Research Paper Summary/Notes

SinGAN: Learning a Generative Model from a Single Natural Image, Technion & Google.

Laxman Singh Tomar

November 2019

Research Paper Summary/Notes

SinGAN: Learning a Generative Model from a Single Natural Image, Technion & Google.

Laxman Singh Tomar

November 2019

Contents

1 Introduction **1**

1.1 Motivation 2

1.2 Objectives 2

2 Methods **3**

2.1 Model Architecture 3

2.2 Training 4

3 Results **6**

3.1 Implementation Details 6

3.2 Single Image Frechet Inception Distance 6

3.3 Applications 7

4 Discussion **8**

4.1 Conclusions 8

List of Figures **9**

Bibliography **10**

A Official Pytorch Implementation **11**

B SinGAN for Single Image Animation **12**

Dedication

I would like to dedicate this work to teachers whom I've never met in real but their valuable contribution have made this work possible.

Acknowledgement

(I encountered several resources which helped me in getting through this work)

I would like to express my sincere gratitude to:

- Tamar Rott Shaham et. al.
- Ian Goodfellow

Abstract

This paper proposes a novel GAN[1] training technique to obtain a generative model that can be learned using a single image. Unlike some of the previous works that used single image for training for a single task, the model can be used for unconditional generative modeling (i.e. generates samples from noise), and is not limited to texture images.

Keywords: Unconditional Generation; Single Image

Chapter 1

Introduction

Unconditional GANs have shown remarkable success in generating realistic, high quality samples when trained on class specific datasets viz. faces and bedrooms. But they still require training the model on a specific task when it boils down to capturing the datasets with multiple object classes e.g. Imagenet.

Our topic of interest it to propose a training method for generative modeling by using a single training image. Our argument is that internal statistics of patches of an image carry enough information for learning a powerful generative model. It is done by using pyramid of fully convolutional GANs capturing the patch distribution at different scales. In a nutshell, now we don't need to rely on having the requirement of database of images from a class.

Prior Work: While the idea of using single training image has been there for quite some time(mostly being for a specific task like harmonization, style transfer, etc) the proposed procedure takes it one step forward by making the generative model unconditional(generating real-like images sampled from training distribution by passing random noise) and is not limited to texture images.

1.1 Motivation

Existing architectures require a large amount of images for each object class in order to generate realistic and high quality samples.

1.2 Objectives

Introduce architecture that can be used to generate samples using a single natural image.

Chapter 2

Methods

Goal: To learn an Unconditional Generative Model that captures the internal statistics of a single training image x where training samples are patches of a single image rather than image itself.

To capture statistics about complex strutcures in an image hierarchy of patch-gans aka Markovian Discriminators are used.

2.1 Model Architecture

The model follows a pyramid structure with each level being fed a different scale of the training image, with x_n being the downsampled version of the image with factor r^n ($r > 1$). At the lowest (coarse) level, random noise is passed through the generator and the output of this (after upsampling by a factor of r) is passed onto the next level ($N = n - 1$).

The generation from the previous level is residually added to random noise at this level and passed through the generator at this level. The same process keeps on continuing till the level with the original image size is reached. The different levels are responsible for generation of different kind of details in the image, with the lowest level able to generate coarser details and the higher levels able to manipulate more finer details.

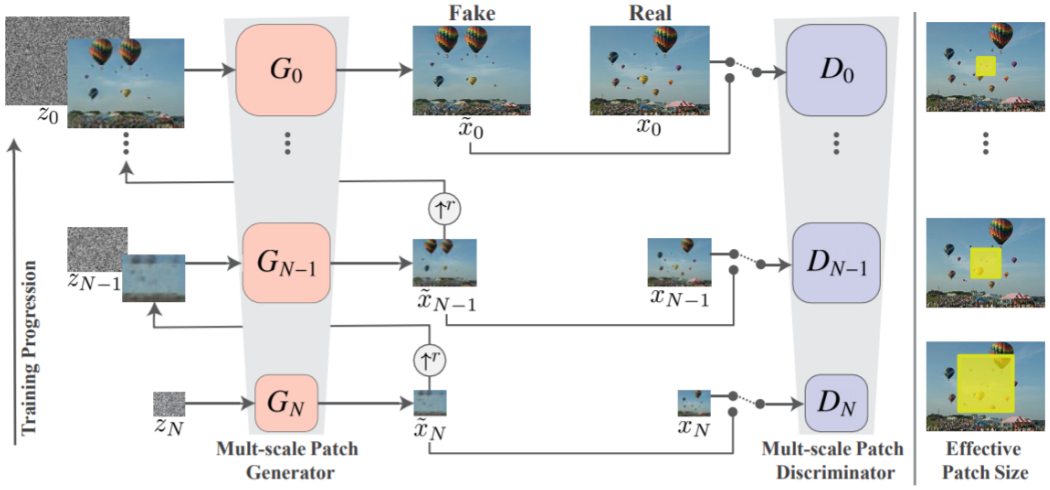

Figure 1: **SinGAN's multi scale pipeline.** Our model consists of a pyramid of GANs, where both training and inference are done in a coarse-to-fine fashion. At each scale, G_n learns to generate image samples in which all the overlapping patches cannot be distinguished from the patches in the down-sampled training image, x_n, by the discriminator D_n; the effective patch size decreases as we go up the pyramid (marked in yellow on the original image for illustration). The input to G_n is a random noise image z_n, and the generated image from the previous scale $\tilde{x_n}$, upsampled to the current resolution (except for the coarsest level which is purely generative). The generation process at level n involves all generators $\{G_N...G_n\}$ and all noise maps $\{z_N, ..., z_n\}$ up to this level.

2.2 Training

The training of the network takes place sequentially from coarser scale to finer scale. The loss terms can be seen as follows:

1. **Adversarial Loss:** WGAN is used for a stabilized training where final discrimination score is the average over the patch discrimination map. Also the loss is defined over the whole image rather than over random crops.

2. **Reconstruction Loss:** Ensures the existence of noise that can be used to generate original training image. So basically simultaneous to what we have been doing (adding noises at various levels as we have seen till now) we try to see whether the initial noise can be used to generate original image. To perform this, we are gonna take

$$\{z_N^{rec}, z_{N-1}^{rec}..., z_0^{rec}\} = \{z^*, 0, ..., 0\}$$

What that means is, we are gonna take zero noise in the later levels. So after this we just calculate the squared difference loss between the generations at each level (with zero noise being assigned to every level except the lowest) and the image at the size scale. This also serves another purpose to determine standard deviation of noise z_n at each scale, as we take the standard deviation to be proportional to RMSE between one-step upsampled version of generated x_{n+1}^{rec} and x_n, giving an indication of amount of information to be added at each scale (which is added as noise).

Chapter 3

Results

3.1 Implementation Details

1. The minimal dimension at the coarsest scale is set up to 25px and the number of scales N is chosen such that the scaling factor is close to $4/3$.

2. The generators are fully convolutional net of the form Conv(3*3)-BatchNorm-LeakyReLU. There are 32 kernels per block at the coarsest scale and the number is increased by a factor of 2 every 4 scales.

Results for various cases (over random sample images, high-resolution image generation, generation from different scales (at inference) and training with different number of scales) with generation being purely unconditional are quite impressive, and also the same network being used for conditional generation task with input of the image to be manipulated being passed onto the later levels rather than the lowest one.

3.2 Single Image Frechet Inception Distance

This is a variant of Frechet Inception Distance (die to the obvious reason being that the training size is just of a single image) which measures the deviation between the distribution of deep features of generated images and that of real images, known as

Single Image Frechet Inception Distance. For this, instead of using the activation vector after the last pooling layer in Inception Network, the internal distribution of deep features at output of convolutional layer just before the second pooling layer is used.

3.3 Applications

To explore how SinGAN performs over a wide variety of tasks, we've kept our model after training and haven't induced any changes. The idea is to utilize the fact that at inference, SinGAN can only produce images with the same patch distribution as the training image. Thus, manipulation can be done by injecting (a possibly downsampled version of) an image into the generation pyramid at some scale $n < N$, and feed forwarding it through the generators so as to match its patch distribution to that of the training image. Different injection scales lead to different effects.

Various applications include:

1. Super Resolution: Increase the resolution of an input image by a factor s.

2. Paint-to-Image: Transfer a clipart into a photo-realistic image.

3. Harmonization: Realistically blend a pasted object with a background image.

4. Editing: Produce a seamless composite in which image regions have been copied and pasted in other locations.

5. Single Image Animation: Create a short video clip with realistic object motion, from a single input image.

Chapter 4

Discussion

4.1 Conclusions

The proposed technique to train unconditional generative model using a single image is undoubtedly quite astounding and there being different level of scales in the architecture just makes it easier to get representations of various patches of the image.

The fact that a single network can be used for various tasks like harmonization, in painting, etc with no explicit training is also great.

However, one of the major drawbacks of this technique (quite visibly) is also its usage of a single training image, which leads to a bottleneck in the variability of the generations produced by the network. The purely unconditional generations being successful for varied representations of the same kind (as in the same class of data) of image, would fail to generate various different classes of data.

List of Figures

1 **SinGAN's multi scale pipeline.** Our model consists of a pyramid of GANs, where both training and inference are done in a coarse-to-fine fashion. At each scale, G_n learns to generate image samples in which all the overlapping patches cannot be distinguished from the patches in the down-sampled training image, x_n, by the discriminator D_n; the effective patch size decreases as we go up the pyramid (marked in yellow on the original image for illustration). The input to G_n is a random noise image z_n, and the generated image from the previous scale $\tilde{x_n}$, upsampled to the current resolution (except for the coarsest level which is purely generative). The generation process at level n involves all generators $\{G_N...G_n\}$ and all noise maps $\{z_N, ..., z_n\}$ up to this level. 4

Bibliography

[1] Goodfellow, I. Generative adversarial nets. *Advances in Neural Information Processing Systems, pages 2672–2680* (2014).

Appendix A

Official Pytorch Implementation

`https://github.com/tamarott/SinGAN`

This is official implementation of the paper along with mentioned applications from author in Pytorch.

Appendix B

SinGAN for Single Image Animation

From author's own Youtube Channel showcasing how SinGAN works on the task of single image animation:

`https://youtu.be/xk8bWLZk4DU`