ProSeminar: "Synthetic Image Generation"

Koray Ulusan (5788277) koray.ulusan@student.uni-tuebingen.de

December 18, 2023

Abstract: *Image generation is very influential in today's world, affecting the daily lives of many people. The innovations of GANs have led to many advances which contributed to other fields. In this report, the generator-discriminator structure of GANs is explained with its training algorithm and the problems it poses. It is summarized in a way that is understandable to a novice in the field. Capabilities of GANs and CycleGAN are explored to give a deeper insight into the subject and then related to today's models. Many GAN architectures are presented, and concepts are linked to other aspects of machine learning to give a comprehensive overview of the topic.*

Keywords: GANs, Image Generation, CycleGAN

Contents

1 Introduction

Image generation is an old field dating back to the 1980s. Many breakthroughs have changed the field since then. Bridging the older methods to modern ones are generative adversarial networks (GANs) [Goodfellow *et al.* (2014)]. GANs were highly influential in synthetic image generation and led to many innovations. Their ability to generate distributions of images has many applications in computer vision as in medicine, agriculture, and numerous other areas.

2 Earlier approaches

2.1 Deep Belief Networks

In the mid-2000s, Deep Belief Networks (DBNs) [Hinton *et al.* (2006)] utilized Restricted Boltzmann Machines to generate images. As seen in Figure 1, generated images have artifacts and are sometimes unrecognizable. Using Markov Chain Monte Carlo (MCMC) methods leads to problems in training [Makhzani *et al.* (2016)] and requires many steps, hence great computational power [Thompson *et al.* (2022)].

Figure 1: Samples from a DBN [Hinton *et al.* (2006)]

Following DBNs, deep generative models gained popularity, which use latent vectors.

2.1.1 latent vectors

Latent vectors can be thought of as compressed, low-dimensional representations of an image. In plain English, “latent” means “hidden”. Learning latent representation enables

models to generate all images of a distribution, e.g. all cat photos, by sampling new latent vectors from a random distribution, most commonly a multivariate Gaussian.

After training, new images are created by feeding latent vectors to the model. Simply put, if the same latent vector is used, the model will generate similar-looking photos. Latent vectors introduce variability to the generated images in the output space.

Latent vectors have a meaning. If the model is well-trained, movements in the latent space correspond to meaningful changes in the output space, that is, one can do latent vector arithmetic. Figure 2 was created by transforming the latent vector. If only one row is considered, it can be created by interpolating in the latent space. Notice how close points in the latent space are mapped to similar-looking outputs.

When latent space does not hold meaning, it is called posterior collapse. It is worth mentioning that the latent space is sometimes called a manifold. The paper [Shao *et al.* (2017)] delves into the geometry of latent space.

2.2 Variational AutoEncoders

Variational AutoEncoders (VAEs) [Kingma and Welling (2013)] has solved some of the problems caused by MCMC methods.

VAEs have an encoder-decoder structure. The layer between the encoder and the decoder is called the bottleneck layer. After feeding an image into a VAE, the latent representation of the image will be in the bottleneck layer. Despite the architectural similarities with the autoencoder, they differ substantially in their objective function.

In Figure 2, one can see that the numbers are blurry, and sometimes they are a mix of two numbers (e.g. between 9 and 2). In more complicated images (e.g. faces), this blur becomes more problematic.

3 Generative Adversarial Networks

Generative Adversarial Nets [Goodfellow *et al.* (2014)] introduced the concept of GANs. It took two years until this idea gained momentum and researchers in various fields started using it. It is considered a highly influential paper in the area of image generation.

A GAN consists of 2 parts, a **g**enerator G and a **d**iscriminator D. Their interaction with each other can be thought of as a game, where G generates images and the D tries to distinguish them from real images (coming from a dataset). Gs goal is to fool D and D tries to find flaws in Gs images to differentiate them from real images. As the game progresses, G and D become increasingly better. In the end, G generates images that resemble the images from the dataset. Because of their competitive nature, G**A**Ns are called **a**dversarial. This adversarial approach separated them from the rest of the field and led to their success.

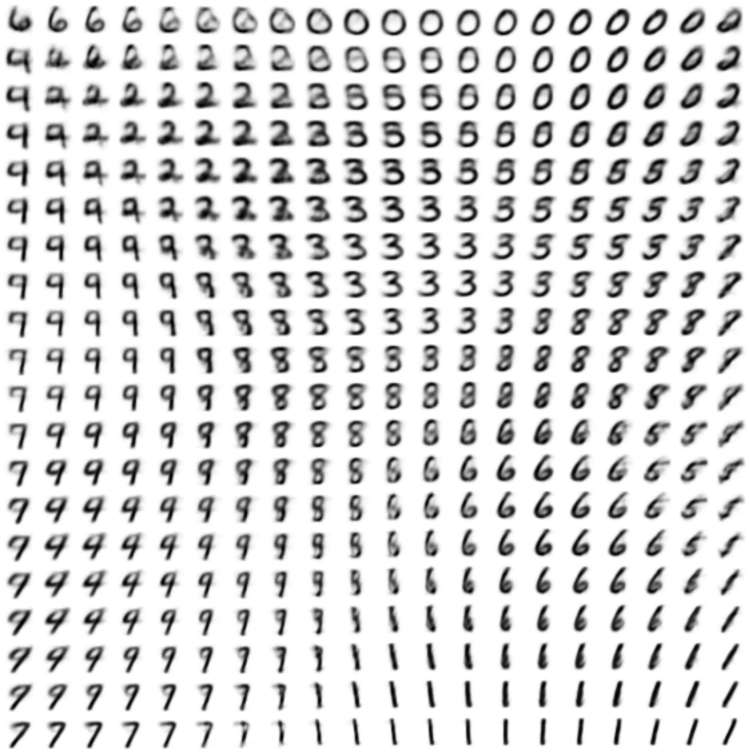


Figure 2: Learned MNIST manifold using a VAE **[**Kingma and Welling (2013)**]**

Analogous to VAEs' decoder, G generates images from latent vectors, whose advantages and properties are covered in section 2.1.1.

3.1 Value Function

The objective can be formalized as a minimax game on a value function V, where D tries to maximize and G tries to minimize it.

$$\min_G \max_D V(D,G) = \underbrace{\mathbb{E}_{x\sim p_{data}(x)}[\log D(x)]}_{(*)} + \underbrace{\mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]}_{(**)} \tag{1}$$

In the Equation 1, $(*)$ ensures the discriminator is classifying real images as real, and in $(**)$ the adversarial competition takes place. In both, the expected value $\mathbb{E}[\,\cdot\,]$ is calculated. In $(*)$, x is sampled from the data distribution $p_{data}(x)$ (i.e. dataset), and in $(**)$ noise z is sampled from noise prior p_z (e.g. multivariate Gaussian).

The output of $D(x) \in [0,1]$ is the confidence of D whether the image x is a real image (i.e. coming from the dataset) ; where 1 denotes real and 0 denotes fake. Any image x with $D(x) < 0.5$ is considered fake (created by G). The expression $D(G(z))$ means that from a latent vector z, an image $G(z)$ is generated and then passed to the discriminator.

3.2 Training Algorithm

The mathematical formulation of $V(D, G)$ is not sufficient. A training algorithm is also needed to make practical applications of the developed theory. Figure 1 shows the proposed training algorithm.

for number of training iterations **do**
  for k steps **do**
    • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
    • Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
    • Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for
  • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  • Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

Algorithm 1: Minibatch stochastic gradient descent training of generative adversarial nets **[**Goodfellow *et al.* (2014)**]**.

Algorithm 1 consists of two steps. First, the discriminator is trained for k steps while the weights of the generator are frozen. Then the discriminator is frozen and the generator gets trained for 1 step. This iterative approach, where they take turns, resembles the minimax game they play.

Code blocks highlighted in cyan and pink are responsible for training the discriminator and the generator, respectively. θ_d are the parameters of D. θ_g are the parameters of G. k is a hyperparameter that ensures that both networks are in balance and that one is not better than the other by a great margin, which will cause problems. In **[**Goodfellow *et al.* (2014)**]**, they used $k = 1$ because it required the least amount of resources.

The gradient expression contains $\frac{1}{m}\sum_{i=1}^{m}[\,\cdot\,]$ as it is the explicit representation of the expected value $\mathbb{E}[\,\cdot\,]$ of a minibatch . The gradient of the generator only contains the $(**)$ from Equation 1 because for G the expression $(*)$ is a constant. Notice that the gradient has D, which means one needs to backpropagate from D into G. Figure 3 illustrates the expected behavior of training.

After introducing these concepts, **[**Goodfellow *et al.* (2014)**]** delves into theoretical results,

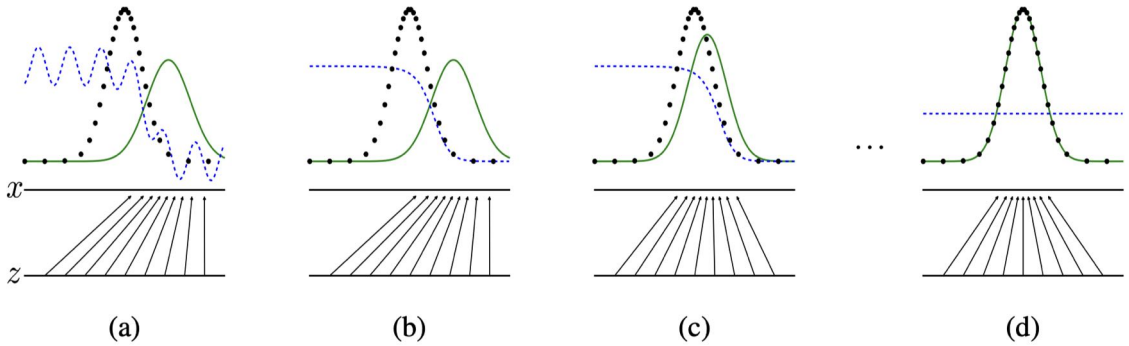


Figure 3: Training of a GAN [Goodfellow *et al.* (2014)]. The blue line and the green line represent the discriminator D and the generator G, respectively. On (a), the weights are randomly initialized. Hence, D cannot distinguish between images. So it outputs a random value throughout. On (b), the D is updated so it can differentiate between real and fake. On (c), the G is updated so it maps the latent vectors to the data distribution more closely. On (d), $D(x) = \frac{1}{2}$ for all images, as the real and fake distributions are the same. This situation is referred to as Nash equilibrium – the solution of a non-cooperative game, which is a concept from Game Theory.

which provide much insight into why this approach works. One can read it to understand concepts more deeply. Then, the paper proceeds with practical results.

3.3 Initial Findings

After training, one can create any image from the learned distribution. In Figure 4, new samples from MNIST [Deng (2012)] are created by interpolating a latent vector z in latent space. Compared to VAEs outputs in Figure 2, these images are not blurry without artifacts.

Figure 4: Interpolation in latent space [Goodfellow *et al.* (2014)]

[Goodfellow *et al.* (2014)] laid the foundations of GANs, however, there was much more to be discovered. In Figure 5, generated samples are blurry and barely recognizable because CIFAR-10's images are more complex than MNIST's. In subsequent years, researchers have developed new techniques to overcome the encountered problems.

[Goodfellow *et al.* (2014)] mentions the lack of a measure to evaluate the results. They fit a Gaussian as it is the best method to their knowledge. As the research progressed, the Inception Score (IS) [Salimans *et al.* (2016)] and other measures were created.

Figure 5: Generated samples using a fully connected network (left) and a convolutional network (right). The green ones are the closest images in CIFAR-10 [Goodfellow *et al.* (2014)]

3.3.1 Fréchet inception distance

Fréchet inception distance (FID) [Heusel *et al.* (2017)] is one of the default quality measures in image generation. It compares how close the generated images are to the dataset at a distributional level. A low FID score is desired.

FID is based on Fréchet distance. Although FID is statistically biased and is unable to assess domain adaptation tasks (e.g. section 3.6), it is widely used.

3.4 Training Problems

As researchers developed new models and techniques, they encountered various problems. The following three are the most significant ones that contributed to progression of the field:

- **Mode collapse:** When the generator only creates a subset of all available objects, it is called mode collapse. For example in a multi-object dataset (a dataset with 2 or more distinct objects) of fruits this is not desirable. If the fruit dataset contains apples and oranges and the generator only creates apples, then the discriminator can confidently assign the label "real" to all oranges. Furthermore, the Nash equilibrium is not reached.

  LSGAN [Mao *et al.* (2017)] uses least squares loss in the value function to prevent mode collapse. With the least squares, the created samples are of higher quality.

- **Vanishing gradients:** Vanishing gradients occur when the discriminator cannot provide sufficient gradient to backpropagate into the generator. This can occur when the discriminator can distinguish between generated images and real ones with high confidence.

  To prevent this, hyperparameters should be correctly selected so the "goodness" of the discriminator and the generator are in balance and the adversarial process works.

- **(Non-)convergence:** Although the existence of Nash equilibrium is proven theoretically, practically converging to it is not a straightforward process. Convergence issues go much more in-depth in GANs, which this report won't explore further. It's worth mentioning the names of the Kullback-Leibler divergence D_{KL} and the Jensen-Shannon divergence D_{JS}

Wasserstein GAN (WGAN) [Arjovsky *et al.* (2017)] is a model that uses the Wasserstein metric to stabilize GAN training. When used with appropriate methods, it prevents the mentioned issues, but it is only an improvement, not the answer to every problem. Each problem has much more depth.

3.5 Capabilities

GANs are capable of performing numerous tasks, from super-resolution to domain adaptation. GANs provide a better approach than traditional algorithms because of their ability to generate a distribution of images. The following bullet points will provide a good stepping stone into GANs:

- **Image super-resolution:** As with traditional algorithms, the image is upsampled, meaning that the resolution is increased and missing information (i.e. pixels) is filled in. The training consists of downsampling images from the dataset and attempting to upsample them. After training, the results are good if the image we are trying to upsample comes from the same or similar distribution. In Figure 6, the second GAN up-samples the image.
- **Image denoising:** The noise is removed with GANs. It has a similar training procedure and reasoning to image super-resolution.
- **Image inpainting:** A region of the image is masked, and the GAN tries to predict the missing areas. It has a similar training procedure and reasoning to image super-resolution.
- **Image fusion:** Two (or more) images are combined to create a final image. These images are mostly of the same scene. The resulting image contains important information from all inputs.
- **Image captioning:** The images are annotated with text. The dataset contains paired images and captions. The generator only receives images and attempts to create captions for them. Note this also falls into the realm of natural language processing (NLP).
- **text-to-image translation:** A sequence of tokens (e.g. a prompt, text) is passed to the generator (called condition variables), which creates images that resemble the

given prompt. The data set contains paired images and labels. The discriminator receives both the tokens and the generated image and determines whether the image matches the tokens. This ensures the relation between the tokens and the generated image.

- **Domain adaptation:** Domain adaptation is best introduced using an example: Creating city maps from aerial images. The images are paired, i.e. for each aerial image, there is a city map that resembles the image, with streets and houses clearly marked.

  If the images are not paired, the task becomes more difficult. This concept is explained in section 3.6.

Researchers develop specific architectures to solve particular tasks. New architectures are cultivated from ideas from previous models, with one or two novel ideas. Coming up with original ideas is not easy and requires many revisions. The following model combines many concepts, from NLP to image generation.

StackGAN [Zhang *et al.* (2016)**]** is a 2-stage model that uses 2 GANs to create images from text prompts. The first GAN receives the prompt and outputs a low-resolutional 64×64 image. The second GAN receives the output of the first GAN, along with the prompt to upsample it to 256×256 pixels. This approach is reported to cover a larger distribution of images and produce more diverse, high-quality images.

Figure 6: Flowers created using StackGAN with a text prompt **[**Shamsolmoali *et al.* (2020)**]**

With each new model introduced, GANs become more capable. This creates new application areas. In medicine, GAN has proven to be useful:

Patient privacy is a concern in medical research, and with GANs, one can create synthetic data that represents patient data at a distributional level. This protects patient privacy and enables researchers to share the generated samples, as they do not come from a specific patient. This matter goes into data argumentation.

These images can be used for many applications, such as training a CNN to detect a disease in chest radiographs. [DuMont Schütte *et al.* (2021)] highlights the effectiveness of this approach.

3.6 CycleGAN

CycleGAN [Zhu *et al.* (2017)] translates images between domains. For many image-to-image translation tasks, paired images are not available. [Zhu *et al.* (2017)] overcomes this challenge by introducing so-called cycle-consistency loss.

Working with unpaired images comes with challenges. However, it is important to solve this problem because it is not possible for some tasks to pair images. For example, in Figure 7, one CycleGAN is trained to translate between Monet paintings and photorealistic images. It is simply not possible to create a paired image dataset for this. There are some cases where pairing is possible, but creating datasets takes too much time, and unpaired images are more available.

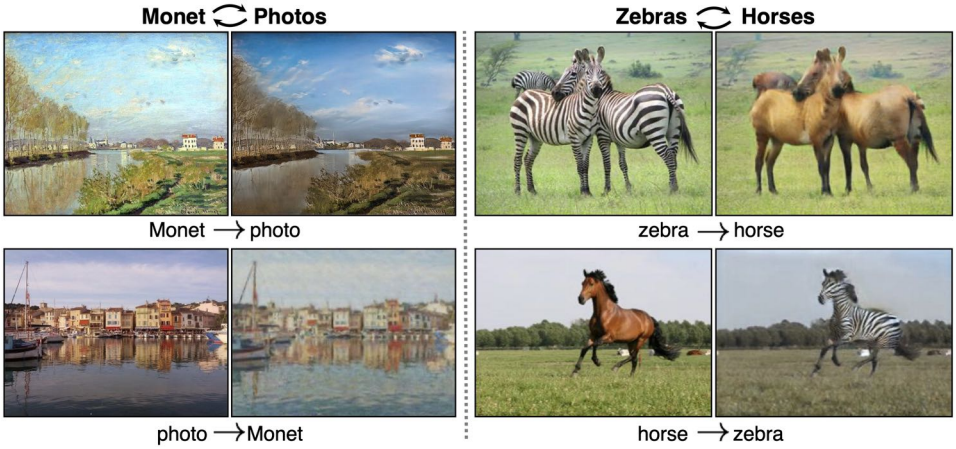


Figure 7: Transition between domains using CycleGAN [Zhu *et al.* (2017)]

A small abstraction helps to formalize this problem. Figure 8 explains everything in a nutshell.

Consider the horses and zebras in Figure 7, and name their domains X and Y, respectively. Since two-way translation is desired, using 2 GANs is an intuitive solution. A generator $G : X \to Y$ with the discriminator D_Y for domain Y. Given a zebra image in X, G will generate a horse image. Similarly $F : Y \to X$ with D_X. This formulation is not sufficient. G can completely ignore the input image and output any zebra image. D_Y would have no way of knowing it. Cycle-consistency loss is introduced to solve this problem.

Cycle-consistency loss works by mapping an image twice. When you convert a horse to a zebra and back, the original and the generated must look similar for it to be a proper translation, i.e. $F(G(x)) \approx x$.

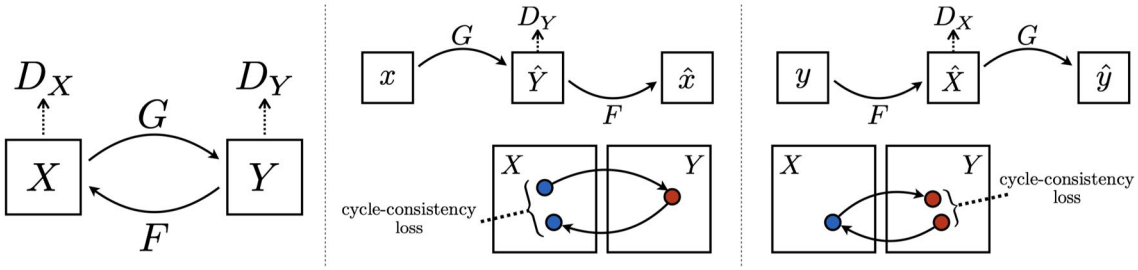


Figure 8: Illustration of CycleGAN [Zhu *et al.* (2017)]

Formalizing this mathematically is straightforward. For both GANs, the model uses the standard GAN loss. This is the loss for images from domain Y:

$$\mathcal{L}_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{data}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{data}(x)}[\log(1 - D_Y(G(x)))]$$

The cycle-consistency loss is the pixelwise difference between the original image and the twice-translated image. It is done for both domains.

$$\mathcal{L}_{cyc}(G, F) = \mathbb{E}_{x \sim p_{data}(x)}[||F(G(x)) - x||_1] + \mathbb{E}_{y \sim p_{data}(y)}[||G(F(y)) - y||_1]$$

Combining these losses, we get the final loss of:

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, X, Y) + \lambda \cdot \mathcal{L}_{cyc}(G, F)$$

where λ is the importance of cycle consistency. In the paper, $\lambda = 10$ is used.

PatchGAN [Isola *et al.* (2016)] is chosen as the discriminator. As the name suggests, PatchGAN works with $n \times n$ patches of images and discriminates them. This enables CycleGAN to work with arbitrarily sized images.

CycleGAN succeeds in color and texture translations but fails at geometric changes. For example, it fails at translating between cat and dog pictures as their bodily sizes and shapes are significantly different. When the model is trained on horses without a rider, it fails to convert the horse to a zebra in case a "horse with a rider" image is given.

The CycleGAN paper [Zhu *et al.* (2017)] is a recommended read as it gives much more insight. Some details are left out in this section because it goes deeper into the problem, such as the identity loss $\mathcal{L}_{identity}$ to prevent tinting of images.

3.6.1 Similar GANs

The following list contains three models that deal with the same problem.

- **Contrastive Unpaired Translation** (CUT) [Park *et al.* (2020)] is created by the same research group and utilizes the fact that image patches should contain similar

information. For example, a horse's face should contain information from the zebra's face but not the grass and trees in the background. By doing so, better results are achieved.

NB: Assessing the quality of created images is hard, and currently, there is no metric. For example, among all possible zebras, a generated zebra is not quantitatively measurable to compare it to other potential zebras. The judgment is left to the human eye.

- **DiscoGAN [**Kim *et al.* (2017)**]** is another model that does the same job. It uses another loss called reconstruction loss, which works similarly.
- **StarGAN [**Choi *et al.* (2017)**]** can be thought of as the generalized CycleGAN. In CycleGAN we must train a different model for each task. StarGAN is 1 model that can transfer between multiple domains (i.e. > 2). The paper highlights StarGAN's capabilities by transforming human faces.

4 Modern Methods

Nowadays, the field of image generation has substantially evolved. Diffusion models were introduced in 2015 and it took 5 years for them to gain popularity **[**Ho and Saharia (2023)**]**. Since 2020, diffusion models have superseded GANs and can generate better, more versatile images.

As people began to use them, diffusion models grew in popularity. Today, DALL-E, Midjourney, and Stable Diffusion **[**Rombach *et al.* (2022)**]** are among the most popular models with text-to-image translation, as it is the most natural way for humans to communicate. Popularity of generated images has increased on the internet as generating them became more accessible to a larger population.

Diffusion models have a completely different architecture than GANs. They work by adding noise to the training data and trying to denoise it. The training is more stable than GANs' and mode collapse is not a big issue.

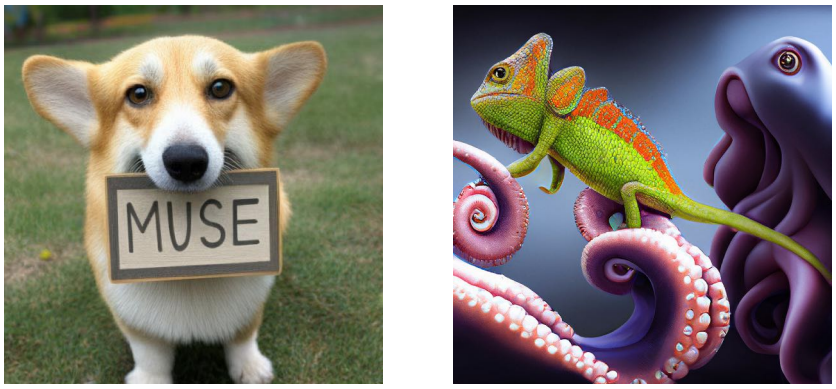
As of the end of 2023, one of the state-of-the-art models is **CM3Leon [**Yu *et al.* (2023)**]** (pronounced "chameleon") from Meta AI and it leverages **CM3 [**Aghajanyan *et al.* (2022)**]**. It is an autoregressive model. The generated images are more versatile, but require high processing power compared to other approaches. **[**Yu *et al.* (2023)**]** scales it up and shows the advantages of such an approach.

These new models do not mean that GANs have become obsolete. Image tokens generated by a VQVAE-GAN were used in the training of CM3. This idea of using an older model to improve another model can be found throughout neural networks.

Another state-of-the-art model is **MUSE [**Chang *et al.* (2023)**]** from Google Research. It takes a different approach by using vision transformers. They also perform better than

GANs, but the generated images cannot keep up with autoregressive models in terms of diversity. The images produced are excellent by today's standards, and the computational cost is less because the transformer architecture requires fewer steps than diffusion models. For these reasons, Google Research has decided to invest resources in research [Amini (2023)].

In Figure 9, the capabilities of both models are to be seen.



(a) "Corgi Holding a Writing" by MUSE

(b) "Chameleon and Octopus" by CM3Leon

Figure 9: Images that MUSE and CM3Leon created that were published by the authors.

5 Conclusion

GANs have played an important role in image generation. This report covered GANs, focusing on introducing the concepts to a novice. First, the DBNs and VAEs were introduced, and their capabilities were demonstrated. A motivation for latent vectors is given. Then, GAN is introduced with its value function and training algorithm, comparing its capabilities to previous models. The training problems and capabilities were presented, along with various GAN models. CycleGAN was explored as it is intuitively understandable. This provided an overview of how domain adaptation works and an overview into the model's inner workings. Similar models are noted to provide an insight into how models are related to each other. Then the modern methods are explained to bridge the topic to more popular diffusion models. Although both GANs and VAEs have been superseded by newer methods, their usefulness remains to this day.

References

Aghajanyan, A., Huang, B., Ross, C., Karpukhin, V., Xu, H., Goyal, N., Okhonko, D., Joshi, M., Ghosh, G., Lewis, M., and Zettlemoyer, L. (2022). Cm3: A causal masked multimodal model of the internet.

Amini, A. (2023). MIT 6.S191: Text-to-Image Generation. [Online; accessed 18. Dec. 2023].

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein gan.

Chang, H., Zhang, H., Barber, J., Maschinot, A., Lezama, J., Jiang, L., Yang, M.-H., Murphy, K., Freeman, W. T., Rubinstein, M., Li, Y., and Krishnan, D. (2023). Muse: Text-to-image generation via masked generative transformers.

Choi, Y., Uh, Y., Yoo, J., and Ha, J.-W. (2017). Stargan v2: Diverse image synthesis for multiple domains.

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, **29**(6), 141–142.

DuMont Schütte, A., Hetzel, J., Gatidis, S., Hepp, T., Dietz, B., Bauer, S., and Schwab, P. (2021). Overcoming barriers to data sharing with medical image generation: a comprehensive evaluation. *npj Digital Med.*, **4**(141), 1–14.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, **27**.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium.

Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, **18**(7), 1527–1554.

Ho, J. and Saharia, C. (2023). High Fidelity Image Generation Using Diffusion Models. [Online; accessed 14. Dec. 2023].

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2016). Image-to-image translation with conditional adversarial networks.

Kim, T., Cha, M., Kim, H., Lee, J. K., and Kim, J. (2017). Learning to discover cross-domain relations with generative adversarial networks.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. (2016). Adversarial autoencoders.

Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., and Smolley, S. P. (2017). Least squares generative adversarial networks.

Park, T., Efros, A. A., Zhang, R., and Zhu, J.-Y. (2020). Contrastive learning for unpaired image-to-image translation.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training gans.

Shamsolmoali, P., Zareapoor, M., Granger, E., Zhou, H., Wang, R., Celebi, M. E., and Yang, J. (2020). Image synthesis with adversarial networks: a comprehensive survey and case studies.

Shao, H., Kumar, A., and Fletcher, P. T. (2017). The riemannian geometry of deep generative models.

Thompson, N. C., Greenewald, K., Lee, K., and Manso, G. F. (2022). The computational limits of deep learning.

Yu, L., Shi, B., Pasunuru, R., Muller, B., Golovneva, O., Wang, T., Babu, A., Tang, B., Karrer, B., Sheynin, S., Ross, C., Polyak, A., Howes, R., Sharma, V., Xu, P., Tamoyan, H., Ashual, O., Singer, U., Li, S.-W., Zhang, S., James, R., Ghosh, G., Taigman, Y., Fazel-Zarandi, M., Celikyilmaz, A., Zettlemoyer, L., and Aghajanyan, A. (2023). Scaling autoregressive multi-modal models: Pretraining and instruction tuning.

Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., and Metaxas, D. (2016). Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks.

Zhu, J., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, **abs/1703.10593**.