Using GANs to generate photo-realistic fundus images manifestating diabetic retinopathy.

Rushikesh Pedgaonkar (1501CS32), Raghav Jindal (1501CS36), Dhawal Gupta (1501CS54)

Project abstract

Creating a new data set requires the presence of a large amount of skilled manpower, time and money. Generative Adversarial Network (GAN) can help mitigate this problem as it can be used to generate new data from random noise and given small data set. In this project, we shall use GAN to generate a data set of images of fundus manifesting various diseases. The data set thus generated will be immensely useful to classify the diseases present in fundus at a later stage.

1 Introduction

In this new trend of deep learning, data has gained a lot of importance. In deep learning, a large amount of training data is required to get satisfactory results. However, for many real world problems, the data is not readily available. Creating and gathering this data requires a large amount of manual labour and time. Many a times, a professional in the field of the problem is required to gather proper data. Hence this costs a lot of money. The Generative Adversarial Networks (GANs) devised by Goodfellow et al [1] in 2014, aims to mitigate this issue. GANs can help to generate new data taking only random noise and small amount of data as input.

Diabetic retinopathy is a diabetes complication that affects eyes. It's caused by damage to the blood vessels of the light-sensitive tissue at the back of the eye (retina). Diabetic retinopathy is the leading cause of the blindness in the working age population. If the disease is detected early and treated promptly, much of the visual loss can be prevented. Fundus images are the images of the interior surface of the eye opposite the lens and includes retina, optic disc etc. We can detect diabetic retinopathy from these fundus images. In this project, we are creating photo-realistic fundus images manifesting diabetic retinopathy. There are subtle differences in the fundus images of an eye of normal person and a person having some disease. So, by testing an image of fundus we should be able to diagnose whether a person has that disease by applying image classification. Since the difference in the fundus images of a normal person vs in the case of diabetic retinopathy is very subtle, we require a large amount of fundus images manifesting diabetic retinopathy. This is done using GANs.

2 Literature survey

GANs have been used heavily since there introduction in 2014. They have been used to generate photo-realistic images of people, anime characters [2], colorize black and white photo [3], generate a photo of a person in a different posture given a posture and the person's image as input [4], generate super resolution images from low resolution ones [5]. Thus we can easily observe that GANs have tremendous potential and applications in various areas.

Detecting diabetic retinopathy from fundus images has been going on since early 2000s. [6] proposed a method to detect lesions in retinal images which can assist in early diagnosis and screening of diabetic retinopathy. But the dataset used consisted of only about 500 images. Similarly, [7] proposed a method to extract features from fundus for diabetic retinopathy.

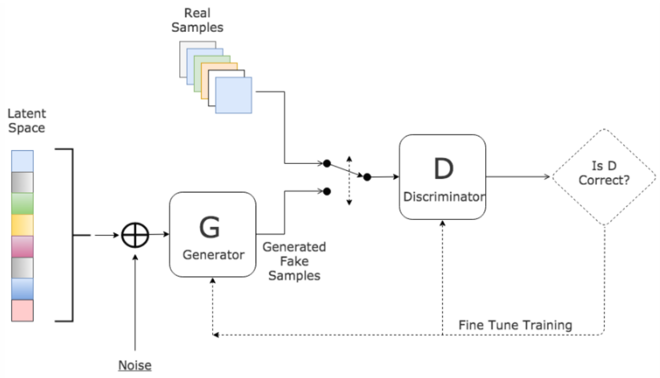

Figure 1: Generative Adversarial Network

3 Contribution

We have developed a model which can generate new realistic data on its own. This will be immensely useful in generating new fundus images with diabetic retinopathy manifestation. The dataset generated from our model can be used in the future to help determine if a person suffers from diabetic retinopathy using an automated system with a high accuracy. This can significantly reduce the workload on the already overburdened doctors.

4 Theory

GANs belong to a set of algorithms called generative models, which are widely used for unsupervised learning tasks which aim to learn the underlying structure of the given data. As the name suggests, GANs allow you to generate new unseen data that mimic the actual given real data. However, GANs pose problems in training and require carefully tuned hyperparameters.

Generative Adversarial Networks can be broken down into two parts :

- **Generator:** Generates artificial data from random noise.
- **Discriminator:** This model takes in data as input. This data can be real world data or the one generated from the generator. This models goal is to recognize if an input data is real - belongs to the original data set or if it is fake - generated by a generator.

Figure 1 shows a basic GAN. The interaction between the generator and discriminator can be thought of the generator as having an adversary, the discriminator. The Generator needs to learn how to create data in such a way that the Discriminator is not able to distinguish it as being fake anymore. The competition between these two teams is what improves their knowledge, until the Generator succeeds in creating realistic data.

4.1 Mathematical modeling of GAN

Let's say a neural network $G(z, W_1)$ is used to model the generator. It's task is to map input noise samples z to the desired data space x (images). Another, Second neural network $D(x, W_2)$ represents the discriminator and outputs the probability that the data came from the real data set, in the range (0,1). In both cases W_i represents the weights and parameters that define each neural network.

As a result, the Discriminator is trained to correctly classify the input data as either real or fake. This means its weights are updated as to maximize the probability that any real data input x is classified as belonging to the real dataset, while minimizing the probability that any fake image is classified as belonging to the real dataset. In more technical terms, the loss/error function used maximizes the function $D(x)$, and it also minimizes $D(G(z))$.

Furthermore, the Generator is trained to fool the Discriminator by generating data as realistic as possible, which means that the Generators weights are optimized to maximize the probability that any fake image is classified as belonging to the real data set. Formally this means that the loss/error function used for this network maximizes $D(G(z))$.

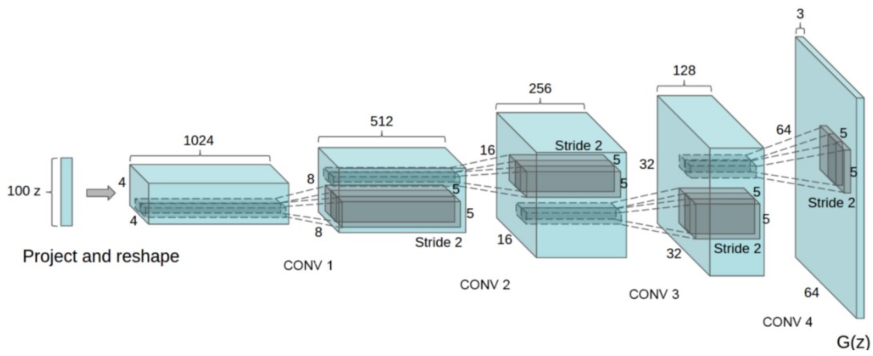

Figure 2: Deep Convolutional Generative Adversarial Networks

After several steps of training, the Generator and Discriminator have enough capacity and they will reach a point at which both cannot improve anymore. At this point, the generator generates realistic synthetic data, and the discriminator is unable to differentiate between the two types of input.

Since during training both the Discriminator and Generator are trying to optimize opposite loss functions, they can be thought of two agents playing a minimax game with value function $V(G,D)$. In this minimax game, the generator is trying to maximize its probability of having its outputs recognized as real, while the discriminator is trying to minimize this same value.

$$\min_{G}\max_{D} \quad V(D,G) = \quad \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z))] \tag{1}$$

4.2 Deep Convolution GAN (DCGAN)

Deep Convolution GAN (DCGAN) is one of the most popular and successful network design for GAN. It was proposed by [8] in 2015. The figure 2 shows the diagram for the generator. Summary of DCGAN is as follows:

- Replace all max pooling with convolutional stride
- Use transposed convolution for upsampling.
- Eliminate fully connected layers.
- Use Batch normalization except the output layer for the generator and the input layer of the discriminator.
- Use ReLU in the generator except for the output which uses tanh.
- Use LeakyReLU in the discriminator.

5 Method

To generate the data, we need a small amount of real world data. We use a small dataset found on kaggle [1] to get the real world dataset. This dataset consists of about 700 fundus images with diabetic retinopathy manifestation.

To generate the images, we used the DCGAN model described in the previous section. The kaggle dataset is used as the real data for the discriminator. Binary categorical cross entropy was used as a loss function for training the model. Adam [9] optimizer was used in place of Stochastic Gradient Descent for both the generator and the discriminator with the learning rate = 0.0002 and $\beta_1 = 0.5$ and $\beta_2 = 0.999$. Due to resource constraint, we feed in 360 x 360 images as input and our generator generates the same size output. Pytorch framework [10] was used for coding the model. The code for our model can be found in the **github repository**[2]

[1] https://www.kaggle.com/c/diabetic-retinopathy-detection/data

[2] https://www.kaggle.com/c/diabetic-retinopathy-detection/data

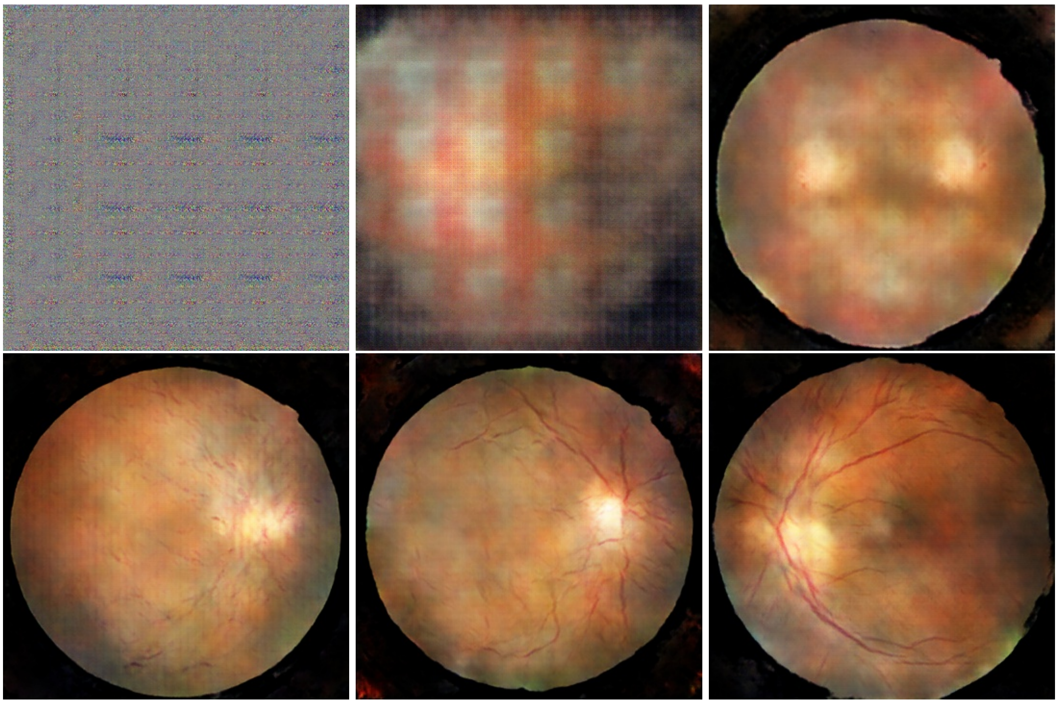

Figure 3: Gradual Improvement in Images

6 Results

In this section we present the results of our experiments. We experiments with different image sizes ranging from $64x64$ pixels to $720x720$. The images available with us were of varying size and we downsamlped them to meet our requriement. We found that 720p images took very long time to be trained particularly because of higher requirement of GPU memory which had forced us to keep small batch sizes at the same time requiring very large number of epochs. 64p images are too small to present any detail, hence after extensive experimentation we decided to go with 360p image. The results for the image generated over different epochs is present in the figure 3. These images tend to show incremental improvement in the details like nerves and almost circular white optic disc and the formation of macula in the last image, with perfect formation of a circle. For e.g. we can see that the model learns to generate a single optical disc after the 3rd image.

We also found some images produced that though were were not perfect shaped and perfectly filled but had greater amount of detail for example the image present in figure 4.

Comparing our images to the disease manifested images shows us that inconsistencies in the colouring which show's possible signs of diabetes. Again due to constrained resolution one may not be able to perfectly make out the disease. We would say that our images are able to replicate the disease level upto 2-3 level. Due to unavailability of a classification model or an expert, we are not able to give the quantitative analysis on the images produced but they due produce the white irregularities that are present in a diabetes infected fundus.

7 Conclusions

In this project, we have created a GAN based model which can be used to generate new fundus images manifesting diabetic retinopathy. The method shows promising results in generating images that are otherwise present in a scarce quantity. This method will help us to produce larger data-sets for image classification tasks. We found that the problem data-set biases while training a classifier where the unavailability of sufficient quantity of level 4 diabetes hindered the accuracy of training model.

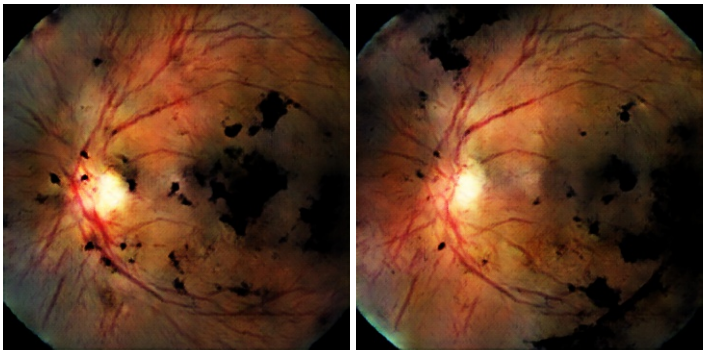

Figure 4: Imperfect images with greater detail

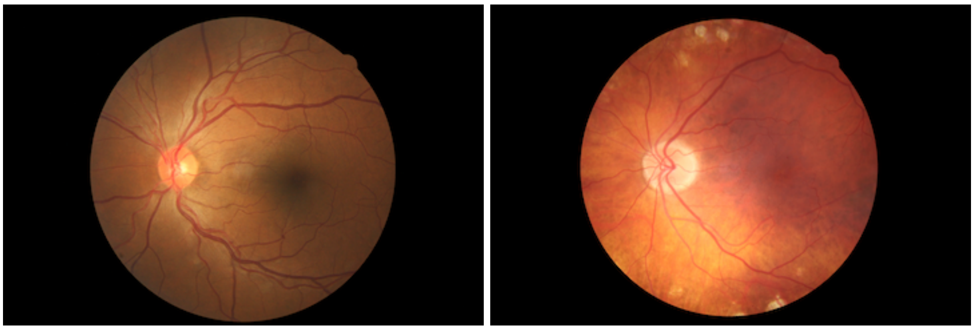

Figure 5: No Diabetes Manifestation (Left) Disease Manifestation (Right) (Real)

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[2] Yanghua Jin, Jiakai Zhang, Minjun Li, Yingtao Tian, Huachun Zhu, and Zhihao Fang. Towards the automatic anime characters creation with generative adversarial networks. *arXiv preprint arXiv:1708.05509*, 2017.

[3] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[4] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Advances in Neural Information Processing Systems*, pages 406–416, 2017.

[5] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[6] Saiprasad Ravishankar, Arpit Jain, and Anurag Mittal. Automated feature extraction for early detection of diabetic retinopathy in fundus images. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 210–217. IEEE, 2009.

[7] Chanjira Sinthanayothin, James F Boyce, Tom H Williamson, Helen L Cook, Evelyn Mensah, Shantanu Lal, and David Usher. Automated detection of diabetic retinopathy on digital fundus images. *Diabetic medicine*, 19(2):105–112, 2002.

[8] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[10] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.