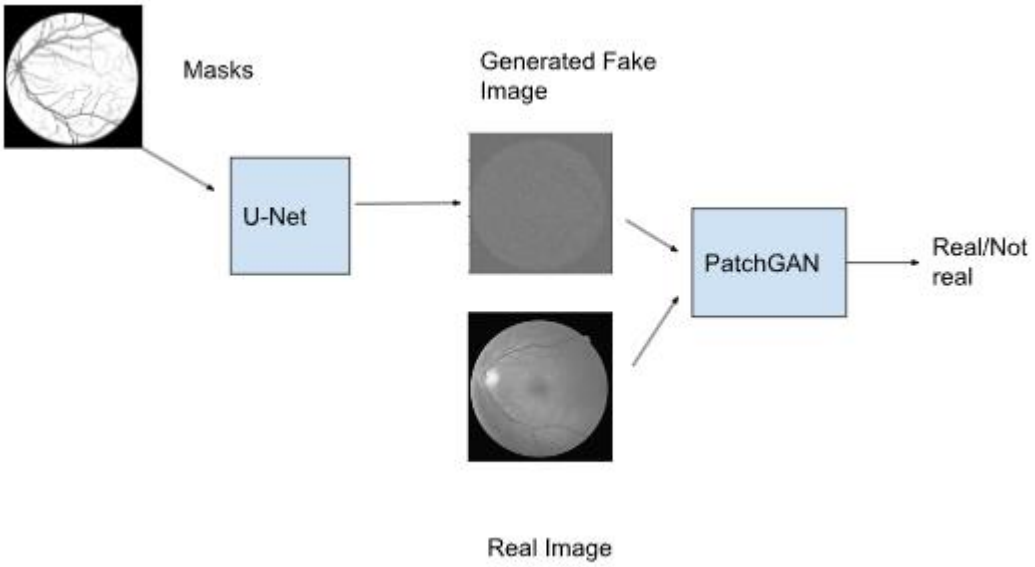

Figure 7. Diagram of the Vess2Image Model. Vess2Image consists of a U-Net generator and PatchGAN as a discriminator. The U-Net takes in a mask and learns to generate images using adversarial loss.