Text To Image Synthesis Using DC-GAN

Mahmoud Tarek Samir, Muhammad Sharafeldeen

Faculty of Engineering, Alexandria University, Computer And Systems Engineering Department

Abstract

A picture is worth a thousand words!". What if we can generate a realistic image from its description? In recent years, a lot of research and interest focused on text-to-image generation using GANs. In this work, we illustrate the problem and the data we used, build on a work from literature, then state the conclusions and possible future works.

Introduction

Text-to-image generation is to translate text in the form of human-written description into an image that is indistinguishable from realistic one. Examples of input/output are provided below.

Text-to-image generation has many practical and useful applications including, but not limited to:

Computer-aided tools
Language learning
Literacy development
Storytelling
Art generation

Examples of input/output

A flower with long pink petals and raised orange stamen.

A sheep standing in an open grass field.

Methodology

Tackling this problem requires learning a text feature representation that captures the important visual details (Natural Language Representation), then using these features to synthesize a pseudo-real image (Image Synthesis). With the advances in Deep Learning, the two areas have evolved and new methods are introduced, particularly using generative adversarial networks (GANs).

The work of Reed et. al (2016) used deep convolutional generative adversarial network (DC-GAN) conditioned on text features encoded by a hybrid character-level convolutional recurrent neural network. The architecture is shown below. We build on this architecture (in Tensorflow).
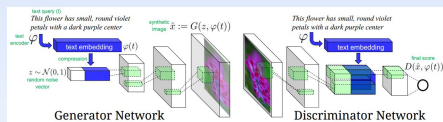

Figure 2. Our text-conditional convolutional GAN architecture. Text encoding $\varphi(t)$ is used by both generator and discriminator. It is projected to a lower-dimensions and depth concatenated with image feature maps for further stages of convolutional processing.

Training phase
* Sentence embedding is generated for all the captions
* Sentence embedded vector is fed to the CNN as its condition
* Text-to-image GAN is then trained over the image dataset

Generative phase
* Sentence embedding the required Input caption
* Feed the input caption vector to our trained model

Results

We trained our final model for 200 epochs over **Oxford-102** dataset which contains 102 categories of flowers with 40-258 images each (a total of 8,189 images), along with their text descriptions, we got the following

this flower has petals that are yellow, white and purple and has dark lines

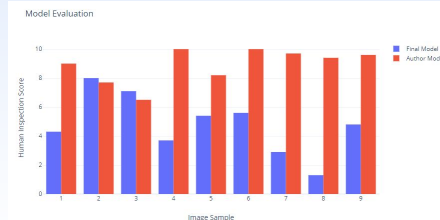this flower has a lot of small round pink petals.

the center is yellow surrounded by wavy dark purple

Evaluation

We evaluated our final model compared with the author's model using average Human Inspection score

Conclusion

* Overall, it is obvious that the author's model is better. This can be due to the learning with manifold interpolation he used (GAN-INT)

* Sent2Vec encoder highly affects the training phase

* The model takes too much time to be trained so it is very hard to make the appropriate hyperparameter tuning and even more harder to change in model architecture

Future work

* Using pyTorch instead of Tensorflow[1] to implement
* progressive augmentation
* More better hyperparameters tuning
* Training over CUB-200 birds dataset
* Trying different text encoder

References

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. (2016b). Generative adversarial text to image synthesis. Proceedings of the International Conference on Machine Learning (ICML).
https://arxiv.org/pdf/1605.05396.pdf

Paarth Neekhara text-to-image implementation in Tensorflow.
https://github.com/paarthneekhara/text-to-image

Oxford 102-flowers dataset.
https://www.robots.ox.ac.uk/~vgg/data/flowers/102/

Jamie Kiros skip-thoughts.
https://github.com/ryankiros/skip-thoughts#getting-started

Generative Models, CS231n Stanford course. Available:
http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture11.pdf