Milestone 2

Harvard AC215: Advanced Practical Data Science

Dongyun Kim, Vasco Meerman

1. Problem Definition

With the advent of Artificial Intelligence, it has changed the world in different ways such as an autonomous car in computer vision and language translation in natural language processing. However, AI is only limited for those who can know how to code, have statistical knowledge and mathematics background even though it has a lot of potential to be used in many fields. In particular, Generative Adversarial Network (GAN) has shown interesting features such as latent space and feature extraction, and skilled artists or affordable artists who have in-house engineering teams take advantage of GANs as a tool to help their artworks or create their own artworks, for example, Refik Anadol while normal artists have not had opportunities to take advantage of novel technologies so far.

2. Proposed Solution

In this project, we want to help artists and designers who are marginalized due to the technology gap and help those who are in the blind spot in technology. To open the AI gate for them, we propose a simple application that allows them to upload and/or scrap images and experience art-related AI techniques with an easy and intuitive interface. The application is a new kind of AI-powered creative tool for artists, designers, and creators. No need to have an AI background.

One of main advantages of using GANs is to extract features and create latent space based on input's characteristics. It means that it could be used to understand existing artworks in different ways through the lens of artificial intelligence and it can give us a new way to interpret its characteristics and unveil and discover hidden insights.

The first thought that comes to mind is to use style transfer. By projecting the style from style image to target image, it directly gives clues as to how the two styles could be combined. Also, instead of 1 to 1 style transfer, generalized style by trained network in certain dataset could be achieved, using different architecture such as CycleGAN and Pix2Pix.

In addition, artists sometimes encounter a moment where they cannot continue to paint their works because of exhaustion of ideas. In this stage, our application can help them figure out which artist's painting is most similar to their painting through a classification model and give hints to help artists develop their work and be inspired. Furthermore, if they find two images they like and want to explore in-between, latent walk could help them traverse it. For example, I am painting cubism-style artwork, but also want to paint impressionist brushstrokes in my artwork. By navigating images between Pablo Picasso and Van Gogh, I might be able to come up with an idea, how to add gogh-style brush touch in my painting.

3. Data preparation

3. 1. Existing dataset

3. 1. 1. Collection of Paintings of the 50 Most Influential Artists of All Time (https://www.kaggle.com/ikarus777/best-artworks-of-all-time)

This dataset is made up of various artists and their paintings. It is useful to observe their artworks' characteristics.

3. 1. 2. Wiki-Art : Visual Art Encyclopedia

(https://www.kaggle.com/ipythonx/wikiart-gangogh-creating-art-gan)

This dataset is made up of various painting genres such as abstract, landscape, portrait, etc. It is good to see differences between art genres.

3. 2. Web image scraper

Scrape Flickr -> Pulls data from their api

Flags:

- -t -> Tag to search for on flickr
- -mc -> Max count, meaning max downloaded images

```
python -m data_collection.flickr -t "dotpainting" -mc 500
```

Scrape Pinterest -> Scrapes the url with selenium

Flags:

- -t -> Tag name to give the folder of images e.g. the board name like "animals_horses"
- -u -> Url to the actual images you want to scrape
- -mc -> Max count, meaning max downloaded images.
  - This is not super specific since chrom will scroll and grab all the new images it can find, meaning that the first scroll will already yield about 9 images.

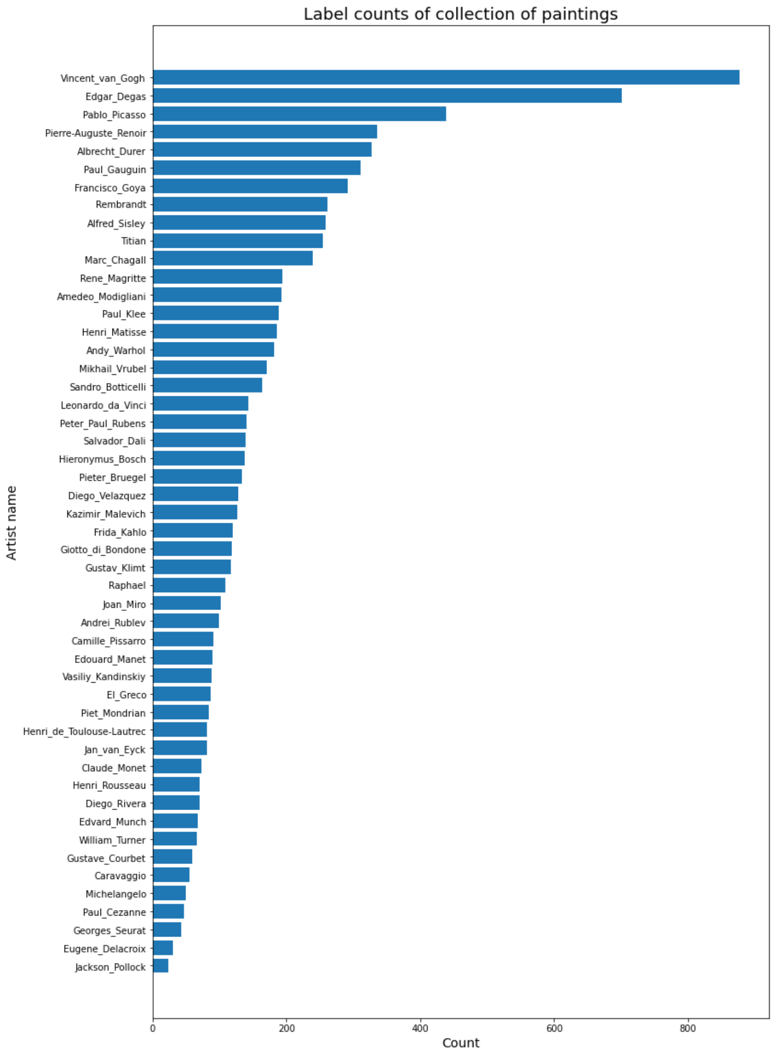
```
python -m data_collection.pinterest -t "dotpainting_on_rocks" -mc 3 -u "https://nl.pinterest.com/
```

Due to limitations of existing art-related dataset, web image scraper is needed. Among the various image sharing websites, Pinterest and Flickr are selected because they have a lot of images and Flickr supports API. In addition to creating our own dataset, these web scrapers will be able to be used in different stages, for example, it can allow an user to find interesting images through keyword search without uploading an image.

The scraper we build makes it easy to scrape pinterest via a Selenium based approach and Flicker through it's API with simple CLI commands we created. We have implemented the function that downloads the images to be multi-threaded whilst also maintaining a randomized pause between batches that are being downloaded.

4. Basic Exploratory Data Analysis on Data

4. 1. 1. Collection of Paintings of the 50 Most Influential Artists of All Time



5. Exploration of various models relevant to the problem

5. 1. Neural Style Transfer (https://www.tensorflow.org/tutorials/generative/style_transfer)

Neural Style Transfer manipulates digital images, or videos, in order to adopt the appearance or visual style of another image. It can mainly be utilized for 1 to 1 style transfer. This is our baseline model and showed very interesting results (displayed below) with a relatively short training time. This model along with Google's Deep Dream seem to be the first papers achieving exciting results in the art domain and style transfer within the Deep Learning space.

5. 2. Pix2Pix (https://phillipi.github.io/pix2pix/)

Pix2Pix uses a conditional generative adversarial network to learn a mapping from an input image to an output image. Main advantage of this architecture is that it allows us to use various image types as an input and output. Both can be a sketch, segmentation map, realistic image, etc.

5. 3. CycleGAN (https://junyanz.github.io/CycleGAN/)

Although Pix2Pix shows awesome results, there is a limitation in the training dataset which requires a labelled dataset. To tackle this problem, CycleGAN allows us to use an unpair dataset. However, to train the model with high resolution, it requires high-performance computer resources. For example, to generate 1024 by 1024 output, we generally need 2 x RTX 3090.

5. 4. StyleGAN2-ADA (https://github.com/NVlabs/stylegan2-ada)

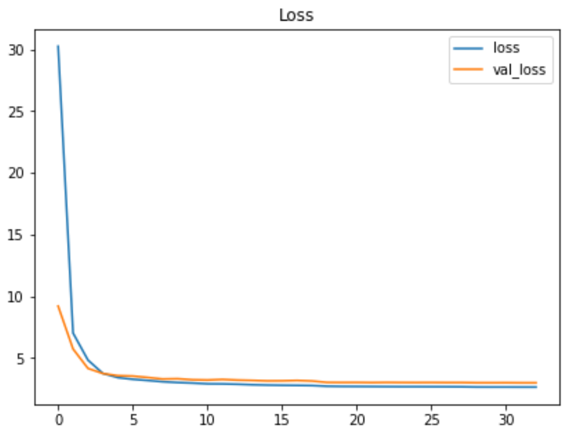
StyleGAN is a model that learns features from dataset images and creates latent space based on its features. It could be used to generalize features in the dataset and explore the features in the latent space. Because it learns the characteristics of the dataset, it has the advantage of being able to generate images that are not in the dataset.

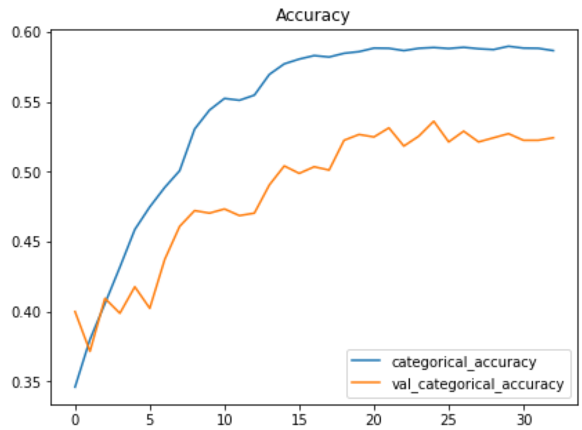
5. 5. Conditional GAN (https://arxiv.org/abs/1411.1784)

The principle of cGAN is similar to basic GAN, but it adds the label y as a parameter to the input of the generator and tries to generate the corresponding data point. Compared to standard GAN, this labeling makes it possible to specify an image class and helps create specific images we aim to generate.

6. **Baseline models trained**

6.1. Classification (MobileNetV2)





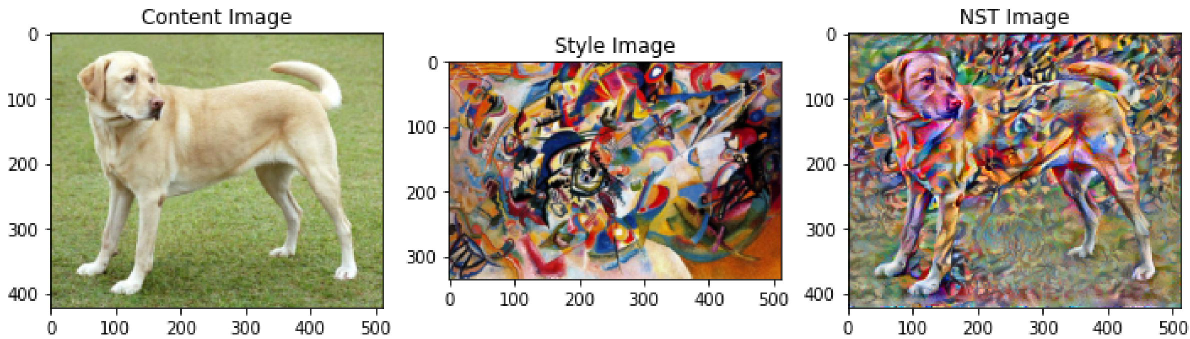
For classification, we try basic CNN classification, using a pretrained network (MobileNetV2). It shows about 52% accuracy since there are so many classes in the dataset (50 classes). To increase accuracy, we will try to utilize different models such as EfficientNet.

6.2 Neural Style Transfer

For our baseline NST model, we used most of the setup of the original paper called, A Neural Algorithm of Artistic Style. The model's architecture consists of the pre-trained VGG19 without it's classification head. From this model different layers are used for the content representation, last layers, and for the style which are multiple layers throughout the network. We use a Gram Matrix to average the correlation over the outer layers of all locations to capture the style of the original image.

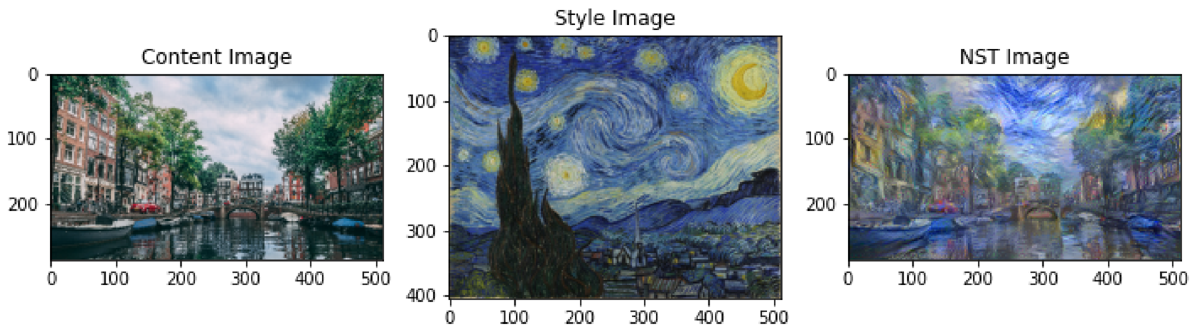
6.2.1 NST Experiment 1

In this experiment, we combined an image of a Yellow Labrador with Wassily Kandinsky's Composition 7.



6.2.1 NST Experiment 2

In this experiment, we combined an image of the Amsterdam Canals with Vincent van Gogh's Starry Night painting



6.2.3 NST Experiment 3

In this experiment, we tried working with a different type of 'art', we used a form of art called 'dot-art' for which an object is painted in with colorful dots. We also ran this experiment to determine how well this setup would work with the styled image being an actual image compared to an image of a painting. We used a relatively simple image with similar shapes as our content image, an open avocado.
We did also play around with the hyperparameters like style weights and the Adam learning rate to get a more visually stable image, since the output also looks more like a painting then an image in some particular style.

6.2.4 NST Results
Our experiments showed promising results from our baseline model. We noticed that the original architecture didn't perform as when we used a real image as 'styled' input compared to an image of a painting. We played around with multiple hyper parameters, like our weighted losses and Adam optimizer learning rate.

```
[39] opt = tf.optimizers.Adam(learning_rate=0.005, beta_1=0.99, epsilon=1e-1)
```

To optimize this, use a weighted combination of the two losses to get the total loss:

```
[40] style_weight=1e-3
     content_weight=1e4
```

This unfortunately did not improve our third experiment as much as would have hoped. Also, using this setup of Neural Style Transfer requires the model to learn specifically for the content and style image, which took a few minutes on a macbook without GPU support, making it not a very effective option to use in a live application.
This all concludes that we will need to further investigate different newer Neural Style Transfer models with the following capabilities:

- Relatively short inference time
- Able to predict on data is hasn't seen before
- Optionally: A one model approach to generate different styles for an input image

7. Main functions being considered

7.1. Classification

We're looking at implementing an extra optional first step in our application where a user can upload an image, of their own work or from someone else, and we can then classify which Style their image most resembles. This helps artists define their style and this can also be used to determine which type of Style Transfer would work best for their image.

7.2. Style transfer

We want to use a form of Style Transfer in our application so that users can upload images, which our application can instantly apply some style to. We want to train a few Style Transfer models or perhaps look at conditional Style Transfer for multiple styles, so that the user can pick any of the styles to apply to their images.
We also want to take a look at a real-time Style Transfer so the user can either upload a video which we can then process or even a functionality where they can film their surroundings for which a real-time style will be applied.

7.3. Image generation and latent walk across different classes

The final step in our project is to create a new image using gan, tentatively StyleGAN. Based on the user's purpose, the image may be created by a random vector or an uploaded image, projecting the image into the latent space and finding the most similar image. Artists sometimes want to be inspired, but they are also worried that they may subconsciously plagiarize other's works. However, GAN model preserves the characteristics of its original dataset, but it is newly created. It indicates that they can be one step away from these worries. Furthermore, traversing between classes is possible if a conditional option is added to GAN model like cGAN. This allows users to control and observe the change of features beyond simply creating an image,showing a wider range of applications by users. For example, artists can create an image with 70% Van Gogh and 30% Picasso styles.