I'm Something of a Painter Myself - Milestone

Jakub Kučera

ČVUT - FIT

kucerj56@fit.cvut.cz

1. prosince 2022

1 Introduction

This report in on an assignment based on a Kaggle competitions (TODO make link) https://www.kaggle.com/c/gan-getting-started/overview . The goal of competition is to train a model, which will take an existing real photograph and transoform into a Monet style painting.

2 kind of unstructured text - only for milestone

First I just kind of started blindly implementing a basic GAN without any thinking, which is where the implementation is now. At that point I didnt really look into discussions about this Kaggle chalenge nor anybody elses solution, since then I would make a solution too similar to someone else, even thought I would try otherwise. Plus I also would have not learned as much.

When later I started researching I came across multiple articles, one of which was this one, which might be actually a bit too similar to my task, which is kind of what I tried to prevent, but too late for that. Since this article talks directly about using Cycle GANs to transform real images into artworks in some specified art style. In this article, it was said that a basic GAN for this use case would work poorly mainly because of a loss function between a real Monet style paintings and fake ones. Because of that I will change my GAN implementation into a CycleGAN.

https://towardsdatascience.com/transforming-real-photos-into-master-artworks-with-gans-7b859a43e6ea

If this will seem too easy for me and not take that much time, I am also considering upgrading my implementation to a Conditional CycleGAN, which could for example allow me to convert images into multiple art styles, depending on what is specified in a query.

https://arxiv.org/abs/1611.07004

https://machinelearningmastery.com/how-to-develop-a-pix2pix-gan-for-image-to-image-translation/

3 Data preparation

The structure of the original Kaggle dataset looks like this (TODO add link the dirtree):

- `gan-getting-started..` Root dataset directory
  - `photo_tfrec` Real photos in the TFRecords format
  - `photo_jpg .` Real photos in the JPG format
  - `monet_tfrec .....` Monet paintings in the TFRecords format
  - `monet_tfrec ..` Monet paintings in the JPG format

The photos in the JPG and TFRecords format are duplicated and the latter is a TensorFlow format, and since I decided to use PyTorch, I deleted the latter versions.

To load the data in Pytorch I used TorchVisions *ImageFolder*. Since it automatically labels data based on subdirectories, I had add an aditional nested directory. The final structure looks like this (TODO add link the dirtree):

- `data-monet.....` Root dataset directory
  - `monet`
    - `monet` Monet paintings in the JPG format
  - `photo`
    - `photo...` Real photos in the JPG format