Face Morphing using Cartoon GAN

Akash Mehta
Computer Science and Engineering
PES University
Bangalore,India
akashmehta556@gmail.com

Hariharasudan S A
Computer Science and Engineering
PES University
Bangalore,India
hari30082001@gmail.com

Suhas K Pai
Computer Science and Engineering
PES University
Bangalore,India
suhaspai100@gmail.com

Dr. Mamatha H R
Computer Science and Engineering
PES University
Bangalore,India
mamathahr@pes.edu

Abstract—

This paper takes on the problem of transferring the style of Anime images to real-life photographic images by implementing previous work done by CartoonGAN. We trained a Generative Adversarial Network(GAN) on over 3000 images. 9007

Keywords—anime,cartoonGAN,style transfer
DOINumber:10.14704/nq.2022.20.8.NQ44920 **NeuroQuantology 2022; 20(8): 9007-9011**

I. Introduction

Anime are creative works of art with a wide range of styles and themes. Anime characters usually take a long time to create and need a lot of creativity. The idea of transferring style has been around for a long time. Neural networks can be used in the field of machine learning to automate the conversion of real-world representations made in a cartoon style. Neural networks that transmit information in two forms: paired and unpaired. A source image (in this case, a real-life image) and a target type image (in this case, a cartoon image) to be added to the source image are given in the paired method. This method necessitates a very unique set of pre- and post-stylized photographs. As a result, data sets for this method can be difficult to come by. The unpaired solution, on the other hand, uses two different collections of photographs: source (real-life) images and goal (anime) images. A model is trained to construct images from the two image sets, retaining the source material structure but adding the target form.

A popular method for unpaired style transfer is Generative Adversarial Networks (GAN). The main idea of GANs is that they allow us to generate samples from a new, unknown distribution of data, using previously seen data. It has been found that with enough data and computing power it is possible to recreate visual features with striking similarity to the target.

CartoonGAN is a tool for automating the task of making cartoon pictures. By using a convolutional eural network to analyse real-life images and transferring the visual style of current anime to them.

Their system relies on the use of convolutional neural networks (CNN) in accordance with GANs.

II. RELATED WORK

GANs were introduced in 2014 by Ian Goodfellow and since the introduction of GAN[4],Generative Adversarial Networks (GAN) have been under study and improvement In the field of machine learning. The attempts were Designed to allow GANs to be used for image to image tasks.

[5] Suggested encoder-decoder architecture which proved to produce better quality style transfer images using GANs.

[6] introduced style transfer, The objective of[6] was to produce a synthesised image using 2 images which belong to different domains, one domain selected such as it is the content target,and other being the style which we want to transfer to the content target.

Many papers[7,8] have given solutions for image-to-image synthesis. These implementations require paired image datasets for training which is practically impossible to obtain such paired images for stylization, This has led to [5] in which a framework has been introduced which is capable of forming image synthesis using unpaired dataset.

III. DATASETS

Our datasets have two classes, each consisting of test and train folders for anime and human faces. Preferentially we custom built them from scratch using a variety of sites like kaggle and socrata etc. Each class contains 3400 images each for both aforementioned folders in a resolution of 256x256 pixels. They are easily interchangeable as the second class is the style class, for which its images' style will be used to rebuild the first class images. Following CartoonGAN's implementation [1], we also removed the smoothed dataset from our cartoon dataset. This was done to reduce the stress that went into identifying edge pixels using a canny filter with a threshold between 150 and 500 and thereafter applying a Gaussian blur filter on a 3x3 pixel dilation of the identified edges. Each anime image was taken in consideration by the following strict criteria: the characters should acquire distinct features, long hair, a mix of female and male traits and initially a light skin tone for easy recognition.

IV. ARCHITECTURE

We implemented the model using PyTorch . All code is available on our Github. A generator G and discriminator D are used in our implementation, as they are in all GANs. The definitions in [3] are used to implement both networks. To classify the images as original cartoons, the discriminator is trained on both the style images and the generator output. The generator is trying to find a style to apply to the source image such that its result is able to fool the discriminator. The discriminator is then trained on the results from the

generator and the cycle continues. The networks are constantly trying to compete with each other, and in doing so trying to fool the other, while improving over time. The generator network, used to transform real-life images to cartoon versions, consists of 14 layers or blocks (see Fig. 1); 1 flat convolution block, 2 blocks for down-convolution, 8 residual blocks [8], 2 blocks for up-convolution, and 1 convolution layer. The residual blocks are adapted from the layout proposed. The discriminator network, used to classify an image as either cartoon or real-life, consists of 5 layers; 1 flat convolution layer, 2 down-convolutions, 1 feature construction block, and 1 convolution layer. The discriminator is what is called a patchGAN [3] which means that rather than giving a single classification per image, it is instead used on smaller, cropped sections (patches) of each image to produce a list of classifications results, one for each patch. Here we use Leaky ReLU (LReLU) with $\alpha = 0.2$ after every normalisation as per [1].

A loss function is used to evaluate the error, and therefore the performance of a model. In the context of GANs it is used to evaluate the performance of both networks [1]. [3] proposes the use of a combined loss function L(G, D) which takes into account both adversarial loss Ladv(G, D) and content loss Lcon(G, D) for both networks G and D (see Eq. (1)).

L(G, D) = Ladv(G, D) + ωLcon(G, D) (1)

ω is a scalar used to balance the two loss functions. Per [3], we set $\omega = 10$. The adversarial loss can be seen as a measure of how well the combined networks can transfer the cartoon style to the target image whereas the content loss can be seen as a measure of how well the content structure of the source image is preserved. Optimising the adversarial loss is a minimax problem [8] where the discriminator network tries to minimise the chance to make a wrong prediction, whereas the generator network aims to maximise the probability of fakes that the discriminator misclassified as real.

V. EXPERIMENT

A. *Training*

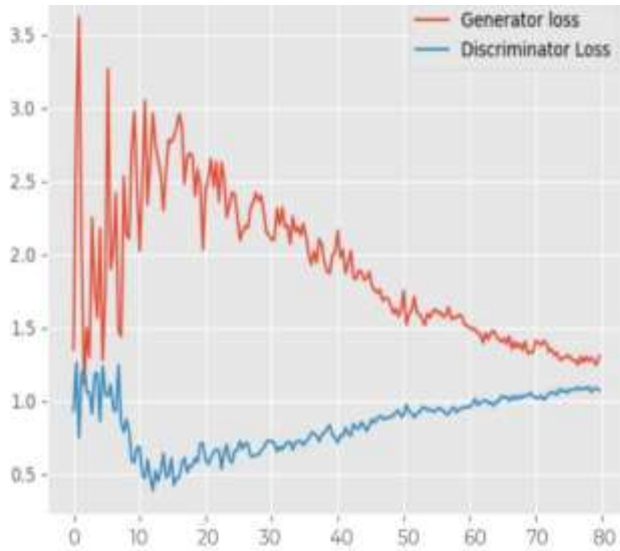
The training was conducted using a random high processing GPU assigned to us by Google Colab. Each image was resized to a size of 224x224 pixels. We replicated the initialization phase done by [3] and trained it for 80 epochs.

Each iteration was trained on a batch of 4 images. The optimizer used was AdamW which is an improvement . The learning rate was set to a constant 1 × 10−3 and a weight decay of 1 × 10−4 . We also used a cyclic learning
rate scheduler with the maximum learning rate set to 1 ×
10−2 .

B. ***Evaluation***

GAN evaluation can be difficult, particularly in the case of style transition, where the end result is a changed picture. [5] suggested a comparative assessment in which two neural networks are used to assess how well the source image's meaning is retained (Content-CNN) and how well the target image's form is transmitted (Style-CNN). They also propose a method for qualitative evaluation, comparing 4 different style transfer GANs to identify which model has the best ”style and content identifiability” [5]. The validity and consistency of using CNNs to test quantitative data is debatable. The Style-CNN of [5] and the discriminator mentioned in Fig. 1 are very close in that they both use patchGAN. We opted to do only a qualitative assessment using human subjects because the discriminator already tests on both style and material.

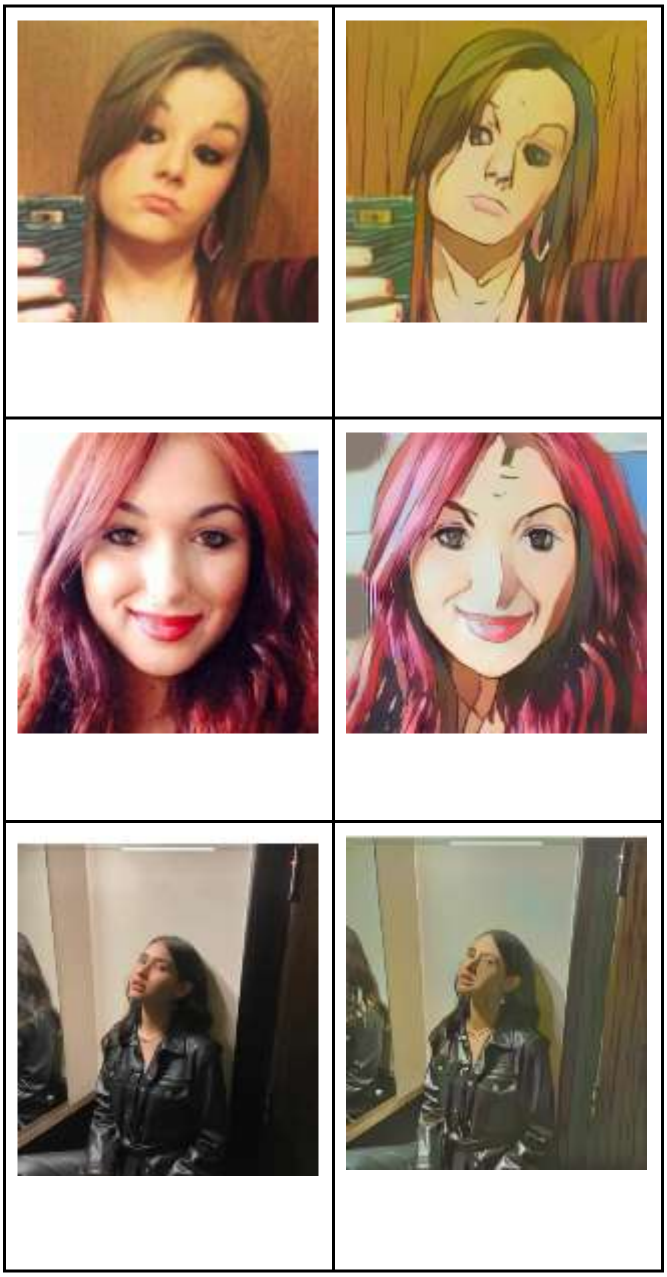
C. ***Figures and Tables***



The above graph is a loss plot which lets us know if the model is correct or not. As you can see the generator in the initial epoches has very high loss values because it is not able to generate the images it is expected to generate but as it is trained more the generator starts learning and is able to generate good images therefore the loss keeps on decreasing.

Whereas for the discriminator the loss initially decreases since it is easily able to distinguish the bad images generated by the generator. After some epoches loss steadily increases because the generator is able to produce good images and hence making it difficult for the discriminator to distinguish between the images

INPUT	OUTPUT
	



VI. Conclusion and Future Work

In this paper we implemented cartoon-gan which was used to transform scenic images or selfies or any image to anime styled images. Our primary goal was to create a product which allows you to see the world from the perspective of a manga, aka, a japanese anime artist.

In the future, time will be spent to increase and diversify the existing dataset including all races and ethnicities so there is no room for error.. Currently we have implemented style transfer for images and videos, so we will invest out time in working on real time style transfer which would let us to see the anime style transfer taking place in real time,which will work like snapchat or instagram filters and this also can be made into a web application or a mobile app which would let a user to use our implementation in a very user friendly manner.

Going ahead, the enhancements necessitates the refinement of the dataset and trying it with other architectures and comparing the results.

VII. Acknowledgment

This wouldn't have been possible without the support of our mentor Dr.Mamatha H R and panel Dr Natrajan whose feedback proved to be quite valuable . We would like to thank PES university which provided a platform to work on this paper.

Reference

[1] Kim, J., Kim, M., Kang, H. and Lee, K., 2019. U-GAT-IT: unsupervised generative attentional networks with adaptive layer-instance normalisation for image-to-image translation. *arXiv preprint arXiv:1907.10830*.

[2] Zhu, J.Y., Park, T., Isola, P. and 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. 2223-2232).

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

[4] I. 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 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

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

[6] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural

networks," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016, pp. 2414–2423.

[7] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017

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