cSinGAN - Improving Generative Model Using Conditional GAN and Segmented Data

Yunke Zhao, Zhilin Guo, Xuejing Wang
Department of Computer Science
School of Engineering and Applied Science
Columbia University
New York, NY 10027
yz3831/zg2358/xw2668@columbia.edu

March 30, 2020

Abstract

Image generation combined with deep learning has become a hot topic these years. Inspired by the SinGAN developed by Shaham et al., we will implement a new optimized conditional GAN model, named cSinGAN, in this project to explore some more potential solution on image generation, and compare them with current models.

1 Introduction

Unconditional Generative Adversarial Nets (GAN) has seen great success on generating high fidelity natural images. Particularly, training GAN on large data set [1] and employing orthogonal regularization to the generator allows fine tweaking between image quality and image variety, such tasks remains a significant difficulty and very often asks for conditioning image formulation on some specific task or signal.

More recently, Shaham et al. developed a new GAN named SinGAN [2] which trains on a single natural image and uses unconditional generation on the internal statistics and structures of that single image to generate natural images. Without the need for a large database or labeled data, SinGan employs a set of connected fully convolutional GANs that captures the internal structures of an image at each different locations, and produces state-of-the-art natural images as results.

However, one limitation of SinGAN is that it only receives an unsegmented image when generating new images, and can lead to failure cases that can only be resolved by manually fine-tweaking at which scale the generation starts. Hence, we propose to introduce an improvement on SinGAN that employs conditional

GAN (cGAN) that takes in segmented data and generates high-fidelity natural images.

2 Problem formulation and goal

The goal of this project is to create and optimize a cGAN generative network model to learn the inner texture information of the given segmented image. SinGAN network gives us a good head-start on solving this problem. It uses single image to randomly generate training data by applying random image manipulations to the input without destroying the texture features. The model learns texture information and detect objects from the image and generate fake image from the giving input. Based on the SinGAN network, we will explore the possibility of combining a conditional GAN to the SinGAN network.

We propose to use conditional GAN and segmented data to refine the model. We believe that by providing clear object segmentation information to the generator, it will help the model to better clarify different object and texture in the image as well as more efficiently preserve the global structure, and in result generate better output.

3 Dataset, method, and algorithm proposed

3.1 Data

We will be using images taken from the Berkeley Segmentation Database (BSD)[3]. The dataset contains 500 natural images, ground truth human annotations and benchmarking code for segmentation. Since we do not need to train on large amount of input images, BSD would be sufficient enough. The ground truth segmentation provided by BSD may further be used as another feature for model improvement.

The algorithm that we will implement is basically a pyramid of fully convolutional light-weight GANs, each is responsible for capturing the distribution of patches at a different scale. Both training and inference are done in a coarse-to-fine fashion. The effective patch size decreases as we go up the pyramid.

3.2 Formula

In General, We will remain the same structure that is used in SinGNN:

$$\min_{G_n} \max_{D_n} \mathscr{L}_{adv}(G_n, D_n) + \alpha \mathscr{L}_{rec}(G_n)$$

But the G_n and D_n are modified to merge with condition on the input, where $G_n$is a special multi-scale generator that generate the output:

$$x_n = G(z_n, \bar{(}x_{\bar{n}+1}) \uparrow^r, s_n)$$

First two input of G_n are the same with SinGAN, where the third input s_n is the image segmentation of the current scale. We use segmentation of the image as a conditional label. Same for the D_n:

$$y_n = D(\bar{x_n}, s_n)$$

Where y_n and $\bar{x_n}$ are same with SinGAN, and s_n are the image segmentation of the current scale.

3.3 Training Time

SinGAN has an estimated training time of 30 minutes using Nvidia 1080Ti Graphics card, and image generation time of less than 1 minute for each image. We estimate that cSinGAN will have approximately the same training time of 30 minutes and image generation time of less than 1 minute.

4 Evaluation criteria

We will test our method both qualitatively and quantitatively on a variety of images spanning a large range of scenes including urban and nature scenery as well as artistic and texture images. Qualitatively, we'll evaluate whether the generated image successfully preserves global structure of objects, as well as fine texture information.

Quantitatively, we'll use two metrics: Amazon Mechanical Turk (AMT) "Real/Fake" user study, and a single-image version of the Fréchet Inception Distance. For AMT perceptual study, we'll perform perceptual experiments in 2 settings: paired (real vs. fake) and unpaired (either real or fake). We'll report the confusion rates for each setting. A score near to 50% would mean perfect confusion between real and fake. For single image FID, we'll calculate the FID between the statistics of internal features of real images and in the generated sample. A small SIFID is a good indicator for a large confusion rate.

5 Current Method

By the time we created this document, we still haven't finish implementing the whole model we proposed to create discussed above. Currently we have done the following tasks:

1) Convert the model into a conditional GAN model.

2) Currently using the input image itself as the label

3) Found and tested available image segmentation model

4) Test performance based on current implementation

And in the next phase we will going to do the following tasks:

1) Replace the current label (image itself) by the result from the image segmentation

2) Test performance based on the final model

3) Compare the performance with the original model

4) Optimize the model for better output

We still have some problems to fix in the future, including:

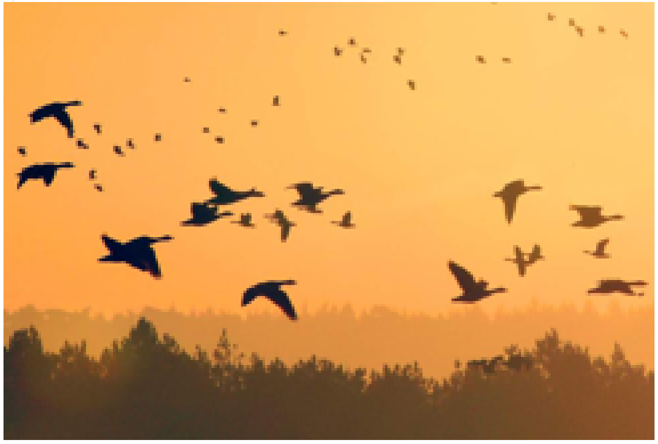
1) Fix the conflict between TensorFlow and Pytorch:While the GAN use pytorch and the segmentation model use Keras, one of the model will lock the graphic card and as a result the other model will fail due to the lack of graphic card resource.

2) Need further manipulation on the image segmentation to make sure it won't influence the color of the generated image, and find better image segmentation model since the current model is not working very well on image with relatively small size.

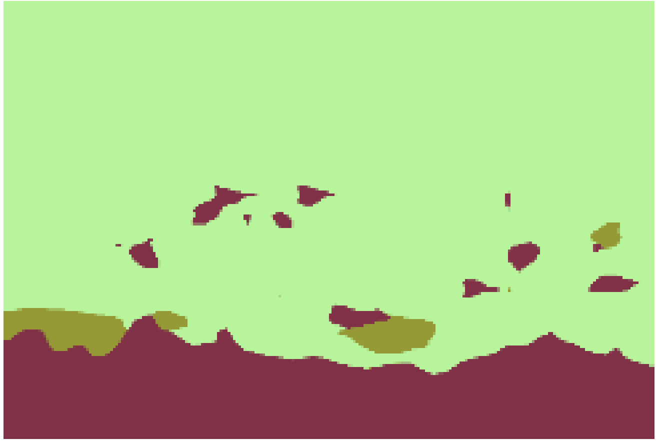
3) Reconstruct the GAN application structure to generalize it for further applications.

6 Current Result

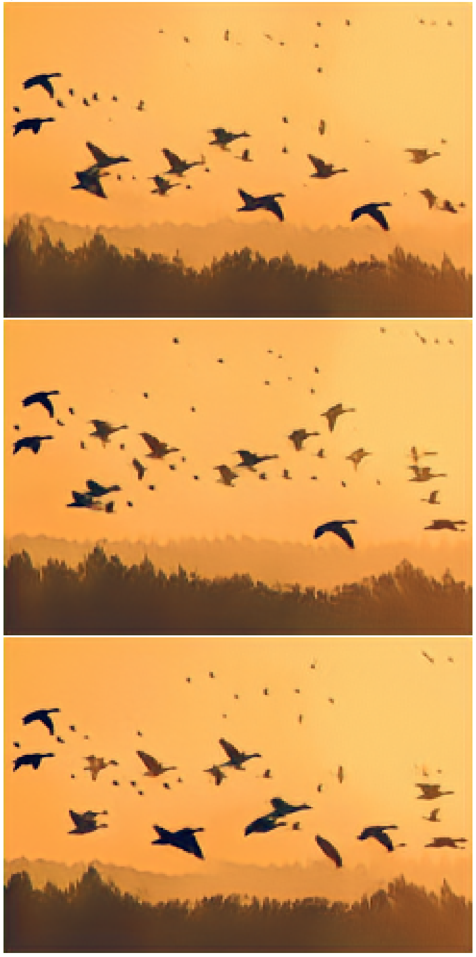
Since there are multiple applications in SinGAN, we use the basic random-sample generation to test our new model. We use the following picture as the input image:



The result of the image segmentation:



Some of the results of our model are:



References

[1] Brock, A., Donahue J., & Simonyan K. *Large Scale GAN Training For High Fidelity Natural Image Synthesis.* OCLR 2019.

[2] Shaham, T.R., Dekel, T., & Michaeli, T, *SinGAN: Learning a Generative Model from a Single Natural Image.* ICCV 2019.

[3] Martin D., Fowlkes C., Tal D., & Malik J., *A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics.* page 416. IEEE, 2001.