Sketch to Photo Conversion using Cycle-Consistent Adversarial Networks

Kommalapati Abhiroop Tejomay, Kishore Kumar Kamarajugadda

***Abstract*: It is proposed to use the cycle-consistent adversarial network as a way to convert images of sketches to images of photos. The network learns to perform the mapping from the domain of sketches to the domain of photos and due to its architecture also learns the inverse mapping from the domain of photos to the domain of sketches. The network converts sketches to photos by reducing a weighted sum of the validity, reconstruction and identity losses. The advantage of using a cycle-consistent adversarial network over other network architectures is that it is not mandatory to have aligned image pairs as its training set and works in an unpaired setting.**

***Keywords:* Cycle-Consistent Adversarial Networks, Image-to-Image Translation, Generative Adversarial Networks, Generator and Discriminator.**

I. INTRODUCTION

Deep Learning research today has advanced significantly ever since its early days in the 1980s. From early perceptrons that could be used to implement simple boolean logics such as AND, OR and NOT, today's deep learning algorithms are quite adept at performing complex tasks such as image classification, text and music generation, facial recognition etc. Examples of such algorithms are convolutional neural networks [1, 2], recurrent neural networks and the BERT [3] algorithm. The ILSVRC challenge, more popularly known as the ImageNet challenge gave rise to several state of the art architectures like ResNet [4], SENet [5] which are variants of the base convolutional neural networks. These algorithms perform so well that many consider the ImageNet challenge a solved problem.

But the last 3 to 5 years have seen a significant rise in applications of deep learning to generative modeling tasks. A generative model involves generating examples that are almost indistinguishable from the instances in the original training set.

A significant advancement in the generative modeling field was the introduction of the generative adversarial network [6] at the 2016 NIPS Conference where the ideas presented are now considered fundamental turning points for generative modeling. Ever since then, several variations of generative adversarial networks have been proposed which has propelled the field to great heights. A generative adversarial network is comprised of two neural networks, one being the generator and the other, the discriminator. The generator tries to synthesize or generate examples that bear similarities to the training set instances where as the discriminator tries to distinguish where a given instance is real, i.e., from the original training set, or fake, one of the generator's forgeries. The key point to note here is that the generator generates these example from random noise. The generator and the discriminator battle against each other and at the end of training, the generator becomes so good at generating examples that closely resemble that of the original dataset that the discriminator is unable to distinguish between real and fake images anymore. Essentially, we need to find the probability density function that mimics that of the training dataset and sample data from it. Figure 1 shows the architecture of a generative adversarial network.

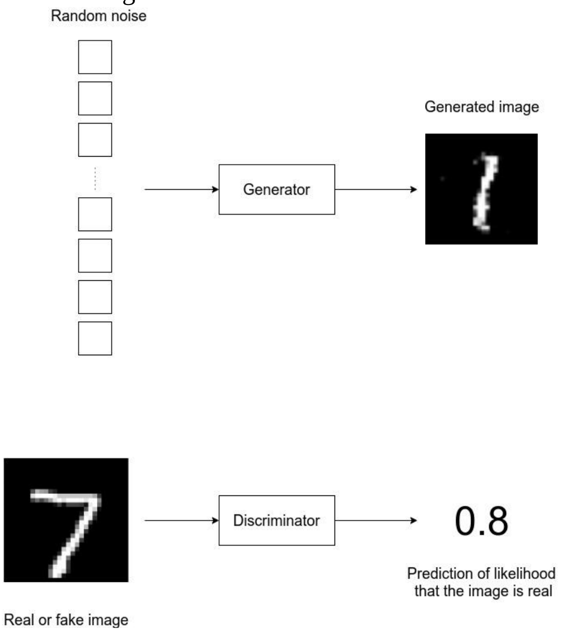

Fig. 1. Generator and Discriminator inputs and outputs in a GAN

One of the interesting problems in generative modeling is Image-to-Image translation. It involves trying to learn a mapping between input and output images using a training set of image pairs that are aligned.

Revised Manuscript Received on February 28, 2020.
* Correspondence Author

Kommalapati Abhiroop Tejomay*, Department of ECE, Faculty of Science and Technology, ICFAI Foundation for Higher Education, Hyderabad, India, abhirooptejomay@gmail.com

Kishore Kumar Kamarajugadda, Department of ECE, Faculty of Science and Technology, ICFAI Foundation for Higher Education, Hyderabad, India, kkishore@ifheindia.org

We try to learn a mapping from *X* to *Y*, i.e., *F: X* → *Y* where images from the distribution *F (X)* are so similar to that of *Y* so much so that they are indistinguishable from one another. This is done by reducing an adversarial loss. Examples of some applications of Image-to-image translation are converting paintings to photos, black-and-white images to colour images and many more. A major drawback of early techniques to solve the Image-to-Image translation problem is that they required having pairs of training set images consisting of both inputs and outputs. Such a task is also called Paired Image-to-image translation. But gathering input images along with their corresponding output image is difficult and is usually not readily available. The Cycle-Consistent Generative Adversarial Network (CycleGAN) [7] alleviates this problem by achieving unpaired image-to-image translation. The CycleGAN network also tries to learn a mapping *F : X* → *Y* but due to their nature, they also provide an inverse mapping *G : X* → *Y* and introduce a cycle-consistency loss such that *G(F (X)) ≈ X*. There is no requirement to have pairs of images in this case. We propose to use this network to learn a mapping between images of sketches and photos to convert sketches to photos. For our problem, we use sketches of human faces.

II. LITERATURE SURVEY

A. Generative Adversarial Network

The fundamental idea of GANs is establishing a game where the generator and the discriminator act as two players. The generator tries to synthesize samples that looks to have come from the probability distribution of the original training data. The discriminator distinguishes between real and fake images. The generator takes in noise as its input and generates examples. By reducing an adversarial loss, the images synthesized by the generator are virtually indistinguishable from the original dataset. The discriminator loss in a traditional GAN is given below:

$$J_D\left(G, D\right) = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log(1 - D(G(z))) \qquad (1)$$

Here J_D is the cost function or loss of the discriminator taking as the discriminator and generator parameters as input. □$_x$ represents real instances where as □$_y$ represents the domain of random noise vectors fed to the generator. This loss is almost identical to the typical cross-entropy loss used when training a sigmoidal binary classifier.

The generator loss is just the negative of the discriminator loss, given by:

$$J_G = -J_D \qquad (2)$$

B. Image-to-Image Translation

Image-to-Image translation is the task of learning a mapping between input images and output images. Prior techniques like Conditional GANs [8] required input-output pairs. Conditional GANs learn a conditional generative model instead of just a generative model, work well for tasks involving image-to-image translation. Traditional GANs learn a mapping between a random noise vector *z* and output image *y*, that is, *G: z* → *y* where as a conditional GAN's objective is to try to learn a mapping *G: {x, z}* → *y*. Here *x* is the input image and *z* is the random noise vector and *y* is the output image. The conditional GAN loss varies from the traditional GAN loss slightly:

$$\mathcal{L}_{condGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z))] \qquad (3)$$

The difference here is that a condition is applied on the generator to convert images from domain of *x* to domain of *y* (the generator is fed both *x* and *z* as input).

C. Unpaired Image-to-Image Translation

There are a number of other solutions to the Unpaired Image-to-Image translation problem like CoGANs [8] and cross modal networks [9] which use a strategy of sharing weights to find the representation across domains. But the drawback of the above networks is that the loss used is not general-purpose and hence cannot be used for a wider range of vision tasks.

D. Neural Style Transfer

Image-to-Image translation is also possible using Neural Style Transfer [10], but it does so without the need for a training dataset. It instead transfers the style of one image to another image. It achieves this by reducing a loss function comprising of three parts: The content loss, the style loss and the variance loss. By reducing the content loss, we make sure that the translated image has the same content as the base image. That is, it preserves the high level features of the base image. By reducing the style loss, we make sure that the style of the style image gets transferred to the base image to form the translated image. The variance loss makes sure that the translated image is smooth instead of pixelated. To compute the style loss, the algorithm computes a gram matrix for a set of layers throughout the network for the base and translated images and compares their similarity using sum of squared errors.

III. DATASET

The dataset used is the CUHK Face Sketch Database [11]. It consists of 188 images of human faces along with their corresponding sketches. The faces are of students belonging to the Chinese University of Hong Kong (CUHK). A sketch is drawn for each face and the photos were taken under normal lighting conditions with a frontal only pose and neutral expressions. Figure 2 shows examples of sketches and photos taken from the CUHK face sketch database.

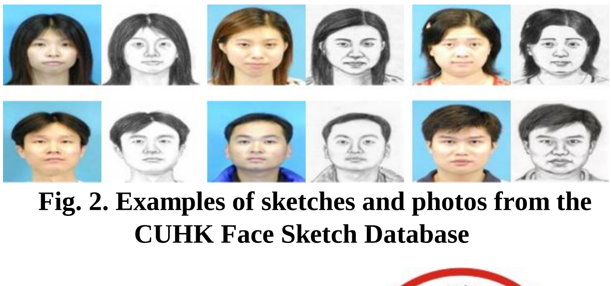

Fig. 2. Examples of sketches and photos from the CUHK Face Sketch Database

The training set consists of 88 images and the remaining 100 images comprise the test set.

IV. PROPOSED METHOD

For the problem, we used Cycle-Consistent Adversarial Networks, more popularly known as CycleGANs. A CycleGAN is actually comprised of four models, two generators and two discriminators. Considering two domains *X* and *Y*, in our case, *X* being the sketch domain and *Y* being the photo domain, the first generator G_{xy} converts images from the sketch domain, *X*, to the photo domain, *Y*. The second generator G_{yx} converts images from the photo domain, *Y*, to the sketch domain, *X*. This is a consequence of the working of a CycleGAN. For our problem, we are more interested in the mapping from the sketch domain to the photo domain, i.e., $G: X \rightarrow Y$.

The two discriminators will determine whether the images produced by the generators are convincing. The first discriminator D_x is trained to tell apart real images of sketches (domain X) and fake images reproduced by the generator G_{yx} . Similarly the second discriminator D_y is trained to tell apart real images of the photo domain, *Y*, and fake images reproduced by the generator G_{xy} . The relationship between the two generators G_{xy} , G_{yx} and two discriminators D_x and D_y is shown in Figure 3:

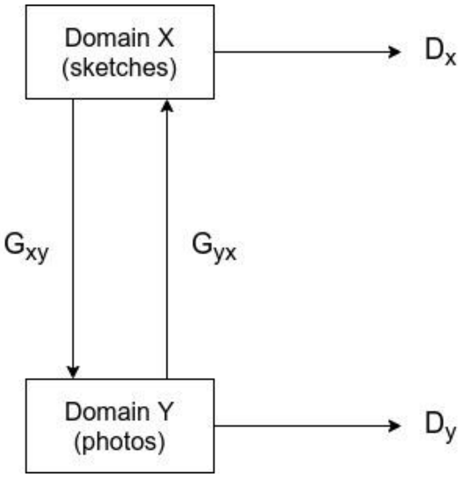

Fig. 3. Relationship between the two generators and discriminators in a CycleGAN

A. The Generator

To implement the generator, we used the U-Net architecture, consisting of two halves. The downsampling and upsampling half. There are also skip connections implemented between the downsampling layers with their corresponding up sampling layers (downsampling layers are connected to those up sampling layers that have identical shape). As the image gets convoluted through the down sampling half, it loses spatial resolution but learns the content of the image. The lost spatial resolution is gained back when the image convolutes through the up sampling half. Skip connections are placed so that the high level abstract information captured by the down sampling process combines with the spatial information fed back from the previous layers during the upsampling process. Figure 4 illustrates the U-Net architecture implemented for the generator.

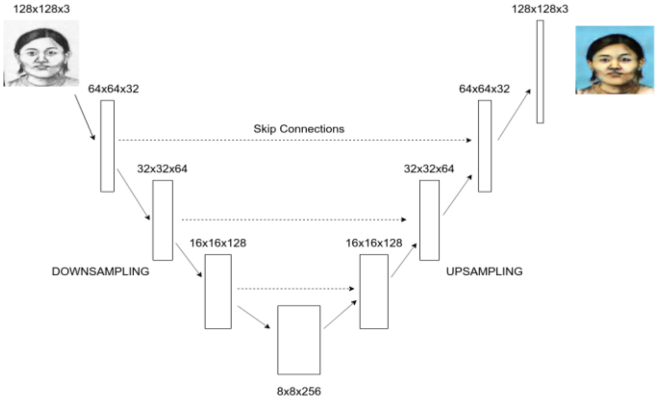

Fig. 4. U-Net Architecture Diagram

CycleGANs use Instance Normalization [12] layers instead of Batch Normalization [13] layers as in style transfer problems, they lead to more satisfactory results.

B. The Discriminator

The discriminator resembles the architecture of a PatchGAN discriminator and outputs a 16x16 single channel tensor instead of a single number. The PatchGAN splits the image into overlapping patches and gives a prediction for each patch, instead of giving a prediction for the image as a whole. A point to note is that the patches are predicted simultaneously as an image is passed through the network. The division into patches occurs naturally as a consequence of the convolution operation and is not done manually.

The advantage of using a PatchGAN discriminator is that the loss function would use the style of the image instead of its content. As each patch of the image would only be a small square in the whole image, each element of the discriminator would need to use the style of the image to make a decision. This is our requirement as we need to discriminate images based on style rather than content. Figure 5 shows the architecture of the PatchGAN implemented.

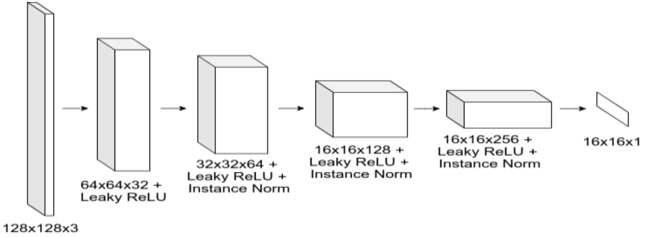

Fig. 5. PatchGAN Architecture

V. TRAINING AND EVALUATION

The two discriminators are compiled directly as the discriminator just takes an input image gives out a binary output (1 if the image is real, 0 if it is a generated fake). The loss used for the discriminator is the mean squared error. The generator on the other hand, cannot be compiled directly as we consider the unpaired setting. The generators are compiled and evaluated based on three criteria:

Validity - Whether a generator fools its relevant discriminator. This is the adversarial loss used, given by equation 4, but for stability during training, the negative log-likelihood function in equation 4 is replaced with least square errors, given by equation 5.

$$\mathcal{L}_{\mathrm{GAN}}(G, D_Y, X, Y) = \mathbb{E}_y[\log D_Y(y)] + \mathbb{E}_x[\log(1 - D_Y(G(x)))] \quad (4)$$

$$\mathcal{L}_{\mathrm{GAN}}(G, D_Y, X, Y) = \mathbb{E}_y[D_Y(y) - 1]^2 + \mathbb{E}_x[D_Y(G(x)) - 1]^2 \quad (5)$$

Reconstruction - Whether the original image can be generated by applying two generators one after the other. This is also known as the cycle consistency loss given by equation 6. We can observe that both the forward and backward cycle consistency losses are included in equation 6. The forward cycle consistency loss is computed for the mapping *F(G(x))* so that *F(G(x)) ≈ x.* The backward cycle consistency loss is computed for the mapping *G(F(y))* so that *G(F (y)) ≈ y.*

$$\mathcal{L}_{\mathrm{cyc}}(G, F) = \mathbb{E}_x[\|F(G(x)) - x\|_1] + \mathbb{E}_y[\|G(F(y)) - y\|_1] \quad (6)$$

Identity - Whether the image remains unchanged when the image is from the generator's own target domain. Identity loss helps in preventing changes in background colours.

A.Training Details

The validity loss for both the generators are implemented using mean squared error whereas mean absolute errors are used for reconstruction and identity losses of the generators. The final loss is the weighted sum of the validity, reconstruction and identity losses. The weights used for validity, reconstruction and identity losses are 1, 10 and 2 respectively. The learning rate we used during was 0.0002. The skip connections in the U-Net of the generator were implemented using concatenate layers. The model was trained for 200 epochs with a batch size of 1, using a Tesla K80 instance.

VI. RESULTS

Figure 6 shows the output images reproduced by the network after training the first batch of the first epoch. The corresponding translated (validity), reconstructed and identity loss image outputs are shown. The network does not perform well just yet but still preserves the content of the sketch image like the eyes, nose and mouth.

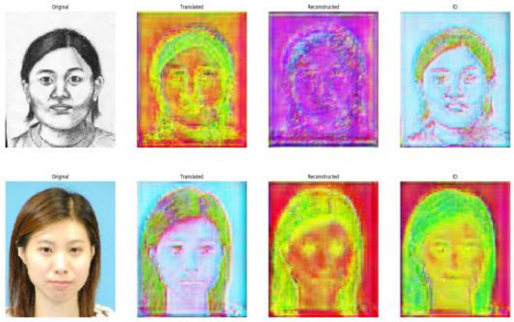

Fig. 6. Output images generated after the training the first batch of the first epoch

Figure 7 shows the output images generated by the network after the training has been completed. It can observed that the sketch to photo conversion is quite convincing. The colours of the skin, hair and eyes are correctly translated in the output image. There is no observable background colour bleed into the face area. The reconstructed image also very closely resembles the original sketch image and identity image is practically unchanged from the original image, which is desired. The same can be observed in the backward cycle, i.e., photo to sketch conversion. But some artifacts can be seen in the sketch generated from the original photo. Although the reconstructed and identity images are significantly similar to the original photograph. Figure 8 shows the graph between validity, reconstruction and identity losses against the number of batches trained. We can observe that the losses gradually decrease as the number of epochs increase.

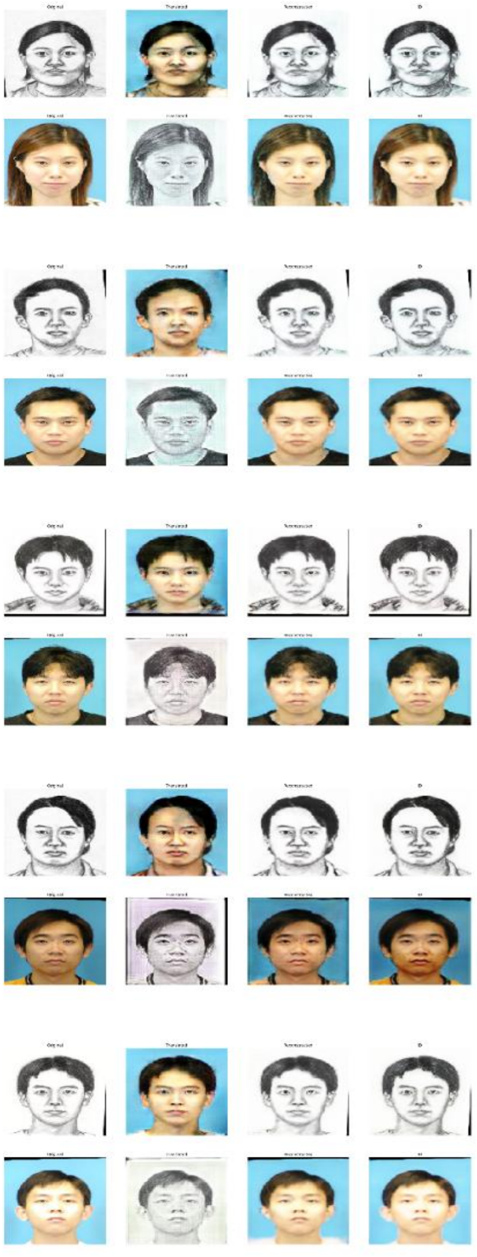

Fig 7. Output images generated after training is completed

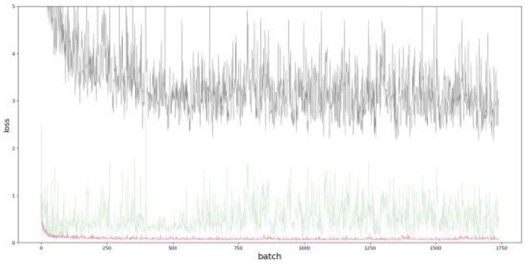

Fig 8. Validity, Reconstruction and Identity Losses vs Number of Batches

VII.CONCLUSION

The above results suggest the problem of converting sketches to photos, when considered as a task of Image-to-Image translation can be solved relatively well using Cycle-consistent adversarial networks. The CycleGAN converts the sketch to a photo by training 4 models (2 generators and 2 discriminators) and finds a mapping between two domains, as well as an inverse mapping between them. During training, the network reduces a final loss which is a weighted sum of the validity, reconstruction and identity losses. The conversion of sketches to photos can be quite advantageous in fields such investigation of suspects of crime and in digital art.

VIII.REFERENCES

1. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition.
2. Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. NIPS.
3. Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL-HLT.
4. Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2016). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. AAAI.
5. Hu, J., Shen, L., & Sun, G. (2017). Squeeze-and-Excitation Networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7132-7141.
6. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., & Bengio, Y. (2014). Generative Adversarial Nets. NIPS.
7. Zhu, J., Park, T., Isola, P., & Efros, A. (2017). “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks”. ArXiv.
8. Isola, P., Zhu, J., Zhou, T., & Efros, A.A. (2016). Image-to-Image Translation with Conditional Adversarial Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 5967-5976.
9. Liu, M., & Tuzel, O. (2016). Coupled Generative Adversarial Networks. NIPS.
10. Gatys, L.A., Ecker, A.S., & Bethge, M. (2016). Image Style Transfer Using Convolutional Neural Networks. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2414-2423.
11. X. Wang and X. Tang, “Face Photo-Sketch Synthesis and Recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol. 31, 2009.
12. Ulyanov, D., Vedaldi, A., & Lempitsky, V.S. (2016). Instance Normalization: The Missing Ingredient for Fast Stylization. ArXiv, abs/1607.08022.
13. Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. ArXiv, abs/1502.03167.

AUTHORS PROFILE

Kommalapati Abhiroop Tejomay, graduated from ICFAI Foundation of Higher Education with a Bachelor of Technology degree in the field of Electronics and Communication Engineering. He is a student member of IEEE since december 2018.

Kamarajugadda Kishore Kumar is associated with Department of ECE, Faculty of Science and Technology, ICFAI Foundation of Higher Education, Hyderabad, India.