International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 08 | August - 2022 Impact Factor: 7.185 ISSN: 2582-3930

CartoonifyGAN: Generative Adversarial Networks for Image Cartoonization

Siji Jose , Jasmin M R

Abstract: In this paper, a solution forconverting real-world pictures into cartoonified pictures that is both useful and exciting in computer vision is proposed. Our method is organised as a knowledge-based plan that has recently gained popularity as a method of stylizing images in creative forms such as painting. Existing artistic style methods on the other hand do not produce satisfactory results because (1) cartoon styles have distinct features such as elite resolution and generalizability (2)cartoon images have smooth edges with obvious color changes. In this paper, it provide cartoonifyGAN,a Generative Adversarial Network (GAN) methodology for cartoonization. It utilize mismatched photographs and hilarious images for teaching cartoonization which is a simple process and makes excellent cartoon drawings from real-world pictures

Indexterms:*GAN, Generator, Discriminator, Cartoonization .*

I. INTRODUCTION

Cartoons are a popular style of art that we see every day. Their uses span from publication in printed media to narrative for children's education, in addition to aesthetic interests. Many iconic cartoon images, like other kinds of art, were based on real-life occurrences. Manually replicating real-world situations in cartoon styles, on the other hand, is time-consuming and requires a high level of creative ability. Artists must draw every single line and shade every color region of target scenes in order to produce high-quality cartoons. Existing image editing software/algorithms with conventional capabilities, on the other hand, are unable to provide adequate results when it comes to cartoonization. As a result, specifically created systems that can automatically turn real-world pictures into high-quality cartoon style images are quite useful, and artists may save a significant amount of time.These tools are also a good complement to picture editing applications like Instagram and Photoshop.

In the field of non-photorealistic rendering, the art of stylizing pictures has received a lot of attention. Traditionally, dedicated algorithms for distinct styles are developed. Fine-grained styles that replicate specific artists, on the other hand, need a significant amount of effort. Learning-based style transfer approaches, in which a picture is styled depending on specified examples, have recently attracted a lot of interest. The ability of cyclically formed Generative Adversarial Networks (GANs) to accomplish high-quality style transfer is investigated, with the unique feature that the model is trained utilising unpaired pictures and styled images.

Despite the fact that learning-based stylization has had great success; current approaches fail to create cartoonized images of acceptable quality. There are two main causes behind this. To begin with, cartoon pictures are very simplified and abstracted from real-world photography, rather than adding textures such as brush strokes as in many other genres. Second, despite the diversity of techniques among artists, cartoon pictures have a distinct look—clear edges, smooth color shading, and generally basic textures—that distinguishes them from other types of art.

This research presents CartoonifyGAN, a GAN-based technique to picture cartoonization, in this study. For training, it uses a set of pictures and a set of cartoon images. It doesn't require matching or correspondence between two sets of photos to get high-quality results while keeping the training data simple to collect. The purpose of cartoon stylization, from the standpoint of computer vision algorithms, is to transfer pictures from the photo manifold into the cartoon manifold while keeping the content unaltered. This suggests using a specialized GAN-based architecture in conjunction with two basic but effective loss functions to achieve this aim.

II. RELATED WORKS

i. Non-Photorealistic Rendering

Many NPR algorithms have been built to replicate certain aesthetic styles, including
cartoons, either automatically or semi-automatically. Some works use basic shading to represent 3D forms, creating a cartoon-like look . Cel shading is a method that may save artists a significant amount of time and has been utilized in the development of games, cartoon videos, and movies . However, converting existing images or movies into cartoons, like in the case of the subject investigated in this study, is even more difficult.

To make images with flat shading that resemble cartoon styles, a number of approaches have been devised. Image filtering or formulations in optimization problems are two examples of such strategies.Winnemöller *et al.* [8] use image filtering to render a given image to produce a cartoon image.Simple mathematical methods, on the other hand, make it impossible to portray complex aesthetic styles. Applying uniform filtering or optimization to the whole image, for example, does not achieve the high-level abstraction that an artist would achieve, such as making object borders evident. Alternative approaches depend on image/video segmentation to increase outcomes , but this comes at the cost of some user engagement. Dedicated approaches for portraiture have also been developed, in which semantic

segmentation may be produced automatically by recognizing face components. However, such approaches are incapable of dealing with common photos.

ii. **Neural Style-Transfer Method**

Convolutional Neural Networks (CNN) is always considering as a problem solver in case of image or computer vision areas. According to traditional style transfer algorithms, which require both style / non style images, in last few researches shows that VGG network trained for object recognition has better ability to carry out semantic features of objects and it is one of the important part of stylization. Another format is Image to Image translation where it deals with transferring image from one domain to another domain. It provide image quality enhancement, stylizing photos into paints, cartoon images and sketches. Bi- directional models are proposed for inter domain translation before few days. Zhu et. Al [9], performs transformation of Rain to winter and sketch to paint of unpaired images.

iii. **GAN-Based Methods**

Generative Adversarial Network (GAN), has produced cutting-edge achievements in text-to-image translation, picture inpainting, image super-resolution, and other areas. However, when undertaking image synthesis, generic GAN[10] requires matching picture sets, which is frequently impractical because it is difficult to locate data sets that contain both actual and synthetic images.

AnimeGAN: Anovel GAN-based anime-face translator, called AnimeGAN [5], to synthesize high-qualityanime-faces. Specifically, a new generator architecture is proposed to simultaneouslytransfer color/texture styles and transform local facial shapes into anime-likecounterparts based on the style of a reference anime-face, while preserving the globalstructure of the source photo-face. New normalization functions are designed for thegenerator to further improve local shape transformation and color/texture style transfer.

CycleGAN : This methodadopts a cyclic training strategy. It can generate vivid artistic images through unpaired training data. However, CycleGAN [4] can only learn the style of one artist at a time, and simply training multiple pairs of CycleGAN models willlead to huge computational costs if multiple artists are needed. Its splits the encoder and decoder of the generator, and solves the above problems by reusing the encoder and decoder. It achieves multi-style transfer by using mask vector and a single model (i.e., using only one generator and one discriminator). It realize multi-domain image-to-image translation by sharing common information among multiple domains through a shared knowledge module.

CartoonGAN : A Generative Adversarial Network to transform real-world photos to high-quality cartoon style images. Aiming at recreating faithful characteristics of cartoon images, it propose (1) a novel edge-promoting adversarial loss for clear edges, and (2) an$\ell 1$ sparse regularization of high-level feature maps in the VGG network for content loss,which provides sufficient flexibility for reproducing smooth shading. Also it proposes a simple yet efficient initialization phase to help improve convergence. The experiments show that CartoonGAN[2] is able to learn a model that transforms photos of real world scenes to cartoon style images with high quality and high efficiency, significantly outperforming the state-of-threat stylization methods.

CartoonLossGAN : This framework, which generates vividCartoon images by learning surface and coloring of images to imitate the cartoon creation process of sketching first and then coloring. The proposed cartoon loss function canimitate the process of sketching to learn the smooth surface of the cartoon image, and imitate the coloring process to learn the coloring of the cartoon image. In addition, it reuse the encoder part of the discriminator without using a cyclic manner and a complexmulti-scale discriminator to build a compact generative adversarial network (GAN) [1] based cartoonization architecture. Furthermore, it also propose an initialization strategy, whichis used in the scenario of reusing the discriminator to make our model training easier and more stable.

III. PROPOSED MODEL

This model learns to convert real-world photographs to cartoon pictures as a mapping function that maps the photo manifold P to the cartoon manifold C. The mapping function is learnt using training data $S_{data}(p) = \{p_i \mid i = 1 \ldots N\} \subset P$ and $S_{data}(c) = \{c_i \mid i = 1 \ldots M\} \subset C$, where N and M are the number of picture and cartoon images, respectively as shown in Figure.1, in the training set.
Similar to previous GAN frameworks, a discriminator function D is trained to assist G in achieving its objective by differentiating pictures in the cartoon manifold from other images and providing adversarial loss for G. Let L represent the loss function, while G and D represent the network weights. Our goal is to solve the minimum-maximum problem ,using the equation(1),

$$(G^*, D^*) = \arg\min_G \max_D L(G, D) \quad (1)$$

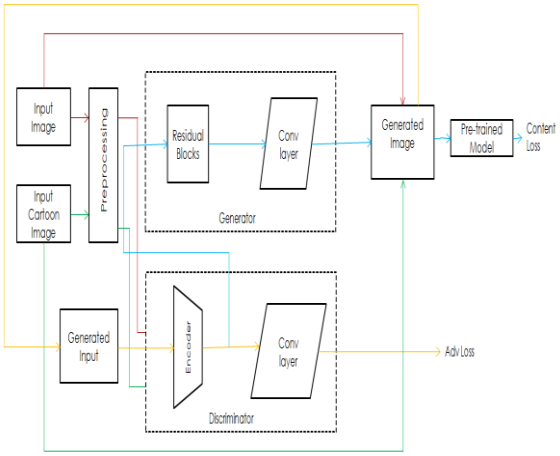

Figure 1 : Architecture of CartoonifyGAN (Training Phase)

Two datasets, such as cartoons and real-world photos, are preprocessed with a variety of smoothing techniques before being used in the training phase as shown in Architecture of CartoonifyGAN Figure.1. This prepares the datasets for use in the training phase. After that, the encoder of the discriminator will convert it, and then it will be sent to the residual blocks and convolutional layers of the generator .The data that is fed into deep neural networks leads those networks to generate images that are consistent with the data that was fed into them. The pre-trained VGG-16 model [6] is applied to the newly produced photographs in order to extract attributes from them. The discriminator is used to determine the adversarial loss based on the input that was produced. The generator model has been saved after going through a certain number of training 210 epochs.

The generator creates cartoon-like representations of the real-world input during the generating phase as shown in Figure.2. The input picture is altered using a few preprocessing stages, and then it is fed through the CartoonifyGAN to produce the desired outcomes.

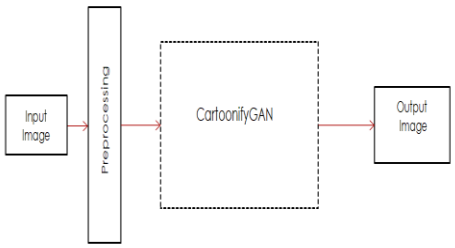

Figure 2: Generating Phase

➢ Data set

The cartoon dataset provided by AnimeGAN is used . Among them, the training data includes realworld photos and cartoon images, while the test data only has real-world photos,and all training data is processed as 256 x 256.

- Real-World Photos: In this experiment, it use 6656 real- world photos fortraining and 790images for testing, and they are all processed as 256 x 256. In addition, it also use 45high-resolution images to show the cartoonized results of our model.
- Cartoon Images: Cartoon images include color cartoon images, grayscale cartoon images, andsmooth cartoon images. Cartoon images are used for training only.

➢ Data Preprocessing

First, it process all cartoon images into 256 x 256. Then, smooth the edges of the cartoon images using smoothing techniques .Then basic morphological like Erosion is done . Here, erosion is used for removing small white noises and to detach two connected objects. It erodes away the boundaries of the foreground object and diminishes the features of an image.

Smoothing Techniques: The premise of data smoothing is that one is measuring a variable that is both slowly varying and also corrupted by random noise. Filtering has an effect of smoothing which removes noises in images, another representative approach, the Gaussian kernel, has been applied. Gaussian kernel, as its name implies, has the shape of the function ‘Gaussian distribution’ to define the weights inside the kernel, which are used to compute the weighted average of the neighboring points (pixels) in an image.

Erosion : It shrinkens the image pixels i.e. it is used for shrinking of element A by using element B. It removes pixels on object boundaries.The value of the output pixel is the minimum value of all the pixels in the neighborhood. A pixel is set to 0 if any of the neighboring pixels have the value 0.

➢ Training the model

In the preprocessing step, two datasets, such as cartoons and real-world photos, are preprocessed with a variety of smoothing techniques before being used in the training phase as shown in Figure.3. This prepares the datasets for use in the training phase. After that, the encoder of the discriminator will convert it, and then it will be sent to the residual blocks and convolutional layers of the generator to undergo additional processing there. Following processing, the data that is fed into deep neural networks leads those networks to generate images that are consistent with the data that was fed into them. The pre-trained VGG-16 model is applied to the newly produced photographs in order to extract attributes from them. The discriminator is what is used to determine the adversarial loss in Figure.4 ,based on the input that was

produced, and it does this using the information that was provided. The generator model has been saved after going through a certain number of training epochs, and it is now getting ready to be made available to the general public.

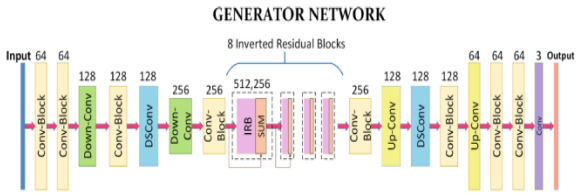

Figure 3

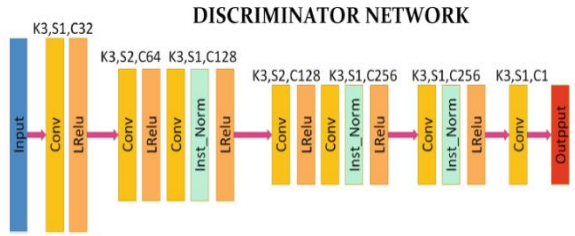

Figure 4

➢ Testing the model

The generator creates cartoon-like representations of the real-world input during the generating phase as shown in Figure.2. The input picture is altered using a few preprocessing stages, and then it is fed through the CartoonifyGAN to produce the desired outcomes.

➢ Pre-trained VGG -16 Architecture

The 16 in VGG16 refers to 16 layers that have weights as shown in Figure.5. In VGG16 there are thirteen convolutional layers, five Max Pooling layers, and three Dense layers which sum up to 21 layers but it has only sixteen weight layers i.e., learnable parameters layer.VGG16 takes input tensor size as 224, 244 with 3 RGB channel. Most unique thing about VGG16 is that instead of having a large number of hyperparameters they focused on having convolution layers of 3x3 filter with stride 1 and always used the same padding and maxpool layer of 2x2 filter of stride 2. The convolution and max pool layers are consistently arranged throughout the whole architecture . Conv-1 Layer has 64 number of filters, Conv-2 has 128 filters, Conv-3 has 256 filters, Conv 4 and Conv 5 has 512 filters. Three Fully-Connected (FC) layers follow a stack of convolutional layers: the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer.

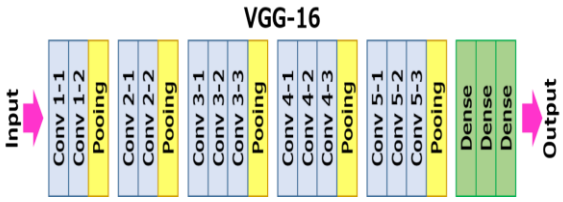

Figure 5

IV. PERFORMANCE AND EVALUATION

How to evaluate the results of creative style transfer is a difficult problem because people's perspectives on the matter vary greatly and are difficult to quantify. In this paper Frechet Inception Distance (FID) [3] is used to evaluate our model.FID collects visual traits and establishes the separation between two pictures using a trained VGG-16model. Distributions utilizing the extracted image properties in order toobserve how close the two image distributions are, compare them. The FID score distribution is worse, which suggests that the distribution of the created image is more comparable to that of the referencereal image. In other words, the final picture is morelike the image itself.
Cartoon and content photos are given separate FID ratings. FID scores generated using Shinkai approach are used mainly for comparison. As shown in Table 1, in order to determine if the created pictures are able to maintain the content images' semantic information, the FID score is calculated.

Table 1

Model	FID Score to content loss
CycleGAN	91.95
AnimeGAN	77.36
CartoonGAN	58.71
CartoonLossGAN	53.95
CartoonifyGAN	52.81

V. EXPERIMENTAL RESULTS

The discriminator primarily reused to develop compact generative adversarial network-based cartoonization architecture and to prevent the waste of the discriminator; however, the re-use of the discriminator also enables our CartoonifyGAN to generate better pictures that have a cartoon look , as in Figure.5. When the discriminator is not employed again, several peculiar vertical lines are produced . In the scenario in which the discriminator is used again, our model produces a lawn that is complete and uniform. It do not compare Gatys *et al.* [7] , because for each input content image very similar to it. When an input cartoon picture is

received, the discriminator is applied to it in order to assess whether or not the image was produced by the generator or was an actual cartoon. Through the process of training the discriminator, the encoder of the discriminator is able to glean more significant information that may be used to differentiate between cartoon pictures. This feels that using an encoder of this kind allows us to produce better graphics in the cartoon style.

Input Image Generated Image

Figure 5

The graph below shows the FID scores of different GANs

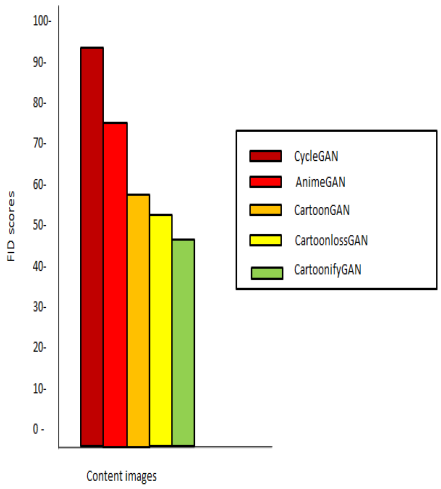

Figure 6 : GAN comparisons

CONCLUSION

In this research, proposed employing a Generative Adversarial Network (named CartoonifyGAN) to create cartoon-like images from real-world photographs. A novel adversarial loss that promotes clean edges is proposed, as is a sparse regularization of high-level feature maps in the VGG-16 network for content loss, allowing for smooth shading reproduction. The ultimate objective is to produce visuals that resemble cartoons in every way. With addition, itrecommends a quick and easy setup procedure to aid in convergence. Tests have shown that a model that accurately and efficiently transforms photographs of realistic scenes into cartoons may be learned by CartoonifyGAN and a reduction in FID Score (in Figure 6) by one. It's much superior to other "state-of-the-art" stylization systems in this regard.

REFERENCES

[1] Yongsheng Dong , Wei Tan, Dacheng Tao , Lintao Zheng , and Xuelong Li , “CartoonLossGAN: Learning Surface and Coloring of Images for Cartoonization” in IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 31, 2022 485

[2]Y. Chen, Y.-K. Lai, and Y.-J. Liu, “CartoonGAN: Generative adversarial networks for photo cartoonization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 9465–9474.

[3] 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 Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 6629–6640.

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

[5] J. Chen, G. Liu, and X. Chen, “AnimeGAN: A novel lightweight GAN for photo animation,” in Proc. Int. Symp. Intell. Comput. Appl., 2019, pp. 242–256.

[6] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556, 2014.

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

[8] Y. Jing *et al.*, “Stroke controllable fast style transfer with adaptive receptivefields,” in *Proc. Eur. Conf. Compu. Vis.*,2018, pp. 238–254.

[9] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, “Neural style transfer: A review,” *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 11, pp. 3365–3385, Nov. 2020.

[10] I. Goodfellow *et al.*, “Generative adversarial nets,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

★★★