Novel Facial Image Generation via Deep Convolutional Generative Adversarial Networks

Tu Yang

***Abstract*—In the modern era of high computational power and advanced technological advancements, we are able to achieve many of the complex tasks that would once be deemed impossible. Generative networks are one of the strongest talking points of the previous decade due to the gravity of the fabulous results that they have been able to generate successfully. One of the tasks that we will discuss in this article, which was also once considered quite complex to achieve, is the generation of realistic faces with the help of generative adversarial networks. There have been multiple successful deployments of models and different neural architectural builds that have managed to accomplish this task effectively. This article [1] will focus on understanding some of the basic concepts of facial generation with the help of generative adversarial networks. We will try to accomplish the generation of realistic images with the help of Deep Convolutional Generative Adversarial Networks (DCGANs) that we constructed in one of our previous works. Before moving further with this article, I would recommend checking out two of my preceding works to stay updated with the contents of this article. To get started with GANs, check out the following link - "Complete Guide to Generative Adversarial Networks (GANs)" and to gain further knowledge on DCGANs, check out the following link - "Getting Started With DCGANs." To understand the numerous core concepts that we will cover in this article, check out the table of contents. We will start off with an introduction to the generation of facial structures with GANs and then proceed to gain further knowledge on some of the pre-existing techniques that have been successfully deployed to achieve the best results possible for these tasks. We will have a quick overview of the DCGANs architecture and methodologies to approach the following task. And then have a code walkthrough to discuss the various steps to follow to achieve a desirable result. Finally, we will conclude with some much-needed sections of the future works and the types of improvements and advancements that we can make to the built architecture model.**

I. Introduction

The growth of Generative Adversarial Networks (GANs) is rapid. The continuous advancements in these dual neural network architectures, consisting of a generator model and discriminator, help to stabilize outputs and generate real images, which become almost next to impossible for the human eye to differentiate. The above 1024x1024 sized image shows the picture of a cheerful woman. Looking at the following image, it would come as a surprise to people new to this spectrum of deep learning that the person does not actually exist. It was generated from the website called this X does not exist, and you can also generate random faces on your own by refreshing the following link. Each time the generator is run, a completely new realistic face is generated.

[1]https://blog.paperspace.com/face-generation-with-dcgans/

When I asked a few of my friends to determine what they thought about the particular picture, almost nobody was able to figure out that the following image was generated with the help of Artificial Intelligence (AI). To be precise, the StyleGAN2 architecture used to generate the following faces performs at an exceptional level producing these results indistinguishable from real photographs. Only upon further introspection will you be able to notice slight abnormalities in the following image depicted above. One of the noticeable issues could be the weird shaded-out background. Another issue could be the difference in the two earrings that are worn by her, but it could also be considered as an interesting fashion sense. Small errors like these are still difficult for It is fun to speculate how the generated image may not be real, but the simple fact we are having the following discussion is enough evidence that the potential for generative networks for a task like facial structure generation or any other similar project is humungous and something to watch out for in the upcoming future. We will explore the enormous potential that these generative networks have to create realistic faces or any other type of images from scratch. The concept of two neural networks competing against each other to produce the most effective results is an intriguing aspect of study in deep learning. Let us start by understanding some of the methodologies and pre-existing techniques that are currently used in the generation of faces. Once we explore these pre-existing techniques, we will go into greater detail on how we can produce these similar results with the help of deep convolutional generative adversarial networks. So, without further ado, let us proceed to understand and explore these topics accordingly.

The progression of the technology of GANs has been fast-faced and extremely successful in a short span of less than a decade. An adventure started out in 2014 with a simple Generative Adversarial Architecture soon found continuous popularity through the numerous years of advancements and improvements. There are tons of new techniques and methodologies that are constantly being discovered by developers and researchers in this field of generative neural networks. The above image is an indication of the rapid pace of development that is achieved by the improvement in the overall technology and methods used in Generative Adversarial Networks. Let us briefly discuss some of the fantastic techniques that are currently deployed in the world of machine learning for the purpose of realistic face generation. One of the major architectures that did a fantastic job of accomplishing this task of generating photorealistic faces from scratch was the StyleGAN architecture developed by NVIDIA. While these initial models produced fabulous results to start with, it was

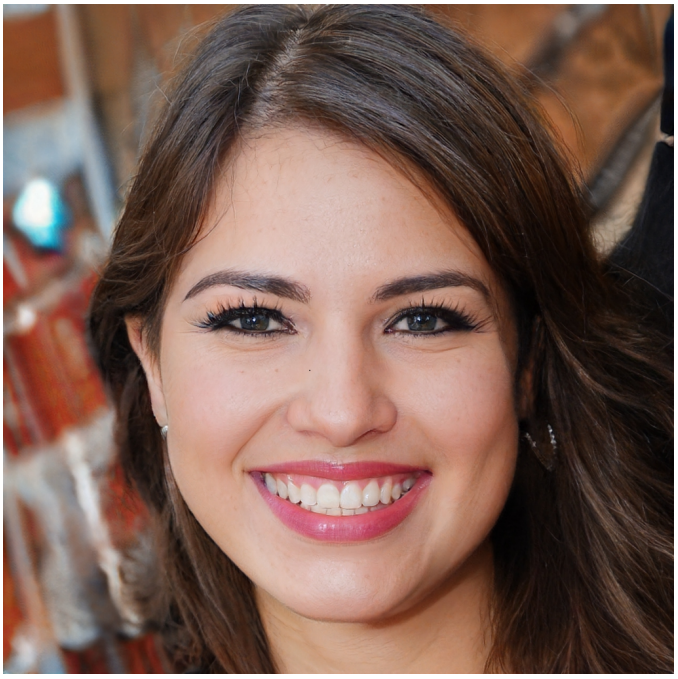

Fig. 1: .

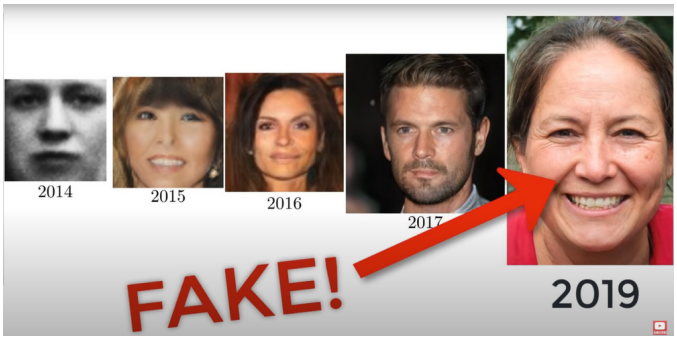

Fig. 2: .

in the second iteration of the StyleGAN architectures that these models gained immense popularity. They fixed some of the characteristic artifacts features after thorough research, analysis, and experiments. In 2019, this new version of Style GANs 2 was released with an advanced model architecture and several methods of improvement in their training procedure to achieve the best results till that time. To understand and learn more about this particular architecture, I would suggest looking at the following research paper. Recently, the third iteration of the StyleGAN architecture was released with improved results and further advancements on the type of tasks it is typically built to perform. Another intriguing concept is of the Face App that performs changes in the type of age of a person by altering the faces is also built with the help of a GAN. Typically to perform these types of tasks, a Cycle GAN architecture is used, which has gained immense popularity as of late. These GANs perform the action of learning the transformation of various images of different styles. The Face App, for example, transforms the generated or uploaded faces into faces of different age groups. We will explore both the Cycle GAN and the Style in future works, but for the purpose of this article, we will stick to our previously discussed DCGANs architecture to achieve the task of generating faces.

II. Related Work

Image to Image translation have been around for sometime before the invention of CycleGANs. One really interesting one is the work of Phillip Isola et al in the paper Image to Image Translation with Conditional Adversarial Networks where images from one domain are translated into images in another domain. The dataset for this work consists of aligned pair of images from each domain. Generative image models are well studied and fall into two categories: parametric and nonparametric. The non-parametric models often do matching from a database of existing images, often matching patches of images, and have been used in texture synthesis, super-resolution and in-painting. Parametric models for generating images has been explored extensively (for example on MNIST digits or for texture synthesis). However, generating natural images of the real world have had not much success until recently. A variational sampling approach to generating images has had some success, but the samples often suffer from being blurry. Another approach generates images using an iterative forward diffusion process. Generative Adversarial Networks generated images suffering from being noisy and incomprehensible. The maximum mean discrepancy (MMD) [1] is a measure of the difference between two distributions P and Q given by the supremum over a function space $\mathcal{F}$ of differences between the expectations with regard to two distributions. MMD has been used for deep generative models [2], [3], [4] and model criticism [5]. WGAN [6] conducted a comprehensive theoretical analysis of how the Earth Mover (EM) distance behaves in comparison with popular probability distances and divergences such as the total variation (TV) distance, the Kullback-Leibler (KL) divergence, and the Jensen-Shannon (JS) divergence utilized in the context of learning distributions.Based on W-div, Wu et al. [7] introduce a Wasserstein divergence objective for GANs (WGAN-div), which can faithfully approximate W-div by optimization. CramerGAN [8] argues that the Wasserstein distance leads to biased gradients, suggesting the Cramér distance between two distributions. Other papers related to WGAN can be found in [9], [10], [11], [12], [13], [14].

Rather than utilizing a single unstructured noise vector z, InfoGAN [15] proposes to decompose the input noise vector into two parts: z, which is seen as incompressible noise; c, which is called the latent code and will target the significant structured semantic features of the real data distribution. Maximizing $I(c; G(z, c))$ means maximizing the mutual information between c and $G(z, c)$ to make c contain as much important and meaningful features of the real samples as possible. However, $I(c; G(z, c))$ is difficult to optimize directly in practice since it requires access to the posterior $P(c|x)$. Fortunately, we can have a lower bound of $I(c; G(z, c))$ by defining an auxiliary distribution $Q(c|x)$ to approximate $P(c|x)$. The discriminator of original GANs [16] is trained to maximize the log-likelihood that it assigns

to the correct source [17]. Discriminators of most cGANs based methods [18], [19], [20], [21], [22] feed conditional information y into the discriminator by simply concatenating (embedded) y to the input or to the feature vector at some middle layer. cGANs with projection discriminator [23] adopts an inner product between the condition vector y and the feature vector.

Two-domain I2I can solve many problems in computer vision, computer graphics and image processing, such as image style transfer (f.) [24], [25], which can be used in photo editor apps to promote user experience and semantic segmentation (c.) [26], [27], which benefits the autonomous driving and image colorization (d.) [28], [29]. If low-resolution images are taken as the source domain and high-resolution images are taken as the target domain, we can naturally achieve image super-resolution through I2I (e.) [30], [31]. Indeed, two-domain I2I can be used for many different types of applications as long as the appropriate type and amount of data are provided as the source-target images. Therefore, we refer to the universal taxonomy in machine learning, such as the categorizations used in [32], [33], [34], and classify two-domain I2I methods into four categories based on the different ways of leveraging various sources of information: supervised I2I, unsupervised I2I, semi-supervised I2I and few-shot I2I, as described in following paragraph. We also provide the summary of these two-domain I2I methods including method name, publication year, the type of training data, whether multi-modal or not and corresponding insights.

In the earlier I2I works [35], researchers used many aligned image pairs as the source domain and target domain to obtain the translation model that translates the source images to the desired target images. Training supervised translation is not very practical because of the difficulty and high cost of acquiring these large, paired training data in many tasks. Taking photo-to-painting translation as an example, it is almost impossible to collect massive amounts of labeled paintings that match the input landscapes. Hence, unsupervised methods [24], [36], [37] have gradually attracted more attention. In an unsupervised learning setting, I2I methods use two large but unpaired sets of training images to convert images between representations. In some special scenarios, we still need a little expensive human labeling or expert guidance, as well as abundant unlabeled data, such as those of old movie restoration [38] or genomics [39]. Therefore, researchers consider introducing semi-supervised learning [40], [41], [42] into I2I to further promote the performance of image translation. Semi-supervised I2I approaches leverage only source images alongside a few source-target aligned image pairs for training but can achieve more promoted translated results than their unsupervised counterpart. Nonetheless, several problems remain regarding translation using a supervised, unsupervised or semi-supervised I2I method with extremely limited data. In contrast, humans can learn from only one or limited exemplars to achieve remarkable learning results. As noted by meta-learning [43], [44] and few-shot learning [45], [46], humans can effectively use prior experiences and knowledge when learning new tasks, while artificial learners usually severely overfit without the necessary prior knowledge. Inspired by the human learning strategy, few- and one-shot I2I algorithms [47], [48], [49], [50] have been proposed to translate from very few (or even one) in the limit unpaired training examples of the source and target domains. Although learning settings may differ, most of these I2I techniques tend to learn a deterministic one-to-one mapping and only generate single-modal output. However, in practice, the two-domain I2I is inherently ambiguous, as one input image may correspond to multiple possible outputs, namely, multimodal outputs. Multimodal I2I translates the input image from one domain to a distribution of potential outputs in the target domain while remaining faithful to the input. These diverse outputs represent different color or style texture themes (i.e., multimodal) but still preserve the similar semantic content as the input source image. Therefore, we actually view multimodal I2I as a special two-domain I2I and discuss it.

Supervised I2I aims to translate source images into the target domain with many aligned image pairs as the source domain and target domain for training. In this subsection, we further divide the supervised I2I in two categories: methods with single-modal output and methods with multimodal outputs. The idea of I2I can be traced back to Hertzmann et al.'s image analogies [51], which use a non-parametric texture model for a wide variety of "image filter" effects with an image pair input. More recent research on I2I mainly leverages the deep convolutional neural network to learn the mapping function. Isola et al. [35] first apply conditional GAN to an I2I problem by proposing pix2pix to solve a wide range of supervised I2I tasks. In addition to the pixelwise regression loss $\mathcal{L}_1$ between the translated image and the ground truth, pix2pix leverages adversarial training loss $\mathcal{L}_{cGAN}$ to ensure that the outputs cannot be distinguished from "real" images. The objective is: Pix2pix is also a strong baseline image translation framework that inspires many improved I2I works based on it, as described in following parts. Wang et al. [52] claim that the GAN loss and pixelwise loss used in pix2pix often lead to blurry results. They present discriminative region proposal adversarial networks (DRPANs) to address it by adding a reviser (R) to distinguish real from masked fake samples. Wang et al. [53] argue that the adversarial training in pix2pix [35] might be unstable and prone to failure for high-resolution image generation tasks. They propose an HD version of pix2pix that can increase the photo realism and resolution of the results to 2048×1024. Moreover, AlBahar et al. [54] take an important step toward addressing the controllable or user-specific generation based on pix2pix [35] via respecting the constraints provided by an external, user-provided guidance image. Unfortunately, pix2pix [35] and its improved variants [52], [53], [54] still fail to capture the complex scene structural relationships through a single translation network when the two domains have drastically different views and severe deformations. Tang et al. [55] therefore proposed SelectionGAN to solve the cross-view translation problem, i.e., translating source view images to target view scenes in which the fields of views have little or no overlap. It was the first attempt to combine the multichannel attention selection module with GAN to solve the I2I problem. What's more, SPADE [26] proposes the spatially-adaptive normalization

layer to further improve the quality of the synthesized images. But SPADE uses only one style code to control the entire style of an image and inserts style information only in the beginning of a network. SEAN [27] therefore designs semantic region-adaptive normalization layer to alleviate the two shortcomings. Having said that, Shaham et al. [56] claim that traditional I2I networks [35], [53], [26] suffer from acute computational cost when operating on high-resolution images. They propose to design a more lightweight but efficient enough network ASAPNet for fast high-resolution I2I. Recently, Zhang et al. [57], [58] proposed an exemplar-based I2I framework to translate images by establishing the dense semantic correspondence between cross-domain images. However, the semantic matching process may lead to a prohibitive memory footprint when estimating a high-resolution correspondence. Zhou et al. and Zhan et al. [59], [60] therefore proposed to reduce the memory cost with hierarchy PatchMatch or bilevel feature alignment while building correspondence.

Actually, this multimodal translation benefits from the solutions of *mode collapse problem* [61], [6], [62], in which the generator tends to learn to map different input samples to the same output. Thus, many multimodal I2I methods [63], [64] focus on solving the mode collapse problem to lead to diverse outputs naturally. BicycleGAN [63] became the first supervised multimodal I2I work by combining cVAE-GAN [65], [66], [67] and cLR-GAN [15], [68], [21] to systematically study a family of solutions to the mode collapse problem and generate diverse and realistic outputs. Similarly, Bansal et al. [64] proposed PixelNN to achieve multimodal and controllable translated results in I2I. They proposed a nearest-neighbor (NN) approach combining pixelwise matching to translate the incomplete, conditioned input to multiple outputs and allow a user to control the translation through on-the-fly editing of the exemplar set. Another solution for producing diverse outputs is to use *disentangled representation* [15], [69], [70], [71] which aims to break down, or disentangle, each feature into narrowly defined variables and encodes them as separate dimensions. When combining it with I2I, researchers disentangle the representation of the source and target domains into two parts: domain-invariant features *content*, which are preserved during the translation, and domain-specific features *style*, which are changed during the translation. In other words, I2I aims to transfer images from the source domain to the target domain by preserving *content* while replacing *style*. Therefore, one can achieve multimodal outputs by randomly choosing the *style* features that are often regularized to be drawn from a prior Gaussian distribution $N(0, 1)$. Gonzalez-Garcia et al. [72] disentangled the representation of two domains into three parts: the *shared* part containing common information of both domains, and two *exclusive* parts that only represent those factors of variation that are particular to each domain. In addition to the bi-directional multimodal translation and retrieval of similar images across domains, they can also transfer a domain-specific transfer and interpolation across two domains.

GANs can be extended to a conditional model [73] if both the discriminator and generator are conditioned on some extra information y. By comparing and, we can see that the generator of InfoGAN is similar to that of cGANs. However, the latent code c of InfoGAN is not known, and it is discovered by training. Furthermore, InfoGAN has an additional network Q to output the conditional variables $Q(c|x)$. Chrysos et al. [74] proposed robust cGANs. Thekumparampil et al. [75] discussed the robustness of conditional GANs to noisy labels. Conditional CycleGAN [76] uses cGANs with cyclic consistency. Mode seeking GANs (MSGANs) [77] proposes a simple yet effective regularization term to address the mode collapse issue for cGANs. GANs are also utilized to achieve image composition [78], [79], [80], domain adaptation [81], [82], [83]. Based on cGANs, we can generate samples conditioning on class labels [17], [84], text [85], [86], [87], bounding box and keypoints [88]. In [87], [89], text to photo-realistic image synthesis is conducted with stacked generative adversarial networks (SGAN) [90]. cGANs have been used for convolutional face generation [91], face aging [92], multi-modal image translation [55], [93], exemplar-based image synthesis [94], [57], synthesizing outdoor images having specific scenery attributes [95], natural image description [96], and scene manipulation [97], [98]. Image/video colorization [99], [28], [100], [101], [102], [29], image inpainting [103], [104], image super-resolution. Isola et al. [35] used cGANs and sparse regularization for image-to-image translation. The corresponding software is called pix2pix. In GANs, the generator learns a mapping from random noise z to $G(z)$. In contrast, there is no noise input in the generator of pix2pix. A novelty of pix2pix is that the generator of pix2pix learns a mapping from an observed image y to output image $G(y)$, for example, from a grayscale image to a color image. As a follow-up to pix2pix, pix2pixHD [53] used cGANs and feature matching loss for high-resolution image synthesis and semantic manipulation. With the discriminators, the learning problem is a multi-task learning problem. Image-to-image translation is a class of graphics and vision problems where the goal is to learn the mapping between an output image and an input image using a training set of aligned image pairs. When paired training data is available, reference [35] can be used for these image-to-image translation tasks. However, reference [35] can not be used for unpaired data (no input/output pairs), which was well solved by Cycle-consistent GANs (CycleGAN) [24]. CycleGAN is an important progress for unpaired data. It is proved that cycle-consistency is an upper bound of the conditional entropy [105]. CycleGAN can be derived as a special case within the proposed variational inference (VI) framework [106], naturally establishing its relationship with approximate Bayesian inference methods. Specifically, GAN models has been broadly applied in image synthesis [26], [27], [107], [108], The basic idea of DiscoGAN [36] and CycleGAN [24] is nearly the same. Both of them were proposed separately nearly at the same time. The only difference between CycleGAN [24] and DualGAN [109] is that DualGAN uses the loss format advocated by Wasserstein GAN (WGAN) rather than the sigmoid cross-entropy loss used in CycleGAN. Some other works utilize GAN to achieve image transfer [24], [36], [109], [110], [111].

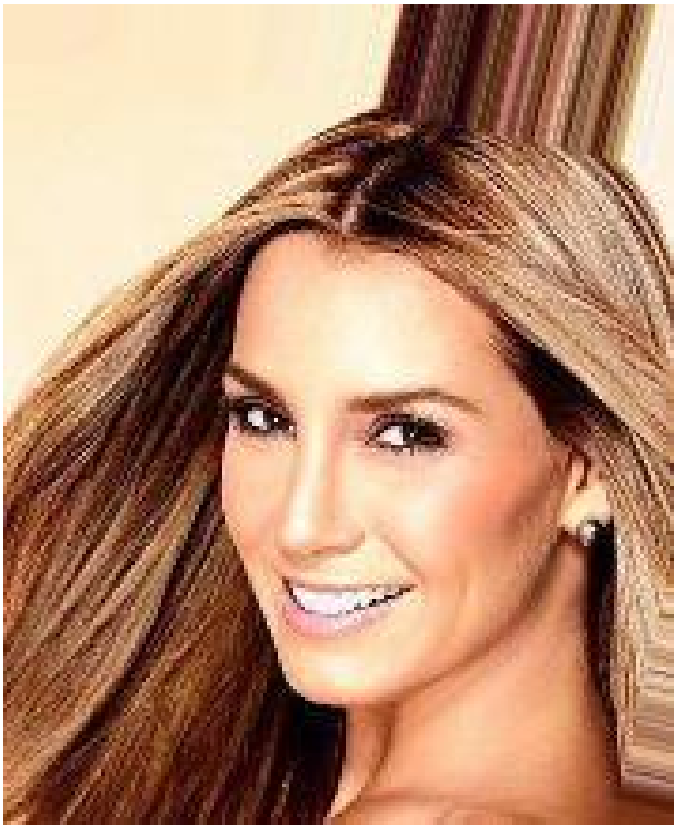

Fig. 3: .

III. Methods

In this section of the article, we will further explore the concept of DCGANs and how to generate numerous faces with the help of this architecture. In one of my previous articles on DCGANs, we have covered most of the essential concepts required for understanding and solving a typical task with the help of these generative networks. Hence, we will have a brief overview of the type of model we will utilize for solving the task of face generation and achieving the best results. Let us get started with the exploration of these concepts.

While the basic or vanilla Generative Adversarial Networks produced some of the best works in the early days of generative networks, these networks were not highly sustainable for more complex tasks and were not as computationally effective for performing certain operations. For this purpose, we use a variety of generative network architectures that are built for specific purposes and perform a particular action more effectively than others. In this article, we will utilize the DCGANs model with a deep convolutional generator architecture to generate images. The generator with a random latent noise vector will generate the facial structures, and the discriminator, which acts as an image classifier, will distinguish if the output image is real or fake.

While the working procedure of the deep convolutional generative adversarial networks is similar to the workings of a typical GAN type network, there are some key architectural improvements that were developed to produce more effective results on most datasets. Most of the pooling layers were replaced with strided convolutions in the discriminator model and fractional-strided convolutions in the case of the generator models. The use of batch normalization layers is extensively used in both the generator and discriminator models for obtaining the stability of the overall architecture. Most of the later dense layers, i.e., the fully connected layers, were removed for a fully convolutional type architecture.

All the convolutional layers in the discriminator model are followed by the Leaky ReLU activation function other than the final layer with Sigmoid for classification purposes. The generator model makes use of the ReLU activation function in all layers except the final layer, which uses the tanh function for generative purposes. However, it is noticeable that we have made certain small changes in our generator architecture as these are some minute alterations in modern time that have shown some better results in specific use cases. The viewers and developers can feel free to try out their own variations or stick to the default archetypes for their respective projects according to their requirements.

The procedure we will follow for the effective construction of this project is to ensure that we collect the best datasets for the required task. Our options are to utilize high-quality datasets that are available on the internet or stick to some other alternatives for performing these tasks. We will utilize the Celeb Faces Attributes (CelebA) Dataset to develop our facial recognition generative networks. The following dataset is available on Kaggle, and it is recommended that you download it to continue with the remaining contents of the article. If you have an advanced system with the resources to compute high-end problems, then you can try out the Celeb HQ datasets. Also, remember that the Gradient platform on Paperspace is one of the best options for these extensive GPU-related tasks.

Once we have the dataset downloaded, ensure that you extract the img align celeba zip file in the ”Images” directory to proceed with the next steps. We will utilize the TensorFlow and Keras deep learning frameworks for this project. If you don’t have complete knowledge of these two libraries or want to refresh your basics quickly, I would recommend checking out the following link for TensorFlow and this particular link for Keras. Once we access the dataset, we will construct both the generator and the discriminator models for performing the task of face generation. The entire architectural build will focus on creating the adversarial framework for generating the image and classifying the most realistic interpretations. After the construction of the entire architectural build, we will start the training procedure and save the images accordingly. Finally, we will save the generator and discriminator models so that we can reuse them some other time.

In this section of the article, we will explore how to code our face generator model from scratch so that we can utilize these generator models to obtain high-quality results similar to some of the state-of-the-art methods. It is crucial to note that in this project, we will utilize sizes of lesser dimensions (64x64) so that people at all levels can construct this work and won’t have to suffer from the obstacle of graphics limitations. The Gradient platform provided by Paperspace is one of the best utility options for these complex computational projects. If you

have better equipment or are looking to produce better-looking results, then I would recommend trying out more HD datasets that are available on the internet. Feel free to explore the options and choose the best ones accordingly. For the purpose of this article, we will stick with a simple smaller dimensional dataset that does not consume too much space. Let us get started with the construction of this project. The starting step of most projects is to import all the essential libraries that we will utilize for the development of the required project. For the construction of this face generation model, we will utilize the TensorFlow and Keras deep learning frameworks for achieving our goals. If you are comfortable with PyTorch, that would also be another valid option to develop the project as you desire. If you aren't familiar with TensorFlow or Keras, check out the following two articles that will guide you in detail on how you can get started with these libraries - "The Absolute Guide to TensorFlow" and "The Absolute Guide to Keras." Apart from these fabulous frameworks, we will also utilize other essential libraries, such as matplotlib, numpy, and OpenCV, for performing visualizations, computations, and graphics-related operations accordingly. You can feel free to utilize other libraries that you might deem necessary or suitable to perform the following task. You can also add numerous options of graphics tools or visualization graphs to track your progress and ultimately help you in improving the results. Once we have successfully imported all the required libraries that we need, we can proceed to explore the data and construct our model for solving the following task.

In this step, we will load the data into a Dataset format and map the details accordingly. The pre-processing class in the Keras allows us to access the flowing of data from a particular directory so that we can access all the images stored in the particular folder. Ensure that once you extract the celeb dataset containing over 200000 images, you place the extracted directory into another directory titled "Images" or any other folder name of your choice. The dataset will flow from the following directory, and we will assign the image size of 64 x 64 for the computation of the following data. The successful execution of the above code cell block should show a result similar to the following statement - "Found 202599 files belonging to 1 classes." You can choose to randomly display an image to check if you have loaded the dataset precisely as planned. The code block below is one of the ways of visualizing your data. With the standard exploration of data and combining them effectively into your code, you can proceed to construct the discriminator and generator blocks for the computation of the face generation. Let us cover these in the upcoming sections of the article. Firstly, we will construct the discriminator model for the generative network to classify the images as real or fake. The model architecture of the discriminator is exactly as discussed in the previous section. We will use the image shape of (64, 64, 3), which is the same size as the original images. The leaky ReLU will follow after each convolutional layer of varying filter sizes. Finally, after three blocks of the following combination, we will add a flatten layer, a dropout layer, and finally add the Dense layer with the sigmoid activation function to make the predictions of 0 or 1, for distinguishing between either real or fake images. Note that if you are performing the following activities for different image sizes, especially for a higher resolution one, you will need to modify some of the parameters, such as the filter size, for a more precise result. However, note that there might be several GPU limitations, and you might encounter a resource exhausted error depending on the type of machine you use on your local system. The Gradient platform on Paperspace is a great option worth considering for higher resolution tasks. The generator model of the DCGANs will be built, similarly as discussed in the overview section of this article. A few changes we will make are the modifications in the general activation function that are utilized in the generator model. Note that the ReLU functions suggested in the paper are replaced by the Leaky ReLU, similar to the ones that we used in the discriminator, and the final tanh function is replaced with the sigmoid activation. If you are looking at a higher quality of images to construct the project (like 512x512 or 1024x1024), ensure that you make the essential changes in the deconvolutional layer strides accordingly to suit the particular architecture.

Once you have created the generator and discriminator models, we can begin the training process of the GANs. We will also generate the images at intervals to ensure that we are receiving a brief idea of the performance of our architectural build. The final step of constructing our project is to train the models for a large number of epochs. Note that I have used only five epochs for the following train and saved an image at an interval of 100. The Gradient Tape function will automatically compile both the models accordingly, and the construction phase for the following project can be started. Keep training the model until you are satisfied with the results produced by the generator. Once your training procedure is complete, you can proceed to save the generator and the discriminator models in their respective variables either in the same directory or another directory of your creation. Note that you can also save these models during the training process of each epoch. If you have any system issues or other similar problems that might affect the complete computation of the training process, the best idea might be to save the models at the end of each iteration. Feel free to perform other experiments on this project. Some of the changes I would recommend trying out is to construct a custom class for the GANs and build a tracker of discriminator and generator loss accordingly. You can also create your checkpoint systems to monitor the best results and save the generator and discriminator models. Once you are comfortable with the given image sizes, it is worth trying out higher resolutions for better-interpreted results. Let us now look at some of the possible future works that we can focus on for further developments in this sector of neural networks and deep learning.

Face generation is a continuously evolving concept. Even with all the discoveries and the new modern technologies that have been developed in the field of generative adversarial networks, there are is an enormous potential to exceed far beyond the confines of simple face generations. In this section of the article, we will focus on the type of future works that are currently happening in the world of deep learning and what your next steps to progress in the art of face generation

with generative networks and deep learning methodologies must be. Let us analyze a couple of ideas that will help to innovate and elevate our generative models to even further superior levels. We know that the model that we currently built is capable of generating pretty realistic training images after a good few epochs of training, which could potentially take anywhere between several hours of training to a few days, depending on the type of system that you use. However, we can notice that even though the model performs quite well in generating a large number of faces, the amount of control that we possess upon the choice of generation is quite less as the generator model randomly generates a facial structure. A few major improvements that we can make to our models is to be able to construct them in a way such that a lot of parameters are under our control, and we have the option to stabilize the received outputs.

Some of the parameters and attributes that you can make adjustable are human traits such as gender, age, type of emotion, numerous poses of the head, ethnicity, toning of the skin, color of the hair, the type of accessories worn (such as a cap or sunglasses), and so much more. By making all these characters variable with the Generative Adversarial Network you construct, a lot of variations are possible. An example of performing such action is noticeable in the Face Generator website, where you can generate almost 11,232,000+ variants of the same face with numerous combinations. We can also make such advancements in our training model to achieve superior results while having decent control over the type of facial structures we choose to generate.

In one of the previous sections, we discussed how the Face App can change the age of a person from a specific range with the help of generative adversarial networks, namely using Cycle GANs. To take these experiments one step further would be to utilize Deepfakes in our projects. By making effective use of artificial intelligence and machine learning, namely using generative networks such as autoencoders and generative adversarial networks, it is possible to generate synthetic media (AI-generated media). This powerful technique can manipulate or generate faces, video content, or audio content, which is extremely realistic, and almost impossible to differentiate if fake or not. With modern Deepfakes computations, you can generate a face and make the same generated face perform numerous actions such as an intriguing dialogue delivery or a facial movement that actually never existed. There is enormous scope for combining our generative neural network architecture with other similar technologies to create something absolutely fabulous and unexpected. One of the future experiments worth trying out is to construct a generative model that can generate high-quality faces with high precision and realism. Using this generated face and combining it with Deepfakes technology, a lot of accomplishments and fun projects are possible. Feel free to explore the various combinations and permutations that these topics offer the developers to work with accordingly.

IV. Conclusion

The generation of faces with exceptional precision and the realistic display is one of the greatest accomplishments of generative adversarial networks. From deeming the following task of a facial generation to be almost impossible a few decades ago, to generating an average facial generation in 2014, to the continuous progression of these facial structures that look so realistic and fabulous is a humungous achievement. The possibilities for these generative neural networks to take over a wide portion of deep learning and create new areas of study are enormous. Hence, it is of utmost significance for modern deep learning researchers to start experimenting and learning about these phenomenal model architectures.

In this article, we understood the significance of generative adversarial networks on achieving a complex task such as generating realistic faces of humans that have never actually existed. We covered the topic of different pre-existing techniques that are continuously deployed in the modern development phase to achieve the following task successfully. We then studied a brief overview of our previous DCGANs architecture and understood how to deal with face generation with these networks. After an extensive code review for the following, we understood the procedure of face generation with DCGANs. Finally, we explored the enormous potential that these networks have by gaining more knowledge about the future developments and improvements that we can make to these models.

In the upcoming articles, we will further explore the topic of generative adversarial networks and try to learn the different types of GANs that have been developed through the years. One of the next works that we will explore will involve the utility of super-resolution generative adversarial networks (SRGANs). Until then, enjoy researching, programming, and developing!

References

[1] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, no. Mar, pp. 723–773, 2012.

[2] G. K. Dziugaite, D. M. Roy, and Z. Ghahramani, "Training generative neural networks via maximum mean discrepancy optimization," *arXiv preprint arXiv:1505.03906*, 2015.

[3] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, and B. Póczos, "Mmd gan: Towards deeper understanding of moment matching network," in *Neural Information Processing Systems*, 2017, pp. 2203–2213.

[4] Y. Li, K. Swersky, and R. Zemel, "Generative moment matching networks," in *International Conference on Machine Learning*, 2015, pp. 1718–1727.

[5] D. J. Sutherland, H.-Y. Tung, H. Strathmann, S. De, A. Ramdas, A. Smola, and A. Gretton, "Generative models and model criticism via optimized maximum mean discrepancy," *arXiv preprint arXiv:1611.04488*, 2016.

[6] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International Conference on Machine Learning*, 2017, pp. 214–223.

[7] J. Wu, Z. Huang, J. Thoma, D. Acharya, and L. Van Gool, "Wasserstein divergence for gans," in *European Conference on Computer Vision*, 2018, pp. 653–668.

[8] M. G. Bellemare, I. Danihelka, W. Dabney, S. Mohamed, B. Lakshminarayanan, S. Hoyer, and R. Munos, "The cramer distance as a solution to biased wasserstein gradients," *arXiv preprint arXiv:1705.10743*, 2017.

[9] H. Petzka, A. Fischer, and D. Lukovnicov, "On the regularization of wasserstein gans," in *International Conference on Learning Representations*, 2018, pp. 1–24.

[10] F. Juefei-Xu, V. N. Boddeti, and M. Savvides, "Gang of gans: Generative adversarial networks with maximum margin ranking," *arXiv preprint arXiv:1704.04865*, 2017.

[11] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks," in *Interspeech*, 2017, pp. 3364–3368.

[12] J. Adler and S. Lunz, "Banach wasserstein gan," in *Neural Information Processing Systems*, 2018, pp. 6754–6763.

[13] Y.-S. Chen, Y.-C. Wang, M.-H. Kao, and Y.-Y. Chuang, "Deep photo enhancer: Unpaired learning for image enhancement from photographs with gans," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6306–6314.

[14] S. Athey, G. Imbens, J. Metzger, and E. Munro, "Using wasserstein generative adversial networks for the design of monte carlo simulations," *arXiv preprint arXiv:1909.02210*, 2019.

[15] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Neural Information Processing Systems*, 2016, pp. 2172–2180.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Neural Information Processing Systems*, 2014, pp. 2672–2680.

[17] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *International Conference on Machine Learning*, 2017, pp. 2642–2651.

[18] E. L. Denton, S. Chintala, R. Fergus *et al.*, "Deep generative image models using laplacian pyramid of adversarial networks," in *Neural Information Processing Systems*, 2015, pp. 1486–1494.

[19] G. Perarnau, J. Van De Weijer, B. Raducanu, and J. M. Álvarez, "Invertible conditional gans for image editing," *arXiv preprint arXiv:1611.06355*, 2016.

[20] M. Saito, E. Matsumoto, and S. Saito, "Temporal generative adversarial nets with singular value clipping," in *IEEE International Conference on Computer Vision*, 2017, pp. 2830–2839.

[21] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville, "Adversarially learned inference," *arXiv preprint arXiv:1606.00704*, 2016.

[22] K. Sricharan, R. Bala, M. Shreve, H. Ding, K. Saketh, and J. Sun, "Semi-supervised conditional gans," *arXiv preprint arXiv:1708.05789*, 2017.

[23] T. Miyato and M. Koyama, "cgans with projection discriminator," *arXiv preprint arXiv:1802.05637*, 2018.

[24] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *International Conference on Computer Vision*, 2017, pp. 2223–2232.

[25] H.-Y. Lee, H.-Y. Tseng, Q. Mao, J.-B. Huang, Y.-D. Lu, M. Singh, and M.-H. Yang, "Drit++: Diverse image-to-image translation via disentangled representations," *International Journal of Computer Vision*, pp. 1–16, 2020.

[26] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[27] P. Zhu, R. Abdal, Y. Qin, and P. Wonka, "Sean: Image synthesis with semantic region-adaptive normalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[28] P. L. Suárez, A. D. Sappa, and B. X. Vintimilla, "Infrared image colorization based on a triplet dcgan architecture," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 18–23.

[29] J. Lee, E. Kim, Y. Lee, D. Kim, J. Chang, and J. Choo, "Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[30] Y. Yuan, S. Liu, J. Zhang, Y. Zhang, C. Dong, and L. Lin, "Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[31] Y. Zhang, S. Liu, C. Dong, X. Zhang, and Y. Yuan, "Multiple cycle-in-cycle generative adversarial networks for unsupervised image super-resolution," *IEEE transactions on Image Processing*, vol. 29, pp. 1101–1112, 2019.

[32] R. Navigli, "Word sense disambiguation: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 2, pp. 1–69, 2009.

[33] G.-J. Qi and J. Luo, "Small data challenges in big data era: A survey of recent progress on unsupervised and semi-supervised methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[34] L. Schmarje, M. Santarossa, S.-M. Schröder, and R. Koch, "A survey on semi-, self- and unsupervised learning for image classification," 2020.

[35] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1125–1134.

[36] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *International Conference on Machine Learning*, 2017, pp. 1857–1865.

[37] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for image-to-image translation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2849–2857.

[38] A. Mustafa and R. K. Mantiuk, "Transformation consistency regularization – a semi-supervised paradigm for image-to-image translation," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 599–615.

[39] M. Shi and B. Zhang, "Semi-supervised learning improves gene expression-based prediction of cancer recurrence," *Bioinformatics*, vol. 27, no. 21, pp. 3017–3023, 2011.

[40] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in neural information processing systems*, 2014, pp. 3581–3589.

[41] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, "Semi-supervised learning with ladder networks," in *Advances in neural information processing systems*, 2015, pp. 3546–3554.

[42] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *Advances in Neural Information Processing Systems*, 2019, pp. 5049–5059.

[43] R. Zhang, T. Che, Z. Ghahramani, Y. Bengio, and Y. Song, "Metagan: An adversarial approach to few-shot learning," in *Advances in Neural Information Processing Systems*, 2018, pp. 2365–2374.

[44] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 403–412.

[45] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in neural information processing systems*, 2017, pp. 4077–4087.

[46] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1199–1208.

[47] M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz, "Few-shot unsupervised image-to-image translation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[48] J. Lin, Y. Pang, Y. Xia, Z. Chen, and J. Luo, "Tuigan: Learning versatile image-to-image translation with two unpaired images," in *European Conference on Computer Vision*. Springer, 2020, pp. 18–35.

[49] J. Lin, Y. Wang, T. He, and Z. Chen, "Learning to transfer: Unsupervised meta domain translation," *arXiv preprint arXiv:1906.00181*, 2019.

[50] J. Lin, Y. Xia, S. Liu, T. Qin, and Z. Chen, "Zstgan: An adversarial approach for unsupervised zero-shot image-to-image translation," *arXiv preprint arXiv:1906.00184*, 2019.

[51] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001, pp. 327–340.

[52] C. Wang, H. Zheng, Z. Yu, Z. Zheng, Z. Gu, and B. Zheng, "Discriminative region proposal adversarial networks for high-quality image-to-image translation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[53] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8798–8807.

[54] B. AlBahar and J.-B. Huang, "Guided image-to-image translation with bi-directional feature transformation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[55] H. Tang, D. Xu, N. Sebe, Y. Wang, J. J. Corso, and Y. Yan, "Multi-channel attention selection gan with cascaded semantic guidance for cross-view image translation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2417–2426.

[56] T. R. Shaham, M. Gharbi, R. Zhang, E. Shechtman, and T. Michaeli, "Spatially-adaptive pixelwise networks for fast image translation," *arXiv preprint arXiv:2012.02992*, 2020.

[57] P. Zhang, B. Zhang, D. Chen, L. Yuan, and F. Wen, "Cross-domain correspondence learning for exemplar-based image translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5143–5153.

[58] F. Zhan, Y. Yu, K. Cui, G. Zhang, S. Lu, J. Pan, C. Zhang, F. Ma, X. Xie, and C. Miao, "Unbalanced feature transport for exemplar-based image translation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[59] X. Zhou, B. Zhang, T. Zhang, P. Zhang, J. Bao, D. Chen, Z. Zhang, and F. Wen, "Full-resolution correspondence learning for image translation," *arXiv preprint arXiv:2012.02047*, 2020.

[60] F. Zhan, Y. Yu, R. Wu, K. Cui, A. Xiao, S. Lu, and L. Shao, "Bi-level feature alignment for semantic image translation & manipulation," *arXiv preprint*, 2021.

[61] I. Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," 2017.

[62] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Neural Information Processing Systems*, 2017, pp. 5767–5777.

[63] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman, "Toward multimodal image-to-image translation," in *Neural Information Processing Systems*, 2017, pp. 465–476.

[64] A. Bansal, Y. Sheikh, and D. Ramanan, "Pixelnn: Example-based image synthesis," in *International Conference on Learning Representations*, 2018.

[65] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[66] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *stat*, vol. 1050, p. 1, 2014.

[67] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *International conference on machine learning*. PMLR, 2016, pp. 1558–1566.

[68] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," *arXiv preprint arXiv:1605.09782*, 2016.

[69] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *ICLR*, 2017.

[70] H. Kim and A. Mnih, "Disentangling by factorising," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2649–2658.

[71] E. L. Denton and v. Birodkar, "Unsupervised learning of disentangled representations from video," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4414–4423.

[72] A. Gonzalez-Garcia, J. Van De Weijer, and Y. Bengio, "Image-to-image translation for cross-domain disentanglement," in *Advances in neural information processing systems*, 2018, pp. 1287–1298.

[73] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[74] G. G. Chrysos, J. Kossaifi, and S. Zafeiriou, "Robust conditional generative adversarial networks," *arXiv preprint arXiv:1805.08657*, 2018.

[75] K. K. Thekumparampil, A. Khetan, Z. Lin, and S. Oh, "Robustness of conditional gans to noisy labels," in *Neural Information Processing Systems*, 2018, pp. 10 271–10 282.

[76] Y. Lu, Y.-W. Tai, and C.-K. Tang, "Conditional cyclegan for attribute guided face image generation," *arXiv preprint arXiv:1705.09966*, 2017.

[77] Q. Mao, H.-Y. Lee, H.-Y. Tseng, S. Ma, and M.-H. Yang, "Mode seeking generative adversarial networks for diverse image synthesis," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1429–1437.

[78] C.-H. Lin, E. Yumer, O. Wang, E. Shechtman, and S. Lucey, "St-gan: Spatial transformer generative adversarial networks for image compositing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9455–9464.

[79] S. Azadi, D. Pathak, S. Ebrahimi, and T. Darrell, "Compositional gan: Learning image-conditional binary composition," *International Journal of Computer Vision*, vol. 128, no. 10, pp. 2570–2585, 2020.

[80] F. Zhan, H. Zhu, and S. Lu, "Spatial fusion gan for image synthesis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 3653–3662.

[81] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim, "Image to image translation for domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[82] J. Cao, O. Katzir, P. Jiang, D. Lischinski, D. Cohen-Or, C. Tu, and Y. Li, "Dida: Disentangled synthesis for domain adaptation," 2018.

[83] A. H. Liu, Y.-C. Liu, Y.-Y. Yeh, and Y.-C. F. Wang, "A unified feature disentangler for multi-domain image translation and manipulation," in *Advances in neural information processing systems*, 2018, pp. 2590–2599.

[84] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski, "Plug & play generative networks: Conditional iterative generation of images in latent space," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4467–4477.

[85] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *International Conference on Machine Learning*, 2016, pp. 1–10.

[86] S. Hong, D. Yang, J. Choi, and H. Lee, "Inferring semantic layout for hierarchical text-to-image synthesis," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7986–7994.

[87] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *IEEE International Conference on Computer Vision*, 2017, pp. 5907–5915.

[88] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," in *Neural Information Processing Systems*, 2016, pp. 217–225.

[89] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas, "Stackgan++: Realistic image synthesis with stacked generative adversarial networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1947–1962, 2019.

[90] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie, "Stacked generative adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5077–5086.

[91] J. Gauthier, "Conditional generative adversarial nets for convolutional face generation," *Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester*, vol. 2014, no. 5, p. 2, 2014.

[92] G. Antipov, M. Baccouche, and J.-L. Dugelay, "Face aging with conditional generative adversarial networks," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 2089–2093.

[93] F. Zhan, C. Xue, and S. Lu, "Ga-dan: Geometry-aware domain adaptation network for scene text detection and recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9105–9115.

[94] X. Zhou, B. Zhang, T. Zhang, P. Zhang, J. Bao, D. Chen, Z. Zhang, and F. Wen, "Cocosnet v2: Full-resolution correspondence learning for image translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 465–11 475.

[95] L. Karacan, Z. Akata, A. Erdem, and E. Erdem, "Learning to generate images of outdoor scenes from attributes and semantic layouts," *arXiv preprint arXiv:1612.00215*, 2016.

[96] B. Dai, S. Fidler, R. Urtasun, and D. Lin, "Towards diverse and natural image descriptions via a conditional gan," in *IEEE International Conference on Computer Vision*, 2017, pp. 2970–2979.

[97] S. Yao, T. M. Hsu, J.-Y. Zhu, J. Wu, A. Torralba, B. Freeman, and J. Tenenbaum, "3d-aware scene manipulation via inverse graphics," in *Neural Information Processing Systems*, 2018, pp. 1887–1898.

[98] F. Zhan, S. Lu, and C. Xue, "Verisimilar image synthesis for accurate detection and recognition of texts in scenes," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 249–266.

[99] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros, "Real-time user-guided image colorization with learned deep priors," *arXiv preprint arXiv:1705.02999*, 2017.

[100] M. He, D. Chen, J. Liao, P. V. Sander, and L. Yuan, "Deep exemplar-based colorization," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–16, 2018.

[101] B. Zhang, M. He, J. Liao, P. V. Sander, L. Yuan, A. Bermak, and D. Chen, "Deep exemplar-based video colorization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8052–8061.

[102] Z. Xu, T. Wang, F. Fang, Y. Sheng, and G. Zhang, "Stylization-based architecture for fast deep exemplar colorization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9363–9372.

[103] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[104] Y. Song, C. Yang, Z. Lin, X. Liu, Q. Huang, H. Li, and C.-C. J. Kuo, "Contextual-based image inpainting: Infer, match, and translate," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[105] C. Li, H. Liu, C. Chen, Y. Pu, L. Chen, R. Henao, and L. Carin, "Alice: Towards understanding adversarial learning for joint distribution matching," in *Neural Information Processing Systems*, 2017, pp. 5495–5503.

[106] L. C. Tiao, E. V. Bonilla, and F. Ramos, "Cycle-consistent adversarial learning as approximate bayesian inference," *arXiv preprint arXiv:1806.01771*, 2018.

[107] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "Maskgan: Towards diverse and interactive facial image manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5549–5558.

[108] H. Tang, D. Xu, Y. Yan, P. H. Torr, and N. Sebe, "Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[109] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for image-to-image translation," in *International Conference on Computer Vision*, 2017, pp. 2849–2857.

[110] Y. A. Mejjati, C. Richardt, J. Tompkin, D. Cosker, and K. I. Kim, "Unsupervised attention-guided image-to-image translation," in *Advances in Neural Information Processing Systems*, 2018, pp. 3693–3703.

[111] M. Tomei, M. Cornia, L. Baraldi, and R. Cucchiara, "Art2real: Unfolding the reality of artworks via semantically-aware image-to-image translation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5849–5859.