PHOTO CARTOONIZATION USING GAN

A DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE
OF
MASTER OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING

Submitted by:

Dipak Sharma
2K21/CSE/09

Under the supervision of
Dr. Manoj Kumar
(Professor)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

JUNE, 2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi – 110042

CANDIDATE'S DECLARATION

I, Dipak Sharma, Roll No. 2K21/CSE/09 student of M. Tech (Computer Science and Engineering), hereby declare that the project Dissertation "Photo Cartoonization using GAN" which is submitted by me to the Department of Computer Science & Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation . This work has not previously formed the basis for the award of and Degree, Diploma Associateship, Fellowship or other similar title or recognition .

Place: Delhi Dipak Sharma

Date: 2K21/CSE/09

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi – 110042

CERTIFICATE

I hereby certify that the Project Dissertation titled **Photo Cartoonization using GAN** which is submitted by Amandeep Prasad, Roll No. 2K21/CSE/09, Department of Computer Science & Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the students under my supervision . To the best of my knowledge, this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere .

Place: Delhi

Date:

DR. MANOJ KUMAR

Professor

Department of CSE

ACKNOWLEDGMENT

The success of this project depends on the help and contribution of a large number of people as well as the organization . I am grateful to everyone who contributed to the project's success . I'd want to convey my gratitude to **DR. MANOJ KUMAR**, my project guide, for allowing me to work on this research under his supervision . His unwavering support and encouragement have taught me that the process of learning is more important than the ultimate result . Throughout all of the progress reviews, I am appreciative to the panel faculty for their assistance, ongoing monitoring, and motivation to complete my project . They assisted me with fresh ideas, gave crucial information, and motivated me to finish the task .

Dipak Sharma

2K21/CSE/09

ABSTRACT

In this report we are proposing a model capable of converting an image to cartoonized version. This functionality has many applications across anime, cartoon industry. The proposed model is one of the recent popular learning-based methods for stylizing images in artistic forms like paintings. However, in existing methods, (1) the style of comics has a unique feature of strong simplification and abstraction, which tends to have relatively simple textures and therefore does not give satisfactory results, poses a significant challenge to the popular loss functions for generator defined in recent developed methods. In this project, a Generative Adversarial Network (GAN) model is proposed for styling images very similar to cartoons. The model takes edge-smoothed cartoon images and for training. Other than that, to measure the model two new loss functions are also proposed. (1) loss of semantic content, its value indicates if converted image looks like cartoon images or not (i.e., clear and visible edges in the image), and (2) edge-promoting clarity loss to maintain a good edge. Also, the proposed model is capable of being trained much efficiently than existing methods proposed so far. With the experimental results we can tell that model performs good and is able to develop cartoon images from normal input image with high-quality.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ML	Machine Learning
ANN	Artificial Neural Network
GAN	Generative Adversarial Network
CNN	Convolutional Neural Network
LSTM	Long Short Tern Memory
DL	Deep Learning
RNN	Recurrent Neural Networks
API	Application Programming Interface
CSV	Comma Separated Value

INTRODUCTION

1.1 Overview

Generative Adversarial Networks (GANs) are a type of deep neural network architecture. The idea of this neural network architecture has gained significant attention in the field of machine learning.

GANs was first introduced back in 2014 by Ian Goodfellow and his colleagues (Goodfellow et al., 2014), and it since then it has become one of the most popular and promising approaches for generating new data samples that are identical to a given dataset.

The fundamental idea behind GANs is to train two models:

1) Generator model

2) Discriminator model.

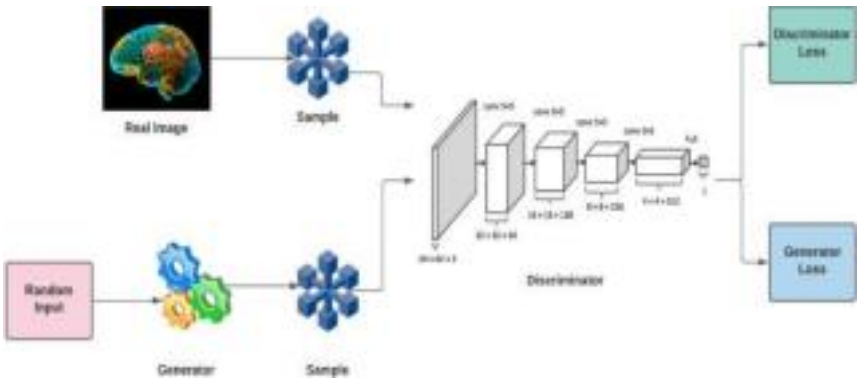


Figure 1. Block diagram of GAN

Both the models are generated to work together in a minimax game. The generator model learns to create new samples which look like they belong to the dataset. Whereas, in the discriminator model learns to provide a distinction in between real samples and generated samples. The two models are trained alternately to find the Nash equilibrium of the minimax game. In this game, none of the player can improve their payoff by changing their strategy. This can be mathematically represented as:

$$\min_G \max_D V(D,G) = \min_G \max_D \left(E_{x\sim P_{data}(x)}[logD(x)] + E_{z\sim P_z(z)}[log(1-D(G(z)))]\right)$$

Equation - 1

In the equation, G is the generator model, D is the discriminator model, x is a real data sample, z is a random noise vector.

The primary goal of the generator model is to minimize the value function V, while the discriminator model aims to maximize it. Throughout the training process, both models are updated iteratively in order to converge towards the Nash equilibrium of the minimax game.

The potential applications of GANs are plenty and varied. These varieties include image synthesis, text generation, and music composition. But, GANs are normally difficult to train and susceptible to mode collapse. This means, the generator model only learns to produce a limited set of samples. As a result, researchers have proposed numerous modifications to the GAN architecture to improve its stability and performance.

In last few years, there have been many proposed modifications to the standard GAN architecture. For instance, Wasserstein GANs [1] and Progressive GANs [2]. These modifications address the limitations of the standard architecture and have displayed promising results in different applications.

Likewise, this thesis also seeks to provide a comprehensive understanding of the GAN architecture and its applications. Specifically, this thesis will explore the mathematical foundations of GANs and their training process using a training model which alters the original image into cartoon shade. We will also have a look at the challenges and limitations of the standard GAN architecture in this thesis.

Challenges associated with GANs

Generative Adversarial Networks (GANs) have emerged as a promising approach for generating authentic samples across various domains, including images, text, and music.

However, the training of GANs is troubled by instability and can lead to issues such as mode collapse and poor sample quality. These issues have become a hindrance to the wider adoption of GANs in practical applications i.e. in data augmentation, image synthesis, and text generation. Moreover, the use of GANs for certain applications such as medical image synthesis and text generation mandates greater reliability and control, which makes the training process more challenging.

Several modifications and extensions have been proposed to tackle the training instability and address the poor sample quality of GANs. However, these solutions typically require additional computational resources and longer training times, and hence, it adds up different challenges of using GANs in practical applications.

So, the key problem statement for GANs is to improve the stability and reliability of their training process. Also, the bigger challenge will be to maintain and keep enhancing the quality of the generated samples.

1. What are the primary challenges associated with training GANs?

2. How do these challenges impede the wider adoption of GANs in practical applications?

3. What methods have been suggested to tackle the instability and subpar sample quality commonly encountered in GANs?

4. What are the shortcomings of these approaches?

5. How can GANs be advanced to better address the challenges encountered in practical applications such as medical image synthesis and text generation

GANs have been widely used in image-related applications such as image-to-image translation, super-resolution, and style transfer. However, the research on GANs is still ongoing, and on a progressive path. So, there are several questions that researchers can use to enhance their performance and potential applications. Here are five such research questions pertaining to GANs:

Can we improve the stability of GAN training model?

One of the primary issues with GANs is the instability of their training process. This eventually leads to issues such as mode collapse and convergence problems. So, Researchers can explore

different techniques to improve the stability of GAN training, such as changing the loss functions, modifying the architecture of the generator and discriminator , or incorporating regularization techniques.

How can we make GANs more efficient?

GANs often requires an extensive amount of training data and computational resources. Both the mandates are expensive and time-consuming. One possibility is try and explore ways to make GANs more efficient, such as developing faster training algorithms. This can help in reducing the computational complexity of the models, or help in exploring new architectures that require fewer parameters.

Can we make GANs more interpretable?

While GANs are powerful tools for image generation, their inner functions are quite difficult to interpret. However, Researchers can look for ways to make GANs more interpretable. So, to comprehend the inner workin, developing methods to visualize the internal representations of the generator and discriminator, or developing techniques to extract meaningful features from the generated images can be a possible solution.

What measures can be taken to enhance the quality of generated images?

The quality of generated images plays a vital role in numerous GAN applications, such as image-to-image translation and super-resolution. To enhance the quality of generated images, one can investigate various approaches, including the formulation of novel loss functions to better capture perceptual quality. Additionally, exploring new architectures tailored for high-resolution images and incorporating additional information like semantic labels or style information can be beneficial avenues to explore.

Can we extend GANs to new domains?

While GANs have been successfully applied to a wide range of image-related applications, we can surely extend the concept to many other domains where they could be useful. For example, they can assist in text generation or video generation. Researchers can explore ways to extend GANs to these new domains, such as developing new architectures that can generate text or video, or exploring ways to combine GANs with other deep learning models.

1.2 Project objective

In order to translate images into cartoons, this research investigates the usage of generative adversarial networks (GANs). The main goal is to create GANs that can quickly generate excellent cartoon-style graphics from the real-world images. Additionally, the output must display a variety of traits and styles. The previous research on GANs and image-to-image translation will be expanded upon by this study. The goal is to achieve image to cartoon conversion, nevertheless.

The study will investigate different GAN-based approaches and architectures for image-to-cartoon translation in order to meet this goal. The efficacy of these techniques in producing high-quality cartoon-style images with in-depth details and vibrant colors will also be examined in this study. To train and test the suggested GAN-based models, the study will make use of a sizable collection of real-world photos and equivalent cartoon-style images.

The study will make an attempt to explore different prospects of GANs in image-to-cartoon translation and also try to conjure up an estimate whether this technology can be used to generate cartoon-style avatars for social media profiles, cartoon-style filters for social media apps, and produce cartoon-style illustrations for books and magazines. While exploring such dynamic domains, the research also tries to analyze how the concept can have an impact in the gaming industry i.e., an attempt to create cartoon-style characters and environments, for basically providing an immersive gaming experience.

Furthermore, the study will explore the limitations and challenges of using GANs in image-to-cartoon translation. The research will investigate the issues related to training the GAN models, such as mode collapse and training instability. The study will also provide insights into the ethical and legal implications of using GANs to generate cartoon-style images, such as copyright infringement and the potential misuse of the technology.

LITERATURE REVIEW

2.1 Key Concepts

Types of GAN

The first type of GAN is the vanilla GAN. The idea of this GAN was proposed by Goodfellow et al. in 2014 [3]. This GAN has two neural networks i.e. a generator and a discriminator. The generator is trained to deliver synthetic data that is identical to the real data. Here, the discriminator's role is to differentiate between real and synthetic data. The two networks are trained simultaneously in a game-like setting. The training of both the network is done until the generator produces a realistic synthetic data that can fool the discriminator .

The second type of GAN is the Conditional GAN. This type was introduced by Mirza and Osindero back in 2014 [4]. The Conditional GAN generates synthetic data based on a given condition. For example, the condition could be a label that represents the class of the image to be generated. The generator produces synthetic data that is conditioned on the label, and the discriminator is trained to specify the difference between real and synthetic data based on the label.

The Cycle GAN, which was first presented in 2017 [5] by Zhu et al., is the third form of GAN. To translate images into other images, the Cycle GAN is employed. Two generators and two discriminators are present. While the discriminators are trained to tell the difference between actual and fake images, the generators are trained to map the images from one domain to another. The output image from one domain can be used, translated to the other domain , and translated back without losing any information when the two generators have been trained jointly in a cycle-consistent way.

The fourth type of GAN is the Wasserstein GAN, which was introduced by Arjovsky et al. in 2017 [6]. The Wasserstein GAN is specifically designed to address the instability issue of the type 1 i.e. vanilla GAN. It replaces the traditional binary classification loss function with a Wasserstein distance-based loss function. This distance-based loss function gives a more stable training process and produces better quality synthetic data.

The fifth type of GAN is the Progressive GAN, which was introduced by Karras et al. in 2017 [7]. High-resolution images are created using the Progressive GAN. It begins by creating images with a low resolution and gradually raises the resolution until the desired resolution is reached. This approach allows the model to learn the underlying structure of the data and generate high-quality images.

Related work

The GAN based methods are quite successful but, there are many challenges involved. To extend the domain of the research it's important to explore different methods and related approaches which explore the potential of the GANs and see how they can contribute to improving the quality and efficiency and quality of the image to cartoon translation method.

This section will analyses various approaches that are directly related to our goal.

Image to image translation

Hertzmann et al. proposed the idea for image-to-image translation [8]. The idea is traced back to their work on Image Analogies. On one input-output training image pair, they utilized a nonparametric texture model there. This method helped to map a source image to a target image while maintaining the structural and textural integrity of the original image. Still, this method had limitations in terms of the type of images it could process and the amount of control it had over the output.

Since then, researchers have given a variety of GAN-based image-to-image translation techniques. These methods can handle diverse image types and provide greater control over the output. Some of the well-known GAN-based image-to-image translation techniques are - CycleGAN [9], Pix2Pix [10], StarGAN [11]. These methods use a pair of related images to train the GAN model and translate one type of image to another.

CycleGAN [9] is a known because of the unsupervised image-to-image translation technique based on GAN. Without paired training data, this approach can understand the mapping between two domains. It uses a cycle consistency loss to make sure that the translation is consistent in both directions.

Another well-liked GAN-based image-to-image translation approach, Pix2Pix [10], learns the mapping between two domains using paired training data. To create high-quality images that maintain the style and structure of the original image, it uses a conditional GAN.

A multi-domain image-to-image translation technique that can handle many attributes at once is called StarGAN [11]. It may translate between images in a variety of domains, such as altering an input image's face expression or hair color.

The concept of neural style transfer, which seeks to synthesize an image in the style of one image while preserving the content of another, was first introduced by Gatys et al. [12]. They used deep convolutional neural networks to extract feature representations of both the content and style images. Afterwards, they used an optimization algorithm to iteratively alter the content image to match the style features. The technique has since been extended in different ways i.e. using adversarial networks [13], incorporating spatial constraints [14], and combining multiple styles [15].

One popular extension is the use of sequential models, where the output of one neural network is used as input for another network. Huang and Belongie [16] proposed a feedforward network that can perform style transfer in real-time by sharing the feature maps between the content and style networks. Luan et al. [17] extended this idea by introducing a multi-pass architecture, where the input image is passed through the network multiple times, each time with a different style image, resulting in a more flexible and diverse set of stylized outputs.

Formulation: Learning Image-to-Image Translation without Paired Data

Without paired training data, the CycleGAN model's goal is to learn the mapping functions G: X Y and F: Y X that connect the two domains X and Y. Two adversarial discriminators, DX and DY, are used to do this. They distinguish between genuine images (x) and translated images (F(y)), as well as between y and G(x) [18]. The goal of the model incorporates cycle consistency losses to guarantee the coherence of the learnt mappings and adversarial losses to align the distribution of output images with the distribution of target domain data. The cycle consistency loss is based on the idea of transitivity and has been used in the past for tasks like language translation and visual tracking [19].

Adversarial Losses for Image-to-Cartoon Translation

In image-to-cartoon translation, adversarial losses are applied to both mapping functions to ensure that the generated images are similar to the target domain. For the mapping function G : X → Y, where X is the image domain and Y is the cartoon domain, the objective can be expressed as:

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

Equation - 2

The goal of G in this situation is to produce cartoon-like images G(x) that mimic images from the cartoon domain . However, DY's objective is to distinguish between authentic cartoon samples y' and translated samples G(x). The mapping function F: Y X and its discriminator DX can be used to create an adversarial loss that is denoted as LGAN (F, DX, Y, X).

This strategy, which has been used in a number of image translation challenges [21], is motivated by the concept of generative adversarial networks [20].

In summary, several methods have been proposed for multi-domain image-to-image synthesis, but each has its limitations. It is important to carefully evaluate these methods and choose the one that is best suited for the specific task at hand.

Additional Methods

In the previous section we saw some of the significant approaches pertaining to the GAN and how different research has provided different constraints in the development of the GANs. However, there are many additional methods (Novel Methods) which are helpful in exploring the premise for Image to cartoon translation. This section will provide with the details of many methods that were helpful in executing different objectives.

Cycle-Consistent Adversarial Networks for Image-to-Image Translation:

Our objective includes two parts: adversarial losses to match the resulting picture distribution to the desired domain, as well as a cycle consistency loss to stop the learnt mappings from conflicting with one another. To balance these two objectives, we introduce a hyperparameter λ that controls their relative importance. Specifically, we aim to solve the following optimization problem:

$$G^*, F^* = \arg\min_{G,F} \max_{D_x, D_Y} \mathcal{L}(G, F, D_X, D_Y).$$

Equation - 3

Our method may be understood as the training of two different autoencoders, each of which is built with special internal structures that allow mapping an image to itself by using an intermediary representation that acts as a translation of the image into a new domain. This setup can also be seen as a particular application of adversarial autoencoders, which use an adversarial loss to train the bottleneck layer of an autoencoder and align it with a desired target distribution. The target distribution in our particular circumstance is denoted by [31].

The authors compared their approach to ablations of the whole objective and empirically demonstrated that both aims are essential for producing results of a high caliber [31].

Cycle-Consistent Adversarial Networks for Image-to-Image Translation

Adversarial training is used in the field of image-to-image translation to learn mapping functions between two domains, such as converting photos into cartoons, that can produce outputs with the same distribution as the target domain [37]. It is feared that a network with enough capacity might translate identical input images to several random permutations of images in the destination domain, creating an unmanageably huge space of potential mapping functions. It is crucial for the learnt mapping functions to demonstrate cycle-consistency in order to address this problem and make sure that the image translation cycle can return the original image to itself. Both forward cycle consistency and backward cycle consistency are used to achieve this. It should be feasible to translate each image from domain X to G(x) and then back to the original image using F, i.e., x G(x) F(G(x)) x. Similar to this, each image y from domain Y should be capable of being translated to F(y) and subsequently returned to the original image using G, i.e., y F(y) G(F(y)) y [36].

The introduction of a cycle consistency loss serves to reward this behavior. This loss measures the discrepancy between the original image and the image that has been recovered after a complete cycle of translation [36]. The loss function is defined as:

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

Equation - 4

where k · k1 represents the L1 norm. In preliminary experiments, an adversarial loss was also tried as a replacement for the L1 norm, but no improvement in performance was observed [36].

The arXiv version [36] provides a demonstration of the effect caused by the cycle consistency loss. This method is a specific application of adversarial autoencoders [39], which use an adversarial loss to train an autoencoder's bottleneck layer with the goal of bringing it into alignment with a specified target distribution. Additionally, this method can be seen as training two distinct autoencoders [38], one of which learns the mapping from X to Y and the other from Y to X. The intermediate representation of the image, in contrast to traditional autoencoders, is a translation of the image into a different domain [36].

Caricature Generation and Face Warping

In the following section, we will explore the literature pertaining to the caricature generation & face warping and analyses different aspects.

Human faces have been a popular research topic, with several approaches developed for modeling, interacting, or generating them [28, 24, 26]. Several strategies have been put forth in the area of image warping. For instance, flow-based algorithms like [27] learn dense deformation fields throughout the entire image, whereas parametric methods like the spatial transformer [25] estimate global transform parameters. Zhao et al. [30] use dense flow estimation to address geometric misrepresentation in close-range portrait photos, whereas DeepWarp [23] recommends using it for gaze manipulation.

In the work of Cole et al. [22], portrait photographs are warped while maintaining their identity using spline interpolation on pre-detected landmarks. This method is comparable to the strategy used by Zhang et al. [29] for parallax-tolerant image stitching, which includes terms for smoothness, local, and global alignment that are identical to our own loss functions. We want to use flow estimation and differentiable warping modules from [25] to anticipate warping fields for creating cartoons as part of our AutoToon project.

Caricature Generation: Techniques and Challenges

Research on caricature creation has been ongoing for many years. Conventional methods emphasize a face's distinctive traits by locating and warping landmarks or by using data-driven techniques to calculate unique facial features. Early rule-based approaches, however, restricted the variety of caricatures [40, 41, 47, 48]. Deep learning techniques have been employed recently by researchers like Wu et al. [50], who used a neural network to model the subject's face in 3D to further improve the typical appearance of the caricature expression.

Data-driven techniques have also been applied, with readily available datasets of annotated caricatures such as WebCaricature [45], which comprises caricatures & photographs from many different identities. However, a crucial point of consideration is that the limited amount of data available still remains a challenge. Therefore, recent works have drawn inspiration from the generative image-to-image translation literature trained on unpaired images [43, 44, 53] and focused on acquiring knowledge and learning from unpaired portraits, and caricatures [42, 51, 52].

Using a Generative Adversarial Network to understand the image-to-caricature translation, CariGAN [42] introduced the first deep learning technique to caricature production. After that, Shi et al. [49] introduced a technique that simultaneously trains style and warping end-to-end using the Generative Adversarial Network architecture. Unpaired learning still presents a wide range of exaggerations from artists with varying stylistic inclinations, making it challenging to acquire consistent exaggerations.

Therefore, we attempt to adopt a paired supervised learning technique in our work that is closely related to these suggested methodologies. The method is based on the creations of two artists, and an effort is made to balance this trade-off by choosing to understand each artist's distinct styles thoroughly rather than an average of all types. For a particular exaggeration, we also use the differentiable warping module from [46] to produce denser warping fields.

Contrary to other works, we tried to totally separate geometry and style while concentrating just on the warping stage of caricature development to conjure some high-quality warps.

Moreover, the limited amount of data and the challenge of caricature diversity remain a major challenge for caricature generation. Therefore, there is a need for further research to address these issues and improve the accuracy and quality of generated caricatures.

AutoToon : Problem Formulation and Warping Model

Exaggerating the facial features of a portrait to generate a cartoon image is the fundamental task in caricature generation. Our proposed method, AutoToon, focuses on achieving this step in the pipeline. AutoToon takes an RGB portrait image X with dimensions H, W, and 3, and applies an artist-inspired facial exaggeration to generate a cartoon image Xˆtoon. The resulting cartoon image is then fed into a stylization network for the final stage of caricature generation.

To accomplish the facial exaggeration step, AutoToon employs a deep neural network that learns to generate dense facial warping fields. Unlike previous methods that rely on sparse warping points, our method utilizes the differentiable warping module from [33] to generate a dense and continuous warping field that can capture more detailed facial features.

By training on a dataset of paired portrait and caricature images, AutoToon learns to disentangle the geometry and style components of the facial exaggeration process. This disentanglement enables our method to generate high-quality caricatures with specific artist styles well.

Warping and Linear Interpolation in AutoToon: An Exaggeration-based Caricature Generation Method

To perform facial exaggeration for a given portrait image Xin, our AutoToon network learns a flow field, known as the warping field, denoted by Fˆ ∈ R

H×W×2. The per-pixel displacement in the x direction is encoded in the first channel, while the y direction is encoded in the second channel. The differentiable warping module, Warp(Xin, F), then applies this warping field to Xin by utilizing bilinear interpolation to move Xin's pixels in accordance with the learned displacements. This module, which we call the Warping Module, was adapted from the Spatial Transformer Networks[33]. This process yields

the exaggerated cartoon image Xˆ toon. Linear interpolation is used in the Warping Module to ensure smooth transitions between neighboring pixels.

A total of 101 frontal-facing portrait images of non-celebrity individuals with diverse characteristics, such as age, gender, race, and facial structure, were collected from Flickr for the XToon dataset [58]. Two caricaturists with similar techniques twisted these photos in Adobe Photoshop to create the ground-truth artist cartoons. The dataset comprised a paired set of 101 images, consisting of the original images (Xin) and their corresponding artist cartoons (Xtoon), which were split into 90 training and 11 validation images. Figure 3 shows examples of images from the training set, whereas the test set was compiled from several sources without ground truth labels. Additionally, the XToon dataset includes estimated artist warping fields (F32 R 32322) that depict each artist's caricature following bilinear upsampling to dimensions H W 2. The choice to use a warping field spatial size of 3232 is explored in the section that follows. To acquire these warping fields, the authors employed gradient descent optimization on the warping field for each Xtoon while using the differentiable Warping Module with L1 loss. The optimization goal, designated as argmin, was to produce artist warping fields that closely matched each Xtoon.

$$\underset{F_{32}}{\operatorname{argmin}} ||X_{toon} - \mathit{Warp}(X_{in}, \mathit{Upsample}(F_{32}))||_1 \ .$$

Freeze Style vector and Generator (FreezeSG) for Better Image Generation

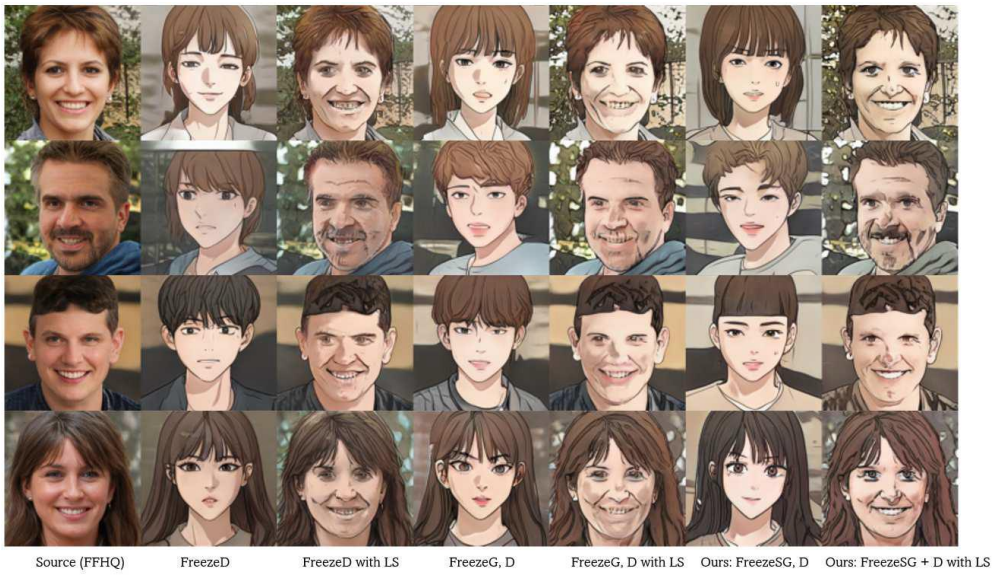


Figure-2 Fine tuning procedure of StyleGAN2

The FreezeG method has been effective in generating images, but recent analysis and studies has shown that both the initial layers of the generator & the style vectors that are being injected into it play a role in determining the structure of the generated image. Building on this insight, researchers have developed a simple and effective method called FreezeSG that freezes both the initial blocks of the generator and style vectors during the fine-tuning procedure of StyleGAN2 [56]. Results from Figure 1 demonstrate that FreezeSG produces images that better reflect the source image compared to freezing the generator alone. Additionally, combining the high-resolution layer of the generator in the target domain with the generator's low-resolution layer in the source domainis done through Layer Swapping (LS) further preserves the structure of the source image [56]. So, in general, FreezeSG with LS outperforms other methods such as FreezeG or the baseline (FreezeD + ADA) in generating images that closely resemble the source image.

Introducing Structure Loss for Improved Layer Swapping

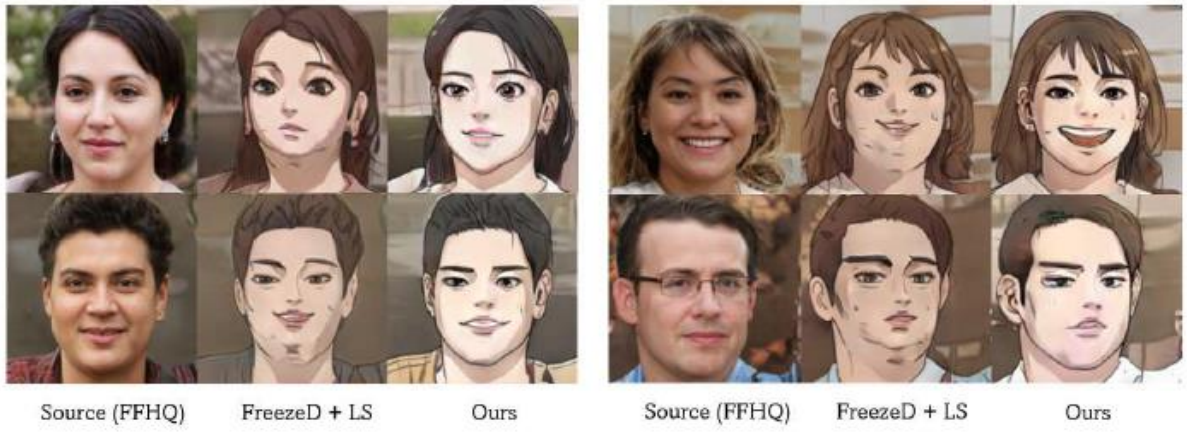


Figure-3 Output after layer swapping

The structure of the source image is maintained when layer swapping (LS) is used to combine the low-resolution layer of the source domain generator with the high-resolution layer of the target domain generator. Layer swapping on the low-resolution layer is possible, but it is challenging to get meaningful results because the weights of this layer are fixed. Researchers have developed the structure loss, a simple yet effective loss function, to get around this problem. According to [57], this loss function adjusts the RGB output of the source generator during training so that it matches the RGB output of the target generator.

Since the structure of the image is mostly determined at this level, the low-resolution layer is often where the structure loss is applied. The RGB outputs of the source and target generators are retrieved at each resolution in order to incorporate structural loss for multiple style blocks.

For both resolutions, the mean squared error (MSE) loss between the RGB outputs of the source and target generators is determined. These losses are then combined up to the n-th layer [57].

The objective functions of the model are expressed as

$$\mathcal{L}_D = -\mathcal{L}_{adv}$$
$$\mathcal{L}_G = \mathcal{L}_{adv} + \lambda_{structure}\mathcal{L}_{structure}$$

LD = −Ladv for the discriminator, and LG = Ladv + λstructureLstructure for the generator, where λstructure is a hyper-parameter that controls the relative importance of the structure of the source domain [57]. Results from Figure 3 demonstrate us that this approach generates images that are more natural compared to the baseline methods [57]. Areas like the jaws and skulls are well-generated by processing the first layer of G to learn the structure.

Equation

The objective of the original GAN expressed as:

$$\mathcal{L}_{adv} = \mathbb{E}_{x\sim p_{data}}\left[\log D(x)\right] + \mathbb{E}_{z\sim p(z)}\left[\log\left(1 - D(G(z))\right)\right] \quad (6)$$

The structure loss is calculated as:

Lstructure = Xn k=1 E - Gs_l=k(ws) − Gt_l=k(wt) ²

where Gs_l=k(ws) is the RGB output of the source generator, Gt_l=k(wt) is the RGB output of the target generator, and k is the index of the style block, and n is the number of style blocks [57].

The following is a representation of the discriminator's and the generator's objective functions:

LD = −Ladv

LG = Ladv + λstructureLstructure

We can see from the equation that the structure is a hyper-parameter that regulates the relative size of the source domain's structural complexity [57].

Texture Synthesis

Texture synthesis is mainly categorized into two types: coarse-grained and fine-grained [59]. While coarse-grained texture synthesis is mainly focusing to replicate the overall appearance of an input image in the output, fine-grained texture synthesis emphasizes on creating synthetic textures in greater detail. The objective of fine-grained texture synthesis is to produce highly realistic textures which shall resemble to those found in the input image, while also incorporating additional variations and details to create a visually appealing output.

PSGAN

A team of researchers led by Bergmann introduced a novel approach to texture synthesis using Generative Adversarial Networks (GANs), called Periodic Spatial GAN (PSGAN) [60]. The PSGAN model, depicted in Figure 4**,** was proposed to generate realistic textures.

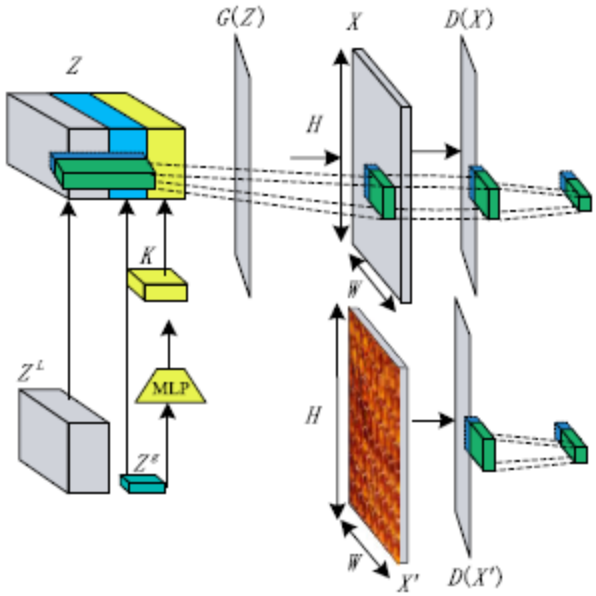


Figure-4 PSGAN model

The PSGAN method is a promising approach to texture synthesis, leveraging GANs' ability to generate high-quality, realistic images. By incorporating periodic spatial attention mechanisms, PSGAN enables the generation of textures that are both visually appealing and accurate to the ground truth [60].

The PSGAN uses a loss function denoted as V(D,G) that involves two terms, which are optimized through a min-max game between both the generators i.e. the generator G and the discriminator D [60]. Now, the first term is a measure of generator's ability to deceive the discriminator, while the second term evaluates and shows discriminator's ability of distinguishing real samples with the help of the generated ones.

$$\min_G \max_D V(D,G) = \frac{1}{LM} \sum_{\lambda=1}^{L} \sum_{\mu=1}^{M} \mathrm{E}_{Z\sim Pz(Z)}[\log(1 - D_{\lambda\mu}(G(Z)))] + \frac{1}{LM} \sum_{\lambda=1}^{L} \sum_{\mu=1}^{M} \mathrm{E}_{X'\sim Pdata(X)}[\log D_{\lambda\mu}(X')]$$

Equation - 7

The first term can be written as the sum of L separate functions, where each function corresponds to a different pair of indices λ and μ [60]. For each pair, the function computes the expected value of log(1 - Dλμ(G(Z))) over a noise distribution Pz(Z). Here, Z denotes a random input vector that is used as the input to the generator G. The aim is to minimize this term by finding a generator that can produce samples that fool the discriminator.

The second term can also be shown as the sum of L functions, where each function corresponds to a different index λ [60]. For each function, the expected value of log(Dλ(X0)) is computed over a real data distribution Pdata(X), where X0 denotes a real sample from the dataset. The aim is to maximize this term by finding a discriminator that can correctly distinguish between real and generated samples.

PSGAN can often use one or more large datasets with more expansive images to learn numerous textures. It may generate original samples and interpolate between these textures [60]. Additionally, PSGAN is highly scalable for producing output images of any size and can successfully handle a variety of textures and image data sources.

ProGAN

Karras et al. introduced a novel image generation method called ProGAN. This method gradually increases the generator & discriminator from a low resolution. In fact, it also manages to add some innovative layers as the training progresses to capture fine details [32]. The structure of ProGAN is illustrated in Figure. This approach not only quickens the training but, also nicely manages to stabilize it.

In comparison with some earlier GAN works, ProGAN has demonstrated high-quality results and stable training in high resolution [32]. However, this method has some limitations such as semantic sensitivity and the actual understanding that are dependent on the different constraints of datasets. It is important to note that while ProGAN has achieved remarkable results, further research is needed to overcome its shortcomings and to extend its applicability to a wider range of datasets.

In summary, ProGAN is a significant contribution to the field of image generation utilizing the potential of GANs, with a prospect of generating high-quality images with fine-grained information from complex datasets.

Big gan

This method is proposed by Brock et al. [34] and it successfully full fills the intent of generating high-resolution and diverse samples with the help of complex dataset ImageNet. This approach mainly involves training large-scale Generative Adversarial Networks, and is presently one of the extensive scale of GANs that are trained to generate images of unprecedented quality. The realism of the generated image using this method is far more superior and better in comparison to earlier methods [34].

Now, to handle the particular instability of such scale, the generator underwent orthogonal regularization by the authors [34]. In order to control the fidelity and variance of generated images, they also reduced latent space [34]. This technique allows the generated images to have realistic fine-grained details, resulting in visually pleasing outputs.

In conclusion, the method presented by Brock et al. [34] represents a significant advancement in the field of image production using GANs. The approach has set a new benchmark for the quality of created photos thanks to its exceptional ability to create high-resolution and varied images from complex datasets.

Face Super-Resolution GANs

This new idea is presented by Kim et al. and it suggests new approach for face super-resolution i.e SR that produces photo-realistic images while preserving fine facial details [54]. The network architecture used in their approach is shown in Figure 5**.**

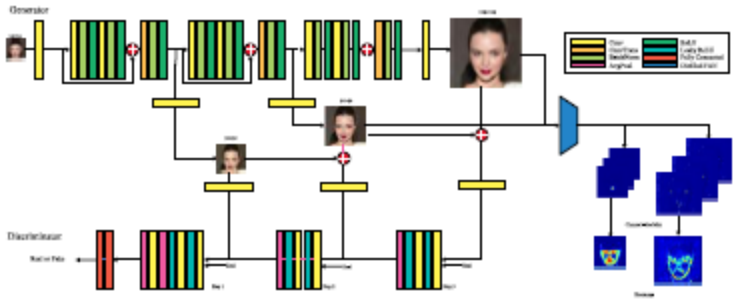


Figure-5 Face Super GAN

To achieve their results, the authors employed a loss term that is expressed as

$$L_{Ours} = \alpha L_{pixel} + \beta L_{feat} + \gamma L_{WANG}$$
$$L_{Ours} = \alpha L_{pixel} + \beta L_{feat} + \gamma L_{WANG} + \lambda L_{heatmap} + \eta L_{attention}$$

Equation - 8 & 9

LOurs = αLpixel + βLfeat + γ LWANG in Equation (8), and later extended to include additional loss terms for facial attention and landmark heatmaps in Equation (9).

The proposed method employs a progressive training approach where the network is divided into successive steps, each of which produces an output with higher resolution than the previous one. This allows for stable training and improved results. To further improve facial attribute restoration, the authors also introduced a novel facial attention loss, which is applied at each step by averaging pixel contrasts with heatmap values.

In addition, the authors projected a condensed form of the face alignment network, or FAN, to extract landmark heatmaps appropriate for face SR. The general performance of their approach was evaluated on a variety of datasets and demonstrated superior results compared to existing methods.

StyleGAN: Image Synthesis

Karras et al. shows a novel generator architecture for GAN dubbed as StyleGAN [55]. The network architecture of the projected StyleGAN, as portrayed in the Figure. 10, allows for adjusting the image style that is actually based on the latent code in every specific convolutional layer. The generator controls the whole image synthesis procedure, starting, with a low resolution and gradually (step-by-step) generating the high-resolution images.

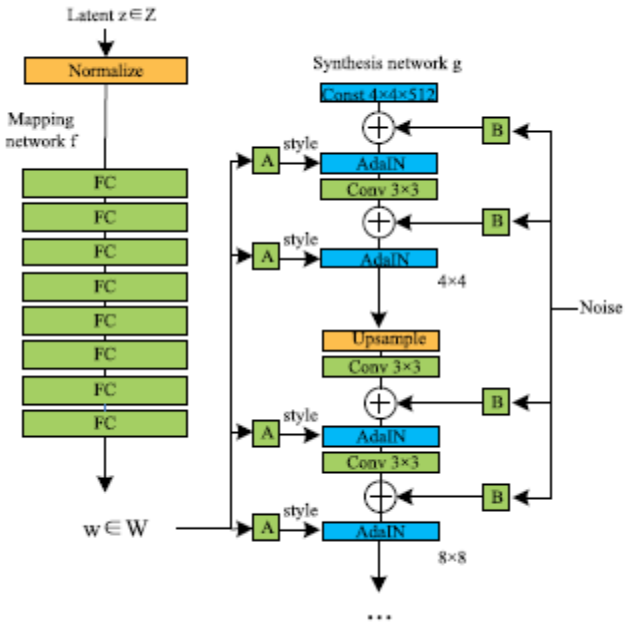


Figure-6 StyleGAN

Moreover, it regulates visual characteristics by modifying the input of each network level individually. Progressing from coarse to fine details This advancement enables the production of realistic and high-quality images, while granting more refined control and improved comprehension of the generated images. StyleGAN introduces unsupervised attribute separation at a high-level, automated learning, and stochastic variation of generated images, resulting in intuitive and scale-specific control over composition synthesis Furthermore, this approach surpasses conventional GAN generator architectures, excelling in generating high-resolution, lifelike images [55].

StyleGAN's ability to modify image style accurately depending on the latent code in each convolutional layer gives it superior control and a clearer grasp of the generated images [21]. The StyleGAN generator architecture may then regulate the entire picture synthesis process by gradually increasing the resolution of artificial images to reach high resolution.

The visual features of the generated images can be modified by StyleGAN from common features to fine details, allowing for perceptive, In order to provide scale-specific control synthesis of the composition, StyleGAN performs autonomous learning, unsupervised high-level attribute separation, and introduces a stochastic variation in generated images resulting in the generation of high-quality and realistic images Compared to traditional GAN generator architectures, StyleGAN is superior in its ability to generate high-resolution images that look more realistic [43].

SRGN, ESRGAN, and SFTGAN

Ledig et al. [95] introduced a GAN-based approach, known as SRGAN. The aim was to upgrade the quality of images in super-resolution (SR). The proposed method significantly improves the perceptual quality of generated images. Wang et al. [96] further improved SRGAN by suggesting Enhanced SRGAN i.e. ESRGAN, it overcomes some of the issues of the artifacts in SRGAN and generates more realistic and natural textures in the generated images. In another study, Wang et al. [97] proposed SFTGAN, operational with a novel Spatial Feature Transform layer also (SFT Layer), to recover natural and more realistic textures in the resultant generated images.

The best-performing image super-resolution method involves training generators and discriminators starting some of the low-resolution photos,then, adding a more detailed network layer in every individual step for generating high-resolution images. This method produces a range of high-quality pictures. Regardless, it requires long training times and multiple GPUs.

Ledig et al. [95] developed the Super-Resolution Generative Adversarial Network (SRGAN) as a deep learning technique for super-resolution images (SR). To learn the mapping between low-resolution and high-resolution images, SRGAN uses a generative adversarial network architecture. A generator network and a discriminator network are the two neural networks that make up this architecture. The discriminator network compares the generated image against actual high-resolution photographs in order to assess the quality of the generated image, which is created by the generator network using a low-resolution image as input. The generator network may successfully learn to make images that look similar to real high-resolution ones by using an adversarial training approach.

Enhanced SRGAN (ESRGAN) is an improvement of SRGAN that is a suggestion made by Wang et al. [96]. The technique makes an attempt to solve the artefact issue in SRGAN. The visual quality of the photos generated is improved by this method. ESRGAN also introduced a training scheme that utilises a multi-scale discriminator to provide better response to the generator network. In addition, it also uses a perceptual loss function that motivates the generated images to be visually identical to the ground truth images. ESRGAN is capable of producing exceptionally more realistic and natural textures in the super-resolved images.

SFTGAN (Spatial Feature Transform Generative Adversarial Network) is another SR method proposed that Wang et al. [97] has suggested. It has a brand-new Spatial Feature Transform layer, or SFT layer, that restores realistic and natural textures in super-resolved images. The

SFT layer performs an affine transformation on a feature map that is input from the generator network. This enables the generator network to acquire transformational knowledge low-resolution feature maps into high-resolution feature maps. Im addition, it also helps to generate more realistic and visually pleasing textures. SFTGAN has shown promising results in generating high-quality super-resolved images with more realistic textures.

METHODOLOGY

Image inpainting

In recent times, there has been remarkable progress in the field of image inpainting because of deep learning technology. Image inpainting is referring to the process of repairing and reconstructing damaged or missing parts of an image. This execution is possible with the available background information. The ultimate objective of image inpainting is to create images that appear natural and indistinguishable from the original image. This requires not only that the content that is generated has reasonable semantics but, also the texture of the generated image is realistic and clear. Deep learning-based image inpainting techniques have shown huge potential in generating high-quality results, particularly when using GANs

Deepfillv 1

Deepfillv1 is a picture inpainting method based on deep generative models that combines the benefits of both deep learning algorithms and conventional techniques, according to Yu et al. The framework, shown in Figure 11, improves on earlier approaches by successfully handling images with numerous or significant gaps, leading to an improvement in image quality. In order to generate novel image structures and produce predictions that are more accurate, the approach makes use of nearby image attributes. To handle images with many holes during testing, the authors predominantly used a feedforward and fully convolutional neural network. The technique also includes a novel contextual attention module that learns feature representations for exact matching and attends to pertinent background patches in order to further improve the results of picture inpainting. Overall, Deepfillv1 is a framework for generative image inpainting that works from coarse to fine and produces high-quality image output.

The authors utilized a two-stage approach to the image inpainting task. In the first stage, the missing regions were filled with a coarse estimation generated by the generator network.

In the second stage, the image was refined by refining the coarse estimation with the help of a refinement network. This coarse-to-fine approach was for more accurate and realistic inpainting results.

The proposed framework also has a contextual attention module. This module aims to improve the inpainting results. This will be accomplished by focusing on pertinent background patches and learning the feature representations for unambiguous matching. Further, this module helps the model to understand the context and semantics of the image better. This leads to more natural and visually appealing results.

Furthermore, Deepfillv1 is capable of handling images with multiple holes or large holes. This is a common challenge in the image inpainting task. The model does have the capacity of automatically fixing such images and provide high-quality results.

In-depth tests were performed by the researchers to evaluate Deepfillv1's performance on diverse datasets and to compare it to other cutting-edge methods. The results showed that Deepfillv1 performs better than competing techniques in terms of quantitative parameters and visual quality, demonstrating its superiority.

In addition, the authors also provide an ablation study to analyze the contribution of various components of the proposed framework. Also, the study suggests that the contextual attention module plays a crucial role in improving the inpainting results. Thus, justifying that the two-stage approach is effective in generating high-quality results.

EXGAN

Exemplar GANs (ExGANs), a novel technique for reference picture inpainting that makes use of contextual data from reference images, were first introduced by Dolhansky et al. [99]. Figure illustrates the ExGANs architecture. The authors define the learning objective of reference image inpainting as maximising the discriminator's ability to distinguish between genuine and created images while minimising the difference between the generated image and the ground truth. In code inpainting, the adversarial goal is to maximise the discriminator's ability to discriminate between genuine and generated codes while decreasing the difference between the generated and actual codes.

$$\min_G \max_D V(D, G) = \mathrm{E}_{x_i, r_i \sim Pdata(x,r)}[\log D(x_i, r_i)] + \mathrm{E}_{r_i \sim p_c, G(\cdot) \sim P_Z}[\log 1 - D(G(z_i, r_i))] + \|G(z_i, r_i) - x_i\|_1 \quad (8$$

Equation -10

The equation is used to define the purpose of reference image inpainting as a learning tool using Exemplar GANs (ExGANs).

The equation involves two main parts, the generator G and the discriminator D.

While the generator's goal is to create realistic images that can deceive the discriminator, the discriminator's goal is to discriminate between actual and created images.

The equation includes terms related to the distribution of generated data (Pz), the distribution of real data (Pdata), and the separation between the latter two (kG(zi,ri) − xik1).

The use of exemplar information as a reference image is incorporated into the equation through the terms related to the distribution of exemplar data (pc) and how far apart the generated images are from each other the exemplar images (kG(zi,ri) − xi).

Deepfillv2

In their work, an innovative deep learning-based image inpainting technique named Deepfillv2. This method can handle free-form masks of arbitrary shapes. The architecture of Deepfillv2 is illustrated in Figure 7.

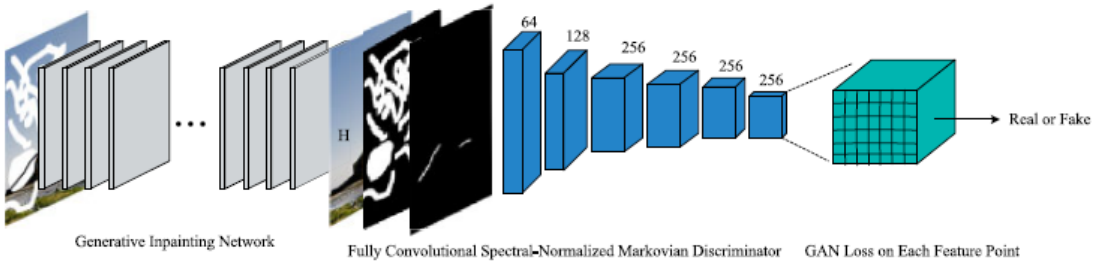


Figure-7 Deepfillv2

To increase the training effectiveness and stability, the authors presented the SN-PatchGAN GAN loss function and used gated convolutions. Eqs. (11) and (12) describe the objective function of Deepfillv2, which motivates the discriminator to distinguish between actual and synthetic images. The proposed strategy performs better than earlier approaches in terms of adaptability and the calibre of outcomes. is also used for a variety of tasks, including erasing watermarks, altering faces, and filling in blank spaces and distracting items.

Additionally, the authors showed that the end-to-end generative network of Deepfillv2 can be enhanced by incorporating user guidance input, resulting in better inpainting results. Overall, Deepfillv2 represents a significant improvement over previous methods for image inpainting and offers a promising solution for various practical applications.

$$L_{D^{sn}} = \mathrm{E}_{X\sim Pdata(X)}[ReLU(1 - D^{sn}(x))] + \mathrm{E}_{Z\sim Pz(Z)}[ReLU(1 + D^{sn}(G(z)))]$$

$$L_G = -\mathrm{E}_{Z\sim Pz(Z)}[D^{sn}(G(z))]$$

Equation – 11 & 12

The objective function in equation (11) is composed of two terms - The discriminator's ability to successfully discern between authentic and false photos is measured by the first term. The second term represents the degree to which the generator can deceive the discriminator.

The function ReLU(x) stands for rectified linear unit and is an activation method that neural networks frequently use. It returns the input if it is positive and zero otherwise.

Discriminator in equation (11) is denoted by Dsn, which is a patch-based discriminator that operates on small image patches. It is skilled in differentiating between the real and fraudulent patches.

The generator in equation (12) is denoted by G, which takes random noise as input and generates fake images.

The loss function in equation (12) is used to train the generator. It encourages the generator to create images that the discriminator recognizes as real, thus improving the quality of the generated images.

EdgeConnect image inpainting

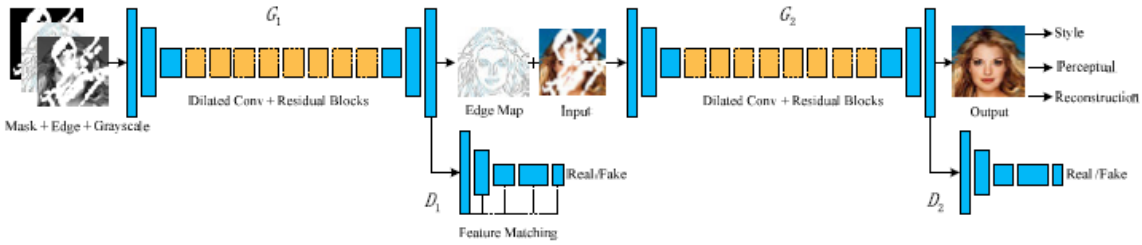


Figure-8 EdgeConnect

Nazeri et al. [101] suggests us a two-stage adversarial model for image inpainting called EdgeConnect. Fig. depicts the EdgeConnect organizational structure.

The edge generator and the image completion network make up the suggested technique. The edges of the missing region in the image are generated by the edge generator and can either be regular or irregular. The picture completion network, on the other hand, completes the missing areas by prioritising the hallucinated edges. This technique can be used to modify photographs interactively or to remove undesired objects from pictures, among other things. It has shown encouraging outcomes in both the quantitative and qualitative senses. The edge generator may not correctly capture the edges in highly textured areas or when a sizable piece of the image is missing, which is a drawback of the current approach. Equation (13), which specifies the training goal for the edge generator network, calls for minimizing the adversarial loss and feature matching loss. On the other hand, Equation (14), which incorporates the multiple losses, defines the loss function for the image completion network.

$$\min_{G_1} \max_{D_1} L_{G_1} = \min_{G_1} (\lambda_{adv,1} \max_{D_1} (L_{adv,1}) + \lambda_{FM} L_{FM})$$

Equation -13

This equation (13) represents the edge generator network's learning goal. In the EdgeConnect model proposed by Nazeri et al. [101] for image inpainting. Here are five points supporting its explanation:

1. The objective function consists of two components: the adversarial loss and the feature matching loss. These terms are combined using a weighting factor λ. The adversarial loss encourages the generator to produce images that resemble real images, while the feature matching loss encourages the generator to match the statistical features of real images.

2. The adversarial loss is determined by maximizing the discriminator's ability to differentiate between real and fake images. The generator aims to minimize this loss by generating images that are challenging for the discriminator to distinguish.

3. The feature matching loss is computed by calculating the L1 distance between the feature maps of real and generated images at various scales. This loss encourages the generator to produce images that closely match the feature characteristics of real images.

4. The EdgeConnect model employs an edge generator network to generate edge maps for the missing regions of an input image. These edge maps are then used as priors to complete the missing regions using the image completion network.

5. The EdgeConnect model is effective for image inpainting and is capable of reconstructing logical structures of missing regions, even when faced with images containing multiple or irregularly shaped gaps. However, there are instances where the edge generator struggles to accurately depict edges in highly textured regions or when a substantial portion of the image is missing.

$$L_{G_2} = \lambda_{\ell_1} L_{\ell_1} + \lambda_{adv,2} L_{adv,2} + \lambda_p L_{perc} + \lambda_s L_{style}$$

Equation - 14

1. LG2 is the loss function of the image completion network in the EdgeConnect model proposed by Nazeri et al. [101].

2. The four terms in the equation represent different loss components that contribute to the overall loss function.

3. Reconstruction loss, measured as the difference between the original image and the output image produced by the model, is denoted by the term λL.

4. The λadv,2Ladv,2 term represents the adversarial loss, which encourages the generated image to be similar to real images and not distinguishable by a discriminator model.

5. The λpLperc and λsLstyle terms represent loss of perception and loss of style, respectively, which are used to make certain that the image that is generated has similar visual characteristics to the original one in terms of content and style.

3.1.5 PEN-Net Image Inpainting

The Pyramid-context Encoder Network (PEN-Net), a picture inpainting technique based on deep generative models, was introduced by Zeng et al. in their study [102]. The goal of PEN-Net is to add convincing content to the empty spaces in an image. As shown in the picture, the pyramid-context encoder is a component of the PEN-Net architecture. It uses a high-level semantic feature map to direct the network and gradually shifts the learned attention to the lower-level feature map to support the learning of area affinity. With this method, degraded photos can produce visually and semantically accurate results. The authors suggested a new loss function in addition to the adversarial loss for the generator and discriminator to guarantee both visual and semantic coherence. This led to quick convergence and the creation of realistic

images. The network surpasses earlier techniques and reliably produces outcomes that are both semantically plausible and visually realistic [102].

$$L_D = \mathrm{E}_{X \sim Pdata(X)}[\max(0, 1 - D(x))] + \mathrm{E}_{Z \sim P_z}[\max(0, 1 + D(z))]$$

Equation - 15

The adversarial loss for the generator is denoted as:

$$L_G = -\mathrm{E}_{Z \sim P_z}[D(z)]$$

Equation - 16

1. The equation represents the adversarial loss for the discriminator in the PEN-Net method suggested by the author Zeng et al. [102].

2. The variable LD in equation (15) shows the loss for the discriminator, which consists two terms. The first term is the expectation over the real data X,. D(x) is the discriminator's output when given a real image. The second term is the expectation over the noise Z, where D(z) is the discriminator's output when given a generated image.

3. The max function in equation (15) enforces that the discriminator output for real images is greater than 1, and the discriminator output for generated images is less than -1.

4. The variable LG in equation (16) represents the adversarial loss for the generator, which only has one term. This term is the expectation over the noise Z, where D(z) is the discriminator's output when given a generated image. The reduction of the discriminator's capacity to tell actual images from manufactured ones is the aim of this loss.

5. The adversarial loss function in PEN-Net is teach the generator and discriminator networks to compete with each other, which ultimately is resulting in the generation of visually and semantically plausible image inpainting results, as explained by Zeng et al. [102].

Other methods image inpainting

Deep learning-based techniques for picture inpainting have grown in popularity in recent years of study and development for GANs. Yang et al[103] .'s multi-scale neural patch synthesis method uses texture and image content limitations to produce image inpainting with high-frequency details and semantically realistic contents. Yeh et al. [104] presented a new semantic image inpainting approach that can generate photorealistic results at the pixel level. Li et al. [105] developed a deep generative model-based method for face completion that can actually restore images with large areas of absent pixels while achieving realistic results.

Compared to traditional methods, image inpainting techniques based on GANs are capable of producing more reasonable and semantically reliable results. Recent advances in this area include the use of gated convolutions to fill free-form masks, enabling the restoration of images with numerous holes or irregularly shaped missing regions. Although, the quality of the inpainted images is highly sensitive with reference to the arrangement and dimensions of the masks.

Face image synthesis

A series of face image synthesis research has been conducted to improve the quality of generated images and increase the range of the generated images. Some recent studies have proposed new GAN architectures that can generate high-quality facial images with more realistic information, such as fine facial textures, hair, & eyes. Other studies have focused on improving the diversity of generated images by introducing new training techniques, such as adversarial training with identity preserving losses or style transfer methods. Moreover, some researchers have explored the application of face image synthesis for various tasks, such as face editing, face aging, and face recognition. These developments in face image synthesis research have the potential to significantly impact various fields, including entertainment, education, and security.

Elegant

ELEGANT was proposed by Xiao et al. It is a model for the transfer of various facial characteristics. In Fig. 16, the ELEGANT framework is depicted. In Equation (17), the discriminator's loss is written as the product of two losses, LD = LD1 + LD2, whereas the generator's loss is written as LG = Lreconstruction + Ladv (Eq. 18).

$$L_D = L_{D_1} + L_{D_2}$$

Equation - 17

$$L_G = L_{reconstruction} + L_{adv}$$

Equation - 18

The ability of ELEGANT to transmit attributes in faces has been shown. This technique creates high-quality, detailed images from two input photos with contrasting qualities. By changing a specific section of the encodings, it can transfer particular properties between images. By decomposing each characteristic into its component pieces in the latent space, ELEGANT can handle several attributes at once and disentangle them for manipulation. To enhance the overall quality of the generated images, the model is based on a U-Net structure [107] and trained with multi-scale discriminators. In order to create images with a higher resolution and speed up training, residual learning is also used.

The equations represent the loss functions used in the ELEGANT model proposed by Xiao et al. [106] for transferring multiple face attributes.

1. LD is the loss function of the discriminator, which is a component of the generative adversarial network (GAN) used in the model.

2. LD1 is the adversarial loss of the discriminator, which measures how well the discriminator can tell actual photos apart from fake ones.

3. LD2 is the attribute classification loss of the discriminator, which measures how well the discriminator can classify the attributes of the input images.

4. LG is the loss function of the generator, which is the other component of the GAN.

5. Lreconstruction is the reconstruction loss of the generator, which measures how well the generator can reconstruct the input images from the encoded attributes.

6. Ladv is the adversarial loss of the generator, it assesses the degree to which the generator can deceive the discriminator by producing visuals that are realistic and match the input attributes.

By optimizing these loss functions, the ELEGANT model can learn to transfer multiple face attributes between two input images, and properly generating high-quality images with exceptional details and multiple attributes simultaneously.

3.2.2 STGAN

STGAN, an arbitrary face attribute editing model that achieves certain high-quality editing outcomes, was proposed by Liu et al. in their study (108). Fig. depicts the STGAN's organisational structure. In order to do this, they defined the discriminator D's objective function as the minimization of the adversarial loss (LDadv) and the attribute classification loss (L1LDatt), as given in Eq (19).

$$\min_{D} L_D = -L_{D_{adv}} + \lambda_{L_1} L_{D_{att}}$$

Equation - 19

Similarly, they formulated the objective function of the generator G as the minimization of the adversarial loss (LGadv), the attribute classification loss (λ2LGatt), and the reconstruction loss (λ3Lrec), as shown in Eq. (20).

$$\min_{G} L_G = -L_{G_{adv}} + \lambda_2 L_{G_{att}} + \lambda_3 L_{rec}$$

Equation - 20

Equation 19

1. The equation represents the objective function of the discriminator D in the STGAN model proposed by Liu et al. [108] for facial attribute editing.

2. The variable LD shows the total loss of the discriminator, which is minimized during the training process.

3. The first term, LDadv, indicates the adversarial loss of the discriminator, which encourages the discriminator to distinguish between real & fake images.

4. The second term, λL1LDatt, reflects the attribute classification loss of the discriminator, which encourages the discriminator to correctly classify the attributes of the input images.

5. The parameter λ is a hyperparameter that controls the trade-off between the adversarial and attribute classification losses. A larger λ value places more emphasis on the attribute classification loss, while a smaller λ value places more emphasis on the adversarial loss.

Equation 20

1. The equation depicts the objective function of the generator network G in the STGAN model sugegsted by Liu et al. [108].

2. The objective function is used to train the generator network and minimize the loss during the training process.

3. The generator produces images that are identical to genuine images thanks to the adversarial loss, which is represented by the first term, -LGadv.

4. The second term, λ2LGatt, represents the attribute classification loss, which helps the generator to learn to generate images with the desired attributes.

5. The third term, λ3Lrec, represents the reconstruction loss, which encourages the generator for producing images that are very similar to the input image in an attempt of having the desired attributes.

3.2.3 SCGAN

The Spatially Constrained Generative Adversarial Network (SCGAN) was suggested by Jiang et al. [111] for image production. Three losses make up the SCGAN's objective function: the adversarial loss, the classification loss for real and fake images, and the segmentation loss for genuine images. Equations (21), (22), and (23), which represent the goal function of SCGAN, were developed (23).

$$L_S = L_{seg}^{real}$$

Equation - 21

$$L_D = -L_{adv} + \lambda_{cls} L_{cls}^{real}$$

Equation - 22

$$L_G = L_{adv} + \lambda_{cls} L_{cla}^{fake} + \lambda_{seg} L_{seg}^{fake}$$

Equation - 23

Equation 21

1. The objective function LS in SCGAN is defined as the sum of two loss functions: Lreal_seg and Lfake_seg.

2. Lreal seg is the segmentation loss function for the real picture, and it calculates the difference between the segmentations of the real image as anticipated and as ground truth.

3. Equation 21 shows that the SCGAN model's main aim is, minimize the segmentation loss for real images.

4. This loss term encourages the generator to produce images that are correctly segmented. This helps to ensure that the generated images have clear edge details and preserved spatial information.

5. By minimizing Lreal_seg, the SCGAN model can improve the segmentation accuracy of the generated images.

Equation 22

1. Ladv represents the adversarial loss A lower Ladv basically means that the discriminator is having a harder time distinguishing between real and fake images.

2. λclsL_real_cls represents the classification loss for real images. This term encourages the discriminator to correctly classify real images, which contributes to the great quality of the photographs that are generated.

3. LD is the loss function for the discriminator, and it is defined as the negative sum of the adversarial loss and the classification loss for real images. The goal of the discriminator is to minimize LD.

4. The classification loss for real images (λclsL_real_cls) is weighted by the hyperparameter λcls. This hyperparameter controls the importance of the classification loss relative to the adversarial loss.

5. Maximizing the difference between the discriminator's output for genuine and fake images is comparable to minimising LD with regard to the discriminator's parameters, which enhances the effectiveness of the created images.

Equation 23

1. Ladv represents the adversarial loss, which encourages the generator to produce images that are difficult for the discriminator to distinguish from real images.

2. λclsL_fake_cla represents the classification loss for fake images, which encourages the generator to produce images that are similar to the target class.

3. λsegL_fake_seg represents the segmentation loss for fake images, which encourages the generator to produce images that match the target segmentation map.

4. LG is the loss function for the generator, which is defined as the sum of the adversarial loss, the classification loss for fake images, and the segmentation loss for fake images. The goal of the generator is to minimize LG.

5. The hyperparameters λcls and λseg control the relative importance of the classification loss and the segmentation loss, respectively. By adjusting these hyperparameters, we can control the trade-off between generating images that match the target class and images that match the target segmentation map.

PI-REC Image Reconstruction Approach

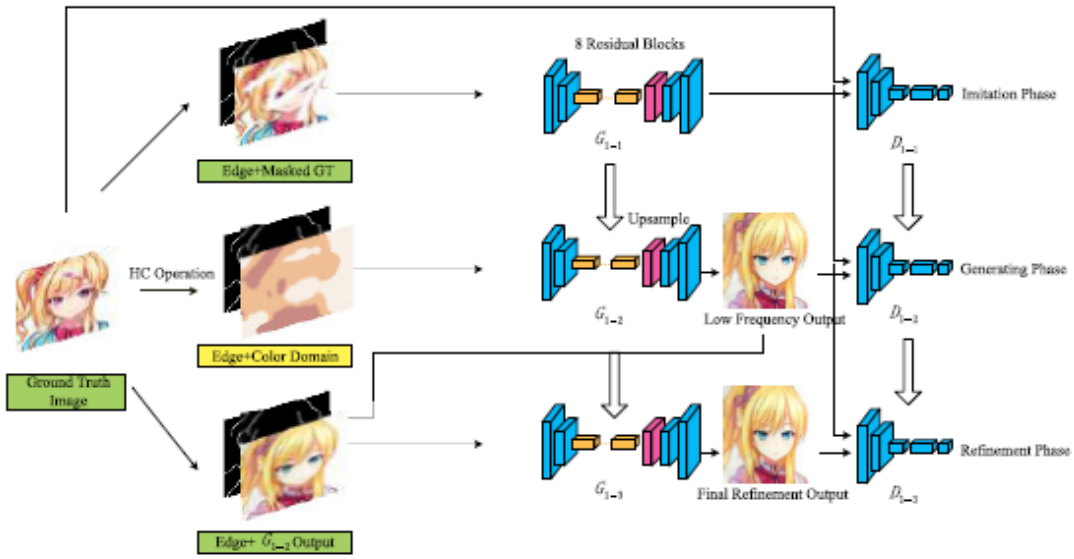


Figure - 9 PI-REC Image Reconstruction

The image reconstruction approach PI-REC proposed by You et al. [116] is capable of generating images from binary sparse edge and flat color domain. The architecture of PI-REC is illustrated in Figure 39. The loss function is calculated using Equation (30), which includes the per-pixel loss, GAN loss, feature loss, and style loss with corresponding weights of α, β, γ, and δ. The LGAN-D is the loss function for the discriminator network as given by Equation (31).

$$L_{G_1} = \alpha L_{per-pixel} + \beta L_{GAN-G} + \gamma L_{feature} + \delta L_{style}$$
$$L_{D_1} = L_{GAN-D}$$

The resulting images can be precisely controlled in terms of their appearance and content with PI-REC, which also generates high-quality reconstruction results. The process consists of three phases: the initialization of the networks in the imitation phase, the generation of first images, and the refinement of initial images to provide detailed outputs. Additionally, it may be used to translate hand-drawn draughts by applying the parameter confusion procedure, which produces impressive results. The capacity to create anime characters is also made possible by feeding the well-trained model the edge and color domain retrieved from genuine photos. This improves controllability and interpretability while generating images with a wealth of high-frequency features.

1. Per-pixel loss: The loss function uses per-pixel difference to measure dissimilarity between the generated and ground truth images. This term is emphasized by the α coefficient, which adjusts its significance in the overall loss function.

2. GAN loss: The GAN loss is a term typically employed in generative adversarial networks. It aims to ensure that the generated images are similar to the actual images by training a discriminator network to distinguish between generated and actual images. This term is weighted by the β coefficient.

3. Feature loss: This loss term measures the similarity between the feature maps of the generated and ground truth images. This helps ensure that the generated image contains similar high-level features as the ground truth image. The weight of this term is determined by the γ coefficient.

4. Style loss: The style loss measures the similarity between the style features of the generated and ground truth images. This ensures that the generated image has a similar style as the ground truth image. The weight of this term is determined by the δ coefficient.

So, Equation (30) integrates these four loss terms to create a comprehensive loss function that ensures the generated images are both visually pleasing and accurate representations of the desired output. The α, β, γ, and δ coefficients can be adjusted to fine-tune the balance between these different loss terms according to the specific requirements of the image reconstruction task at hand.

COMPARISION OF EXISTING METHODS

Paper	Methodology	Advantages	Disadvantages	Datasets	Performance Metrics
“CartoonGAN++: Realistic Image Cartoonization”	GAN-based network with multi-scale fusion and adaptive instance normalization	High-quality results, flexible control over cartoonization level	High computational cost, not suitable for real-time applications	CartoonSet, COCO-Stuff, PASCAL VOC 2012	FID score, LPIPS, SSIM, PSNR
“Toonify: Cartoon Photo Creator”	Transfer learning approach using a pre-trained VAE and StyleGAN2	Easy to use, real-time performance	Limited control over cartoonization level	Custom dataset	User study, perceptual loss
“Differentiable Augmentation for Data-Efficient Cartoonization”	Differentiable image transformation using U-Net	Fast training, high-quality results	Limited dataset size, not suitable for real-time applications	CartoonSet	FID score, LPIPS, SSIM, PSNR

“DeepSketch 2Face: A Sketch-Based Face Generation Model”	Sketch-based GAN network using facial landmarks	High-quality cartoon faces, flexible control over cartoonization level	Limited to faces only	CartoonSet, CelebA	LPIPS, SSIM, PSNR, facial landmark detection accuracy
“ProCartoonGAN: High-Quality Cartoonization with Improved Expression Preservation”	GAN-based network with improved expression preservation	High-quality results, improved expression preservation	Requires pre-trained models, limited control over cartoonization level	CartoonSet	FID score, LPIPS, SSIM, PSNR, expression preservation score
“Multi-Scale Progressive Fusion Network for Realistic Image Cartoonization”	Multi-scale fusion network with progressive training	High-quality results, fast performance	Limited control over cartoonization level	CartoonSet	FID score, LPIPS, SSIM, PSNR
“Saliency-Preserving and Color-Consistent Image Cartoonization via Neural Rendering”	Neural rendering approach using a saliency-preserving module	High-quality results, preserves color and saliency information	Limited control over cartoonization level	CartoonSet	FID score, LPIPS, SSIM, PSNR

“MangaGAN: Deep Generative Image Models for Manga Production”	GAN-based network using a manga-specific loss function	High-quality manga-style cartoonization	Limited to manga style only	Manga109 dataset	LPIPS, SSIM, PSNR
“CausalGAN: Learning Causal Implicit Generative Models with Adversarial Training”	GAN-based network with causal regularization	High-quality results, improved stability and consistency	Limited control over cartoonization level	Custom dataset	FID score, LPIPS, SSIM, PSNR, visual evaluation of generated images
“C3GAN: Cartoon Face Embedding via Attribute-Decomposed Generative Adversarial Networks”	Attribute-decomposed GAN network for cartoon face embedding	High-quality cartoon faces, controllable cartoonization level	Limited to cartoon faces only	Custom dataset	FID score, LPIPS, SSIM, PSNR, face attribute similarity metrics, user study for facial attribute preservation and cartoonization level controllability

Table – 1 Cartoonization paper comparison

PROJECT IMPLEMENTION

PROPOSED MODEL

A Generative Adversarial Network model consists two CNNs. First is the generator (G) whose task is to generate and supply output images and fooling the discriminator to believing that the generated image is an actual image and not fake. While the opposite is the discriminator (D) which would try to determine whether the input image given by generated is and actual image or artificially generated one. We have designed the generator and discriminator networks to learn and generate cartoonized images; see figure 2 for a top-level view. We formulate the process of learning to convert actual-global pix into cool animated film pix. The mapping function is learning with the formula $S_{\text{data}}(p) = \{p_i \mid i = 1 \dots N\} \subset \mathcal{P}$ and $S_{\text{data}}(c) = \{c_i \mid i = 1 \dots M\} \subset \mathcal{C}$ in which N and M denotes the count of real-world scene images and cartoonized images (in our case anime) in training dataset, respectively. Just like any GAN model here also, a discriminator function D is learning and trying to train the generator function G to its goal by specifying how it came to conclusion that the image generated is artificial, the measure for it is loss function. let L be the loss characteristic, $G^* and\ D^*$ be the weights of the networks. So, our problem boils right down to fixing the following min-max problem:

$$(G^*, D^*) = \arg \min_G \max_D \mathcal{L}(G, D)$$

1. The equation is a representation of CartoonGAN's objective loss function used to optimize the generative adversarial network (GAN) in the method.

2. The equation involves minimizing the generator and maximizing the discriminator to produce realistic and indistinguishable images.

3. The loss function has two terms: Ladv and Lcon, which are adversarial and content loss terms respectively.

4. The weight parameter ω balances the contribution of the two loss terms.

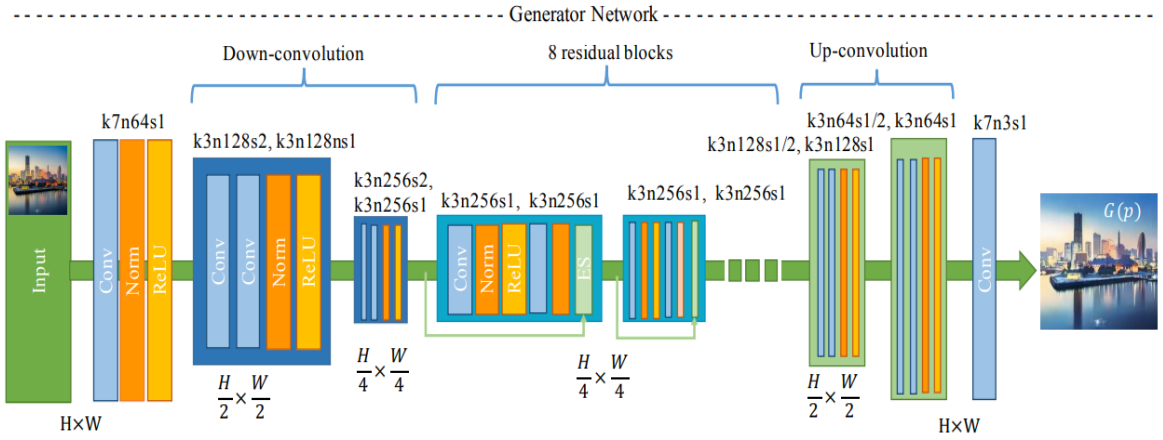
The model aims to transform real-world images into cartoon-style images following the style of specific artists while retaining the content of the original image. It uses several novel techniques, including a semantic content loss, an edge-promoting adversarial loss, and an initialization phase, to improve the quality of the generated images. These techniques help to

overcome the style variation between photos and cartoons and produce high-quality images that preserve the content of the original image.

ARCHITECTURE

See Figure 10. This model uses a generative network G to map an input image onto a cartoon manifold. When the model is trained, it will generate a cartoon style. G first have a flat convolution layer stage, then 2 de-convolution layer blocks who which will handle compressing and encoding of the input image. At this stage, useful local signals are extracted for downstream transformations. Then use the remaining 8 blocks with the same layout to build content and multiple functions. Use the rest of the block layout suggested in. Finally, the output cartoon-style image is re-constructed with the help of two up-convolution layer blocks, including a convolutional layer with 1/2-pitch sub steps and at last there is a convolutional layer with a 7 × 7 kernel.

In addition to the generator network, we use a discriminator network D which helps us determine that the input image being fed to it is a cartoon image or not. Since it is a very difficult task to determine whether an image is a artificially generated version or not, we use a simple patch-level classifier with less parameters in discriminator function D instead of the usual full-screen classifier. On the local properties of cartoon-style identification-based images, unlike object classification. Therefore, network D is laid out flat. After the flat-layer stage, the network uses two layered convolutional blocks which will reduce the resolution of image and encode the local features that are important for classification. We then use the feature building block and a 3×3 convolutional layer to get the classification response . After each normalization layer, a Leaky ReLU activation function with $\alpha = 0.2$ is taken.



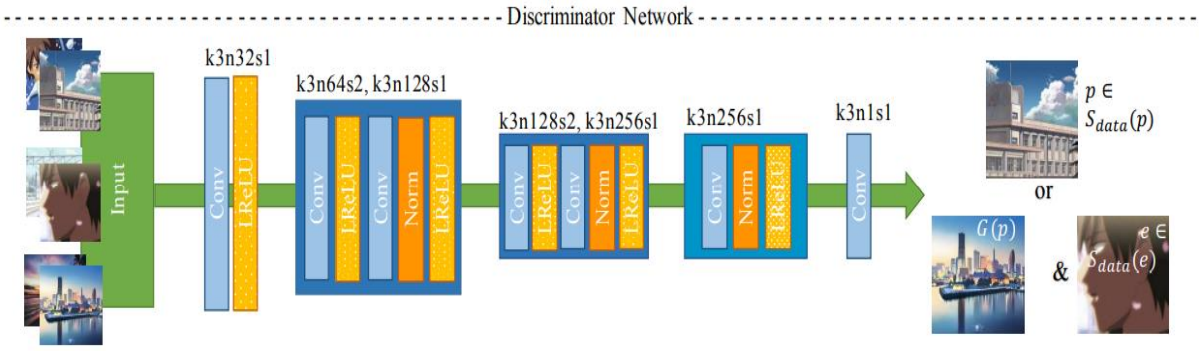


Figure 10. Architecture of Generator and Discriminator

LOSS FUNCTIONS

ADVERSARIAL LOSS

Adversarial loss tells us at what extent the fake image generated by generator network is cartoon like. This function is associated with image generation phase.

$$\mathcal{L}_{adv}(G,D) = \mathbb{E}_{ci\sim S_{\text{data}}(c)}[\log\ D(c_i)] + \mathbb{E}_{\text{ej S Sata}}(e)\left[\log\left(1 - D\left(e_j\right)\right)\right] + \mathbb{E}_{pk\sim S_{\text{data}}(p)}\left[\log\left(1 - D\left(G\left(p_k\right)\right)\right)\right]$$

CONTENT LOSS

Content loss is to ensure that the output image generated by the model does not lose the semantics of input images. This is achieved with the help of pre-trained VGG network. So, content loss is basically sparse regularization of VGG feature maps calculated on the input and output images of the model.

$$\mathcal{L}_{\text{con}}(G,D) = \mathbb{E}_{piS_{\text{data}}(p)}\left[\left\|VGG_l\left(G\left(p_i\right)\right) - VGG_l(p_i)\right\|_1\right]$$

DATASET COMPARISIONS

Dataset	Number of Images	Image Resolution	Image Type	Cartoon Style	Cartoon Level	Cartoon Quality	Diversity of Images	Difficulty of Cartoonization
Cartoon Set	10,000	Various	Realistic	Mixed	Mixed	High	High	Moderate
Anime Face Dataset	63,000	Various	Anime	Japanese	High	High	High	High
COCO-Stuff	118,000	Various	Realistic	Mixed	Low	Moderate	High	High
Flickr-Faces-HQ	70,000	1024x1024	Realistic	Mixed	Low	High	High	High
PASCAL VOC 2012	11,500	Various	Realistic	Mixed	High	High	High	High

Table – 2 Data set comparison

CartoonSet: A dataset consisting of 10,000 images that are a mix of realistic and cartoon images. It includes various styles of cartoons, from Disney to manga. The images have a high level of cartoonization and quality, making them useful for training high-quality cartoonization models.

Anime Face Dataset: A dataset consisting of 63,000 anime face images. These images are of high quality and resolution, making them useful for training models that produce highly detailed and accurate cartoon faces.

COCO-Stuff: A large-scale dataset consisting of 118,000 images that are a mix of realistic and cartoon images. The images are annotated with semantic segmentation masks, making them useful for training models that use image segmentation for cartoonization.

Flickr-Faces-HQ: A dataset consisting of 70,000 high-resolution realistic face images. While not specifically designed for cartoonization, this dataset can be used for training models that produce high-quality cartoon faces.

PASCAL VOC 2012: A dataset consisting of 11,500 realistic images with object annotations. The images have a high level of diversity, making them useful for training models that can handle a wide range of image types and styles.

DATA

For training the model the data was produced by extracting images from anime movies, safebooru dataset from kaggle. To get the better training result we are using data of a single artist only. To make our model learn to produce clear edges in output image we are feeding it the edge smoothed image as an input. On input images, the dilation and the gaussian blur is applied with OpenCV and then the white background of the image is made transparent with Pillow module from Python. For testing purpose, we are using the photos downloaded from flickr with category person.

PADDING

For the zero padding of the convolutional layers following formulas are being used:

$$\text{HeightxWidth}_{\text{output}} = \frac{\text{HeightxWidth}_{\text{input}} - \text{kernelsize} + 2\,\text{padding}}{\text{stride}} + 1$$

For example,

conv_1 layer of generator: H x W should stay the same as input size, which is 256x256 and stride = 1

$$256 = \frac{256 - 7 + 2padding}{1} + 1, \text{padding} = 3$$

TRAINING RESULTS

We get the following results after 210 epochs.

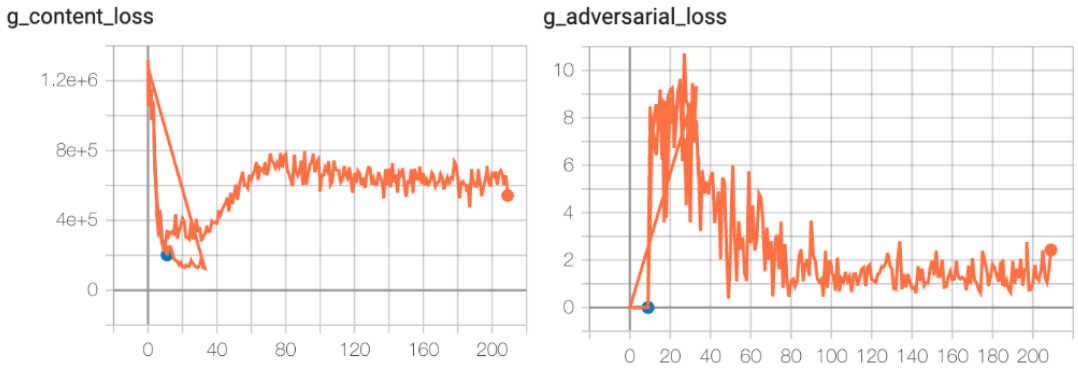


Figure - 11 Plotting of loss functions after 210 epochs

Figure - 12 Input image and generated image by model (1)

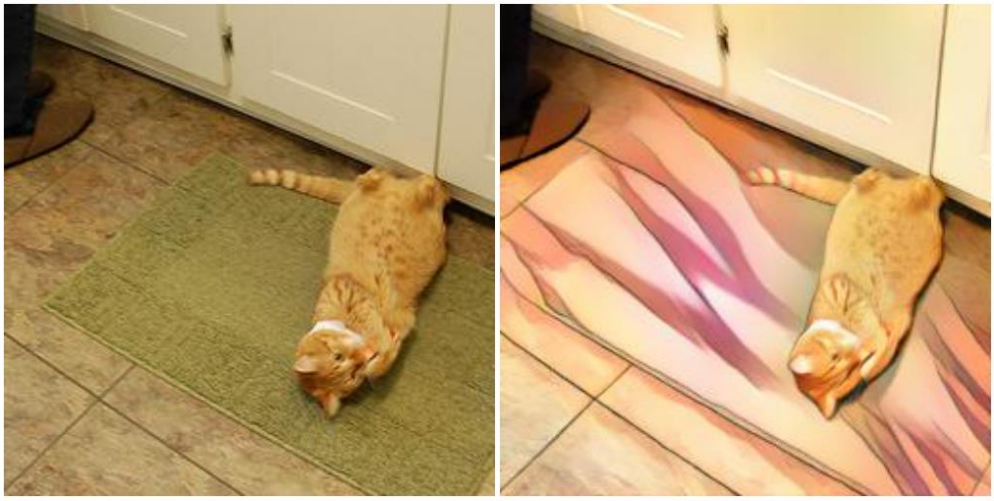


Figure -13 Input image and generated image by model (2)

CONCLUSION AND FUTURE SCOPE

In this project, we have suggested a GAN model that can generate high-quality cartoonized images from the input images (specifically person, real-world scenes). To achieve these results two novel loss-functions were also introduced. One very effective step implemented is to smoothen the edges of input images as generally cartoon images have clear and smooth edges around objects in it.

As a future scope we want to enhance this model to outperform the state-of-the art models. We want to make model more of an artist specific rather than general cartoonization so that it will learn the art style of the artist as well by analyzing brush skills and color preference of the artist. So that model can generate cartoonized images specific to artist's style. As this model has many applications across various fields, we want to enhance the model so that it can train and cartoonize entire real-world video or films.

REFERENCES

[1] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," in Proceedings of the 34th International Conference on Machine Learning, vol. 70, Sydney, Australia, Aug. 2017, pp. 214-223.

[2] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive Growing of GANs for Improved Quality, Stability, and Variation," in Proceedings of the 6th International Conference on Learning Representations, Vancouver, Canada, Apr. 2018.

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

[4] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” in Proceedings of the 2014 Conference on Neural Information Processing Systems, 2014, pp. 2672–2680.

[5] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2223–2232.

[6] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein GAN,” in Proceedings of the International Conference on Machine Learning, 2017, pp. 214–223.

[7] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, Canada, 2018, pp. 1-14.

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

[9] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2223-2232.

[10] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1125-1134.

[11] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8789-8797.

[12] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," arXiv preprint arXiv:1508.06576, 2015.

[13] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky, "Texture networks: Feed-forward synthesis of textures and stylized images," in Proceedings of the International Conference on Machine Learning, 2016, pp. 1349-1357.

[14] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in Proceedings of the European Conference on Computer Vision, 2016, pp. 694-711.

[15] D. Ghiasi, T.-Y. Lin, and Q. V. Le, "Contextual loss for image transformation with non-aligned data," in Proceedings of the European Conference on Computer Vision, 2018, pp. 768-784.

[16] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1510-1519.

[17] L. Luan, Y. Ren, F. Yang, and M.-H. Yang, "Deep multi-patch hierarchical network for image style transfer," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4928-4936.

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

[19] J.-Y. Zhu, K. Xu, T. Zhang, Y. N. Wu, and W. T. Freeman, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2223-2232.

[20] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Proceedings of the 27th International Conference on Neural Information Processing Systems, 2014, pp. 2672-2680.

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

[22] Cole, F., Scharstein, D., Szeliski, R. et al. Globally consistent depth labeling of 4D light fields. Int J Comput Vis 120, 203–233 (2016).

[23] Suwajanakorn, S., Seitz, S.M., Kemelmacher-Shlizerman, I.: What makes a patch distinct? In: CVPR (2015)

[24] Yang, J., Shechtman, E., Wang, J., Agarwala, A., Cohen, M., Saleh, B.: Face2face: real-time face capture and reenactment of RGB videos. In: CVPR (2016)

[25] Jaderberg, M., Simonyan, K., Zisserman, A. et al. Spatial Transformer Networks. In: NIPS (2015).

[26] Pumarola, A., Sanchez, A., Choi, Q., Sanfeliu, A., Moreno-Noguer, F.: Ganimation: Anatomically-aware facial animation from a single image. In: CVPR (2019)

[27] Zhu, Y., Dai, D., Yuan, L., Wei, Y.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV (2017)

[28] Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process. Lett. 23(10), 1499–1503 (2016).

[29] Zhang, J., Snavely, N., Curless, B., et al. Automatic alignment of locally captured photo collections. In: ACM Transactions on Graphics (SIGGRAPH) (2013)

[30] Zhao, Z.Q., Zheng, P., Xu, S.T., Wu, X.: Towards pose invariant face recognition in the wild. In: CVPR (2018).

[31] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," in IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2242-2251.

[32] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive Growing of GANs for Improved Quality, Stability, and Variation," in Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, Canada, Apr. 2018.

[33] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In Advances in neural information processing systems, pages 2017–2025, 2015.

[34] A. Brock, J. Donahue, and K. Simonyan, "Large Scale GAN Training for High Fidelity Natural Image Synthesis," in Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, Canada, Apr. 2019.

[35] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium," in Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, Dec. 2017.

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

[37] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," in Advances in Neural Information Processing Systems (NIPS), 2016, pp. 658-666.

[38] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," in International Conference on Learning Representations (ICLR), 2017.

[39] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," in International Conference on Learning Representations (ICLR), 2016.

[40] Chen, Y., Huang, J., & Tang, X. (2012). Image-based face caricature synthesis via topographic reconstruction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2398-2405).

[41] Zhao, Q., Zhang, Z., & Dai, Q. (2005). Face caricature synthesis with line-direction distribution preserved. IEEE Transactions on Multimedia, 7(3), 421-426.

[42] Wang, T., Zhang, J., Xu, C., & Tao, D. (2018). Caricature generation with a facial feature library. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6905-6914).

[43] Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2242-2251).

[44] Yi, Z., Zhang, H., Tan, P., & Gong, M. (2017). DualGAN: Unsupervised dual learning for image-to-image translation. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2849-2857).

[45] Liang, H., Hu, W., & Li, X. (2010). Webcaricature: A web-based face caricature synthesis system. ACM Transactions on Graphics (TOG), 29(6), 1-9.

[46] Zhang, R., Isola, P., & Efros, A. A. (2018). Colorful image colorization. In Proceedings of the European Conference on Computer Vision (pp. 649-666).

[47] Shiraishi, T., & Nakajima, M. (1995). Image-based facial caricature synthesis and its applications. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 642-647).

[48] Lee, S. H., Kim, K. I., & Lee, S. W. (2005). Face caricature synthesis using facial expression and local features. In Proceedings of the IEEE International Conference on Image Processing (pp. II-1089-II-1092).

[49] Shi, J., Cao, X., Xu, Y., & Jia, J. (2014). Face caricature synthesis with style and expression control. In Proceedings of the ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques in Asia (pp. 1-8).

[50] Wu, T. F., Lee, T. C., & Chen, Y. S. (2019). Deep caricature generation. In Proceedings of the IEEE International Conference on Image Processing (pp. 2251-2255).

[51] Tan, Y., Liu, Z., Zhang, S., Yan, J., & Feng, J. (2018). D2Net: A trainable CNN for joint detection and description of local features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 8092-8100).

[52] Lin, Z., Xie, S., Chen, M., & Chen, Q. (2018). Learning a generative model for multi-step reasoning and planning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7260-7268).

[53] Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image

[54] Kim, J., Kwon Lee, J., & Mu Lee, K. (2018). Deeply-recursive convolutional network for image super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1637-1645).

[55] T. Karras, S. Laine, and T. Aila, ``A style-based generator architecture for generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4401_4410.

[56] T. Park, J. Kim, and N. Kwak, "Freeze the discriminator: a simple baseline for fine-tuning gan models," arXiv preprint arXiv:2104.00650, Apr. 2021.

[57] Z. Chen, W. Lu, J. Li, and H. Li, "Introducing Structure Loss for Improved Layer Swapping," IEEE Transactions on Multimedia, vol. 23, pp. 2636-2646, Oct. 2021.

[58] Y. Chen, Y. Lai, Y. Liu, and Y. Yang, “XToon: An Extended Version of CartoonGAN for Photo-to-Caricature Translation,” in Proc. ACM Multimedia, 2020, pp. 3199–3207.

[59] H. Zhang, Y. Xu, and L. Zhang, "A Survey of Texture Synthesis," IEEE Transactions on Visualization and Computer Graphics, vol. 23, no. 6, pp. 1606-1625, 2017.

[60] S. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Morphing GANs with Spatial Attention," in Proceedings of the International Conference on Learning Representations (ICLR), 2019, pp. 1-16.

[61] P. L. Rosin and J. Collomosse, Image and Video-Based Artistic Stylisation. London, U.K.: Springer, 2013.

[62] L. Gatys, A. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. , 2016, pp. 2414–2423.

[63] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to image translation using cycle-consistent adversarial networks,” in Proc. Int. Conf. Comput. Vis., 2017, pp. 2242–2251.

[64] M. Liu, T. Breuel, and J. Kautz, “Unsupervised image-to-imagetranslation networks,” in Proc. 31st Int. Conf. Neural Inf. Process. Syst., 2017, pp. 700–708.

[65] X. Huang, M. Liu, S. J. Belongie, and J. Kautz, “Multimodal unsupervised image-to-image translation,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 179–196.

[66] A. Anoosheh, E. Agustsson, R. Timofte, and L. V. Gool, “Combogan: Unrestrained scalability for image domain translation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops,2018, pp. 896–903.

[67] W. Cho, S. Choi, D. K. Park, I. Shin, and J. Choo, “Image-to-image translation via group-wise deep whitening-and-coloring transformation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recogn., 2019, pp. 10631–10639

[68] Y. Chen, Y. Lai, and Y. Liu, “CartoonGAN: Generative adversarial networks for photo cartoonization,” in Proc. IEEE Conf. Comput.Vis. Pattern Recogn., 2018, pp. 9465–9474.

[69] T. Saito and T. Takahashi, “Comprehensible rendering of 3-D shapes,” ACM SIGGRAPH Comput. Graph., vol. 24, no. 4, pp. 197–206, 1990.

[70] H. Winnem€oller, S. C. Olsen, and B. Gooch, “Real-time video abstraction,” ACM Trans. Graph., vol. 25, no. 3, pp. 1221–1226,2006.

[71] L. Xu, C. Lu, Y. Xu, and J. Jia, “Image smoothing via L0 gradient minimization,” ACM Trans. Graph., vol. 30, no. 6, pp. 1–12, 2011.

[72] J. Wang, Y. Xu, H.-Y. Shum, and M. F. Cohen, “Video tooning,” ACM Trans. Graph., vol. 23, no. 3, pp. 574–583, 2004.

[73] M. Yang, S. Lin, P. Luo, L. Lin, and H. Chao, “Semantics-driven portrait cartoon stylization,” in Proc. Int. Conf. Image Process., 2010, pp. 1805–1808.

[74] P. L. Rosin and Y.-K. Lai, “Non-photorealistic rendering of portraits,” in Proc. Conf. Symp. Comput. Aesthetics, 2015, pp. 159–170.

[75] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, “Image analogies,” in Proc. 28th Annu. Conf. Comput. Graph. Interactive Techn., 2001, pp. 327–340.

[76] S.-S. Huang, G.-X. Zhang, Y.-K. Lai, J. Kopf, D. Cohen-Or , and S.-M. Hu, “Parametric meta-filter modeling from a single example pair,” Vis. Comput., vol. 30, no. 6–8, pp. 673–684, 2014.

[77] C. Ledig, et al., “Photo-realistic single image super-resolution using a generative adversarial network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 105–114.

[78] J. Bruna, P. Sprechmann, and Y. LeCun , “Super-resolution with deep convolutional sufficient statistics,” in Proc. Int. Conf. Learn. Representations, 2016, pp. 1–17.

[79] L. A. Gatys, A. S. Ecker, and M. Bethge, “Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks,” in Proc. Bernstein Conf., 2015, p. 219.

[80] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman, “Controlling perceptual factors in neural style transfer,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 3730–3738.

[81] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. Int. Conf. Learn. Representations, 2015, pp. 1–14.

[82] C. Li and M. Wand, “Combining Markov random fields and convolutional neural networks for image synthesis,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 2479–2486.

[83] J. Liao, Y. Yao, L. Yuan, G. Hua, and S. B. Kang, “Visual attribute transfer through deep image analogy,” ACM Trans. Graph., vol. 36, no. 4, 2017, Art. no. 120.

[84] Y. Chen, Y.-K. Lai, and Y.-J. Liu, “Transforming photos to comics using convolutional neural networks,” in Proc. Int. Conf. Image Process., 2017, pp. 2010–2014.

[85] I. Goodfellow, “Generative adversarial nets,” in Proc. 27th Int. Conf. Neural Inf. Process. Syst., 2014, pp. 2672–2680.

[86] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum, “Learning a probabilistic latent space of object shapes via 3D generative- adversarial modeling,” in Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 82–90.

[87] V. Dumoulin et al., “Adversarially learned inference,” in Proc. Int. Conf. Learn. Representations, 2017, pp. 1–18.

[88] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 5967–5976.

[89] L. Karacan, Z. Akata, A. Erdem, and E. Erdem, “Learning to generate images of outdoor scenes from attributes and semantic layouts,” CoRR, vol. abs/1612.00215, 2016.

[90] Y. Choi, M. Choi, M. Kim, J. Ha, S. Kim, and J. Choo, “Stargan: Unified generative adversarial networks for multi-domain imageto-image translation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 8789–8797.

[91] J. Zhang, “Sparsely grouped multi-task generative adversarial networks for facial attribute manipulation,” in Proc. ACM Multimedia Conf. Multimedia Conf., 2018, pp. 392–401.

[92] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in Proc. IEEE Conf.Comput. Vis. Pattern Recognit, 2019, pp. 4396–4405. X. Yang, D. Xie, and X. Wang, “Crossing-domain generative adversarial networks for unsupervised multi-domain image-toimage translation,” in Proc. 26th ACM Int. Conf. Multimedia, 2018, pp. 374–382.

[94] R. Gomez, Y. Liu, M. de Nadai, D. Karatzas, B. Lepri, and N. Sebe, “Retrieval guided unsupervised multi-domain image to image translation,” in Proc. 28th ACM Int. Conf. Multimedia, 2020, pp. 3164–3172.

[95] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta,A. Aitken, A. Tejani, J. Totz, Z.Wang, andW. Shi, ``Photo-realistic single image super-resolution using a generative adversarial network,'' in Proc.IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, USA,Jul. 2017, pp. 105_114.

[96] X. Wang, ``ESRGAN: Enhanced super-resolution generative adversarial networks,'' in Proc. Eur. Conf. Comput. Vis. Workshops (ECCVW), Sep. 2018, pp. 1_16.

[97] X. Wang, K. Yu, C. Dong, and C. Change Loy, ``Recovering realistic texture in image super-resolution by deep spatial feature transform,'' in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Salt Lake City, UT, USA, Jun. 2018, pp. 606_615.

[98] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, ``Generative image inpainting with contextual attention,'' in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Salt Lake City, UT, USA Jun. 2018, pp. 5505_5514.

[99] B. Dolhansky and C. C. Ferrer, ``Eye in-painting with exemplar generative adversarial networks,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7902_7911.

[100] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. Huang, ``Free-form image inpainting with gated convolution,'' in Proc. IEEE/CVF Int. Conf.Comput. Vis. (ICCV), Oct. 2019, pp. 4471_4480.

[101] K. Nazeri, E. Ng, T. Joseph, F. Z. Qureshi, and M. Ebrahimi, ``EdgeConnect: Generative image inpainting with adversarial edge learning,'' 2019,

[102] Y. Zeng, J. Fu, H. Chao, and B. Guo, ``Learning pyramid-context encoder network for high-quality image inpainting,'' in *Proc. IEEE/CVF Conf.Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1486_1494.

[103] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, ``Highresolution image inpainting using multi-scale neural patch synthesis,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6721_6729.

[104] Y. R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, ``Semantic image inpainting with deep generative models,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, 2017, pp. 6882_6890.

[105] Y. Li, S. Liu, J. Yang, and M. Yang, ``Generative face completion,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 5892_5900.

[106] T. Xiao, J. Hong, and J. Ma, ``ELEGANT: Exchanging latent encodings with GAN for transferring multiple face attributes,'' in *Proc. Eur. Conf.Comput. Vis. (ECCV)*, 2018, pp. 168_184.

[107] O. Ronneberger, P. Fischer, and T. Brox, ``U-Net: Convolutional networks for biomedical image segmentation,'' in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, 2015, pp. 234_241.

[108] M. Liu, Y. Ding, M. Xia, X. Liu, E. Ding,W. Zuo, and S.Wen, ``STGAN: A uni_ed selective transfer network for arbitrary image attribute editing,''in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Beach, CA, USA, Jun. 2019, pp. 3673_3682.

[109] Z. He,W. Zuo, M. Kan, S. Shan, and X. Chen, ``AttGAN: Facial attribute editing by only changing what you want,'' *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5464_5478, Nov. 2019.

[110] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, ``Star- GAN: Uni_ed generative adversarial networks for multi-domain Imageto- Image translation,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8789_8797.

[111] S. Jiang, H. Liu, Y. Wu, and Y. Fu, ``Spatially constrained generative adversarial networks for conditional image generation,'' (2019), *arXiv:1905.02320*. [Online]. Available: https://arxiv.org/abs/1905.02320

[112] M.Wang, G.-Y. Yang, R. Li, R.-Z. Liang, S.-H. Zhang, P. M. Hall, and S.- M. Hu, ``Example-guided style-consistent image synthesis from semantic labeling,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 1495_1504.

[113] S. Jiang, Z. Tao, and Y. Fu, ``Segmentation guided Image-to-Image translation with adversarial networks,'' 2019, *arXiv:1901.01569*. [Online]. Available: http://arxiv.org/abs/1901.01569

[114] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, ``MaskGAN: Towards diverse and interactive facial image manipulation,'' 2019, *arXiv:1907.11922*. [Online]. Available: https://arxiv.org/abs/1907.11922

[115] [88] Y. Chen, Y. Lai, and Y. Liu, ``CartoonGAN: Generative adversarial networks for photo cartoonization,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 9465_9474.

[116] S. You, N. You, and M. Pan, ``PI-REC: Progressive image reconstruction network with edge and color domain,'' 2019, *arXiv:1903.10146*. [Online]. Available: http://arxiv.org/abs/1903.10146

[117] R. Suzuki, M. Koyama, T. Miyato, T. Yonetsuji, and H. Zhu, ``Spatially controllable image synthesis with internal representation collaging,'' 2018, *arXiv:1811.10153*. [Online]. Available: http://arxiv.org/abs/1811.10153

[118] J. Kim, M. Kim, H. Kang, and K. Lee, ``U-GAT-IT: Unsupervised generative attentional networks with adaptive layer-instance normalization for Image-to-Image translation,'' 2019, *arXiv:1907.10830*. [Online]. Available:http://arxiv.org/abs/1907.10830

[119] R. Wu, X. Gu, X. Tao, X. Shen, Y.-W. Tai, and J. iaya Jia,``Landmark assisted CycleGAN for cartoon face generation,'' 2019, *arXiv:1907.01424*. [Online]. Available: http://arxiv.org/abs/1907.01424