Training Generative Adversarial Networks on Small Datasets by way of Transfer Learning

Hume Center for National Security and Technology

<Conf Name>

Brian Lee

Abstract

The main objective of this project was to explore the capabilities of today's state-of-the-art generative adversarial networks (GANs) in image generation using transfer learning on limited datasets assimilated from Instagram. As a step in this process, a pipeline of tools was assembled to intake and filter image datasets compatible with GAN training requirements. Transfer learning is a technique that leverages a network pretrained on a different, often larger, dataset. Transfer learning provides benefits in training time, computing requirements, and small dataset compatibility. It was found that pretrained networks have noteworthy potential in adapting to new domains of data, even when retraining on small datasets from the target domain.

1. Introduction

Generative modeling is the unsupervised learning task of modeling a distribution of input training data so that samples synthesized from the learned model resemble plausible examples from the real distribution [11]. Generative adversarial networks (GANs) offer an innovative solution to this task with its two-network architecture. The basic GAN architecture as illustrated in Figure 1 consists of a generator network and a discriminator network, pitted against one another as adversaries. The generator is tasked with learning the distribution of a real dataset to generate new examples mimicking examples from the training data. The discriminator network is tasked with learning how to correctly classify real and synthetic examples while providing feedback to the generator model on its success at generating fake examples [3]. Random noise is fed into the generator as a starting point from which the artificial samples are generated in this particular architecture, although that is not always the case.

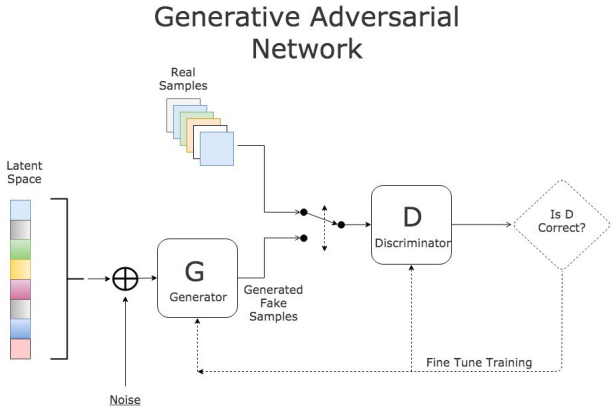

Figure 1. Basic generative adversarial network structure [13].

Together, these networks and their loss functions compete in what is known as a two-player minimax game. While the discriminator learns to maximize its

classification successes, the generator learns to generate examples that minimize the discriminator's classification successes. First, the distribution from which the input random noise is created is defined with a prior $p_z(z)$. The nonlinear function that maps input to the data space is denoted $G(z)$. The resulting distribution of generated data is defined as p_g, which is being learned to resemble the distribution of the training data, defined as p_{data}. Lastly, the probability that a data point is from the training dataset rather than the generated one is denoted $D(x)$. Then, the minimax loss equation is defined with the value function $V(G, D)$, as described in the function as described by Goodfellow [3].

$$\min_G \max_D V(G,D) = E_{x\sim p_{data}(x)} log(D(x)) + E_{z\sim p_z(z)} log(1 - D(G(z))) \quad (1)$$

Currently, image data is the most popular domain of data in which GANs are trained [9]. Not only was the first GANs trained with image data, but their most well-known and exciting applications typically involve some form of image synthesis. Among such foci include image generation for datasets [3], photo-realistic human face generation [5], and text-to-image generation [8].

In many of these published works, GANs are trained from scratch, but doing so is a difficult ordeal. Training on large datasets over extended periods of time can prove costly in both time and computation, especially when training with image data [5]. According to NVIDIA, the ideal GAN is trained with 50,000 to 100,000 images. However, arbitrarily decreasing the size of the dataset in an effort to save computational resources often results in various training failures. The most common failure is when the discriminator overfits to the data. When training a GAN with an insufficient number of images, the discriminator can become overly adept at classifying real and generated images, forcing the generator to synthesize nonsense in an attempt to fool the discriminator [12].

A proposed potential solution to balance training failure minimization and cost efficiency is to leverage transfer learning to train a GAN on the desired target dataset. Transfer learning is the technique of applying a model that has been previously trained on one task or domain of data to another task or domain. In the context of GANs and image data, a pretrained generator from a GAN has already learned the weights of key features from the images of a dataset known as the source dataset, such as edges, curves, coloration, and lighting patterns, to name a few. Ideally, the prior knowledge learned from the source domain enables the network to adapt to a new dataset known as a target dataset when it continues training, requiring fewer training examples.

1.1. Objective and Motivations

The primary objective of this project was to evaluate how efficient and effective transfer learning is as a method to train GANs on small datasets. This entails exploring the method's successes and limitations under various training conditions, namely the combinations of source and target domains. A secondary objective was to assess the viability of using Instagram as an effective platform to assemble datasets for transfer learning.

Given the drawbacks of GANs, the motivation to apply transfer learning to generative learning remains strong. The intuition is to leverage the learning contained in an existing model to do most of the heavy lifting in the training process, achieving similar levels in performance all while requiring less data from the target domain and a fraction of the iterations as does training from scratch. In particular, requiring less data is useful for applying this task to sample generation from domains limited by specificity or an abundance of examples to train from in real life without encountering mode failures. In addition, the relaxed requirements on data and training time also enable more training runs, observing GAN training behavior and testing GAN effectiveness in a more efficient manner.

2. Related Work

2.1. GANs (Goodfellow et al. 2014)

The novel two-network machine learning architecture,the generative adversarial network, is first proposed in this work by Ian Goodfellow and colleagues [3]. Most existing approaches to generative modeling in the machine learning community at the time consisted of single-network architectures, such as deep belief networks or stacked convolutional autoencoders [3].

Goodfellow and colleagues demonstrate the viability of GANs in image generation through experiments on

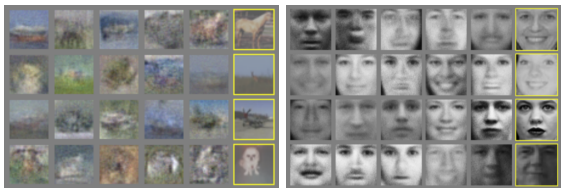

Figure 2. Left to right: Images generated by generative adversarial networks from Goodfellow et al. training on CIFAR-10 and TFD datasets [3]

the MNIST, CIFAR-10 and the Toronto Face Database datasets. From these early results of training GANs, the trained generator synthesized images that are reasonably representative of the real examples, as shown in Figure 2. The work concluded that although it was unclear whether GANs generate better samples than those of existing methods, GANs are "at least competitive" with then, and this served to "highlight the potential of the adversarial framework" [3]. Gaining mainstream attention, GANs have become the state-of-the-art approach to generative modeling using machine learning.

2.2. StyleGAN (Karras et al. 2019)

Although many variations of the original GAN architecture have achieved successful results, a persistent challenge was an inability to control the image synthesis process [7].

To address this issue, researchers from NVIDIA reworked the generator's fundamental structure with state-of-the-art architecture in StyleGAN. Traditional generators take initial latent code defined by a random distribution in the input layer only. In contrast, StyleGAN models take advantage of a standalone mapping network that maps the random latent vectors from Z to an intermediate latent space, W. Simply put, this mapping operation makes the features of the training dataset easier for the generator to learn. Consider that, in many datasets, the distribution of its features is often non-uniform. Consequently, random inputs from Z under-represent more common features while over-representing ones that are less prominent in the data. Figure 3 visualizes the changes made to Z, which now resembles the original data more closely [7].

Learned affine transformations are applied to the vectors in W to specialize them into various styles, as is referenced by the name "StyleGAN." These styles are

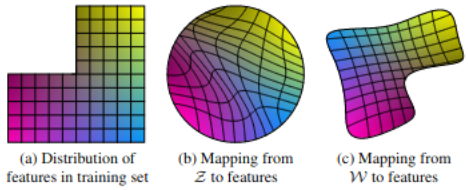

Figure 3. An illustration of the effects of the mapping network [7]. Note that the distribution in c) resembles that of a), the true data.

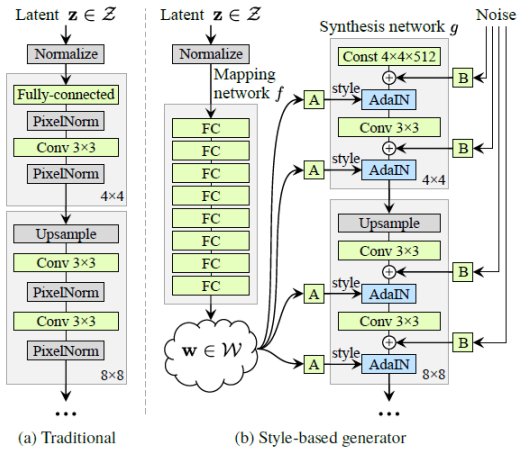

Figure 4. Traditional generator structure compared to that of StyleGAN [7]. Note the use of a mapping network to create an intermediate latent space, W.

then used "to control adaptive instance normalization" (AdaIN) operations after each convolution operation in the generator network. As the image is up-sampled, various styles can be chosen and applied to the image accordingly. As a result, the generator is capable of producing photorealistic images with fine features that can be tuned and adjusted. Furthermore, styles can be mixed across images, allowing for a remarkable variation in generated examples [7].

3. Materials and Methods

This project's investigation on using transfer learning in training GANs involves two components. Firstly, a data pipeline was created to assemble datasets used in training. This component involves the creation of tools that scrape images from Instagram given a search criterion and then run images through a filter, discarding unsuitable images and resulting in a set of ideal examples for training the GAN. Secondly, transfer learning

experiments investigated the impact of various combinations of source networks and target datsets.

3.1. Data Pipeline

An early goal of the project was devising tools and methods to assemble small datasets to be used in transfer learning. The transfer learning experiments involve training the networks on datasets of images topically different from those on which they were previously trained. Therefore, the ideal data source would be a repository of public images with some built-in method of searching for a particular topic or category. Instagram was identified as a candidate source for scraping images to form datasets due to its hashtag search function, allowing for a straightforward method to categorize raw image data by the hashtag an image was scraped from.

Instaloader is a Python package developed by Alexander Graf and Andre Koch-Kramer providing the underlying functionality required to scrape data from Instagram [4]. It uses a combination of other packages, namely Requests, and urllib to access posts by URL and download the image and video files contained in them. JSON files containing post metadata and text files containing captions, comments, and likes can also be downloaded. The scraper is written as a wrapper class that leverages Instaloader functionality to automate the data collection process in a streamlined manner. Passable parameters specify for any given scraping job the target hashtags to scrape posts from, subdirectory organization options, and the number of posts to download from. This way, the user has considerable modularity in how they desire to form a dataset, including its contents and categorization. A value of 1,000 posts to scrape from was selected, yielding a range of 1,500 to 3,000 images per hashtag due to the possibility of a given post containing multiple images.

The scraping process faced various setbacks. Instaloader's built-in Requests rate-controller class would temporarily lock the accounts from downloading post media to avoid being flagged by Instagram automation detection. This slowed down data collection for extended periods of time. In addition, hashtags themselves were more imprecise than anticipated. On one hand, Instagram users will tag a post with a flurry of hashtags regardless of how relevant they are to the media in the post. As long as a hashtag being searched for is in the caption, Instaloader downloads it. On the other hand, the media in a multi-image post gets downloaded, even if off-topic to the hashtag. In either case, these made the collected data of some hashtags far too varied and ambiguous for training purposes where using a dataset of similar images is crucial. To remedy the latter issue, an intentional, manual examination of a hashtag was performed prior to selection for scraping.

Figure 5. Examples of images scraped by Instaloader, after being given the hashtag "#latteart"

Training GANs with image data requires a standardized dataset of suitable images. For the purposes of this project, suitable images are those of sufficient quality and without elements that the generator will have significant trouble reproducing, namely text and human faces [1]. Concerning image quality, training the GAN with a significant number of images with low resolution or significant blurring leads to the generation of blurry samples. Without abundant data and a lengthy training process, neither of which are present in these experiments, the appearance of text and human faces leads to the generator's attempts to generate them, disrupting the photorealism of images for topics originally without human faces or text. Therefore, the training data must have a standard to prevent either phenomenon. A data filtration pipeline was devised to iterate through all collected images and discard those that did not pass various thresholds of certain target criteria. Figure 6 contains examples of images that would be rejected by these filters. Simply put, each criterion acts as a filter that the image has to pass through on the path to standardization and dataset assembly. These filters included:

- Valid image file format (.jpg, .png, .webp, etc...)

Figure 6. Left to right: Images of #fighterjet that would be rejected for low resolution, blurriness, prominent text, human faces. The generator can have considerable trouble when attempting to generate the latter two elements of images in particular.

- Pixel height and widths are both within at least 80% of a target resolution (i.e., 512×512)
- Sufficiently sharp images
- No grayscale images
- No significant or prominent bodies of text
- No visible human faces

Upon passing all filters, images are cropped and resized to a specified square size and then saved as new training examples. Numerous packages are used to process the image both in the filtering and resize operations. Pillow and OpenCV are two general-purpose Python modules used to read, edit, and save image data. For the facial recognition filter, GitHub user *ageitgey*'s Python module provides functionality to detect presence, location, and recurrence of human faces. For the text recognition filter, OpenCV's Frozen Efficient and Accurate Scene Text (EAST) Detection deep learning model provided creative ways to identify text [10]. The package also provides functionality to draw boxes over the identified text, which was used for threshold tuning and debugging the text detector.

Required fine-tuning of various filter thresholds remained a recurring challenge. The pipeline, at various filters, would either reject images that are indeed good training examples or accept images that are poor training examples, filtering too harshly at some checkpoints while filtering too leniently at others. Running tests on toy datasets, outputs would have to be examined to tune parameters. For example, the Frozen Efficient and Accurate Scene Text (EAST) Detection network that was used for the text checkpoint would, at times, falsely classify parts of the image as text, when in actuality, the image simply contains dark edges against a light

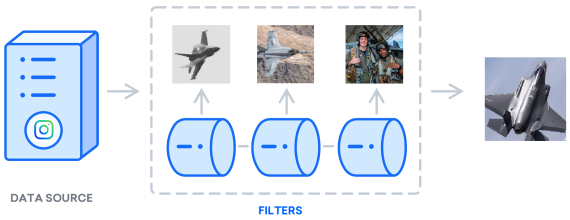

Figure 7. A diagram of the current working data filtration pipeline. Various filters flag unacceptable input images, leaving behind only usable images. Diagram modified from Digital Ocean [2]

.

Figure 8. Parameters need to be tuned when false positive and false negative classifications pass through the filter. Green boxes overlay all instances of text detection, true or false.

background, as depicted in Figure 8. Manual discernment needed to be made on a variety of classifications to find the right parameter. The same applied to the other filters as well, even if to a lesser extent.

3.2. Augmentation

Additionally, a data augmentation technique was required for all transfer learning runs. Data augmentation is a method of artificially increasing the number of training examples by random alteration to the data. Without sufficient data to learn from, previous work has shown that the discriminator network effectively memorizes the real images, the outcome of overfitting [6]. StyleGAN3 makes use of adaptive discriminator augmentation (ADA), an implementation introduced with StyleGAN2-ada. ADA is applied at the beginning of the training job, applying alterations such as isotropic image scaling, random 90 deg rotations, and color transformations, induced by an augmentation probability hyperparameter. With a typical Instagram dataset containing around 500-1,500 images, data aug-

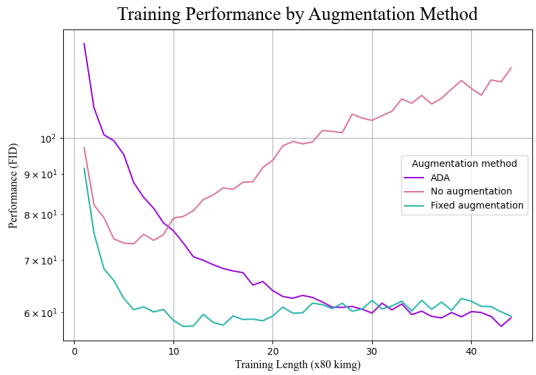

Figure 9. FID scores from runs using each available data augmentation technique.

mentation is a necessity. To observe the impacts of different augmentation methods, preliminary training runs were conducted. Figure 9, illustrates that transfer learning runs that used fixed or no augmentation techniques performed relatively worse, with the run using no augmentation blew up in its FID score, growing away from intended generation improvement.

3.3. Training

Training runs were conducted using the NVIDIA Corporation's official PyTorch implementation of StyleGAN3. The repository it comes from also includes scripts that handle dataset assembly and image generation. These experiments were conducted on a GPU cluster from the Advanced Research Computing (ARC) at Virginia Tech. Three nodes of eight GPUs were allotted for this project at all times. Using these 24 GPU's, each training run was typically conducted for 3,500 steps, measured as "*kimg*." Each *kimg* indicates one thousand images that pass through the generation-discrimination operation, meaning a single transfer learning job generates and discriminates images 3,500,000 times over. Training runs for 512×512 and 1024×1024 resolutions completed in around 15 and 19 hours, respectively. This training length was chosen after observing that a sizeable majority of runs reached an oscillatory state in performance by 3,500 kimg. Model performance was measured with the Fréchet Inception Distance (FID) score, which measures the similarity between two datasets. In the context of GANs, the FID score is used to assess the realism of generated images by comparing a set of synthesized images against its real counterpart, the training dataset. Lower FID scores correspond to greater similarity between two datasets.

3.3.1 Hyperparameters

To find a training run to serve as a control, initial transfer learning runs were conducted using NVIDIA's AFHQv2 (AFHQ) StyleGAN3 network. Training was continued from the network onto a target dataset of 651 beach sunset images scraped from the Instagram hashtag "#beachsunset." Multiple training runs were performed to find the optimal hyperparameters to use. These included adjustments to the generator and discriminator networks' learning rates, the R1 regularization penalty on the discriminator, and the chosen augmentation method. Combinations that resulted in stable training and the lowest FID scores serve as the control run moving forward. The choice to train a control run on beach sunsets was made based on a general dissimilarity to animal faces, which serves to reflect a generic dataset that could be used in transfer learning.

3.3.2 Varying Starting Networks

After selecting hyperparameters, the first training experiment involves the use of different pretrained networks as the starting points from which transfer learning is conducted on a common target dataset. Comparing the performance across runs trained with starting networks provides an indication of how transfer learning performs based on the pretrained network chosen.

Because transfer learning begins from a model's weights that were previously learned from training on a source domain, its performance is inevitably related to the contents of the source domain. This experiment observes any noteworthy behaviors that occur across the networks when continuing training on a target dataset. The hypothesis is that transfer learning performance varies depending on the starting network used. In particular, if the source domain matters in the transfer learning process, then not all pretrained networks will perform equally when training on the same target dataset. A pretrained network might perform better in transfer learning due to the complexity of the source dataset, as captured in the complexity of the model it was used to train. Conversely, similar performances achieved by all starting networks training on the same

Figure 10. A synthesized image of a corgi.

target dataset suggests that all source datasets provided sufficient prior knowledge to enable adaptation to the same target. This shared behavior might occur because each GAN learns similar core features of the image dataset. The latter result, although unlikely, is an interesting possibility, as it suggests that the choice of pretrained network does not affect training and that knowledge gaps between the source and target datasets were equally surmountable by each network.

Two other GANs pretrained by NVIDIA were used as starting networks in addition to the AFHQ network. While the AFHQ network was trained on images of animal faces, the FFHQ network was trained on images of human faces from Flickr photos, and the Metfaces network was trained on images of portraits from the Metropolitan Museum of Art [6, 7]. Table 2 details the sizes and resolutions of each dataset. Image datasets of fighter jets and latte art, both scraped from Instagram, served as the target datasets. The subjects of the target datasets are dissimilar to the subjects of the source datasets to prevent any one network from having an "advantage" over the others when training. In each instance of this experiment, each starting network was simultaneously trained on the same target dataset, and the performances across starting networks were subsequently compared for each target dataset.

Source Dataset	Resolution	Images
Animal Faces	512×512	16,130
Flickr Human Faces	1024×1024	70,000
Met Portraits	1024×1024	1,635

Table 1. Table of pretrained network, resolution, and source dataset size.

3.3.3 Varying Target Datasets

The second training experiment involves the use of different target datasets on which to conduct transfer learning from the same pretrained starting network. Comparing the performance across runs trained on different target datasets provides insight into which kinds of datasets are more effective targets for transfer learning.

Like any other data type, image datasets exist as data distributions as well. As is the task of generative modeling, a GAN, when trained on a given dataset, is learning to model the real distribution of training examples. Intuitively, more complex distributions of image data will be more challenging to model. This experiment explores how substantive differences in dataset content factor into this learnability when conducting transfer learning. However, characterizing the overall contents and complexity of an image dataset remains difficult. Even while image representation is quantitative (e.g., three-dimensional matrices of pixel values that exist across spectrums of measurable characteristics), perhaps it is useful to assess this representation of contents with their qualitative aspects too, as described by their semantically meaningful hashtags. The initial hypothesis is that datasets containing images with well-defined subjects, high contrast, and similar coloration are easier to learn features from and will result in stronger transfer learning performance. Most importantly, images with consistent spatial positioning are also easier to train with, that is, how objects of prominence are situated in the image relative to the background.

Considering these qualities of interest, new target datasets were reviewed and selected for containing the most ideal training examples. The results would confirm or challenge these as factors influencing transfer learning performance. Among the new target datasets that were reviewed and selected for training included corgis, betta fish, bald eagles, fighter jets, trains, pizza, latte art, flowers, and mountains.

4. Result analysis

The generator trained during the control run was capable of synthesizing plausible samples of beach sunsets. However, a sizeable portion of these images contained chaotic, random generations of differing quality.

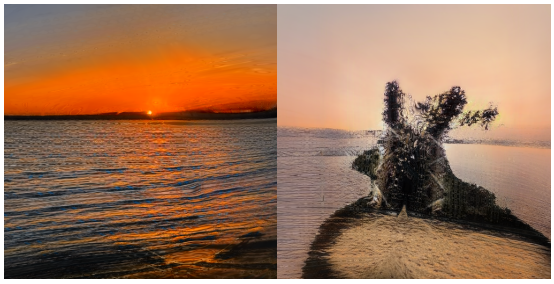

Figure 11. Beach generations.

Examples of good and poor generations are shown in Figure 11.

4.1. Experiment 1: Varying Starting Network

When performing transfer learning to a target domain of latte art, the animal faces generator network appears to cease learning while the Flickr faces and Metfaces networks continue to improve. More specifically, the animal faces network's learned generation does not improve in realism, as indicated by the non-decreasing FID score. While all three networks performed similarly when training on the target dataset of fighter jets, this was not true for the target dataset of latte art. Thus, it is reasonable to suspect that there is source-to-target suitability mapping that is impactful in transfer learning on GANs.

This indicates that there exists some incompatibility between source and target datasets when training. It is unclear as to what the incompatibility is. In terms of the occurrence illustrated in 12, it is inferred that the model weights learned from the Animal Faces dataset are not adapting to the dataset of latte art because of some relationship between the two datasets that are halting the model weight updates, resulting in the impasse. Although this result is not particularly surprising, it is important nonetheless. Such a result calls for a mindful selection of pretrained networks and target datasets that are deemed compatible with each other.

4.2. Experiment 2: Varying Target Datasets

The continued training from the starting network exhibited differential performance in transfer learning depending on the target dataset it was trained on, as illustrated in 13. Certain target datasets performed well relative to the others, like the runs trained on the datasets of betta fish and corgis. These runs exhibited strong performance with the lowest FID scores. In conjunction, these runs resulted in trained generators that could synthesize realistic images consistently. The training run on betta fish in particular achieved the lowest FID score of all runs, and its resulting generations featured in Figure 14 show decently realistic samples in which the sample fish possess anatomically correct fins and eyes. Considering that the purpose of transfer learning is to utilize previously learned information to train the generation of realistic examples with less time and data, this bodes well, as achieving that goal is clearly within the realm of possibility. However, not all transfer learning runs with these target datasets performed as well, such as the run trained on the dataset of latte art. Like the previous experiment, a starting network of animal faces and a target dataset of latte art, the generator trained in this run appeared to reach some block in its learning and stopped generating more plausible samples, as indicated by the largely non-decreasing FID score. An examination of its resulting generations reasonably supports this result, as the backgrounds of these synthesized images contain significant warping and chaotic additions. Rather in-

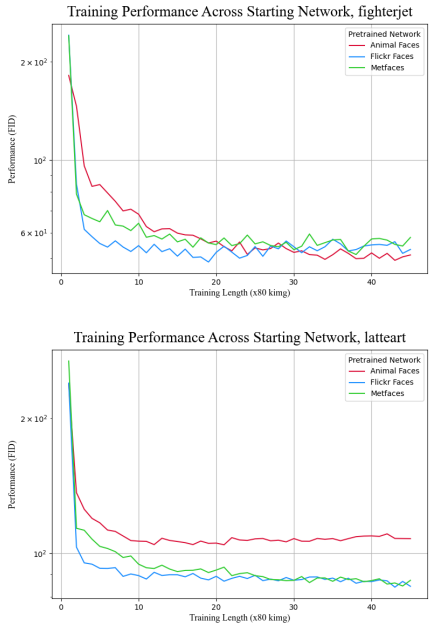

Figure 12. Performances resulting from transfer learning runs starting from different pretrained networks. Three NVIDIA-trained models were available: AFHQ, FFHQ, and Metfaces.

Figure 14. Generations of betta fish generated during the training process (left) and additional generation of betta fish from the generator (right).

Figure 15. Generations of latte art. Note the realistic quality of the detail in the latte art.

teresting, however, is that, while the examples are less realistic based on the FID score, the generator appeared to learn how to generate specific and detailed patterns well, namely the actual latte art on the coffee.

Figure 13. Performances resulting from transfer learning runs trained on differing target datasets.

By the current understanding, it is difficult to state which attributes of a source-target dataset pairing lead to a stronger or weaker transfer learning performance. All target datasets were selected with the expectation that they contained the best average training examples for their respective domains, but as demonstrated, certain transfer learning runs trained on these target datasets underperformed expectations. This differential performance inspires a succeeding hypothesis that there exist certain "attributes" of the image dataset and its contents that significantly affect the GAN learning process. Furthermore, the combinations of these attributes and how they interact also matter greatly. These attributes may be hidden, latent, and obscure while describing the qualitative aspects of the dataset more closely. The manner in which these attributes appear can be speculated upon.

Target Dataset	Minimum FID Score
Beach sunsets	50.949
Fighter jets	49.353
Latte art	104.555
Betta fish	41.417
Bald eagles	59.002
Trains	60.140
Corgis	46.317
Mountains	57.636

Table 2. Minimum FID scores by transfer learning with target datasets.

1) Source and target dataset distance – Firstly, the similarity between the target datasets and the source dataset appeared to impact transfer learning. Considering that this AFHQ network was trained on images of animal faces, it is worth noting that the transfer learning runs that trained on target datasets of corgis and betta fish outperformed most runs trained on target datasets of objects and settings, such as trains and mountains. Furthermore, although examples of fish are not included in the animal faces source dataset, the transfer learning run on betta fish still performed the best, suggesting that the prior knowledge of the network may have enabled robust adaptation to other kinds of animals with animal-like qualities, as seen with the anatomical accuracy in body texture and eye placement that exists in such generations.

2) Variability – Secondly, variability between the images within a given target dataset seemed to have a role in transfer learning. Not only is "variability" of these attributes subjective, but it can apply to many qualities of an image, including lighting patterns, coloration, topic relevance, and even spatial layout.

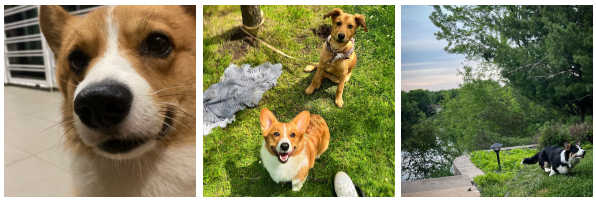

Figure 16. Even when scraped under the same hashtag, "#corgi," training examples can vary greatly in their contents. This could present a possible challenge when training GANs.

The lattermost refers to how elements of an image are positioned, such as main subject of the image. This quality is of particular interest, as the nature of Instagram images as social media content lends itself to more randomness in how image are taken than would be in a curated dataset.

Based on the results, variability in this spatial layout appeared to be considerably impactful, as indicated by the image-scraping process. While the images in the best-performing dataset of betta fish were scraped from the hashtag "#bettaphotgraphy," nearly all of them were posted by a professional account that specializes in betta fish photography. As a result, most examples are the side profiles of betta fish as a sole subject against a black backdrop. On a larger scale, this could be an underlying reason behind the photorealism of generations from the animal faces network, whose training examples were near-uniform in their layout, consisting of the animal's face and little inclusion of the background. This is the stark difference from the Instagram-scraped datasets of fighter jets or latte art, which contain a much wider variety of visual contexts in which the picture was taken.

5. Conclusions and Future Work

This work serves as a preliminary exploration of the applications of transfer learning to train GANs on domains limited by specificity or abundance of examples. It studies its successes and limitations to characterize its effectiveness and efficiency as a training method.

Besides examining transfer learning, this work demonstrates that Instagram has served as a viable method of forming datasets to use in GAN training. A decent degree of realism is achievable by the network when trained on Instagram-scraped examples, indicating their suitability for training. However, a concern when scraping for training examples is the potential for randomness, both in the relevance to the hashtag scraped from and how objects in the image are situated. To offset this issue, a careful selection of the hashtag search criterion is required to reduce the variability in each. In conjunction with filtering tools, a dataset of high-quality images suitable for GAN training can be formed.

This work demonstrates the promising future that transfer learning has in its application to training generative adversarial networks. Attaining near-realistic generations after training from a pretrained GAN is well within the realm of possibility under the right training conditions. However, describing the methodology for achieving such results remains a challenge. The limits on dataset size and the stochastic nature of examples from Instagram add to the unpredictability of training results. Such a challenge is worth overcoming, as it addresses the motivations of transfer learning in the first place. When attempting to train high-performing GANs with fewer training examples and less time, a greater understanding of factors influencing the transfer learning process on GANs will grant future users more systematic methods to do so.

Of course, future work requires a broadening of transfer learning experiments to even more diverse datasets of varying sizes and subjects, as well as selecting more pretrained networks to train with, as the only networks used in this work are those homegrown from NVIDIA. Just like the experiments featured in this work, this will confirm or rebut existing intuition on what source-target pairings are optimal for transfer learning.

However, the future work that remains the most urgent and effective is the definition of metrics and metadata to describe attributes of image datasets. As previously discussed, these attributes and how they interact may be latent while being highly impactful in transfer learning. Based on the results, there is reason to believe that these attributes exist. Defining and applying these metrics to datasets serves to quantify the qualitative aspects of a dataset in order to establish some window of expectation in the effects they and their interaction have on transfer learning.

When used on source datasets, one such metric can be a measurement of differences between source datasets to learn more about which source domains

might be best and most compatible when conducting transfer learning to a target domain. When used on target datasets, metrics should be defined to express the contents captured in an image to make sense of what kinds of images are more suitable beyond just the naked eye. Such metrics can be pixel-based measurements that measure a dataset's consistency in spatial positioning or even topic relevancy. Although it is yet to be seen how such metrics will be calculated, this is undeniably the most helpful step to making transfer learning a viable training method using small datasets.

References

[1] Vincent-Pierre Berges, Isabel Bush, and Pol Rosello. Text generation using generative adversarial networks. 2017.

[2] Justin Ellingwood and Vadym Kalsin. How to install elasticsearch, logstash, and kibana (elastic stack) on ubuntu 16.04. 2016.

[3] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Advances in Neural Information Processing Systems*, 2014.

[4] Alexander Graf and Andre Koch-Kramer. Instaloader. 2016.

[5] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaako Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 2021.

[6] Tero Karras, Miika Aittala, Samuli Laine, Erik Harkonen, Janne Hellsten, Jaako Lehtinen, and Timo Aila. Training generative adversarial networks with limted data. *Advances in Neural Information Processing Systems*, 2021.

[7] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *Advances in Neural Information Processing Systems*, 2019.

[8] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip H. S. Torr. Controllable text-to-image generation. *Computer Vision and Pattern Recognition*, 2019.

[9] Sadrach Pierre. 5 kaggle data sets for training gans. 2020.

[10] Adrian Rosebrock. Opencv text detection (east text detector). 2018.

[11] Lars Ruthotto and Eldad Haber. An introduciton to deep generative modeling. *GAMM-Mitteilungen*, 2021.

[12] Isha Salian. Nvidia research achieves ai training breakthrough using limited datasets. *NVIDIA Blogs*, 2020.

[13] Jan Scholz. Genetic algorithms and the traveling salesman problem a historical review. *Neural and Evolutionary Computing*, 2019.