CS 449 Final Project Update

Due: Feburary 24, 2023 at 11:59pm

1. Names and Net IDs

Lining Mao (LMI7892), Xiaotian Ma (xma6781), Zeqiu Yu (ZYV9962)

2. Abstract

We aim to learn how to use CNN and GAN for image processing on a real life datasets. Specifically, our goal is to first differentiate paintings of Edvard Munch from those of Monet. And than we aim to tranfer the style of Munch to a random daily image or pictures through the implementation of CycleGAN.

3. Big Changes

For the first classification task, we used Monet's painting datasets instead of Van Gogh's as the different style class. Because from what we learned in art history, Munch's painting style is closer to that of Monet, so we thought choose Monet as Van Gogh is more challanging. Other than this, we mostly follow the original prject plan.

4a. Describe your dataset(s)

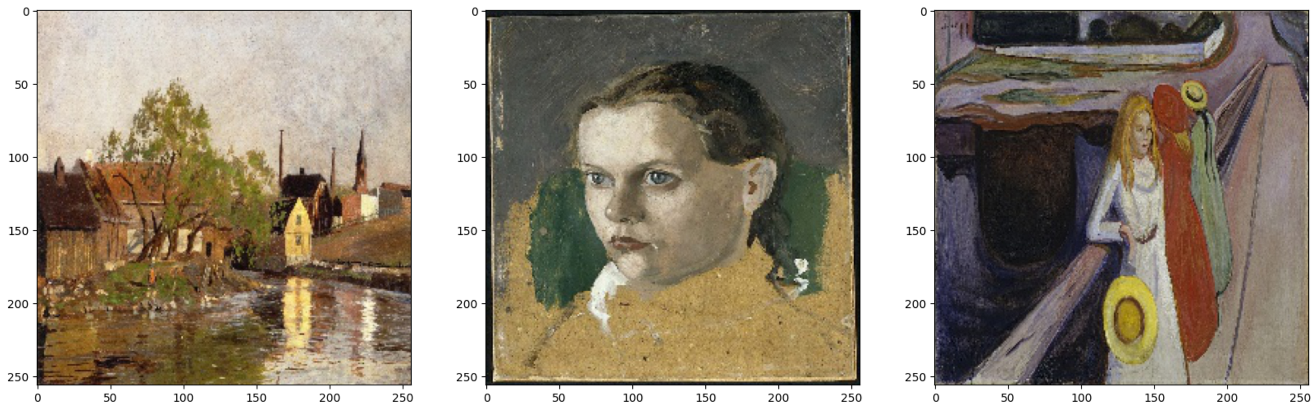
We use an open data set found on kaggle: https://www.kaggle.com/discussions/getting-started/377989 (https://www.kaggle.com/discussions/getting-started/377989) . it contains 1769 images of Edvard Munch's life work. The images are in .jpg format and vary in sizes.

For the data set: https://www.kaggle.com/code/dimitreoliveira/introduction-to-cyclegan-monet-paintings (https://www.kaggle.com/code/dimitreoliveira/introduction-to-cyclegan-monet-paintings), it includes paintings of Monet, from which we want to disentangle the Munch's paintings.

For the data set: https://www.kaggle.com/datasets/prasunroy/natural-images (https://www.kaggle.com/datasets/prasunroy/natural-images), it will be used when we try to transform some of them to the Munch's style.

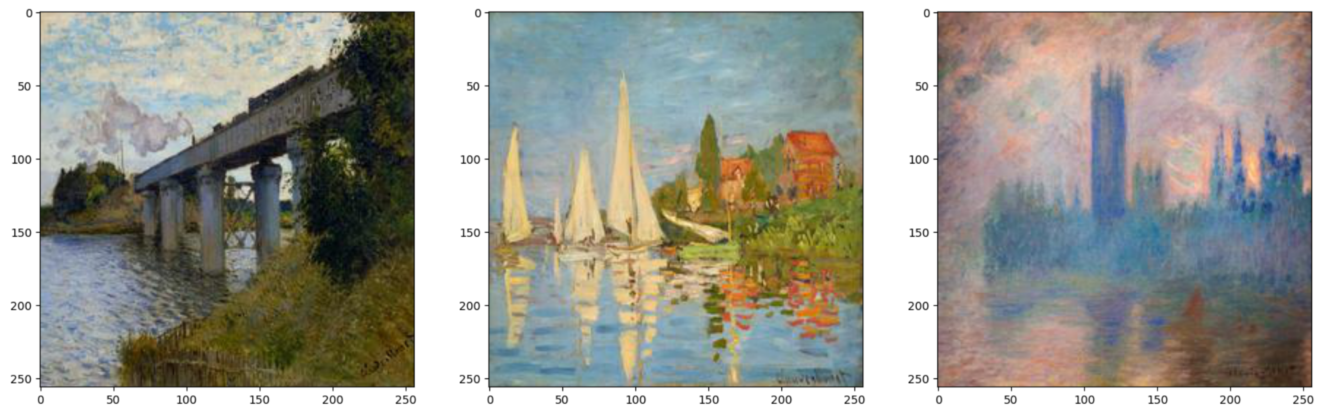
In [3]:
```
## 4b. Show some data
"""
    *Demonstrate that you have made at least some progress with getting your
     dataset ready to use. Load at least a few examples and visualize them
     as best you can*
"""

# Munch Painting dataset

import matplotlib.pyplot as plt
import cv2
import matplotlib.image as mpimg

fig = plt.figure(figsize=(20,10))
rows = 1
columns = 3

img1 = mpimg.imread('./archive/munch_processed/22.jpg')
img2 = mpimg.imread('./archive/munch_processed/48.jpg')
img3 = mpimg.imread('./archive/munch_processed/484.jpg')

fig.add_subplot(rows, columns, 1)
plt.imshow(img1)
fig.add_subplot(rows, columns, 2)
plt.imshow(img2)
fig.add_subplot(rows, columns, 3)
plt.imshow(img3)
```

Out[3]:
```
<matplotlib.image.AxesImage at 0x7f99d1c08550>
```



In [10]:
```
# Monet painting dataset

import matplotlib.pyplot as plt
import cv2
import matplotlib.image as mpimg

fig = plt.figure(figsize=(20,10))
rows = 1
columns = 3

img1 = mpimg.imread("/Users/liningmao/Desktop/archive/monet_processed/1a127acf4d.jpg")
img2 = mpimg.imread("/Users/liningmao/Desktop/archive/monet_processed/1f22663e72.jpg")
img3 = mpimg.imread("/Users/liningmao/Desktop/archive/monet_processed/2e0d0e6e19.jpg")

fig.add_subplot(rows, columns, 1)
plt.imshow(img1)
fig.add_subplot(rows, columns, 2)
plt.imshow(img2)
fig.add_subplot(rows, columns, 3)
plt.imshow(img3)
```

Out[10]:
```
<matplotlib.image.AxesImage at 0x7f99b06c2580>
```



5. Updated Methods

We plan to use CNN for start, as shown in (Gatys et.al, 2016), CNN was first thrived in image classification tasks, later on researhers found out that the convolutional layers can capture image traits and apply to other images. Addtionally, we also tried a more sophisticated model for classification, ResNet.

Certainly we plan to focus on understanding and implementing the Generative Adversarial Network (GAN) and its derivative CycleGAN. GAN is a deep-learing-based generative model and involves two sub-models: a generator model for generating new examples and a discriminator model for classifying whether generated examples are real, from the domain, or fake, generated by the generator model. For the task we are interested in, we first need to train a generater (consists of one or two autoencoder) to generate Munch style painting, and then feed into the discriminator to evaluate the quality. Based on the evaluation, the generator is also updated.

6. Previous Deliverables

Copy the deliverables from your proposal and write a sentence saying whether you completed this goal, made progress on this goal, or abandoned this goal. Unless you completed the goal, give an explanation of how it went.

6.1 Previous Essential Goals

- For the purpose of differentiating Munch's style from others', we decide to include Monet's painting dataset as a distinct style so that the model may gives a binary prediction whether the image is Munch's or not. **We complete this goal**
- Find a more effective evaluation method to score the GAN Model. **Since we havn't finished implementing CycleGAN model, this goal is still in progress.**

6.2 Previous Desired Goals

- Trying to get the best image classification model for our case. **Achieved really good result using ResNet.**
- As the object to transform, natural image (https://www.kaggle.com/datasets/prasunroy/natural-images (https://www.kaggle.com/datasets/prasunroy/natural-images)) is included in our modelling process. Since the original GAN structure can only generate image that is similar to certain type and prefer paired images, we intend to implement the CycleGAN structure to achieve unpaired image transformation. Simply put, CycleGAN architecture includes two GAN, i.e, two generator and two discriminator, and each are trained in typical GAN loss. Considering a pair of image (A,B) with different content and style, one GAN transforms image A to B and another transforms B to A. A novel loss, cycle consistency loss, is introduced to control the performance of whole cycle. Previous research has shown that this architecture can extract certain 'style' and then apply to general images while retaining the content. **We are half way building a working CycleGAN model, there are still more work to be done on hyper parameter tuning**

6.3 Previous Stretch Goals

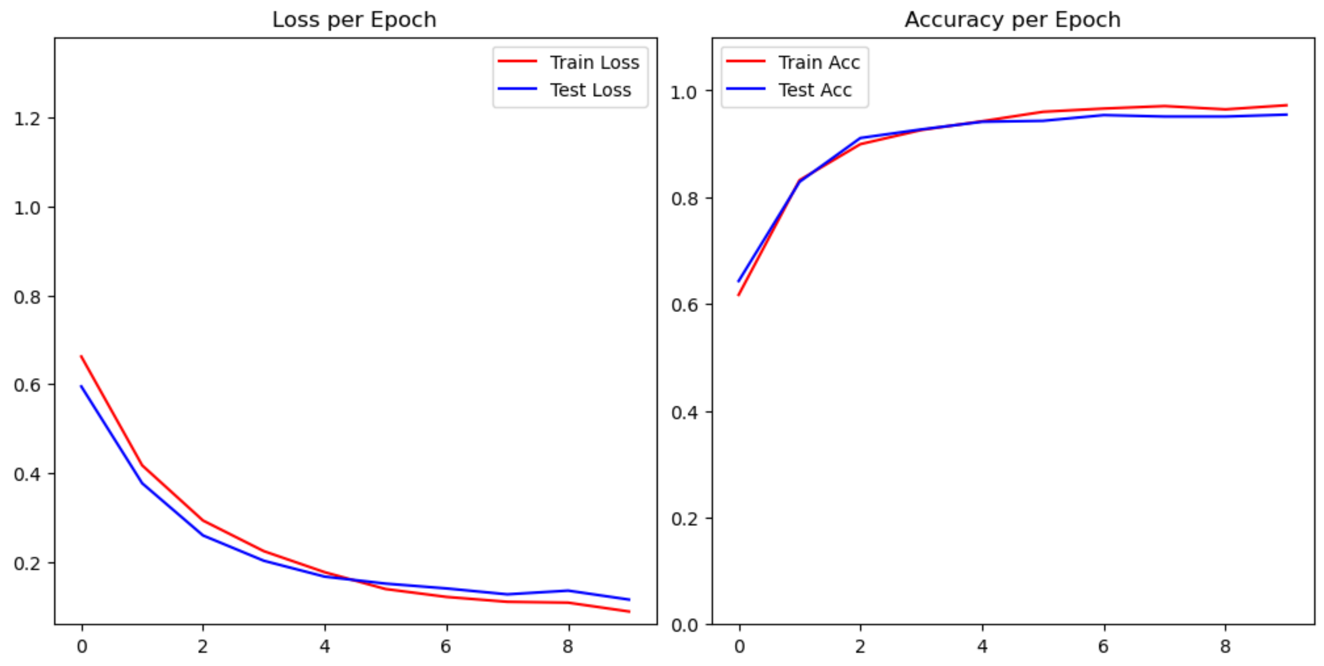
- Compare the performance and difficulty in training for our models. **Still work in progress since the training process is not done yet.**
- Improve the robustness of the models, so that they are able to distinguish a Munch's paiting with colorless input or half a painting. **Haven't started yet.**
- Transform colorless picture to paintings in Munch's style (with coloring). **Haven't started yet.**

7. Results So Far

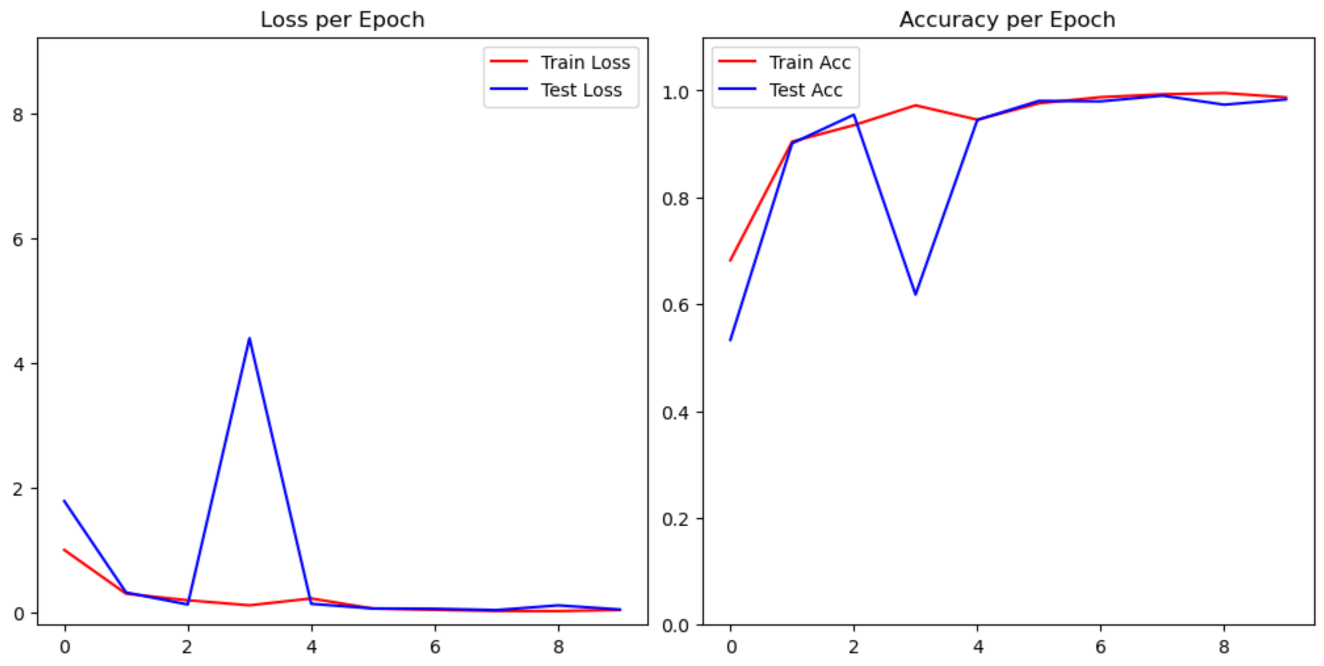
""" Include some initial empirical results on how your model(s) perform on your dataset(s). Ideally, you should not write all your code here, rather you should import from other code in your GitHub repository and just show some initial results. If it's easier, you can just include plots and a description of how you made them, rather than runnable code. """

The solid results we have had so far is on image classification.

For CNN architecture, we trained for 10 epoch and the final test accuracy is 0.9545, the final train accuracy is 0.9720, the training history is attached as follow



For ResNet architecture, we trained for 10 epoch and the final test accuracy is 0.9830, the final train accuracy is 0.9869, the training history is attached as follow



8. New Deliverables

For any deliverables that you did NOT complete, copy them into these lists below. Then add at least one goal per level.

8.1 New Essential Goals

- Find a more effective evaluation method to score the GAN Model.
- Find some adversarial examples of our trained style transfer model.

8.2 New Desired Goals

- Train a satisfactory style transfer model.
- Try a pretrained model to see if the pretraining can improve our final trained performance.

8.3 New Stretch Goal

- Improve the robustness of the models, so that they are able to distinguish a Munch's paiting with colorless input or half a painting.
- Transform colorless picture to paintings in Munch's style (with coloring).

9. Hopes and Concerns

The coding and model architecture hasn't been of much trouble for us, we spent most time on configuring cloud training or online training environment. Which causes trouble for our CycleGAN training and tuning.

10. References

Cite the papers or sources that you used to discover your datasets and/or models, if you didn't include the citation above. For example:

Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image style transfer using Convolutional Neural Networks. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2016.265 (https://doi.org/10.1109/cvpr.2016.265)

Zeiler, M.D., Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8689. Springer, Cham. https://doi.org/10.1007/978-3-319-10590-1_53 (https://doi.org/10.1007/978-3-319-10590-1_53)

Isaienkov, K. (2023, January). Edvard Munch Paintings. Retrieved January 31, 2023 from: https://www.kaggle.com/datasets/isaienkov/edvard-munch-paintings (https://www.kaggle.com/datasets/isaienkov/edvard-munch-paintings)

Innat (2022, March). Van Gogh Paintings:Art-Work Sample of Vincent. Retrieved Feburary 7, 2023 from: https://www.kaggle.com/datasets/ipythonx/van-gogh-paintings (https://www.kaggle.com/datasets/ipythonx/van-gogh-paintings)

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks", in IEEE International Conference on Computer Vision (ICCV), 2017.

Roy, P., Ghosh, S., Bhattacharya, S. and Pal, U. (2018). Effects of Degradations on Deep Neural Network Architectures. Retrieved Feburary 8, 2023 from: https://www.kaggle.com/datasets/prasunroy/natural-images (https://www.kaggle.com/datasets/prasunroy/natural-images)

```
In [ ]:
```