Paintings and Language

Ben Barrett, Austin Cheng, Daniel Choi
University of Virginia, Charlottesville, VA 22904
{bjb2us, ac7rf, dc9db}@virginia.edu

Abstract

The generation of images using GANs or text using RNNs/transformers has seen significant progress in recent years, leading to the creation of photorealistic media popularly known as DeepFakes. While several methods for exclusively generating images or exclusively generating text have been published, there has been little research on generating image-text pairs, such as captioned photos. In this project, we explore two methods for generating new image-text pairs of paintings and their titles using a publicly provided Kaggle paintings dataset (sourced from WikiArt) and a new MET Paintings dataset. Painting titles are more abstract and therefore less semantically grounded than captioned photos, and we believe that capturing this behavior in a generative model would be interesting. We attempt two methods in creating a synthetic painting along with an appropriate title: sequential and parallel. The sequential method is treated as an image generation task using a GAN directly followed by an image captioning task using an RNN, while the parallel method uses a multimodal GAN to generate both the image and text simultaneously. We find that the simpler sequential model generates some reasonable titled paintings.

1. Introduction

Generative neural networks is a challenging field of study in order to make sure the model can generate fabricated data that is plausible for people to believe in. There has been many research in this field of generating images, text and even audio [9] using models such as Generative Adversarial Networks (GAN) or Recurrent Neural Networks (RNN). While these tasks are challenging, if done well, they represent an opportunity to supplement human creativity in a broader way.

In this project, we aim to create model(s) that can generate both image and an associated text and compare the approaches used for this task. We will attempt two separate approaches in accomplishing this task - one sequential and one parallel. For the sequential model, we will have two components to it - a GAN that generates images and a RNN that produces a text label for the generated image - while the parallel model will explore using a multimodal GAN to generate both the image and label simultaneously.

Our project of generating image-text pairs focuses on creating art work and their respective title, trained on two different datasets: a Kaggle Dataset based on WikiArt,[1] and a new MET Paintings Dataset[2].

2. Related Work

An important measure in training is the performance of the model. For GANs, however, it is difficult to have an objective measure because human measures of believability are subjective. The method we use to measure our GAN's performance is the Fréchet Inception Distance (FID) [3]. FID measures the performance by comparing the average Gaussian of the mean and covariance of images from the actual data set to that of generated image sets. Compared to inception score, FID shows a better measurement for performance when evaluating generated images based on a given dataset as the FID compares the generated images to actual images from the dataset, while the inception score simply evaluates the generated image.

With generating images, there are multiple ways to generate the accompanying captions. The most intuitive process would be to train an image captioner and then run it on the outputs of the GAN network as in [2]. On the other hand, there has been little work on jointly generating image-text pairs. One method has been suggested to cast image-text generation as an exclusively image generation task by first rendering the text caption as an image and then placing it next to the original photograph [7]. However, we believe that this method is less able to capture the semantics of the text than with text embeddings more commonly used in natural language processing, such as RNNs or transformers.

[1] https://www.kaggle.com/c/painter-by-numbers
[2] https://www.metmuseum.org/art/collection/

3. Model

The sequential model consists of image generation followed by image captioning on generated images. The parallel model uses a multimodal GAN.

3.1. Image Generation

For image generation, we used PyTorch to implement our model, and used the code from a publicly available implementation of DCGAN as our starting implementation.[3] While we wanted to produce a more higher resolution than the base 64 by 64 version, simply upscaling the existing model resulted in frequent collapse of the GAN model due to the balance between the generator and the discriminator. Therefore features used for the two components were changed to prevent the model from collapsing. By setting the size of the feature maps in the generator(ngf) to 128 and the size of feature maps in the discriminator(ndf) to 32, the model was able to get a good balance between the two components and go through the training without collapsing. The images generated from the GAN can be seen along with actual image. (Figure 1)

3.2. Text Generation

For text generation, our model uses an encoder based on ResNet and a LSTM based decoder. The text generator was inspired by an assignment in a vision and language class [5]. The image encoder uses 4 fully connected layers of ResNet-152. To help prevent over-fitting, dropout layers are included after every fully connected layer [8]. The encoder takes a 64x64 pixel image and converts it to a 1024 sized vector which is fed to the decoder. The decoder is a single directional RNN that uses an embedding size of 512. The results of the encoder and the start token are fed into the decoder as the initial state. The network uses a tokenizer that comprises 2000 of the most common words in the data set.

[3] https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html

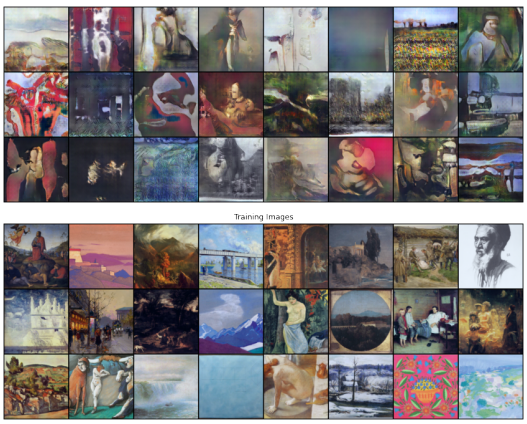

Figure 1. Generated images (above) compared with actual images (below) in the dataset.

3.3. Multimodal GAN

The implementation and training of the multimodal GAN was unsuccessful because the model's outputs appear the same as noise, but a description is included here for completeness. A multimodal GAN is made up of a multimodal generator that decodes image-text pairs from noise and a multimodal discriminator that encodes both image and text to classify the image-text pair as fake or real. In the multimodal generator, a Gaussian noise vector z with a size of 100 is fed into an image decoder taken from DCGAN [6], which uses fractionally-strided convolutions to successively upsample to a 64x64 image. At the same time, z is set as the initial hidden state of an LSTM to autoregressively produce text using a bag-of-words representation, but tokenized by character. This was justified because the average length of titles in the Kaggle dataset is small, being only 22.7 characters. Characters with a frequency of less than 100 in the dataset were replaced with the unknown token. In the multimodal discriminator, ResNet is used to extract the image features, encoded to a size of 64, and each text token is fed into an LSTM is used to extract the text features, also encoded to a size of 64. Finally, the image and text features are concatenated and passed through a linear layer to a final binary classification.

4. Data

We present a new dataset of public domain paintings from the Metropolitan Museum of Art. While the Met has many diverse exhibits ranging from pottery to sculptures to fashion, only entries labeled as paintings were extracted. Metadata on entries available online was obtained from the Met's Open Access Initiative [1]. Entries were webscraped from the Met's website using BeautifulSoup and contain the painting image, painting title, and description paragraph, along with other metadata such as author, medium, and date. Images were high resolution (on the order of MB for each image) and so to reduce download and extraction times, images were resized so that their longest side was at most 1024 pixels. Of these paintings, a subset of oil paintings were extracted because they appear more uniform in style and have less background clutter. In total, $6,664$ paintings were extracted from the Met's website, and $2,788$ were oil paintings.

5. Experiments and Results

We used PyTorch to implement our GAN model, and trained the model on the Kaggle Dataset collected from WikiArt [4] and the Met Dataset from the The Metropolitan

Figure 2. Images generated using the MET Dataset. GANs were trained with paintings(top) and oil art(bottom)

Museum of Art images. While we tried to train our model using the MET Dataset, possibly due to the smaller dataset size, we were unable to train the 128×128 resolution GAN successfully using the MET images. Even after 300 epochs, the images would show up as noise, with no noticable features that looked like the training images.

Instead, it was much more successful in the 64×64 resolution GAN as shown in Figure 2. The GAN that was trained mainly on the paintings of the MET dataset generated plausible images that looked like portraits of people. While the portraits did lack facial details, the GAN was able to capture common characteristics of the portrait, such as the circular frame found in multiple images. The generation done using the oil paintings, however, lacked even more data and it could be seen that the GAN repetitively generated similar images. Additionally, it could be seen

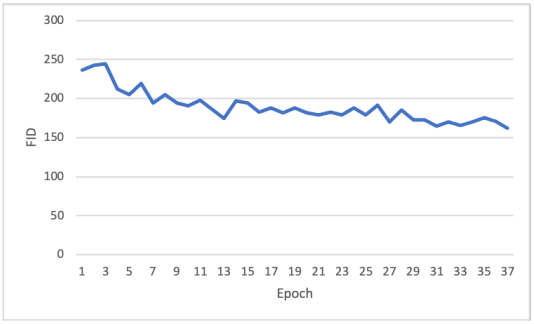

Figure 3. Here we show the FID measured over epochs for GAN training. The FID was measured by comparing 256 generated images and 256 images from the dataset. We can see that it decreases over epochs. A smaller FID value indicates a better model.

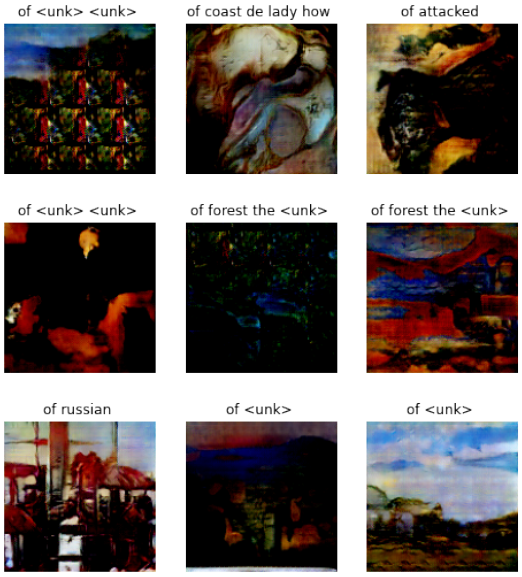

Figure 4. Generated titles of generated images from the Kaggle dataset.

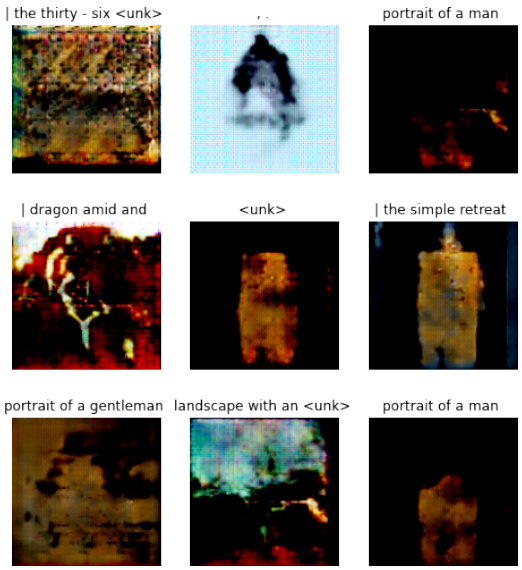

Figure 5. Generated titles of generated images from the MET dataset.

that the GAN also learned features of different art styles, as the images generated by the GAN that was trained specifically using oil paintings seemed to generate images with more texture in the images, where was the GAN that was trained using the entire paintings data also generated some flat looking images.

We trained our model using Adam and Binary Cross Entropy Loss. The learning rate was set to 2e-4 and the GAN was trained up to 40 epochs for the 128×128 resolution. The results of the GAN can be seen in Figure 1.

FID was used to measure the performance of the model.

As shown in (Figure 3), the FID value started around 250 when the generator was not trained, but quickly decreased, and went down to 164 at its minimum with 40 epochs. This indicated great performance considering how the FID score was 123 even with the images within the actual dataset.

We also used PyTorch to implement the captioning model. We trained the model using Adam and Cross Entropy Loss. The learning rate of the encoder was set to 1e-4 and the decoder learning rate was set to 1e-3. Initial training runs revealed that the model was over-fitting the training set and not producing "interesting" output for the GAN-generated images. Thus we added a dropout layer with $p = 0.2$ which led to better generalizations across images. The model was trained on the Kaggle dataset and MET dataset independently and run on the outputs of the GAN-generated images. Some interesting results are shown in (Figure 4 and 5).

The multimodal GAN was trained for 40 epochs at a learning rate of $2e{-}4$. However, outputs generated by the model appeared the same as the initial noise, suggesting that the model was ill-defined, and so we were unsuccessful in this approach.

6. Conclusion and Future Works

In this work we present two models to generate images and captions for image datasets. The results show great performance on our models, generating plausible art works as well as their titles for the sequential model that uses a DCGAN paired with a LSTM Model. Unfortunately, the multimodal GAN for the parallel model did not work out the way we had expected, not reutrning promising results.
We were unable to experiment with the MSG GAN that is said to work better with higher resolution image generation. Given the problem there was with generating high resolution images using the MET dataset, working with different models that can generate higher resolution images with the MET Dataset as the train data would be improvements that could be made. This along with more exploration on different structures for the multimodal GAN would be a great extension to the project.
Additionally, we had found that the GAN was indeed able to imitate some styling of art and art titling, where the GAN that was trained using oil paintings generated images with more textures. This could be extended to have a conditional GAN that would generate images based on the type of painting that was given as input.

References

[1] metmuseum/openaccess, Nov. 2020. original-date: 2016-09-09T18:46:43Z.

[2] S. Gorti and J. Ma. Text-to-image-to-text translation using cycle consistent adversarial networks. 08 2018.

[3] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018.

[4] K. Nichol. *Painter by Numbers*, 2020. `https://www.kaggle.com/c/painter-by-numbers/overview`.

[5] V. Ordonez-Roman, Z. Yang, P. Cascante-Bonilla, and A. Suri. *Assignment on Text Generation and Image Captioning*, 2020.

[6] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[7] B. Shimanuki. *Joint generation of image and text with GANs*. PhD thesis, Massachusetts Institute of Technology, 2019.

[8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 06 2014.

[9] Y. Zhou, Z. Wang, C. Fang, T. Bui, and T. L. Berg. Visual to sound: Generating natural sound for videos in the wild, 2018.