Picture to Picture Morphing using CycleGAN mechanism

Chaitanya Praveen Kumar Gaur
Computer Science and Engineering
SRM Institute of Science and Technology
Chennai, India

Kazi Nizamul
Computer Science and Engineering
SRM Institute of Science and Technology
Chennai, India

Y. Nanda Sai Srivatsav
Computer Science and Engineering
SRM Institute of Science and Technology
Chennai, India

Prof. T. Malathi
Computer Science and Engineering
SRM Institute of Science and Technology
Chennai, India

***

Abstract— In this paper, we propose an efficient *CycleGAN synthesis which takes the input of a semantic image, such as poses of a human, segmented masks, etc and converts it to a photorealistic image. Our model achieves this synthesis via a novel network weight generation module (Keras) mechanism. We conduct validations of the output with comparison to several large-scale image datasets including horse and zebra images, and street-scene images. These small subtasks further used with each modules in the program, and then learn and train simultaneously with a final targeted image. We also use a important algorithm called discriminator for this, as it forces the network to synthesize realistic details.*

Keywords—*cycleGAN, horse2zebra, pix2pix.*

I. INTRODUCTION

A lot of machine learning models perform operation by taking some kind of complicated inputs and produce a simple output out of it. The main objective of a GAN model is something the opposite of it. GAN takes input in a small number, for example, an array of random integers etc, and produce a complete complex output out of it like a photo realistic picture.

II. LITERATURE REVIEW

With the progress of technology over the last few years, a number of methodologies for image shaping have been proposed. In the field of this technology, numerous scientists and individuals have used different rationale and algorithms to implement their ideas. We have examined this document in our literature study to understand in detail the scene behind the image structure.

1. *Title*: Unattached image-2-image conversion using cycle-consistent systems of adversarial

Author: Jun-Yan Zhu, Taesung Park, Phillip

Year: 2018

Description: Image-2-image The aim of the conversion is to acquire a mapping between an image of the input and an image of the output using a pair object training method. However, for such operations the cumulative data on training will not be available. We give a method of learning to transform a image from source region X into the target area Y in the absence of paired instances. A number of tasks with unconnected training details, such as the transmission of the selection sort, object resurrection, season transformation, photo enhancement etc, are listed in statistical terms. Quantitative analogies are evidence of the superiority of our approach.

2. *Title 2: Image morphing using the morphological operating system and pixel segmentation.*

Author : Anupurban Mandi

Year : 2016

Description : The picture displays clearly pixels divided in this text. The segmentation of the threshold and the Watershed is very easy and common but the recent approach to this problem includes morphological operators who have been able to better detect the output image of the other 2. It is very difficult to decide the factor used in thresholding, as the factor used for one image can not work for another image. For different pictures this aspect can be different. The watershed approach is inefficient, because a runoff is formed at each minimum, and it is very responsive to local minimums. This paper proposes an algorithm based on morphology for image processing.

III. PROPOSED METHOD

The proposed algorithm uses cycleGAN. Our suggested methodology is discussed in the following article with more information.

A. GAN

At a core level this makes perfect sense, if you create a machine that produces the same picture each time it runs, it would not be very exciting. Perhaps as significant, however, is that thinking in terms of probabilities also helps one turn the question of producing images into a normal mathematical context. Mathematically, this means modeling a probability distribution on images, that is, a function that tells of images are likely to be faces and which are not. This sort of problem modeling a function in a high-dimensional space is precisely the kind of thing that neural networks are built for.

The great insights which define a GAN are the development of a form of competition as this modeling problem. It is the root of the word "adversarial." No one but two competing networks should be created, one generator and one discriminator. The generator attempts to generate a random simulated output, while the discriminator attempts to differentiate them from actual performances. The goal is to change and strengthen as the two networks approach each other with a generator network delivering practical outputs.

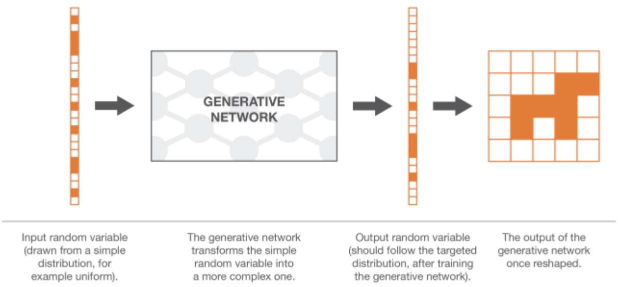

Fig. 1 Generative adversarial *method*

B. Pix2Pix

It is a generative network adversarial, or GAN, designed to generally convert photo to object. The GAN architecture includes a generating model to produce new plausible synthetic pictures and a discriminating model to distinguish real (dataset) or fake (generated) images.

The model discriminator is explicitly modified, while the model generator is modified via the model discriminator. In this way, two models are concurrently trained in an adversarial method in which the generator tries to delude the discriminator better and the discriminator tries to better recognize the false photos.

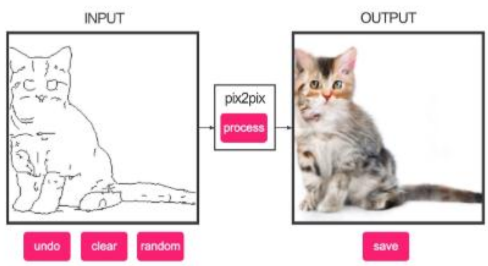

Fig. 2 Architecture of pix2pix method

C. Generator

A function that turns a random input into a synthetic output is called a Generator. The Generator takes the image and compresses into low dimensional tensor representation called as “Bottleneck”. Then the upsampling of the image is learnt by the Generator. The en-coder and de-coder architecture consist of the following:

En-coder architecture: CO64-CO128-CO256-CO512-CO512-CO512-CO512-CO512

De-coder architecture: COD512-COD512-COD512-COD512-COD512-COD256-COD128-COD64

Now, to perform a job of linking the number of output channels that is 2 in colourisation part and 3 in general part, a convolutionary method is used in the decoder, followed by a Tan_h function. To the first CO64 layer in the encoder, the Batch - Normalisation is not applied. Re-LUs in decoder are not leaky but Re-LUs in encoder, with slope 0.20, are leaky. This takes a black and white image in the feed, this goes through a variety of convolution layers. It eventually produces an image source that has the same size as the input with three colored channels. But the generator only generates random output before preparation. The volume next to them represents the tensor dimension. The input image is of the resolution 256*256 and with 3 colours, red, green and blue and they is same for greyscale. The output also has the same characteristics. The image is taken and is reduced to much smaller representation by the encoder using convolution and activation functions. This is done in a hope that we have a higher level representation of the image after the final encoder layer. Using deconvolution and activation ruction, the decoder does the opposite of the encoder to reverse the actions performed by the layers of the encoder.

Instead of the encoder decoder, a “U_Net” architecture is used in order to improve the performance of the image to image transformation. It connects the encoder layers to decoder layers directly with skip connection :

The Generator's structure is also known as “encoder decoder” and in picture -2- picture, its like :

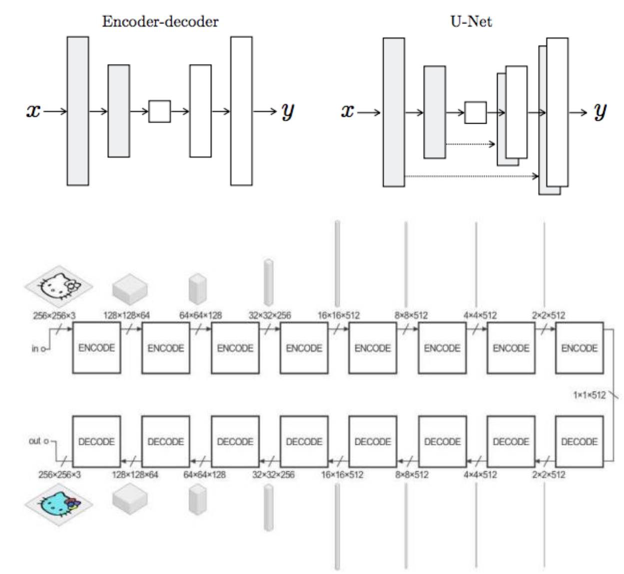

Fig. 3 The UNet Encoder Decoder

D. Discriminator

The Discriminator is responsible for taking two pictures, the input pictures and picture which is either a generator picture, or an output picture to be formed, and determining whether or not the other image is generated by the generator. The architecture of 70by70 discriminating is as given below:

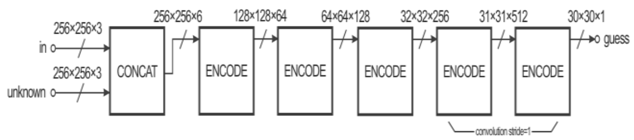

Fig. 4 Working of discriminator

1 x 1 : discriminator: *C[64]--C[128]*

16 x 16 : discriminator: *C[64]--C[128]*

Per 30-to-30 picture pixel of the pix2piximplementation is based on an assumption that the input picture has a 70-to-70 patch (patches are overlapping a lot since the entry images are 256-to-256). It is called a' PatchGAN ' architecture.

E. Discriminator

Although PIX 2 PIX is capable of delivering spectacular effects, training data are difficult. The two picture spaces in which you had to traduce had to be preformatted into a single X / Y view that comprised two closely associated images. This could take time, be inviable, or even be impractical regardless of what two forms of images try to interpret. The CycleGAN drops in here.

The fundamental concept behind CycleGANs is that the strength of the PIX2PIX architecture is used to direct the platform to two discrete unpaid image sets. For eg, Group X, an image series, is full of sunshine pictures at the beach and group Y, a selection of clouded images at the beach. The CycleGAN model will learn to convert the images into a single X / Y training image between the two cosmetics without the need to combine closely connected matches.

To order that CycleGANs are able to learn such amazing translations without clear X / Y training images, the concept of a complete translation cycle is added, to assess how effective the full translation method is such that both generators are strengthened.

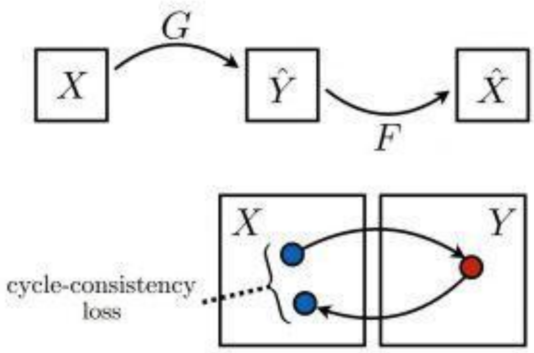

Fig. 5 Feedback mechanism of cyclegan(Discriminator)

F. Loss Function

CycleGAN does not provide combined data to practice on, and no promises are made that inputx and goal y are important during workouts. For implement this, the cycle constancy loss is the correct mapping. The ability of Cycle GANs is to set up the loss function and use the whole cycle loss as an additional optimization goal.

The length of the loop implies that the output will be near the initial value. When, for example, a phrase is translated from English into Japanese and then returned from Japanese into English, the resulting phrase will be the same as the initial phrase.

The cyclic loss from one generator will be preserved in the same image space by using a separate generator, reducing the gap between the original image and the cyclic image. In loop continuity loss:

Image X goes through the generator G, which gives image Y'.

Y ' image is generated via F generator, yielding X ' image cycled.

Among X and X' is determined the mean absolute error.

Continuity loss of loop: X— > G(X)— > F(G(X))~X

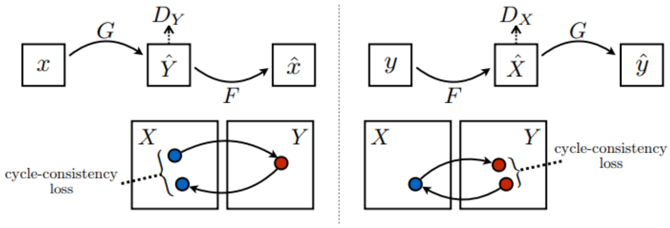

Fig. 4 Returning process continuity loss

Engine G transforms image X into image Y engine. Identity failure notes that if image Y is fed to generator G, the original image Y or anything similar image Y will be fed.

IV. CONCLUSION

In this project, we propose cycleGAN over existing system PIX2PIX. It also refines the morphological output and produces more photorealistic images than pix2pix. When expressed in the measurements and the diagrams, the network efficiency after this is slightly increased. Nevertheless, only a slight benefit makes a big difference in this rapidly rising nation. The picture forming environment is only one year old and thus offers the basis for more study. CycleGAN

V. ACKNOWLEDGEMENT

We found that the ouput tests were prone to initialization during the preparation.

We think that this model is also not ideal for modifying the shape of the target. This effect can only be detected after 10-20 cycles. We attempted to adopt the pattern of transforming a male face into a female one. We used celebA dataset for this, but the findings are not strong and the generated images are very blurred.

Image-to-Image conversion entails controlled picture alteration which includes vast quantities of paired pictures which are difficult or not available to plan.

CycleGAN is a methodology for the training via the GAN system of unexposed image collections from two separate realms of unattached image transfer models.

A variety of applications including seasonal conversion, turning objects, design transfer, and creating pictures of paintings have been demonstrated in CycleGAN.

Also we would like to express our gratitude to our guide, Prof T. Malathi Ma'm, who guided us throughout the project and offered a deep insight into the study. We wish to acknowledge the help provided by the technical and support staff in computer science department of SRM institute of science and technology.

VI. REFERENCES

[1] Andreas et al. learning to compose neural networks for question answering. arXiv preprint arXiv:1601.01705, 2016

[2] Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 2650–2658, 2015.

[3] Gatys et al. Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks. arXiv preprint arXiv:1505.07376, 12, 2015.

[4] Goodfellow et al. Generative adversarial nets. In Advances in neural information processing systems (NIPS), pages 2672–2680, 2014.

[5] Y. Guc¨¸lut¨ urk et al. Convolutional sketch inversion. In European Conference on Computer Vision (ECCV), pages 810– 824. Springer, 2016.

[6] P. Isola et al. Image-to-image translation with conditional adversarial networks. arXiv preprint, 2017.

[7] M. Jaderberg et al. Spatial transformer networks. In Advances in neural information processing systems (NIPS), pages 2017–2025, 2015.

[8] D. Ji et al. Deep view morphing. Computer Vision and Pattern Recognition (CVPR), 2017.

[9] J. Johnson et al. Inferring and executing programs for visual reasoning. arXiv preprint arXiv:1705.03633, 2017.

[10] T. D. Kulkarni et al. Deep convolutional inverse graphics network. In Advances in Neural Information Processing Systems (NIPS), pages 2539–2547, 2015.