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

Brian Lee
Undergraduate Student, Virginia Tech, Blacksburg, VA
Logan Eisenbeiser
Blue Sky Innovators, Fairfax, VA
Erin Lanus, Ph.D.
National Security Institute, Virginia Tech, Arlington, VA

Abstract: This work explores the capabilities of generative adversarial networks (GANs) in image generation using transfer learning on limited datasets assimilated from Instagram and the creation of a pipeline of tools 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. This work demonstrates the potential of pretrained networks in adapting to new domains of data, even when retraining on small datasets.

Keywords: generative adversarial networks, transfer learning, dataset creation, small datasets

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 samples from the real distribution [1]. Generative adversarial networks (GANs) offer an innovative solution to this task with an architecture of a generator network and a discriminator network pitted against one another as adversaries (Figure 1). The generator is tasked with learning the distribution of a real dataset to generate synthetic samples mimicking the training data. The discriminator is tasked with learning to correctly classify real and synthetic samples while providing feedback to the generator on its success [2]. Random noise is often fed into the generator as a starting point for generating the artificial samples, although that is not always the case.
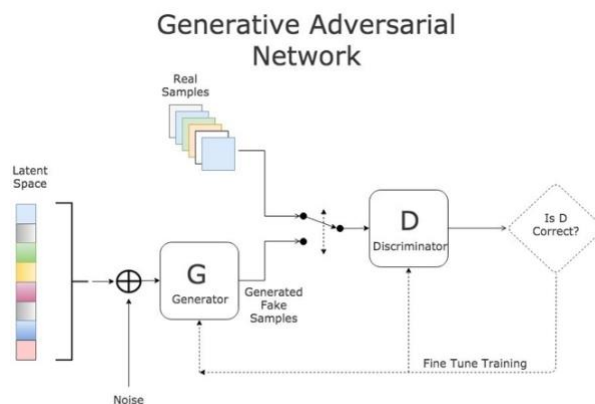

Figure 1. Basic generative adversarial network structure [17].

Together, these networks and their loss functions compete in a two-player minimax game. While the discriminator learns to maximize its classification successes, the generator learns to generate samples 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, p_g, is learned to resemble the distribution of the training data, 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 [2]

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

Currently, image data is the most popular domain for GANs training [3]. Not only was the first GAN 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 [2], photo-realistic human face generation [4], and text-to-image generation [5].

In many published works, GANs are trained from scratch, but doing so is difficult. Training on large datasets over extended periods can prove costly in time and computation, especially with image data [4]. According to NVIDIA, the ideal GAN is trained with 50,000 to 100,000 images. However, arbitrarily decreasing the size of the dataset to save computational resources often results in various training failures. The most common failure is when the discriminator overfits to the data, becoming overly adept at classifying real and generated images, forcing the generator to synthesize nonsense to fool the discriminator [6]. Additionally, datasets from some domains may be limited by specificity or an abundance of samples to train from in real life.

A proposed 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 source dataset, such as edges, curves, coloration, and lighting patterns. Ideally, the prior knowledge learned from the source domain enables the network to adapt to the target dataset when it continues training, requiring less data from the target domain and a fraction of the iterations of training from scratch while achieving similar levels of performance and 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.

This work's primary objective is to evaluate the efficiency and effectiveness of transfer learning to train GANs on small datasets by exploring the method's successes and limitations under various combinations of source and target domains. A secondary objective is to assess the viability of using Instagram as an effective platform to assemble datasets for transfer learning.

2. Related Work

2.1. GANs

Previous approaches to generative modeling in the machine learning community consist of single-network architectures, such as deep belief networks or stacked convolutional autoencoders. The viability of the novel GAN two-network architecture for image generation is demonstrated through experiments on the MNIST, CIFAR-10, and the Toronto Face Database datasets [2]. In the early results, the trained generator synthesized images that are reasonably representative of the real samples (Figure 2), demonstrating the potential of GANs to be competitive with previous methods. Gaining mainstream attention, GANs have become the state-of-the-art approach to generative modeling using machine learning.

Figure 2. Images generated by generative adversarial networks from training on CIFAR-10 (left) and TFD (right) datasets [2].

2.2. StyleGAN

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]. Traditional generators take initial latent code defined by a random distribution in the input layer only. In many datasets, the distribution of features is non-uniform, so random inputs from latent space Z under-represent more common features while over-representing ones less prominent in the data. To address this issue (Figure 3b), StyleGAN models take advantage of a standalone mapping network that maps the random latent vectors from Z to an intermediate latent space, $W,$ (Figure 3c) making the features of the training dataset resemble the original data (Figure 3a) more closely and easier for the generator to learn.

(a) Distribution of features in training set   (b) Mapping from Z to features   (c) Mapping from W to features
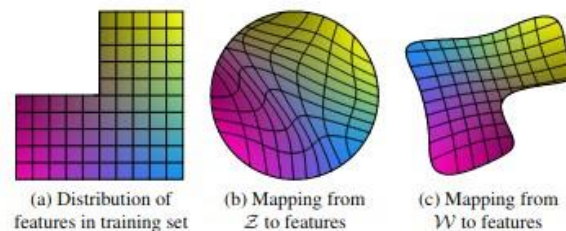
Figure 3. An illustration of the effects of the mapping network [7].

The name StyleGAN refers to the learned affine transformations applied to the vectors in W to specialize them into various styles "to control adaptive instance normalization" (AdaIN) operations after each convolution operation in the generator network (Figure 4). As the image is up-sampled, styles are chosen and applied accordingly, resulting in a generator capable of producing photorealistic images with fine features that can be tuned and adjusted. Furthermore, style mixing allows for remarkable variation in generated samples [7].
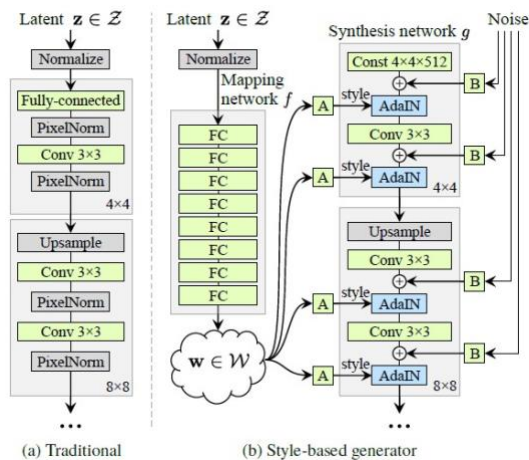

Figure 4. Traditional GAN structure (a) compared to that of StyleGAN (b) with mapping network to create an intermediate latent space, W [7].

3.   Materials and Methods

The current work involves two main efforts: 1) a data pipeline to scrape, filter, and augment images to assemble training datasets for transfer learning and 2) transfer learning experiments to investigate the impact of various combinations of source networks and target datasets. Model performance is evaluated with the Fréchet Inception Distance (FID) score, which measures the similarity between two datasets; lower FID scores correspond to greater similarity. 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. Training runs are measured in *kimg* units of one thousand images passing through the generation-discrimination operation.

3.1.   Data Pipeline

The data pipeline constructed for this work includes tools to scrape images given a search criterion and then run images through filters, discarding unsuitable images and resulting in a small set of ideal samples for training the GAN. The transfer learning experiments involve training the networks on datasets topically different from those on which they were previously trained. The ideal data source is a repository of public images with a built-in method of searching for a topic.

Instagram is a candidate platform for scraping images for datasets as its hashtag search function provides a straightforward method to categorize raw image data by hashtag (Figure 5).

Figure 5. Examples of images scraped by Instaloader with #latteart hashtag.

Instaloader is a Python package providing the functionality to scrape data from Instagram [8]. It uses the packages Requests and urllib to access posts by URL and download the image files contained in them. The scraper tool is written as a wrapper class that leverages Instaloader functionality to automate the data collection process in a streamlined manner. Passable parameters specify subdirectory organization options and the target hashtags and number of posts to download from, supplying considerable modularity over dataset formation, 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 as a given post may contain 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 Instagram automation detection, slowing data collection for extended periods of time. Additionally, hashtags were less precise than anticipated, making the collected data of some hashtags far too varied and ambiguous for training purposes where using a dataset of similar images is crucial. For example, Instagram users often tag posts with many hashtags regardless of topic relevance to the post media, and not all media in a multi-image post is relevant to each hashtag listed. To remedy the latter issue, a manual examination of a hashtag was performed prior to use for scraping.

Training GANs with image data requires a standardized dataset of suitable images. For this work, suitable images are those of sufficient quality and without elements that the generator will have difficulty reproducing, namely text and human faces [9]. Training the GAN with images of 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 inclusion of text and human faces leads to the generator's attempts to generate them, disrupting the photorealism of images for topics without human faces or text. A data filtration pipeline was devised to iterate through collected images and discard those that did not pass various thresholds of target criteria; examples of discarded images are in Figure 6. These filters included:

- valid image file format;
- pixel height and widths are both within 80% of a target resolution;
- sufficiently sharp images;
- no grayscale images;
- no significant or prominent bodies of text; and
- no visible human faces.

Figure 6. Images of #fighterjet that would be rejected for (left to right) low resolution, blurriness, prominent text, and human faces.

Upon passing all filters, images are cropped, resized, and saved as new training samples (Figure 7). 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, a Python module provides functionality to detect presence, location, and recurrence of human faces [10]. For the text recognition filter, OpenCV's Frozen Efficient and Accurate Scene Text (EAST) Detection deep learning model provides creative ways to identify text and functionality to draw boxes over the identified text, used for threshold tuning and debugging. [11].
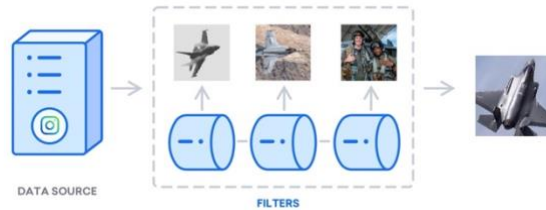

Figure 7. A diagram modified from Digital Ocean [16] of the current data filtration pipeline.

Fine-tuning of various filter thresholds remained a challenge. The pipeline would filter too harshly at some checkpoints and too leniently at others, rejecting good or accepting poor training samples. For example, the Frozen EAST Detection network would occasionally falsely classify parts of the image as text, when in fact the image simply contains dark edges against a light background (Figure 8). Manual discernment on a variety of classifications was required to tune parameters on this and other filters.

Figure 8. Green boxes overlay all instances of text detection, true or false.

3.2.   Augmentation

Data augmentation is a method of artificially increasing the number of training samples by random alteration of the data. Without sufficient data to learn from, previous work has shown that the discriminator network overfits and effectively memorizes the real images [12]. Adaptive discriminator augmentation (ADA) is an implementation introduced with StyleGAN2-ada. ADA starts at the beginning of the training job, applying alterations such as isotropic image scaling, random 90-degree rotations, and color transformations, induced by an augmentation probability hyperparameter. Data augmentation is necessary for a typical Instagram dataset containing 500 to 1,500 images. Preliminary training runs were conducted to observe the impact of augmentation methods. Runs with ADA augmentation consistently improved, while runs with both fixed and no augmentation did well initially but got worse as training went on, and no augmentation resulted in FID scores substantially worse than no training at all (Figure 9).

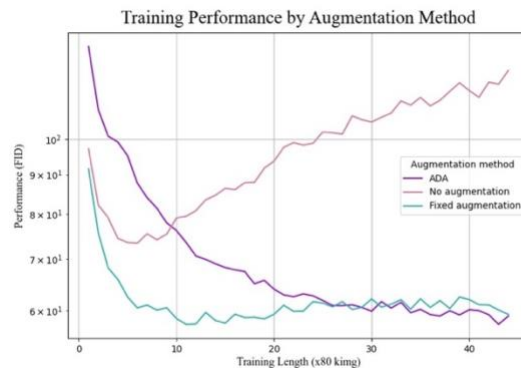### 3.3.   Training and Hyperparameters

Figure 9. FID scores for each data augmentation technique.

Training runs were conducted using NVIDIA's official PyTorch implementation of StyleGAN3 and their scripts handling dataset assembly and image generation. Experiments were conducted on a GPU cluster from the Advanced Research Computing (ARC) at Virginia Tech utilizing three nodes of eight GPUs. Each training run was typically conducted for 3,500 kimg steps, meaning a single transfer learning job generates and discriminates images 3,500,000 times over. This training length was chosen after observing that a sizeable majority of runs reached an oscillatory state in performance by 3,500 kimg. Training runs for 512×512 and 1024×1024 resolutions completed in around 15 and 19 hours, respectively.

Multiple initial transfer learning runs were conducted using NVIDIA's AFHQv2 StyleGAN3 network (AFHQ) to find optimal hyperparameters for experiments. 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. Training was continued from the AFHQ network onto a target dataset of 651 beach sunset images scraped from the Instagram hashtag #beachsunset. Beach sunsets were chosen for the control run based on a dissimilarity to animal faces, reflecting a generic dataset used for transfer learning. The control run generator was capable of synthesizing plausible images of beach sunsets, but a sizeable portion were chaotic and random (Figure 10).

Figure 10. Examples of plausible (left) and chaotic (right) beach generations.

4. Experiments and Results

4.1. Varying Starting Networks

The first experiment used different pretrained networks as the starting points from which transfer learning is conducted on a common target dataset, exploring the impact of source network on transfer learning. Because transfer learning begins from a model's weights that were previously learned from training on a source domain, performance may be related to the contents of the source domain or differences in knowledge captured due to the architecture or hyperparameters. 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. Conversely, similar performance achieved by all starting networks transferring to the same target dataset may suggest that all source datasets provided sufficient prior knowledge captured by the pretrained networks to enable adaptation to the same target. This shared behavior might occur if each GAN learns similar core features of the image dataset or if knowledge gaps between the source and target datasets were equally surmountable by each network. Due to the use of pretrained networks without controlling for architecture and hyperparameters, this experiment cannot attribute cause of performance differences to the source dataset versus the architecture or hyperparameters of the starting 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 [12] [7]. Table 1 details the size and resolution of each dataset. Images of fighter jets and latte art scraped from Instagram served as the target datasets.

Table 1. Source dataset name, resolution, and size for each pretrained network.

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

While all networks performed similarly when training on the target dataset of fighter jets, this was not true for the target dataset of latte art (Figure 11). When performing transfer learning to a target domain of latte art, the animal faces generator network appears to cease learning as indicated by the non-decreasing FID score, while the Flickr faces and Metfaces networks continue to improve. Thus, there may be a source-to-target suitability mapping that is impactful in transfer learning on GANs, but the experiments do not shed light on how to measure the suitability. Source dataset size does not correlate with performance either early or late in training for either target domain, suggesting performance differences are not a caused by size of training data.
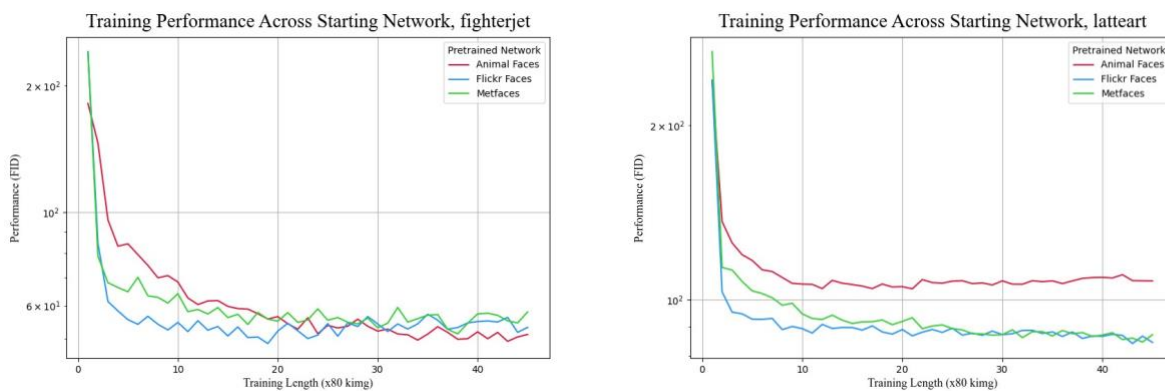

Figure 11. Transfer learning performance of different starting networks to #fighterjet (left) and #latteart (right) datasets.

4.2.   Varying Target Datasets

The second experiment began with the same pretrained starting network, the AFHQ network trained on images of animal faces, and transferred to different target datasets to explore how substantive differences in target dataset content factor into transfer learning. It is expected that datasets containing images with well-defined subjects, high contrast, and similar coloration composed of images with consistent spatial positioning, or how objects of prominence are situated in the image relative to the background, are easier to learn features from and result in stronger transfer learning performance. Considering these qualities of interest, new target datasets were reviewed and those containing the most ideal training samples were selected. These topics included corgis, betta fish, bald eagles, fighter jets, trains, pizza, latte art, flowers, and mountains. The continued training from the starting network exhibited differential performance in transfer learning depending on the target dataset it was trained on (Figure 12 and Table 2). Inspection of resulting images for the lowest FID scores, betta fish and corgis (Figure 13), demonstrated that these generators synthesize realistic images consistently.
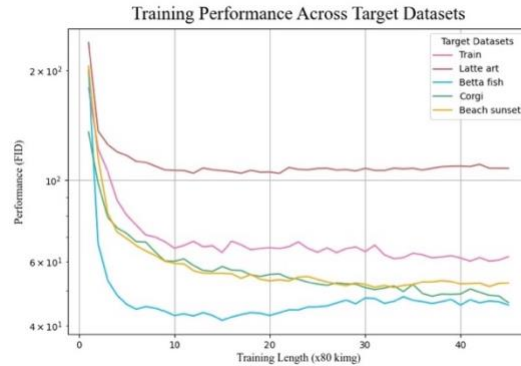
Figure 12. Transfer learning performance of animal faces to different target datasets.

Table 2. Minimum FID scores achieved per target dataset by transfer learning.

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

Figure 13. Generations of betta fish (left, middle) and a corgi (right).

However, not all transfer learning runs with these target datasets performed as well, such as the run with the target dataset of latte art. Like the previous experiment with 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, as indicated by the largely non-decreasing FID score. An examination of its resulting generations supports this result, as the backgrounds of these images contain significant warping and chaotic additions; however, the generator appeared to learn how to generate the art on the coffee (Figure 14).

Figure 14. Generations of latte art with realistic art patterns but chaotic backgrounds or misshapen cups.

These experiments do not allow us to determine 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 samples for their respective domains, yet the target datasets exhibited differential transfer learning performance. We hypothesize that there exist attributes of an image dataset that significantly affect the GAN learning process related to 1) source-target dataset distance and 2) variability.

The similarity between the target datasets and the source dataset may impact transfer learning. Considering that the source dataset in the second experiment was trained on images of animal faces, it is worth noting that the transfer learning to target datasets of corgis and betta fish outperformed runs trained on target datasets of objects and settings, such as trains and mountains. Though samples of fish are not included in the animal faces source dataset, the prior knowledge of the animal faces 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 of the betta fish. The poorer performance on the bald eagle target serves as a counterexample to the hypothesis that topical similarity was important, however.

Variability between the images within a given target dataset may play a role in transfer learning. Not only is variability subjective when not quantitatively measured, but it can apply to many qualities of an image, including lighting patterns, coloration, topic relevance, and even spatial layout. The lattermost refers to how elements of an image are positioned, such as the placement of the 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 images are taken than would be in a curated dataset. Based on the results, variability in this spatial layout appeared to be 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 samples 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 samples 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. The corgi dataset also exhibited more variability in spatial layout (Figure 15) than the betta fish dataset which is expected to present a challenge for training GANs and may explain some the difference between the betta fish and corgi FID scores in early training.
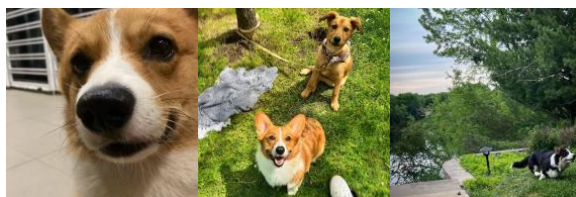

Figure 15. Variability of training samples scraped from the same #corgi hashtag.

5.   Conclusions and Future Work

This work demonstrates Instagram as a viable method for forming datasets to use in GAN training, as a decent degree of realism is achievable by a network when trained on Instagram-scraped

samples. A concern when scraping samples is the potential for randomness, however, both in the relevance to the hashtag scraped from and how objects in the image are situated. In conjunction with filtering tools, a careful selection of the hashtag search criterion is required to reduce the variability and form datasets of high-quality images suitable for GAN training.

This work serves as a preliminary exploration of transfer learning to train GANs on domains limited by specificity or abundance of samples. Attaining near-realistic generations after transfer learning from a pretrained GAN is possible under the right training conditions, but limits on dataset size and the stochastic nature of samples from Instagram lead to unpredictable results. Describing the methodology for achieving good results remains a challenge. A greater understanding of factors influencing the transfer learning process on GANs could yield systematic methods to train high-performing GANs more efficiently in terms of training samples and time.

The future work that remains the most pressing is the definition of metrics and metadata to describe attributes of image datasets. These attributes and how they interact may be latent while being highly impactful in transfer learning. The differential performance demonstrated by this work suggests that these attributes exist. However, characterizing the contents and complexity of an image dataset remains difficult. Image representation itself is quantitative as three-dimensional matrices of pixel values that exist across spectrums of measurable characteristics. One hypothesis is that datasets containing images with well-defined subjects, high contrast, and similar coloration will result in stronger transfer learning performance. Metrics that may be useful for determining suitability for training GANs on small target datasets may be quantitative pixel-based measurements, such as the consistency in spatial positioning across images of a dataset. Defining metrics to quantify the qualitative aspects of a dataset, such as topic relevancy via semantically meaningful hashtags, could establish a window of expectation in the effects they and their interactions have on transfer learning. One metric that can be applied to either latent features or metadata is a measurement of difference between source and target datasets in transfer learning, and might be useful for choosing the most compatible source for a target domain [13] [14] [15].

Author Bios

BRIAN LEE is a senior at Virginia Polytechnic Institute and State University studying for a degree in Computational Modeling and Data Analytics along with minors in Computer Science and Mathematics. His research experience includes an internship with the Virginia Tech National Security Institute's Intelligent Systems Division during Summer 2022 exploring the effectiveness of transfer learning to train GANs on small datasets and as an undergraduate research assistant studying combinatorial coverage of datasets for machine learning model performance during academic year 2022-2023.

LOGAN EISENBEISER is Research Engineer at Blue Sky Innovators, where he supports a wide variety of research projects across different machine learning domains. Logan's primary research interests are generative machine learning and conditional generation, with a focus on applications to real-world problems. Logan has a B.S. and M.S. in computer engineering, both from Virginia Tech.

ERIN LANUS, Ph.D., is a Research Assistant Professor at the Virginia Tech National Security Institute. Her research applies combinatorial testing to security and AI assurance. She leverages metric and algorithm development for constructing test sets, measuring the quality of data sets, and evaluating systems

with embedded AI. Dr. Lanus has a B.A. in Psychology and a Ph.D. in Computer Science, both from Arizona State University.

Acknowledgements

References

[1] L. Ruthotto and E. Haber, "An introduction to deep generative modeling," *GAMM-Mitteilungen,* vol. 44, no. 2, p. e202100008, 2021.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative adversarial networks," *Communications of the ACM,* vol. 63, p. 139–144, 2020.

[3] S. Pierre, "5 Kaggle Data Sets for Training GANs," 2020. [Online]. Available: https://towardsdatascience.com/5-kaggle-data-sets-for-training-gans-33dc2e035161.

[4] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen and T. Aila, "Alias-free generative adversarial networks," *Advances in Neural Information Processing Systems,* vol. 34, p. 852–863, 2021.

[5] B. Li, X. Qi, T. Lukasiewicz and P. H. S. Torr, "Controllable Text-to-Image Generation," in *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, Canada, 2019.

[6] I. Salian, "NVIDIA Research Achieves AI Training Breakthrough Using Limited Datasets," 2020. [Online]. Available: https://blogs.nvidia.com/blog/2020/12/07/neurips-research-limited-data-gan/.

[7] T. Karras, S. Laine and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4396-4405.

[8] A. Graf and A. Koch-Kramer, "Instaloader," 2016. [Online]. Available: https://instaloader.github.io/index.html.

[9] V.-P. Berges, I. Bush and P. Rosello, "Text Generation using Generative Adversarial Networks," Stanford University, 2017.

[10] A. Geitgey, "Face Recognition," [Online]. Available: https://github.com/ageitgey/face_recognition.

[11] A. Rosebrock, "OpenCV Text Detection (EAST text detector)," 2018. [Online]. Available: https://pyimagesearch.com/2018/08/20/opencv-text-detection-east-text-detector/.

[12] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen and T. Aila, "Training generative adversarial networks with limited data," *Advances in Neural Information Processing Systems,* vol. 33, p. 12104–12114, 2020.

[13] E. Lanus, L. J. Freeman, D. Richard Kuhn and R. N. Kacker, "Combinatorial Testing Metrics for Machine Learning," in *2021 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, 2021.

[14] S. F. Ahamed, P. Aggarwal, S. Shetty, E. Lanus and L. J. Freeman, "ATTL: An Automated Targeted Transfer Learning with Deep Neural Networks," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021.

[15] T. Cody, E. Lanus, D. D. Doyle and L. Freeman, "Systematic training and testing for machine learning using combinatorial interaction testing," in *2022 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, 2022.

[16] J. Ellingwood and V. Kalsin, "How To Install Elasticsearch, Logstash, and Kibana (Elastic Stack) on Ubuntu 16.04," 2016. [Online]. Available: https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-elastic-stack-on-ubuntu-16-04.

[17] J. Scholz, "Genetic Algorithms and the Traveling Salesman Problem a historical review," *arXiv preprint arXiv:1901.05737,* 2019.