JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Generation of Cartoon Images Using Deep Generative Model

Mr. Akshay B. Rupnar 1 , Miss. Jyoti Kendule 2
1 Sveri's College of Engineering (Polytechnic), Pandharpur, Solapur Maharashtra, India
2 Sveri's College of Engineering , Pandharpur, Solpaur Maharashtra, India

Abstract : In this paper, we propose a solution to transforming photos of real-world scenes into cartoon style images, which is valuable and challenging in computer vision and computer graphics. Our solution belongs to learning based methods, which have recently become popular to stylize images in artistic forms such as painting. However, existing methods do not produce satisfactory results for cartoonization, due to the fact that cartoon styles have unique characteristics with high level simplification and abstraction, and cartoon images tend to have clear edges, smooth color shading and relatively simple textures, which exhibit significant challenges for texture-descriptor-based loss functions used in existing methods. In this paper, we propose CartoonGAN, a generative adversarial network (GAN) framework for cartoon stylization. Our method takes unpaired photos and cartoon images for training, which is easy to use.

***IndexTerms* - CNN, GAN**

1. Introduction

Cartoons are commonly used in various kinds of applications. As we know cartoons are artistically made it requires elegant and fine human artistic skills**.**

Cartoons are an artistic form widely used in our daily life. In addition to artistic interests, their applications range from publication in printed media to storytelling for children's education. Like other forms of artworks, many famous cartoon images were created based on real-world scenes. Animation movies are currently one of the most popular forms of entertainment. Nowadays, the animation has become more realistic and it is not drawn frame by frame by artist rather it is acted by actors then given animation look to the real video to generate the animation. This process lots of expert graphics a lot of time to complete one animation. This is because the animator has to go through the video frame by frame and change the character and scenarios into the animation form. To help this process we have proposed a neural network based conditional image generation model that takes a real-world image and generate a corresponding animation image.

2. Related work

2.1 Statement

A solution to transforming photos of real-world scenes into cartoon style images, which is valuable and challenging in computer vision and computer graphics. A solution belongs to learning based methods, which have recently become popular to stylize images in artistic forms such as painting. However, existing methods do not produce satisfactory results for cartoonization, challenges as (1) cartoon styles have unique characteristics with high level simplification and abstraction, and (2) cartoon images tend to have clear edges, smooth colour shading and relatively simple textures, which exhibit significant challenges for texture-descriptor-based loss functions used in existing methods. Therefore, we are focusing to generate high-quality cartoon images from real-world photos (i.e., following specific artists' styles and with clear edges and smooth shading) and outperforms state-of-the-art methods.

2.2 GAN

Generative Adversarial Networks (GANs) [13] has two neural networks: a generator (G) and a discriminator (D). The task of the generator (G) is to generate a photorealistic image while the discriminator (D) distinguishes the real image and the generated image of the generator which gives the decision about the real image as well as fake image. Both the generator and discriminator plays the min-max game until there is a confusion of discriminator between the real image and fake image generated by the generator because a generated image is too close to the real one. GANs have many

applications in computer vision like image super-resolution, image colorization [14], image dehazzing [15], etc. The min-max game function is expressed as,

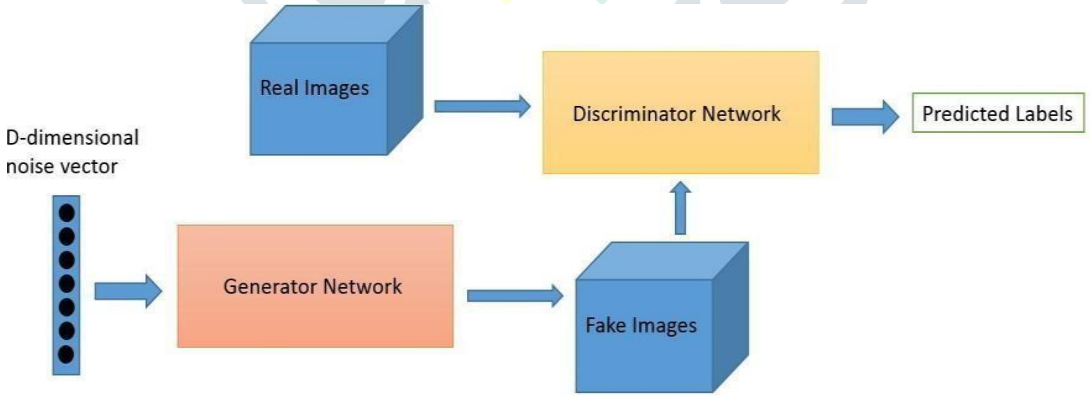
$$\min_{G}\max_{D} \mathbb{E}_{x\sim p_{data}}[\log D(x)] + \mathbb{E}_{z\sim p(z)}\left[\log\left(1 - D(G(Z))\right)\right] \quad (1)$$

A noise vector, z is given to a generator that is sampled from a normal distribution p(z) which maps z to produce synthesized image complement, x. The first part of the equation ($\mathbb{E}_{x\sim p_{data}}[\log D(x)]$) described as log probability of discriminator, D predicts that real-world data is original. The second part of the equation ($\mathbb{E}_{z\sim p(z)}\left[\log\left(1 - D(G(Z))\right)\right]$,) as log probability of discriminator, D predicts G's generated data which is not real. For easy training of Generative Adversarial Networks (GANs), Radford et al. [12] proposed DCGANs

(Deep Convolutional Generative Adversarial Network) for many applications such as video data frame prediction, cross-domain image generation network. M. Mirza et al. designed Conditional Generative Adversarial Networks (CGANs) for image generation, based on the availability of prior information. Recently, CycleGANs is used for many applications such as unpaired image datasets training [16], image2image translation and achieving good performance in terms of loss reduction, accuracy, etc. This work prominently helps to improve the performance of GANs in image generation.

2.3 Architecture

The generator network G is used to map input images to the cartoon manifold. Cartoon stylization is produced once the model is trained. G begins with a flat convolution stage followed by two down-convolution blocks to spatially compress and encode the images. Useful local signals are extracted in this stage for downstream transformation. Complementary to the generator network, the discriminator network D is used to judge whether the input image is a real cartoon image. Since judging whether an image is cartoon or not is a less demanding task, instead of a regular full-image discriminator, we use a simple patch-level discriminator with fewer parameters in D. Different from object classification, cartoon style discrimination relies on local features of the image. Accordingly, the network D is designed to be shallow. After the stage with flat layers, the network employs two stride convolutional blocks to reduce the resolution and encode essential local features for classification.



3. Cartoon GAN

A GAN framework consists of two CNNs. One is the generator G which is trained to produce output that fools the discriminator. The other is the discriminator D which classifies whether the image is from the real target manifold or synthetic. We design the generator and discriminator networks to suit the particularity of cartoon images

3.1 Network Architecture

Generator:

When the dimension of the input vector is lower than the dimension of the output vector, the neural network is equivalent to a decoder. The generator G acts as a decoder corresponding to the convolutional encoder E and takes the face feature vector z, the age vector a, and the gender vector as input, and the face image is reconstructed from the feature information. The gender condition is added because the aging characteristics of different genders are very different. The aging synthesis of a face is carried out on the basis of clear gender, which can avoid the influence of gender on the aging result. The network structure of the generator is shown in Figure 2.

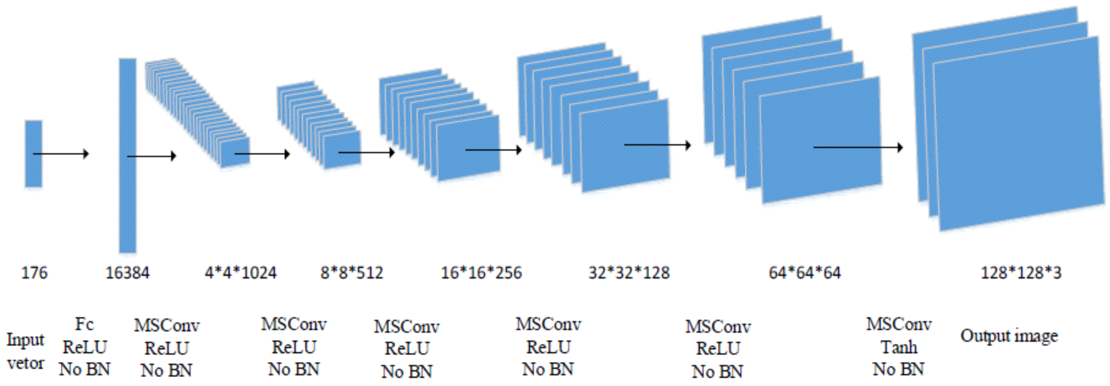


Fig.2 Deconvolution network structure of the generator

The input to the generator is the merge vector of feature vector z, age vector a, and gender vector g. From the structure diagram of the convolutional encoder, z is a 60-dimensional feature vector, and the age information is an eight-dimensional one-hot vector. The gender information is a two dimensional one-hot vector. If directly merged, the age condition and gender condition will have little effect on the generator. In order to balance the influence of eigenvectors and conditional vectors on the composite image, the age vector a is copied seven times before merging to obtain a 56-dimensional vector, and the gender vector g is copied 30 times to obtain a 60-dimensional vector. Then, the conditional input of the

generator is a 116-dimensional vector, and the eigenvector z is combined to obtain a 176-dimensional input. The most important operation of generating a network is the Fractional-Strided Convolution, which is also considered to be deconvolution in many places. In the micro-step convolution operation, it is adopted. A convolution kernel of size 5 × 5 with a step size of 2 × 2. Similar to the convolutional coding network, batch normalization is not used in the generation network, and the Relu activation function is used for all layers except the output layer using the Tanh activation function.

Descriminator:

The role of the discriminator is to distinguish between the real face image and the synthetic face image and, finally, output a scalar value indicating the probability that the discriminator's input image is a real face image. The network structure of the discriminator is shown in Figure 3. As can be seen from Figure 2, the input is an RGB face image (real image or composite image) of size 128 × 128 pixels, and the output is a scalar value in the range of (0, 1). The constraint is connected to the first convolution layer according to the design rules of the condition GAN. Specifically, after the input image passes through the first convolutional layer, a feature map of 16 pixels is output and then connected to the conditional feature map after the extended copy to obtain a feature map of 32 pixels. After the conditional connection is successful, the 32 feature maps are convoluted and fully connected, and finally, a scalar value representing the probability is output.

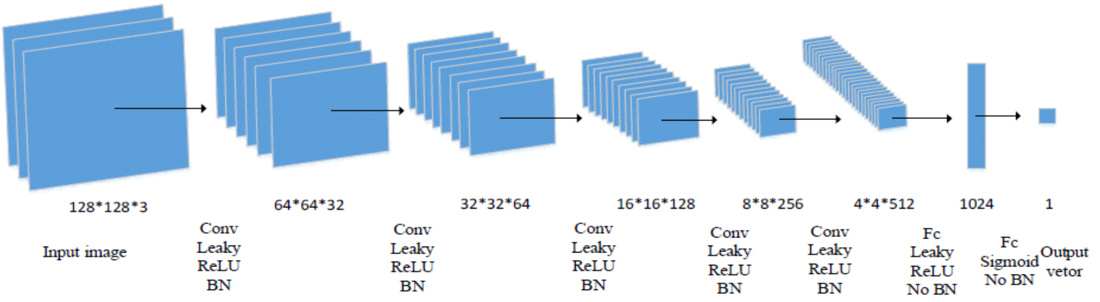


Fig. 3. Convolution network structure of the descriminator

3.2 Convolutional Neural Networks

This network is specially used and designed to handle pixel data; it is kind of an artificial neural network which has applications in _led like image processing and recognition. The make of CNN includes many layers which are the input layer, output layer and a multilayer perceptron layer included as the hidden layer, then we have the fully connected layers and normal layers.

3.5 Loss Function

Generative adversarial networks attempt to replicate a probability distribution. Therefore, they use loss functions to show the gap in the data distribution produced by the GAN and that of realworld. GAN mainly constitutes of two loss functions, one which is for the training of discriminator and the other for the training of generator. The generator loss and discriminator loss derive from a single measurement of separation among the probability distributions. In any case, the generator can just influence one term in the distance measure that will react the distribution of the counterfeit data. Hence, while training the generator the other term is dropped, which will react the distribution of the real data. The generator loss and discriminator loss appear to be unique at the end, despite the fact that they derive from a single formula, the generator at-tempts to limit the function whereas
the discriminator attempts to expand it: Ex[log(D(x))] + Ez[log(1 - D(G(z)))] D(x) Estimation given by discriminator that x is real. Ex- Instances of real world.

4. EXPERIMENTATION

The following experiments were conducted in order to derive the accurate results:

4.1 Cartoon character generation using GAN

Using GAN we can create characters by taking human faces as input and then processing it to get a high density polygon which saves a lot of time in creating cartoon characters from scratch. GAN model creates a high polygon density image of the picture taken as input with the technology used up scaling. GAN model can generate a cartoon character from taken input in lesser time when provided with adequate CPUs and GPUs. GAN takes a set of photos in the form of a cartoon images for training. (CartoonSet100K). The input taken is then passed into the discriminator network. At first, we take a dimensional noise vector and the pass it into the Generator network, where the generator produces a fake image. The fake image is then passed into the discriminator network where the discriminator compares the fake image to the real image from the data set, if the image resembles to the data set then its classified as a predicted image else it returns back to the generator network for retraining. This sequence continues as a zero-sum game till the fake image is able to deceive the discriminator as a real once which is practically very much time consuming.So, when the probability to predict the real image exceeds around 60 to 70 percent, we predict the image as a required output.

4.2 Cartoonist

In the cartoonist application we take real world images taken by the user and process them. The dataset is first cleaned and outlier data is removed. Then the dataset is checked for some missing values and inappropriate values. The dataset is then passed through the discriminator and compared with the random noise generated by the generator. After the process secures a handsome accuracy the output is generated.

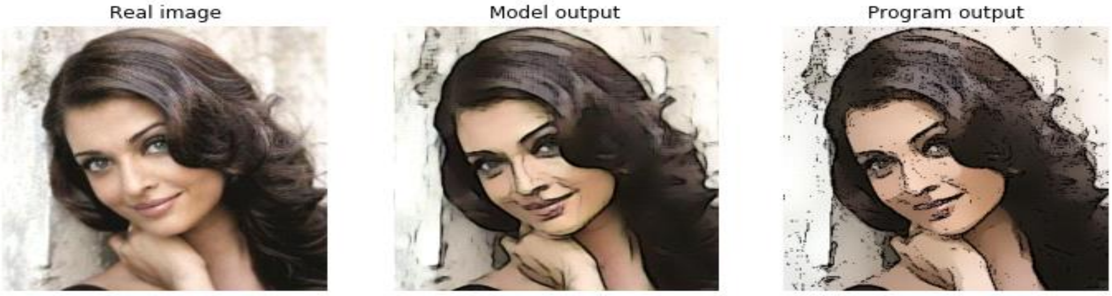
5. Result

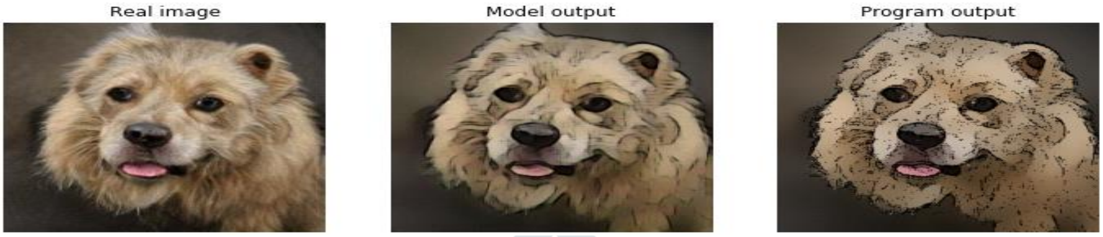
GAN model can generate a cartoon character from taken input in lesser time when provided with adequate CPUs and GPUs.

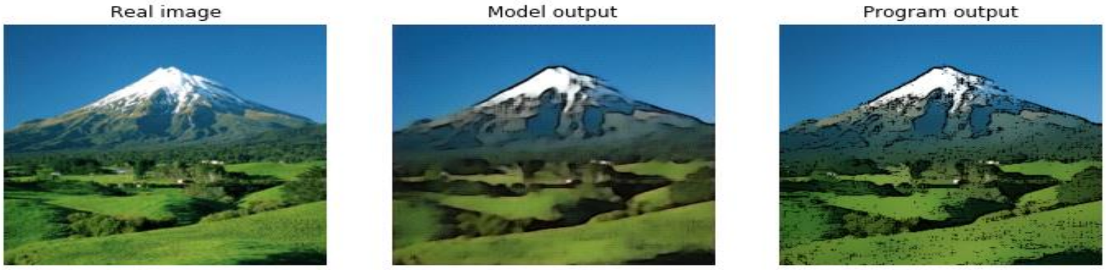
We take here input some different category images

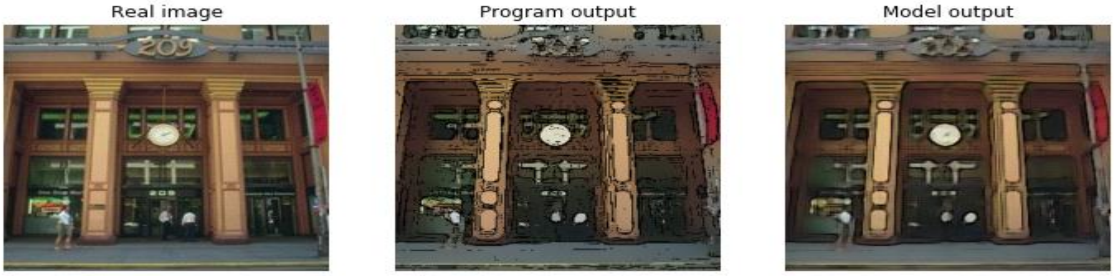
Input Image

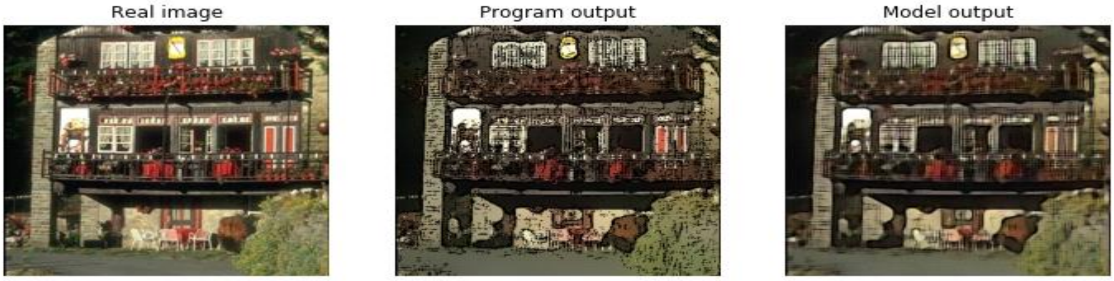
Output Image











Conclusion

Results show that this technique can produce excellent animation pictures from genuine world photographs which have explicit craftsmen's styles and with legible edges and smooth concealing and outflanks best in class strategies. Obtaining labeled data is a manual process and is time consuming too. GANs don't make use of labeled data and thus can be trained using unlabeled data. GAN can be helpful to transform real world photos to high quality cartoon images, outperforming other methods.

Refrences

[1] Shuvendu Roy, "Generating Anime from Real Human Image with Adversarial Training," *ICASERT*, 2019.[]

[2] Kun Xu, Peter Hall Xian Wu, "A Survey of Image Synthesis and Editing with Generative Adversarial Networks," vol. 22, no. 6, dec 2017.

[3] Zengchang Qin, Zhenbo Luo, Hua Wang Yifan Liu, "Auto-painter: Cartoon Image Generation from Sketch by Using Conditional Generative Adversarial Networks," 7 may 2017.

[4] Gauri Gakhar ,D Vanusha Gourab Guruprasad, "Cartoon Character Generation using Generative Adversarial Network," IJRTE, vol. 9, no. 1, may 2020.

[5] Yu-Kun Lai, Yong-Jin Liu Yang Chen, "CartoonGAN: Generative Adversarial Networks for Photo Cartoonization," IEEE Xplore, pp. 9465-9474.

[6] Ashwathy Unnikrishnan, Navjeevan Bomble, Prof. Sachin Gavhane Akanksha Apte, "Transformation of Realistic Images and Videos into Cartoon Images," *IRJET*, pp. 2118-2121, jan 2020.

[7] Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros Phillip Isola, "Image-to-Image Translation with Conditional Adversarial Networks," *IEEE Xplore*, pp. 1125-1134.

[8] Jinjin Gu, Xiaoou Tang, Bolei Zhou Yujun Shen, "Interpreting the Latent Space of GANs for Semantic Face Editing," IEEE Xplore, pp. 9243-9252, 2020.

[9] Jinze Yu Xinrui Wang, "Learning to Cartoonize Using White-box Cartoon Representations," IEEE, pp. 8090-8099, 2020.

[10] Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley,Sherjil Ozair Ian J. Goodfellow, "Generative Adversarial Nets," 10 jun 2014.

[11] Yu Xinyu, "Emerging Applications of Generative Adversarial Networks," *IOP*, 2019

[12] Michael Wand Chuan Li, "Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis," *IEEE xplore*, pp. 2479-2486

[13] Yu-Kun Lai, Yong-Jin Liu Yang Chen, "TRANSFORMING PHOTOS TO COMICS USING CONVOLUTIONAL NEURAL NETWORKS".

[14] Wei Zhang, Tong Shen, Tao Mei Xinyu Li, "EVERYONE IS A CARTOONIST: SELFIE CARTOONIZATION WITH ATTENTIVE ADVERSARIAL NETWORKS," 20 Apr 2019.

[15] Gourab Guruprasad, Gauri Gakhar, D Vanusha “Cartoon Character Generation using Generative Adversarial Network” ISSN: 2277-3878 (Online), Volume-9 Issue-1, May 2020

[16] Mingkui Tan, Yuguang Yan, Chunmei Qing, Qingyao Wu, Zhuliang Yu Junhong Huang, "Cartoon-to-Photo Facial Translation with Generative Adversarial Networks," ACML, 2018.