COMS3168 Final Project

Painted Object Synthesis

Isabel Tu (it2334) & Grayson Newell (gln2109)

https://github.com/isabeltu/DLG-final-project

Abstract

For this project we created a GAN model that can be trained to generate a stylized image of a specific object (e.g. dress, hat). Our model is trained on images of objects, derived from a painted art dataset. To obtain viable training data, we used the Detectron2 framework by Facebook Research, specifically leveraging their Detic pretrained model for segmentation and extraction of object images. This allowed us to train our GAN on clean, labeled images of individual objects. The resulting model can generate an image of a specific object that is consistent with the painted style of the original data.

Introduction

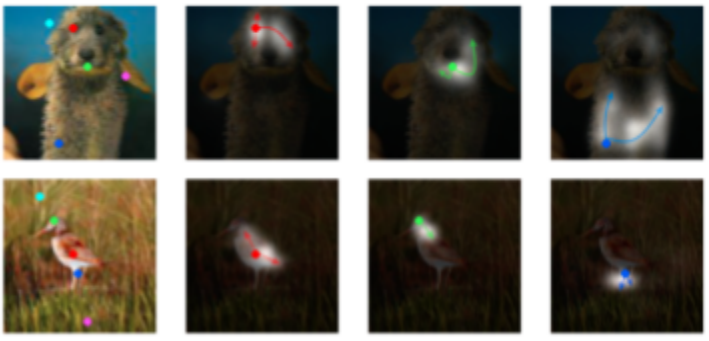
The identification of ‘common objects’ is an important concept within the field of computer vision and generative modeling. Modern object detection frameworks such as Mask R-CNN allow for accurate classification and segmentation of objects within potentially cluttered scenes. Additionally, advances in Generative Adversarial Networks (GANs) have made the synthesis of stylized imagery far more accessible. For this project, we sought to bring these two frameworks together by exploring the usage of a Mask R-CNN to segment object data for training a Self-Attention GAN (SAGAN). Our goal was to develop a SAGAN model that could be trained on images from a specific object class and then generate images of that object. These generated images would ideally reflect the style of the training data, which we observed by using imagery from painted artwork for training. Furthermore, by training a SAGAN individually for each object class, instead of a conditional GAN on all classes, we obtained a model that was far more capable of accurately representing classes.

Background

There are two papers that helped establish the basis for our methodology.

1. “Mask R-CNN” (2017): Iteration upon the preexisting R-CNN to incorporate pixel-level masks, model framework utilized by Detectron2. https://arxiv.org/pdf/1703.06870.
2. “Self-Attention Generative Adversarial Networks” (2019): Iteration on convolutional GANs to include a self-attention module for establishing spatial relations within image data, we included this innovation in our SAGAN. https://arxiv.org/pdf/1805.08318
3. Original SAGAN code https://paperswithcode.com/method/sagan

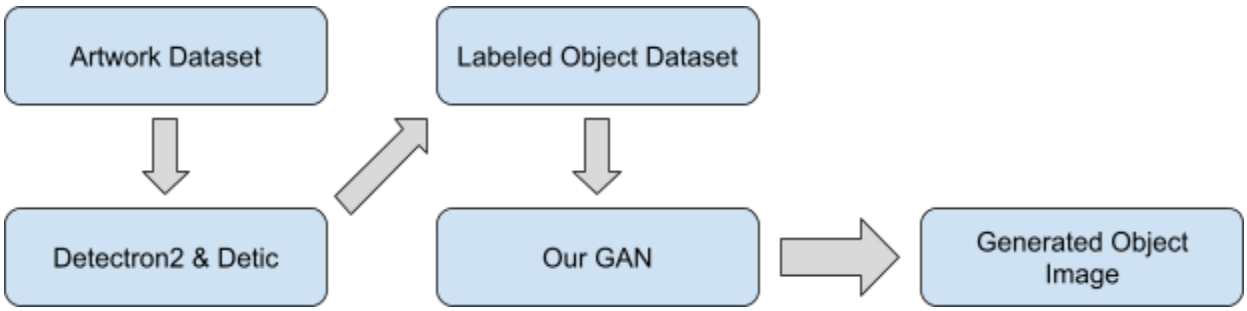
1. Mask R-CNN Segmentation



2. SAGAN Spatial Relations

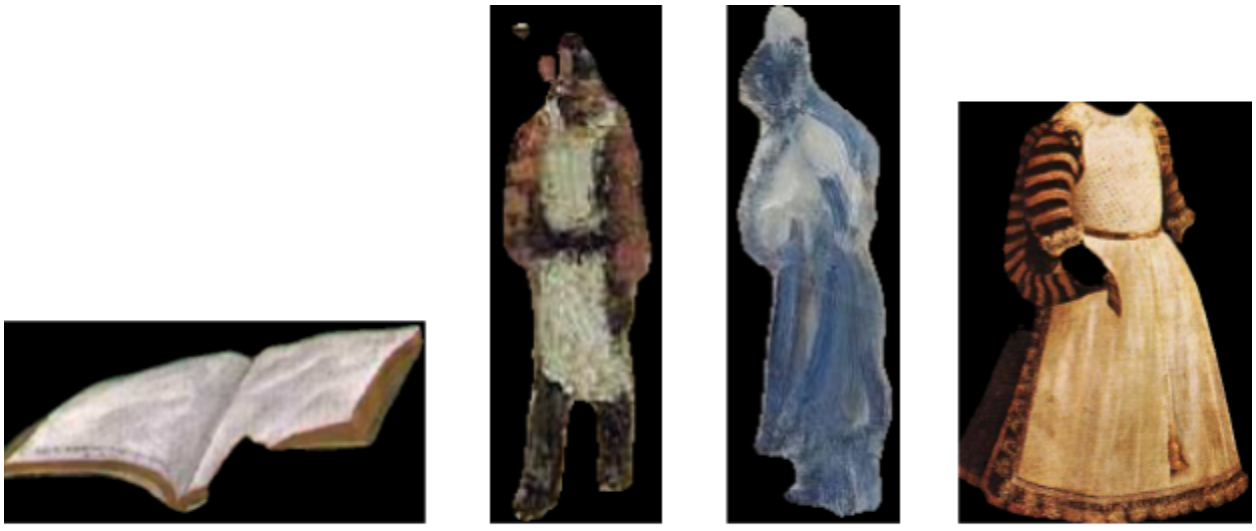
Overview

Our project pipeline began with an artwork dataset, which we segmented using an object identification model. This yielded a dataset containing labeled images of objects, cropped to their outlines. We then trained our GAN model on images of a specific object from this dataset, resulting in a usable image generator for that object. This generator can then take a random noise vector to generate a new object image, marking the end of the pipeline.



Technical Description

Dataset: Our initial artwork dataset consisted of 2026 painting images from the WikiArt dataset. We used a custom list of 2500 object classes generated using WordNet. Using Detectron2, we extracted 18271 cropped object images in 325 unique classes.



Book Person Statue Gown

Segmentation & Labeling: Detectron2 automatically labeled and segmented our object images, this was done using their Detic pretrained model and a list of object classes found in our artwork dataset.

Image Augmentation: Given that the number of crops for a given object decreased sharply after the first few most frequent categories, we used image augmentation techniques, including flipping, cropping, and brightness scaling. For augmented image classes with less than 1000 crops available, we augmented until we had a training set of 1000 crops, which we found to be a sufficient number to achieve good results after training.

Model Architecture:

Self-attention layer

- Helps the model learn long-range spatial dependencies, such as textures and structure
- Query and key are convolutions of dimension/8 to reduce computation complexity
- Value is a 1x1 convolution of dimension
- The attention weights are calculated using $\text{softmax}(\text{query} \cdot \text{key}^T)$
- Outputs the attention-weighted values scaled by a gamma value

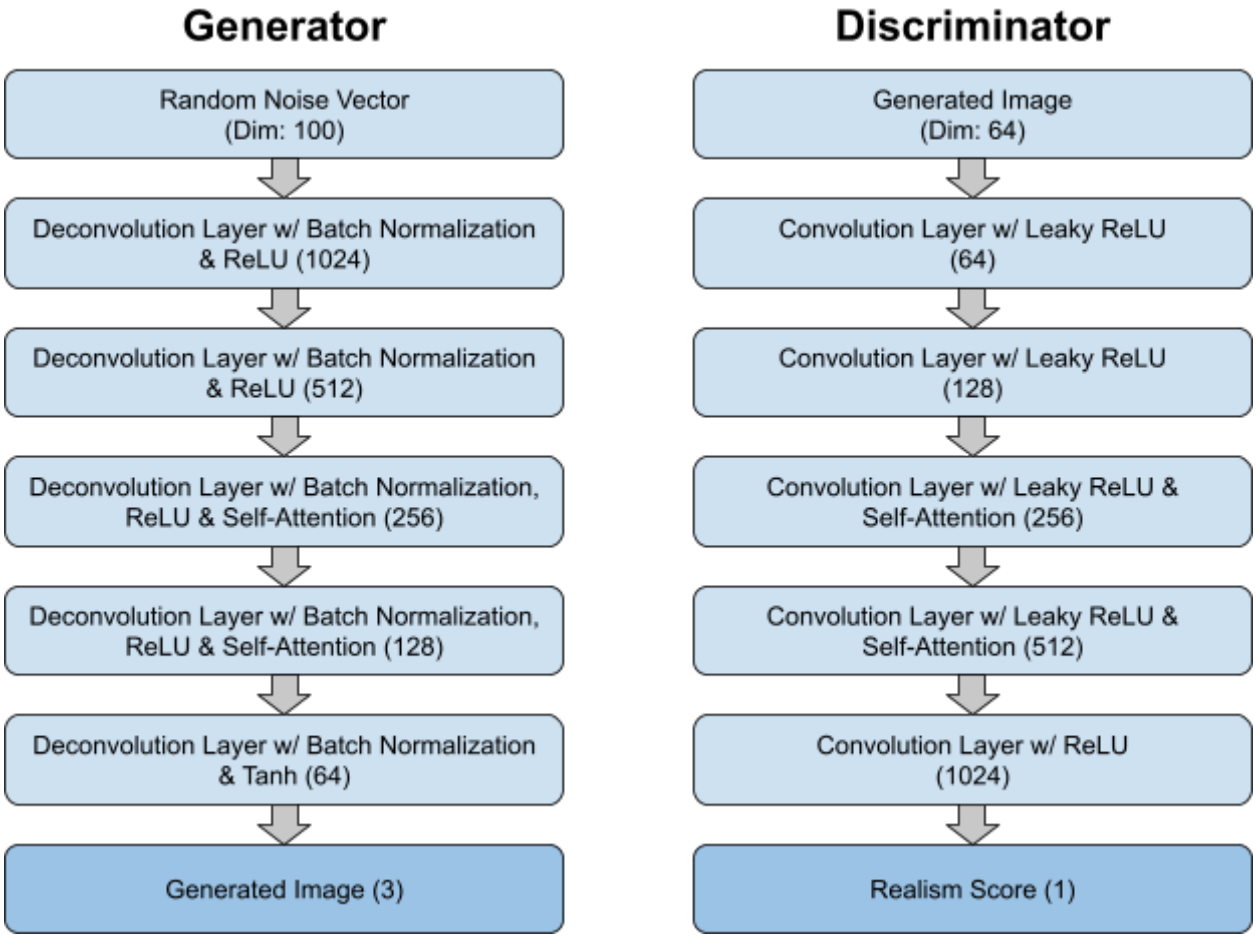
Generator

- Takes a 100-dimensional random noise vector as input
- Uses five layers of deconvolution with batch normalization, using ReLU activation
    - First layer expands the 100-dimensional vector to 1024 channels
    - Each following layer halves the number of channels
- Final layer maps the 64 channels to the image's 3 color channels using tanh activation
- Applies self-attention after the third and fourth deconvolution layers

Discriminator

- Takes a 3 color channel image as input
- Uses five layers of convolution using leaky ReLU activation
    - First layer maps the 3 channels to 64 feature maps
    - Sequentially doubles the number of features in following layers
- Final layer convolves the 1024 features into a single output representing how real the discriminator thinks the image is

- Applies self-attention after the third and fourth convolution layers



Training Parameters:

- Batch Size: 64
- Optimizer: Adam with learning rate = 2e-4, $beta_1 = 0$, and $beta_2 = 0.9$
- Loss function: hinge loss for discriminator
- Epochs: 200 by default with the option to continue training for more after
- Hardware: CUDA-enabled GPU (AWS)

Results & Analysis

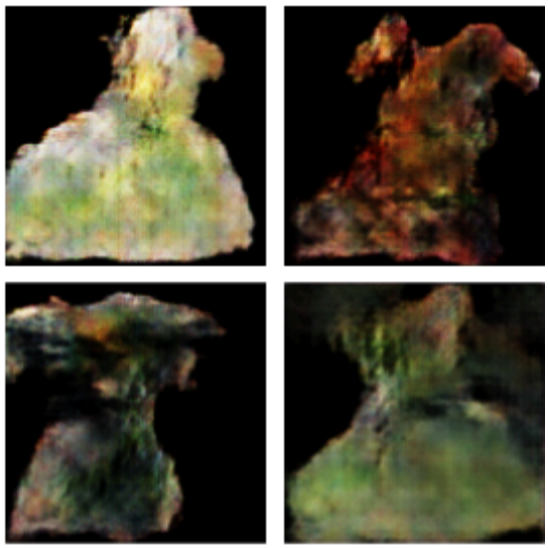
After training our GAN on multiple types of objects, we noticed that the best samples emerged from objects whose crop data was the most narrowly constrained. Gowns, for example, share relatively consistent silhouettes and color palettes, and framing. This enabled the GAN to identify and replicate these features, generating images that closely resembled the proportions of the gowns from the dataset. It performed similarly with the statue class, where it was able to extract the broad shape of the statues, but it was unable to capture the internal intricacies, producing statue-like silhouettes without any texture variation or surface detail.

As we began looking at more complex objects, the GAN generated poorer results. The person dataset was the largest and most diverse dataset that we looked at, and it contained crops of people in various scales, some of just their head from a side profile, and some of an entire body. This diversity in poses made it difficult for the GAN to learn any reliable patterns for generating images of people; thus, the majority of the samples we got lacked clearly defined human features.

An additional thing we noticed while training our GAN on particular objects was its tendency to mode collapse onto a small subset of the total input data. While the book dataset contained various positions of books, including both open and closed books, a portion of the examples were all of a single page of a book. When training the GAN on the books, we saw that it produced only images of single pages. This lines up with the architectural limitations of a GAN approach. Because the discriminator had to be able to identify the individual page images as real since they were present in the dataset, the generator took advantage of this and only learned to generate this particular type of book image. We saw the largest diversity in generated images from the person dataset, likely due to it being the largest, containing 4255 crops, as opposed to the other categories, which only contained a few hundred.

Object: “Gown”

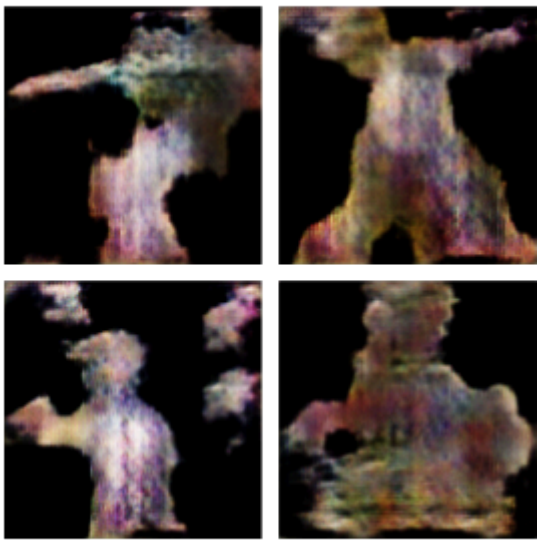
Segmented From Paintings



Generated Images

Object: “Statue”

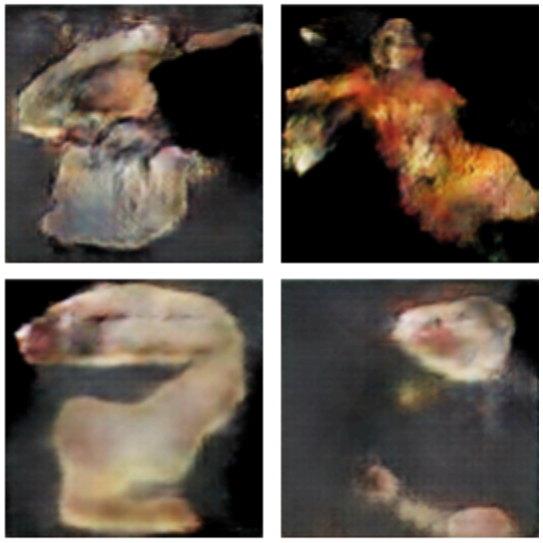
Segmented From Paintings



Generated Images

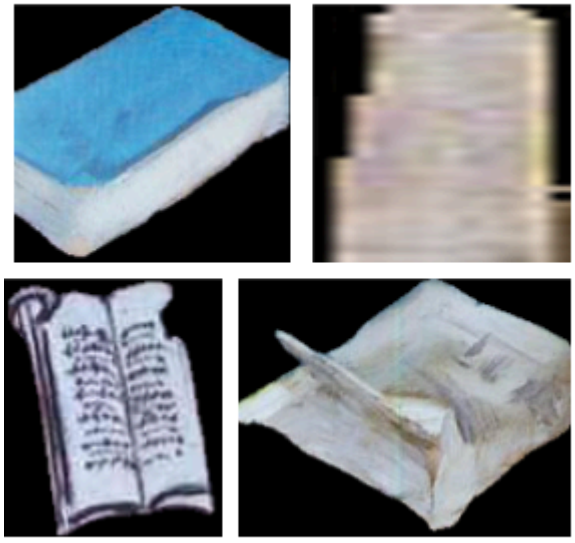
Object: “Person”

Segmented From Paintings

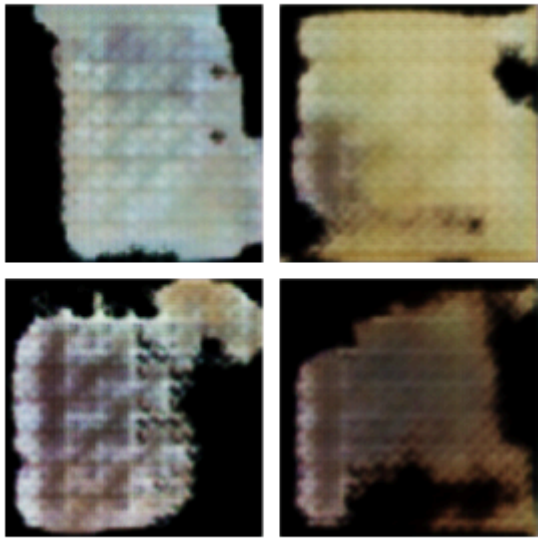


Generated Images

Object: “Book”



Segmented From Paintings



Generated Images

Applications

Image-based GANs are powerful tools for graphic design, allowing anyone to generate imagery without the need for artistic capability. Furthermore, models like ours that are trained on specific datasets (like paintings) allow for stylized generation that reflects the original data. This technology is incredibly useful by making the creation of imagery for projects far more accessible.

Discussion

While the GAN produced recognizable images for the classes with consistent training images, it struggled with the classes containing diverse ones. There was also a portion of our data that contained many incorrectly classified crops from the Detectron2 model. We purposefully avoided these object categories as we didn't believe that training on them would produce reliable results. Additionally, the style differences between the several artists included in our dataset likely forced our GAN to generalize, leading to less defined results. Our work has highlighted the importance of preprocessing and dataset curation with regards to its effects on the output of a generative model.

Future Work

Expanding our GAN functionality to handle descriptors like color and material for the objects would be an interesting avenue for further developing this project. This would require a new detection model, capable of extracting labels for these attributes along with the object itself. Then, a conditional version of our GAN could be trained on the resulting dataset, where each image would be labeled with detected attributes as well as its object. Such an implementation would likely require a large amount of processing power due to the exponential number of classes but would vastly expand the functionality and user agency of our image generator. Though its implementation would be insightful into the cohesion between cGAN and SAGAN.