Coronary Arteries modeling and classification using DCGAN network

Sara Elgayar
Goergen Institute for Data Science
Rochester, NY 14627
selgayar@ur.rochester.edu

Maegan Fish
University of Rochester
Rochester, NY 14627
mfish6@u.rochester.edu

Abstract

The reconstruction of coronary arteries is essential for the localization and diagnosis of blockage in the arteries. One solution is to perform reconstruction directly from coronary angiography images. However, this is a very difficult computer vision task due to the complex shape of coronary blood vessels, and the lack of data set and labels. Recently, more work has been done to reconstruct models of human organs from medical images using deep neural networks. We propose an adversarial and generative model using DCGAN network to reconstruct and classify the coronary artery models. We obtained classification accuracy of 0.83 from the artificial images compared to accuracy of 0.86 from classification of the real images.

1. Introduction

There are many ways in which computer vision has helped humans see beyond their capabilities. One such field is modeling and reconstructing 3D objects based off of 2D images. The medical field is now trying to use these type of methods in order to help doctors 'see' within the body when performing operations.

Cardiovascular diseases are one of the major death threats to human beings. The doctors who perform the necessary operations, need many years of practical experience. In order to help the doctors, computer vision scientists teamed up with medical professionals to create a three-dimensional(3D) model of cardiovascular blood vessels prior to or during the surgeries and thus guide them during complex interventional surgeries. Prior studies have shown that these types of methods help to greatly reduce difficulties of cardiovascular interventional surgeries. In addition, creating these models with help novice doctors to train without clinical experiences.

The 3D reconstruction of arteries from coronary computed tomography angiography (CCTA) is often used in many CT machines however the reconstructions are affected by many things that reduce the sensitivity of detecting and assessing things during surgeries. The types of images needed for reconstruction are shown in figure 1. Due to the reduced sensitivity, CCTA is not often used interventional surgeries. To solve this problem, computer vision scientists have used stereo vision techniques to reconstruct 3D models from multiple 2D images of different view points. With the rise of deep learning in computer vision, scientists believe that it will be able to solve 3D reconstruction with less view points or one view point. The first step in 3D reconstruction is to reconstruct the objects in 2D and verify the accuracy of the images output from the network. In this paper we will discuss our work on this step.

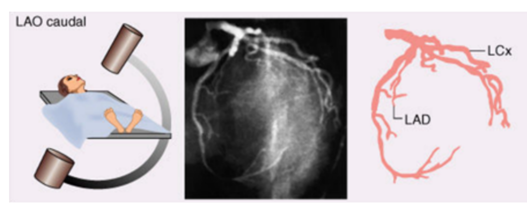

Figure 1. Coronary angiogram takes a series of images (angiograms), offering a look at blood arteries

2. Related Work

The 3D reconstruction of coronary artery, from angiographic image sequences, is an active research area. In Cath Lab, physicians can use only few view angles, because of the patient safety. The goal is to assist interventional cardiology physicians to visualize and localize the diagnosis of stenosis in the artery [2]. Significant efforts have been done in this topic. In this paper [2], they follow a weakly supervised 3d construction from 2 views using Generative Adversarial Networks GAN and WGAN. The objective of this work is to reconstruct a 3d model of the coronary artery using CNNs based on visual codewords. This process requires two steps. First, a bag of visual words is used to extract relevant features. Then the CNNs are used to reconstruct a 3d model. There are multiple challenges, such as the dataset availability, dataset labeling and precisely modeling the complicated details of the arteries.

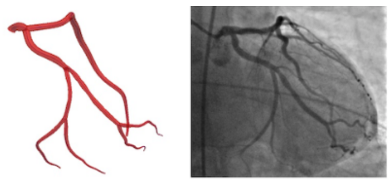

Figure 2. Sample output from previous work [6]. On the left, 3d reconstructed model. On the right, the actual 3d image from the angiogram.

3. Methods

In this paper, we implemented a supervised 2D reconstruction method by generative-adversarial deep learning. First, the raw images are processed, then fed into DCGAN network to model 2D angiography images. Finally, deep learning classifier is used to measure the accuracy of the generated images. The pipeline of the proposed framework is shown in figure 3. The figure shows a semi-supervised algorithm that uses artificial data generated by a DCGAN to improve image classification. In the following sections, we will explain more details about each process of the proposed framework.

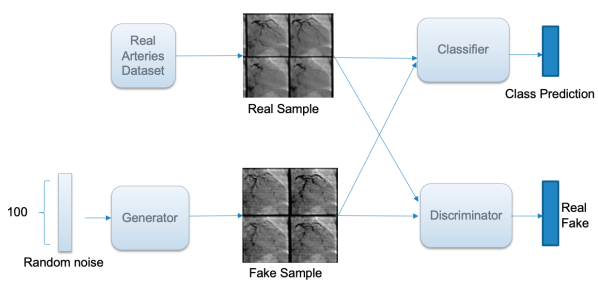

Figure 3. A framework of the DCGAN, and External Classifier DCGAN, Method.

3.1. Data Preprocessing

One of the biggest challenges of this project is the data set, since the training set is small. The data set used in this project is generated and analyzed by [6]. This data is publicly available and it consists of only 30 images. It corresponds to the study: ”Reconstruction and structured meshing of coronary arteries from X-ray angiography” [6]. The input images are grey images in (JPEG) formats of size 512x512 as shown in figure 4.

In order to get better results the training raw images are preprocessed. First, the images are resized from 512x512 to 64x64, this size is easier to be processed by the GAN network. Then, a mask shown in figure 5 is created by thresholding the training images. This is mask is used to segment the arteries. The preprocessed images as shown in figure 6.

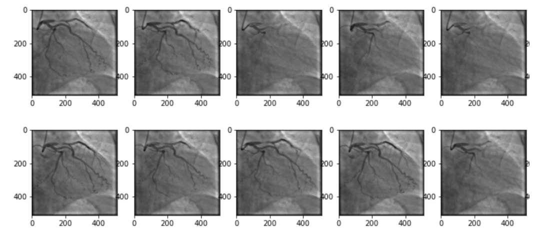

Figure 4. Raw sample from the training dataset images [6].

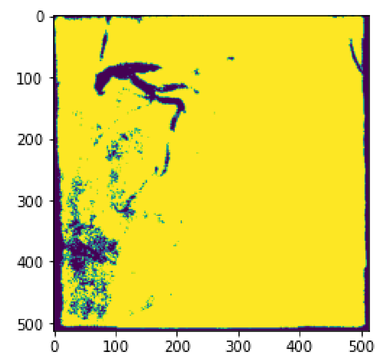

Figure 5. A mask of the arteries is created from thresholding the grey scale images.

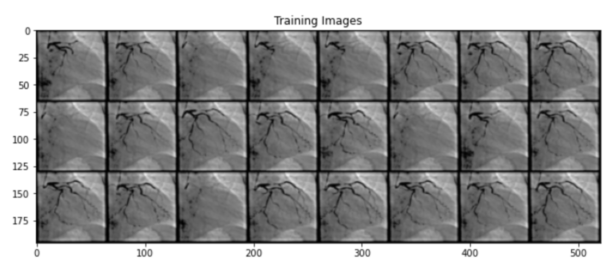

Figure 6. Preprocessed images.

3.2. GAN

The vanilla GAN developed by Ian Goodfellow [1]. The main goal of a GAN network is to train two competitive neural networks. The goal of a generative network is to generate images resembling real training samples that look so real that even an observer finds it hard to distinguish them from real images. In parallel, a discriminative network predicts the probability that a generated image is from the real training set. It is a contest between the two networks. During training, the generator is updated on predictions of the discriminator to fool the discriminator and produce better images, and the discriminator is trying to be more skilled at discriminating images as real or fake. The goal is to have the two networks achieve balance, at which point the generator is creating almost perfect images and the discriminator is left with a 50% chance of discriminating correctly [1].

3.2.1 GAN loss function

In this section, we are going to define some notation to explain the GAN network loss function. First, for the discriminator network: Let x represents an image. D(x) is the discriminator network that outputs probability that x is real image rather than generated by the generator network. The input to the Discriminator D(x) network is an image of size 1x64x64. Therefore, D(x) should be high when x is a real image and low when x is generated by the generator [1]. Second, for the generator's network, the input z is a fixed-length vector. This fix-length vector is selected randomly from a Gaussian distribution is used to seed the generative process.

The goal of Generator is to learn the distribution of the training data set that comes from (Pdata). By doing so G can generate fake samples from that estimated distribution (Pg) [5].

Hence, D(G(z)) is a scalar that represents the probability that the output of the generator G is an actual image. As mentioned in [1], the situation between D and G is a min-max problem. To illustrate, D tries to maximize the probability it correctly classifies reals and fakes (logD(x)), and G tries to minimize the probability that D will predict its outputs are fake log(1-D(G(z))). From the paper, the GAN loss function is stated as:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}\left[logD(x)\right] + \mathbb{E}_{z\sim p_z(z)}\left[log(1-D(G(z)))\right]$$

Theoretically speaking, the solution to the minimax problem is when Pg = Pdata and the discriminator guesses if the inputs are real or fake. However, in reality, the models do not always converge.

3.3. DCGAN

A DCGAN network was first described by Radford et. al [4]. It is a GAN network as described above with some guidelines. In particular: (1) Replacing any pooling layers with strided convolutions for the discriminator network and fractional-strided convolutions for the generator network. (2) Using batchnorm in both the generator and the discriminator. (3) Removing fully connected hidden layers for deeper architectures. (4) Using ReLU activation in generator for all layers except for the output, which uses tanh. (5) Using LeakyReLU activation in the discriminator for all layer. (6) Fractionally strided convolutions, or deconvolutions, transpose images, usually from a minimized format to a larger one. To transpose an image that has been reduced to a 2x2 pixel format up to a larger format, a fractionally strided convolution reconstructs the image's resolution, then performs the convolution. (7) Batch normalization to standardize the activations from a prior layer to have a zero mean and unit variance. This has the effect of stabilizing the training process.

An image of the generator from the DCGAN paper [4] is shown in figure 7.

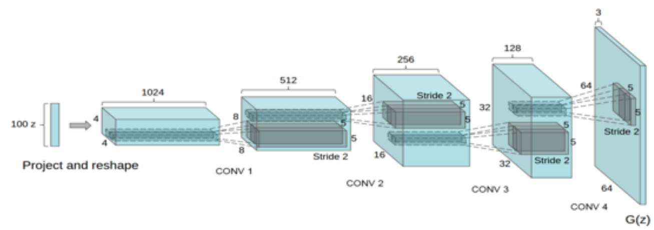

Figure 7. The generator of the DCGAN network consists of convolutional-transpose layers, batch norm layers, and ReLU activations. The input is a latent vector, zz, that is drawn from a standard normal distribution and the output is a 3x64x64 RGB image as described in [4]

3.4. Training of GAN network

The training process of a Generative Adversarial Network is not the same as any other deep neural network. For instance, in GAN, both the discriminator and generator are trained together in an adversarial manner.

First, we start training with the discriminator network. In the meanwhile, the generator network remains idle. This means that the generator has passed forward propagation but not back-propagation yet. At this point, the generator generates a batch of fake samples from a random distribution together with actual samples from the training data set given to the discriminator for classification as real or fake. Now, after training the discriminator neural network, we start the training of the generator network. Similarly, while the generator network is in the training process, the discriminator neural network remains idle. Now when the discriminator network makes predictions on the generated samples, we use these predictions to train the generator network. The training process in shown in figure 8.

3.5. NN Classification

In order to determine the accuracy of the generated images created by the DCGAN network, we wanted to find the classification accuracy, between blockage or no blockage in the arteries, on the generated images. To do this we used 30 generated images and 30 corresponding images from the original data-set. The images from the original data-set were used to train the network for 100 epochs. Then the

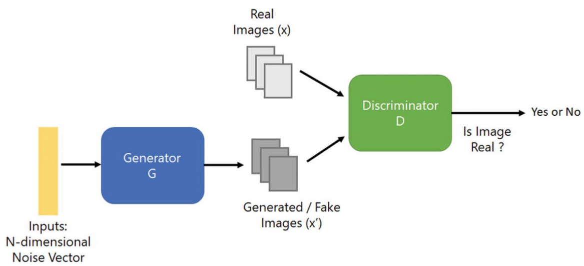

Figure 8. Training of GAN network. Image credit [3].

generated images were input into the model and evaluated. The classification stage is shown in the framework in figure 3. The neural network was composed of 3 blocks of a convolutional layer with Leaky ReLU activation, 2D Max Pooling, and Batch Normalization. After those 3 blocks a 2D Global Max Pooling layer is used followed by 4 fully connected layers with Leaky ReLU activations. A visualization of the basic blocks of the network is shown in 9. The network uses pre-made optimizer and loss functions from keras; Adam Optimizer with learning rate 0.0003 and Binary Cross Entropy Loss.

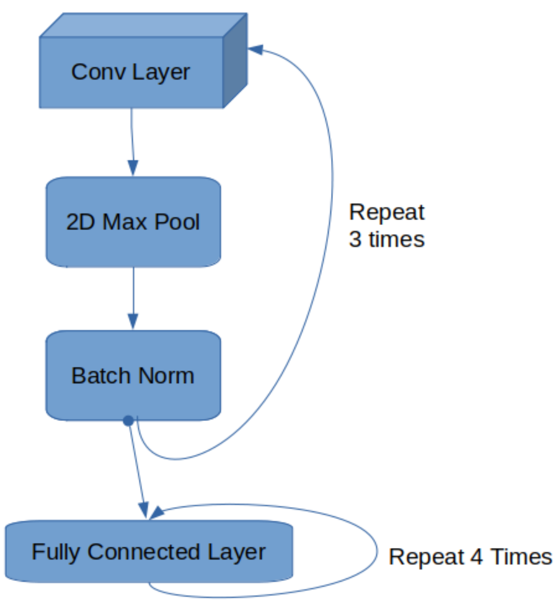

Figure 9. Neural Network Classifier for Determining Blockage or No Blockage of Arteries

4. Experiments

Our input parameters are set as follows: image channels = 1, noise vector size = 100. Also, the dataset is preprocessed and ready for training. According to the DCGAN paper [4], it is mentioned that all weights should be randomly initialized from a normal distribution with mean=0, std=0.02. The generator and discriminator networks are initialized as shown in figures 10 and 11 respectively. The input size of the generator network is the latent space (100, 1, 1) and the output is a grey image with the same size as the training images (i.e. 1x64x64). On the other hand, the discriminator network takes a 1x64x64 input image and outputs the final probability of size 1x1x1 as shown in figure 11.The network is trained for 250 epochs. The proposed framework is shown in figure 3.

Layer (type)	Output Shape	Param #
ConvTranspose2d-1	[-1, 1024, 4, 4]	1,638,400
BatchNorm2d-2	[-1, 1024, 4, 4]	2,048
ReLU-3	[-1, 1024, 4, 4]	0
ConvTranspose2d-4	[-1, 512, 8, 8]	8,388,608
BatchNorm2d-5	[-1, 512, 8, 8]	1,024
ReLU-6	[-1, 512, 8, 8]	0
ConvTranspose2d-7	[-1, 256, 16, 16]	2,097,152
BatchNorm2d-8	[-1, 256, 16, 16]	512
ReLU-9	[-1, 256, 16, 16]	0
ConvTranspose2d-10	[-1, 128, 32, 32]	524,288
BatchNorm2d-11	[-1, 128, 32, 32]	256
ReLU-12	[-1, 128, 32, 32]	0
ConvTranspose2d-13	[-1, 1, 64, 64]	2,049
Tanh-14	[-1, 1, 64, 64]	0

Figure 10. The generator architecture from our implementation. input size = (100, 1, 1) and batch size is set to -1 meaning any batch size can be provided.

```
1 summary(disc, input_size = (1, 64, 64))
```

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 64, 32, 32]	1,088
LeakyReLU-2	[-1, 64, 32, 32]	0
Conv2d-3	[-1, 128, 16, 16]	131,072
BatchNorm2d-4	[-1, 128, 16, 16]	256
LeakyReLU-5	[-1, 128, 16, 16]	0
Conv2d-6	[-1, 256, 8, 8]	524,288
BatchNorm2d-7	[-1, 256, 8, 8]	512
LeakyReLU-8	[-1, 256, 8, 8]	0
Conv2d-9	[-1, 512, 4, 4]	2,097,152
BatchNorm2d-10	[-1, 512, 4, 4]	1,024
LeakyReLU-11	[-1, 512, 4, 4]	0
Conv2d-12	[-1, 1, 1, 1]	8,193
Sigmoid-13	[-1, 1, 1, 1]	0

Figure 11. The discriminator architecture from our implementation. input size = (1, 64, 64) and batch size is set to -1 meaning any batch size can be provided.

4.1. Results

Despite extensive progress during the past decades, no procedure for measuring accuracy of GAN reconstructions from angiogram images has been standardized so far. Instead, a few indicators were frequently used including the measurement of difference between ground truth. In this project, some statistics are reported at the end of each epoch to track the progress of training. These statistics include:

- Loss_D - the generator network loss shown in figure 12

- Loss_G - the discriminator network loss shown in figure 13
- D(x) - the average output of the discriminator for the all real batch.
- D(G(z)) - average discriminator outputs for the all fake batch.

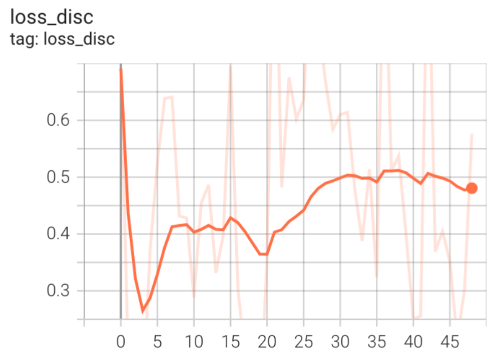

Figure 12. The generator loss function during training.

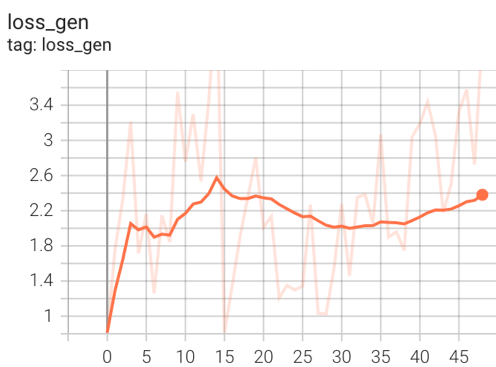

Figure 13. The discriminator loss function during training.

The below plots shown in figures 12 and 13 represent the loss functions of the generator and the discriminator respectively. It can be seen that the two functions are correlated. The generator begins to improve (approximately after 40 epochs). At the same time, the task of the discriminator becomes harder and its performance deteriorates. This is a good sign that the training scheme is working.

Finally, lets take a look at some real images and fake images side by side 14. Visually, it can be seen that the generated artificial images are highly varied and high-quality dataset of coronary arteries.

We then use generated images as inputs for the classification. This is the semi-supervised portion of our algorithm, as the generated images do not have associated labels. To create labels, we use hand-crafted labels based on the most

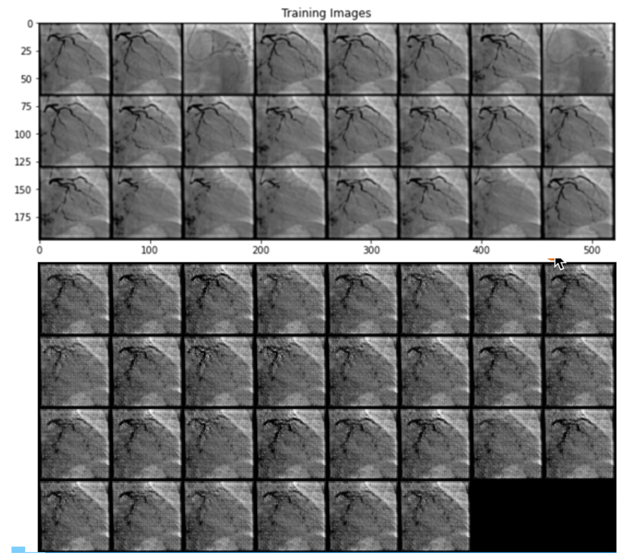

Figure 14. Real images (top) and fake images (bottom), generated by the G network.

likely class according to the training dataset. Regarding the most important results, the classification accuracy of real images was compared to the generated ones. Table 1 contains a summary of the classification results. The accuracy of the generated images show that it is approximately the same as the accuracy of the real images. Traditionally, if a data sample lacks a corresponding label and the dataset only contains a small amount of images, a model cannot learn from it. The proposed framework addresses this problem. It can increase the size of the dataset by generating artificial data, and the results show that the accuracy of both data sets are approximately the same.

Model	Accuracy
Real images	86.67 %
Generated images	83.33 %

Table 1. Classification accuracy's of the real and generated images.

5. Conclusion

GANs have shown tremendous potential in recent years and have been applied in various scenarios, ranging from image synthesis to enhancing the quality of images, image to image translations, text to image generation, and more. In this project, we reviewed a generative model using DCGAN network. Followed by using an external classifier to improve classification performance on unbalanced small datasets. The results show promising performance of the proposed framework for real application to medical images.

References

[1] Vincent Dumoulin-Kai Arulkumaran Biswa Sengupta Antonia Creswell, Tom White and Anil A Bharath. ”generative adversarial networks: An overview.”, 2018. 2, 3

[2] Xiao-lei Yin Jing Qiu Zhi-yun Yang Jun-hui Xing Jianzeng Dong Lu Wang, Dong-xue Liang and Zhao yuan Ma. Weakly-supervised 3d coronary artery reconstruction from two-view angiographic images, 2020. 1

[3] Nayak Manish. Deep convolutional generative adversarial networks(dcgans), NOV 2018. 4

[4] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2015. 3, 4

[5] Joseph Rocca. Understanding generative adversarial networks (gans), Mar 2021. 3

[6] Arso M. Vukicevic and Serk Çimen. Three-dimensional reconstruction and nurbs-based structured meshing of coronary arteries from the conventional x-ray angiography projection images., 2018. 2