Group Members: Dhruv Saligram (NetID: dhruvks2)

Project Title: Reproducing Generative Adversarial Text to Image Synthesis

Problem Definition

I would like to complete an implementation project for my final project. The research paper that I will be focusing on is Generative Adversarial Text to Image Synthesis by Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, & Honglak Lee. This paper revolves around developing a novel deep learning architecture and GAN formulation that can successfully take in sentence descriptions and produce corresponding images.

This paper operates on CUB (a dataset of bird images), Flowers 102 (a collection of images of flowers), and COCO (a large scale object detection dataset). The authors create a GAN with a matching-aware discriminator (GAN-CLS), a GAN with manifold interpolation learning (GAN-INT), and a GAN that combines these two concepts (GAN-CLS-INT), and compares these 3 models to a traditional GAN across the 3 previously listed datasets. This leads to the creation of figures such as this in the paper:

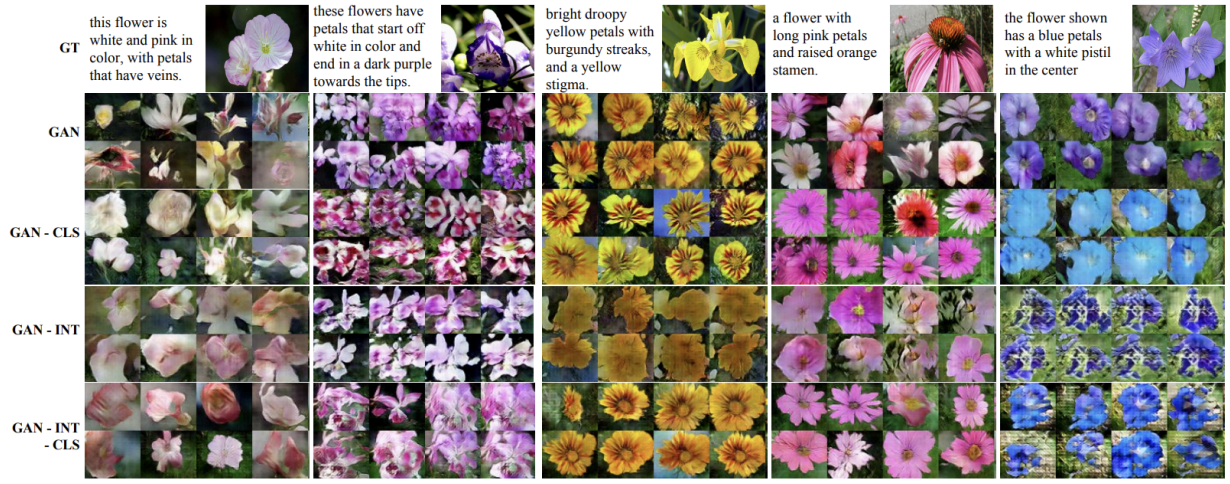

Figure 4. Zero-shot generated flower images using GAN, GAN-CLS, GAN-INT and GAN-INT-CLS. All variants generated plausible images. Although some shapes of test categories were not seen during training (e.g. columns 3 and 4), the color information is preserved.

Similar outputs are created for the CUB and COCO datasets as well.

For my final project, I aim to partially reproduce this paper's results by implementing their GAN-CLS architecture. An outline of the architecture is provided in the paper and the following illustration of the GAN is provided as well:

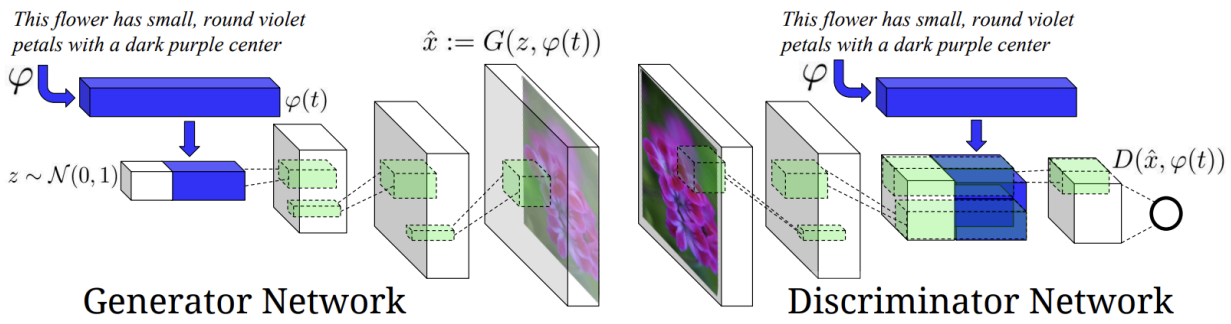

Figure 2. Our text-conditional convolutional GAN architecture. Text encoding $\varphi(t)$ is used by both generator and discriminator. It is projected to a lower-dimensions and depth concatenated with image feature maps for further stages of convolutional processing.

Specifically, my goal in this final project will be to implement the GAN-CLS model with a sole focus on the Flowers 102 dataset. I have decided to slightly limit the scope of this project since I am a beginner in deep learning working alone. My current aim for milestones in this project are:

- Processing image and text data from Flowers 102
- Finding a suitable text encoder
- Implementing the GAN-CLS architecture
- Training and testing on the Flowers 102 data
- Comparing resulting output to the output from the original paper
- Potentially implementing some form of quantitative evaluation metric to determine the quality of the work

Ultimately, the GAN-CLS is shown to produce good results, and my goal will be to implement the GAN-CLS myself and attempt to reproduce these same positive outputs on Flowers 102.

Key References

My main reference is the paper that I am trying to reproduce – Generative Adversarial Text to Image Synthesis.

The Flowers 102 dataset and associated text descriptions that I will be using for training and testing can be found here.

While I have not fully delved into the project yet, I will likely be using an existing text encoder.

Relationship to Background

I am a BS/MCS student in the first semester of my master's program. In the past, I have mainly taken classes in big data (such as database systems and data mining) and my main project experience revolves around website development.

I am a beginner in deep learning and have never completed a deep learning project before. This has led me to somewhat question the feasibility of my proposed project, and I would be open to opinions regarding my project's complexity.

I was unaware that the code associated with this paper was available online – how would this change my project?

- Is reimplementing the approach in PyTorch a good project?
    - What would my final report look like – is it mainly a discussion of how I reimplemented it, and then comparing my results to the paper's?
- Should I shift my project to modify / build on top of this code and analyze it further – ways to structure prompts to create better image output, ways to improve the architecture / output, hyperparameter tuning, testing different text encoders, etc.?
    - Could I use other research and try to find a way to quantitatively "score" the generated image quality (maybe through inception scores)? And then my paper becomes about creating a quantitative evaluation metric for the paper and finding ways to improve its score?

After discussion with prof:

- Solely reimplementing it in pytorch needs to hinge on the proof of some usefulness being derived from a pytorch implementation
- Looking to build upon code might be tough with legacy torch code – would likely need to reimplement in pytorch and then build up
    - Can use existing pytorch code to build up from, but need to do significant enough work on extensions then
    - For scoring, could do more of a research dive

Seems like reimplementing the text encoder from scratch and training my own might be too difficult – try using an existing one for the progress update, and then if time permits, training one from scratch (could add another layer of analysis on improvement)