Retinal Vessel Segmentation in Fundoscopic Images with Generative Adversarial Networks

Sujit Rai, Sagarika Sharma

Problem Statement

- Generation of premise maps of retinal vessels from fundoscopic images

Retinal Image

Vessel Image

Dataset

Retinal Image

Vessel Image

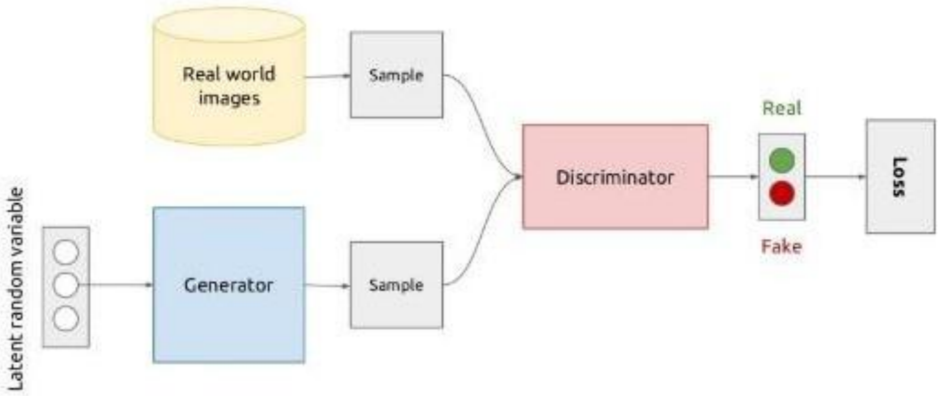
- 20 Annotated Images
  - 10 Train Set
  - 10 Test Set

Augmentation:

- Left-Right Flip
- Rotation
- Split (9:1)

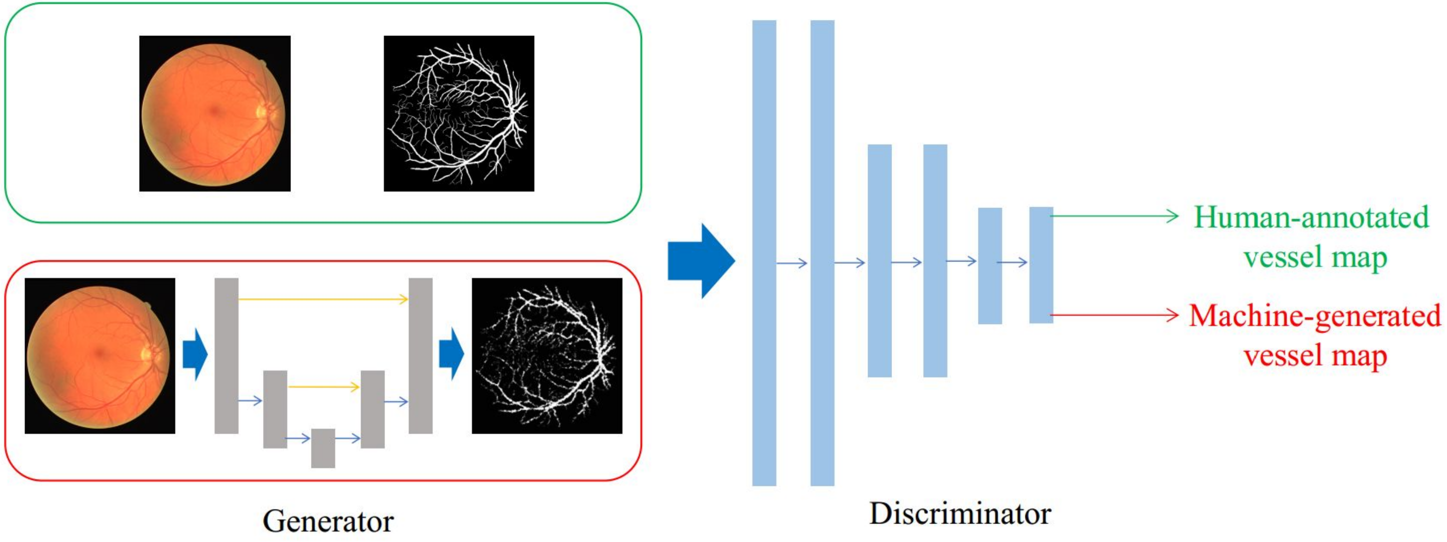
Generative Adversarial Network

Adversarial Learning

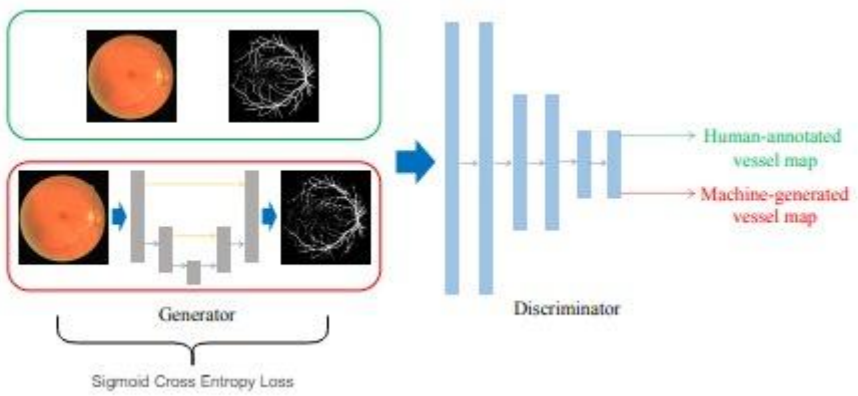


http://www.slideshare.net/xavigiro/deep-learning-for-computer-vision-generative-models-and-adversarial-training-upc-2016

Baseline


Human-annotated
vessel map
Machine-generated
vessel map
Generator
Discriminator

Loss Function

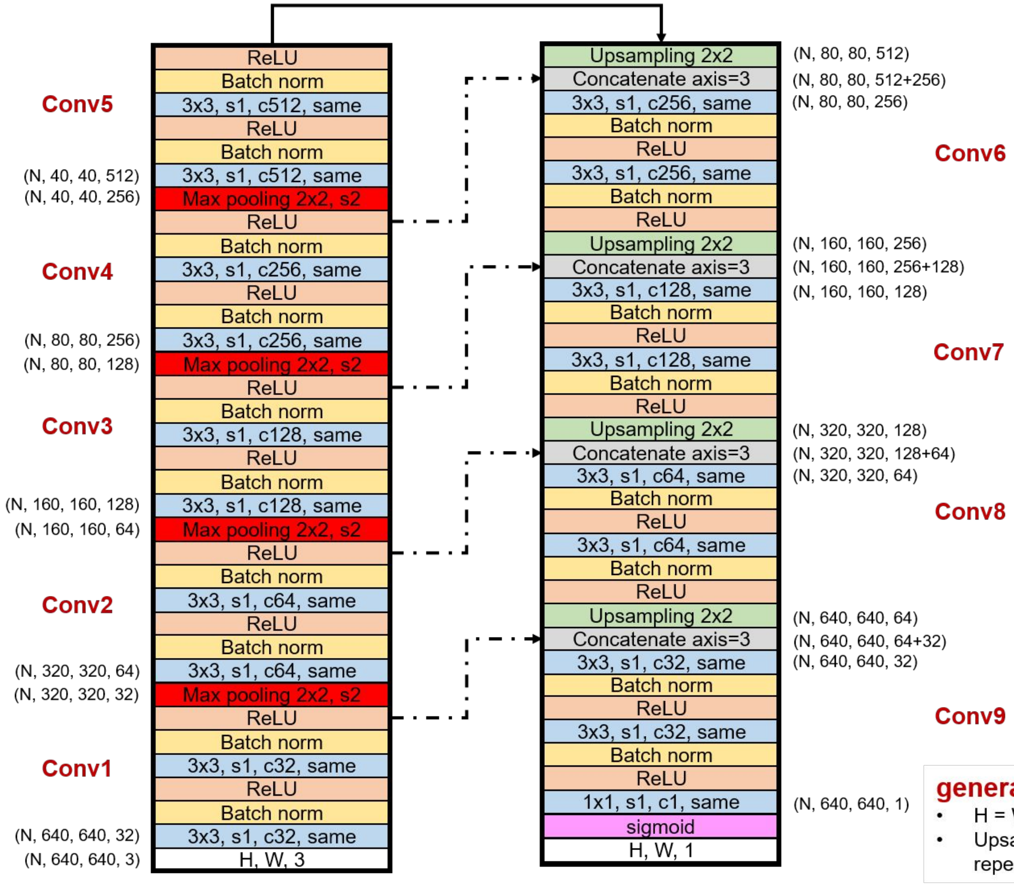


$$G^* = \arg\min_G \left[\max_D \mathbb{E}_{x,y\sim p_{data}(x,y)}[\log D(x,y)] + \mathbb{E}_{x\sim p_{data}(x)}[\log(1 - D(x, G(x)))]\right]$$

$$L_{SEG}(G) = \mathbb{E}_{x,y\sim p_{data}(x,y)} - y \cdot \log G(x) - (1-y)\cdot \log(1 - G(x))$$

$$G^* = \arg\min_G \left[\max_D L_{GAN}(G,D)\right] + \lambda L_{SEG}(G)$$

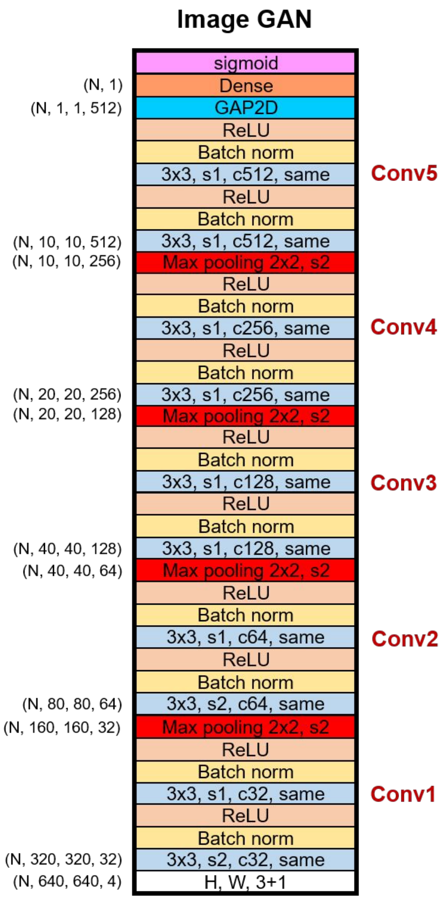
Architecture



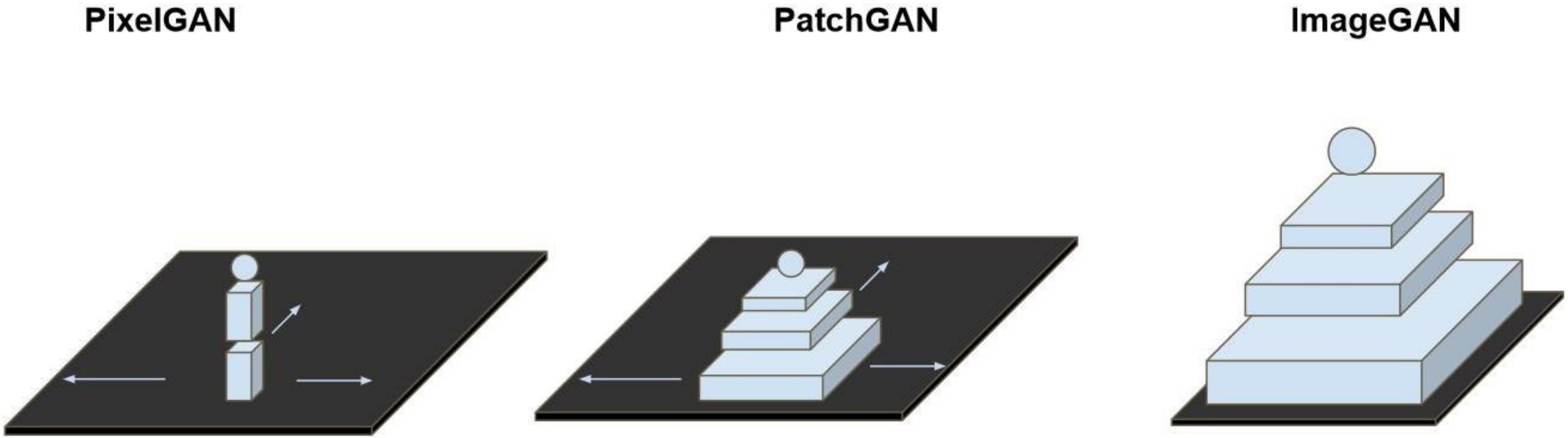
generator

- H = W = 640 or 720
- Upsampling in Keras: repeat rows and cols

Discriminator



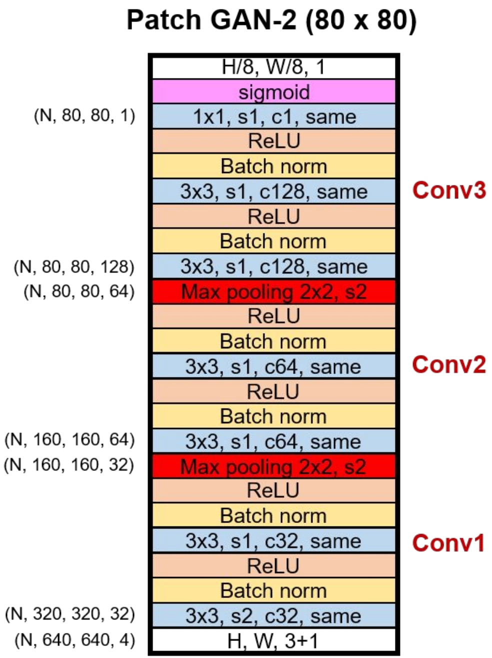
Types of Discriminator



Weighted Segmentation Loss with CGAN

1. Divide Images into Patches
2. For every patch
    a. Obtain Discriminator’s output for the patch
    b. Calculate Segmentation loss

Segmentation Loss = (1 - D(x))*SigmoidCrossEntropyLoss



Theoretical Analysis

$$L^o = E_{x \sim p_{data}(x)}[logD(x)] \tag{1}$$

$$W = 1 - D(x) \tag{2}$$

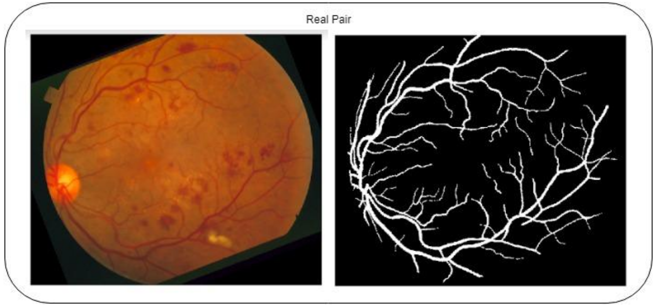
$$L^n = E_{x \sim p_{data}(x)}[(1 - D(x)) * logD(x)] \tag{3}$$

$$L^n = E_{x \sim p_{data}(x)}[logD(x)] - E_{x \sim p_{data}(x)}[D(x)logD(x)] \tag{4}$$

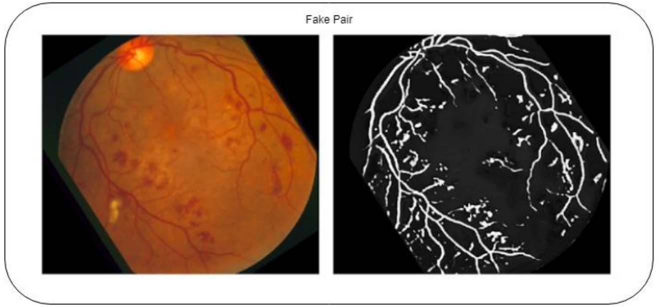
$$L^n = L^o + Entropy \tag{5}$$

$$L^n \geq L^o \tag{6}$$

Negative Sample CGAN



Real Image, Real Output



Real Image, Generated Output

Real Image, False Output

Negative Sample CGAN

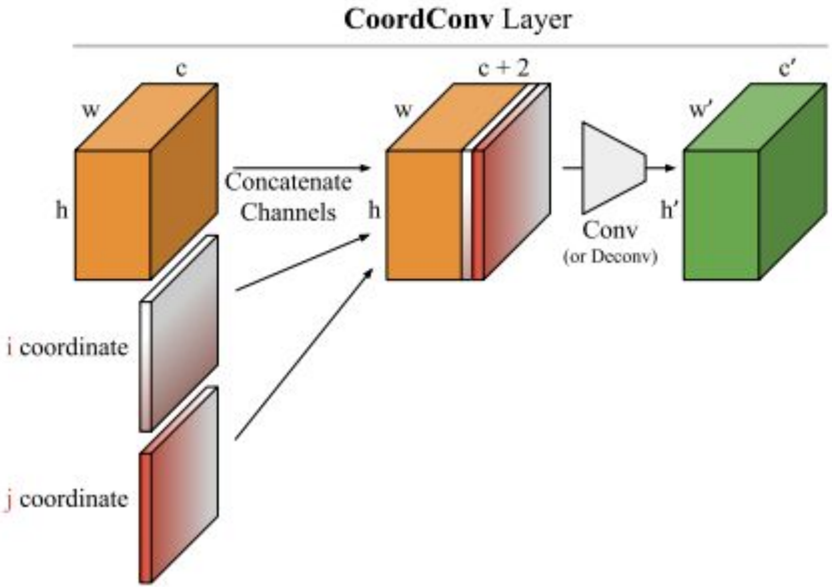
$$G^* = \arg\min_G \left[\max_D \mathbb{E}_{x,y\sim p_{data}(x,y)}[\log D(x,y)] + \mathbb{E}_{x\sim p_{data}(x)}[\log(1-D(x,G(x)))]\right]$$
$$+\left[\arg\max_D E_{x,y'\sim p_{data}(x,y')}[logD(x,y')]\right]$$

$$L_{SEG}(G) = \mathbb{E}_{x,y\sim p_{data}(x,y)} - y \cdot \log G(x) - (1-y) \cdot \log(1-G(x))$$

$$G^* = \arg\min_G \left[\max_D L_{GAN}(G,D)\right] + \lambda L_{SEG}(G)$$

CoConv

An Intriguing Failing of Convolutional Neural Networks and the CoordConv Solution

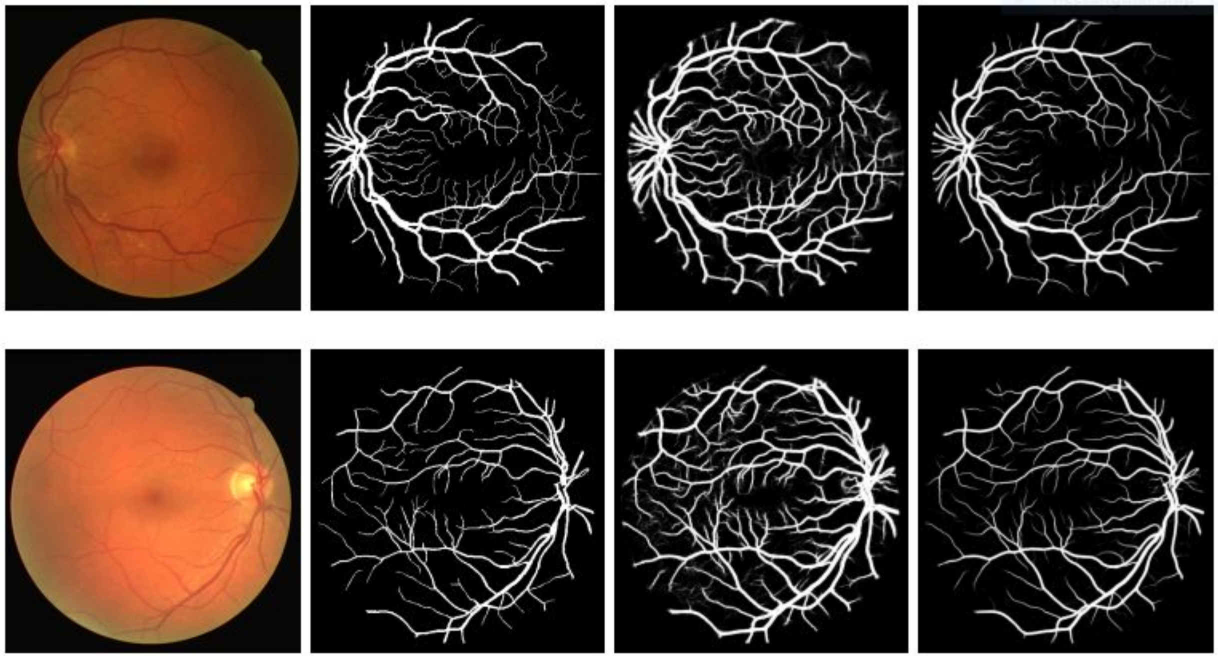


Ref: https://scholar.google.com/scholar?cluster=17251371047104529608&hl=en&as_sdt=0,5&sciodt=0,5

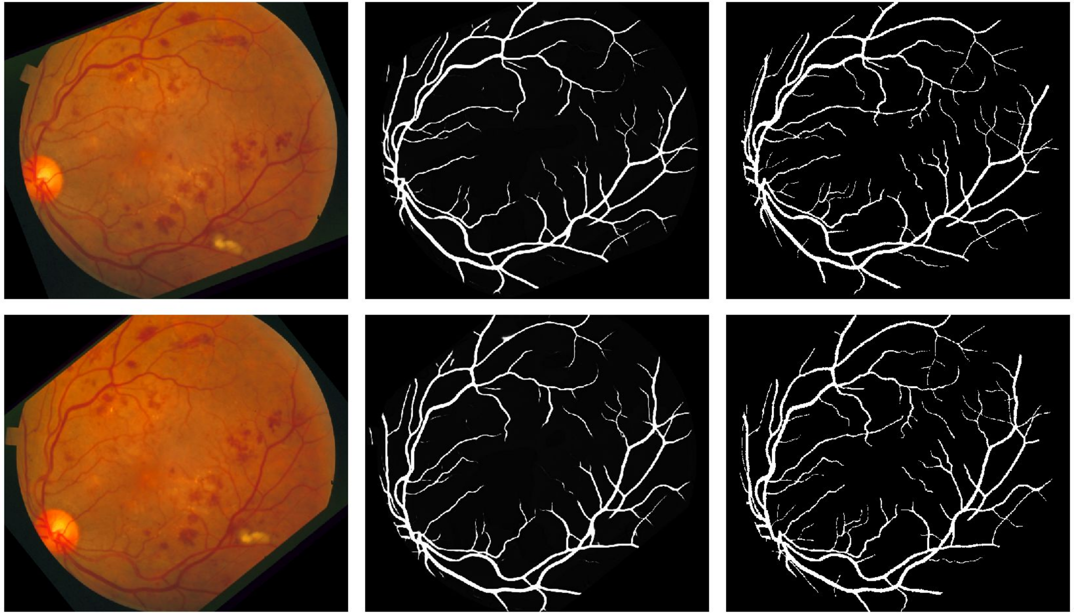
Results

	Baseline	NSGAN	Sample Selection	COCONV
ITERATION	49500	41500	**1500**	**38000**
AUC_PR	0.806	0.757	**0.901**	**0.791**
AUC_ROC	0.912	0.931	**0.967**	**0.945**
DICE_COEFF	0.765	0.745	**0.826**	**0.728**
ACC	0.948	0.943	**0.964**	**0.935**
SENSITIVITY	0.764	0.76	**0.77363**	**0.793**
SPECIFICITY	0.971	0.965	**0.987**	**0.952**
SCORE	5.168	5.104	**5.442**	**5.147**
AUC_SUM	1.718	1.689	**1.868**	**1.736**
AVG_PT	62.679	49.4711	**76.235**	**52.346**

Samples (Baseline)



Samples(WSGAN)



Samples(CoConv)