Face2sketch using Generative Attentional Net

1 Introduction

Image-to-image translation is a class of problem where the objective of the generator is to learn the mapping between input image and output using target images to train. Generative Adversarial Network (GANs) (Goodfellow u.a., 2014) has been extensively used in machine learning community to solve image-to-image translation problem such as style transfer in arts, object transfiguration, season transfer, etc. Despite recent advances, all the previous known method only works well if there are no significant shape differences between the target and input image. A recently released architecture called Generative Attentional Net (Zhu u.a., 2017) was proposed to tackle this shortcoming by using attention modules.

The problem that we try to solve using Generative Attentional Net (U-GAT-IT) is to automate the creation of face sketches mimicking an artist's style. We also propose task specific improvement of the loss function for training that further improve the performance of the network compared to the original loss function and results obtained using CycleGAN.

2 Approach

2.1 Model/Architecture

Generative Attentional Net is based on the commonly used GANs. Generative Attentional Net is also similar to CycleGAN (Kim u.a., 2019), which consists of two generators $G_{face \rightarrow sketch}$, $G_{sketch \rightarrow face}$, and two discriminators D_{face}, D_{sketch}.

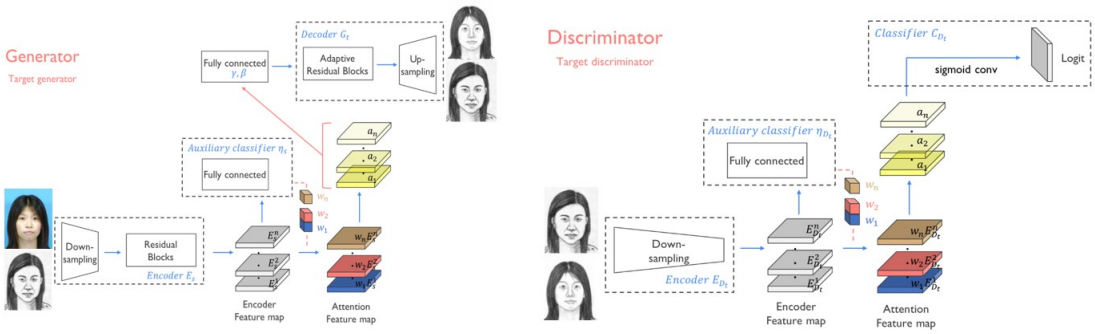

Figure 1: The U-GAT-IT network.

2.2 Major difference from conventional GAN: the Attention Module

The attentional module is what differentiates the attentional network from other GANs proposed for unsupervised image-to-image translation learning. The generators and discriminators both have an auxiliary classifier. The auxiliary classifier is assigns weights to the encoder output (we can think of these as image features), based on how close they are to source or target and figure out where improvement is needed. These weights are then multiplied with the corresponding features so the later part of the network knows which features to pay attention to. These auxiliary classifiers are trained using Class Activation Mapping (CAM) loss (discussed later).

2.3 Loss Function

2.3.1 Discriminator loss

The loss for both discriminators is simply binary cross entropy for the predictions made. Furthermore, CAM loss is used for the predictions made by the auxiliary classifier. The weights used for both losses are equal.

2.3.2 Generator loss

The Generator is trained using the following total loss function:

$$\lambda_1 L_{gan} + \lambda_2 L_{cycle} + \lambda_3 L_{identity} + \lambda_4 L_{cam} + \lambda_5 L_{edge} \tag{1}$$

A more detailed explanation of each term is as follows.

Adversarial loss

$$L_{gan}^{s \to t} = \mathbb{E}_{x \sim X_s}[(1 - D_t(G_{s \to t}(x)))^2] \tag{2}$$

This term is similar to the loss term used in vanilla GANs. It uses predictions made by the discriminator to help the generator match the distribution of the output images to the distribution of the target images.

Cycle loss

$$L_{cycle}^{s \to t} = \mathbb{E}_{x \sim X_s}[|x - G_{t \to s}(G_{s \to t}(x))|_1] \tag{3}$$

This encourages input images in one domain (X_{face}), when put through the two generators to (X_{sketch}) and back, to be the same images. In other words, a sketch generated by the face-to-sketch generator, when fed through the second generator, should return the same face. The loss is simply an L1-Loss.

Identity loss

$$L_{identity}^{s \to t} = \mathbb{E}_{x \sim X_t}[|x - G_{s \to t}(x)|_1] \tag{4}$$

The identity loss encourages generators to not make any changes to the input if the input is from the target domain. For example, if $G_{face \to sketch}$ is given a sketch as an input, it should output the same sketch without any changes. The loss is again, an L1-Loss.

CAM loss

$$L_{cam}^{s \to t} = -(\mathbb{E}_{x \sim X_s}[log(\eta_s(x))] + \mathbb{E}_{x \sim X_t}[log(1 - \eta_s(x))]) \tag{5}$$

$$L_{cam}^{D_t} = \mathbb{E}_{x \sim X_t}[(\eta_{D_t}(x))^2] + \mathbb{E}_{x \sim X_s}[log(1 - \eta_{D_t}(G_{s \to t}(x)))^2] \tag{6}$$

where η_{D_t} and η_s are the auxiliary classifiers in the discriminator and generator respectively. This is the most important loss as it is used to train the auxiliary classifier.

Edge loss This is a new loss that we introduced. The intuition was to help the generator perform better alignment and maintain any facial expressions.

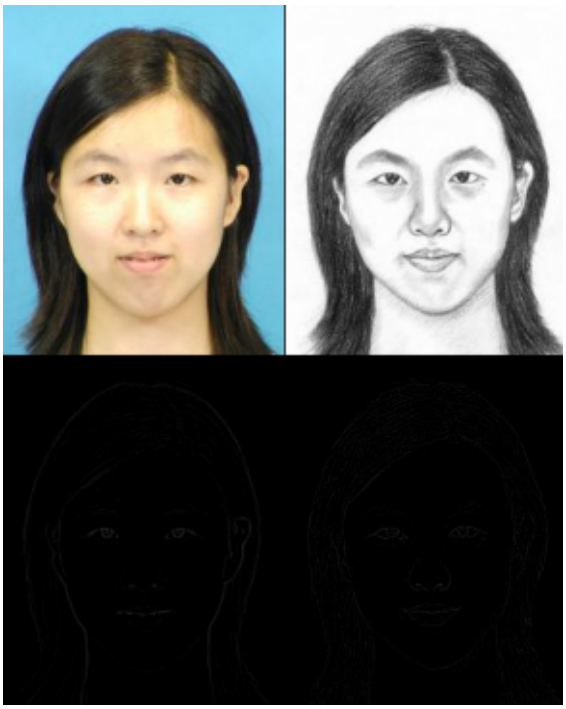

Figure 2: Top row: face and a corresponding sketch drawn by an artist. Bottom row: Edges for the face and sketch. (Best viewed by zooming in).

Figure 2 shows that edges for a face and sketch should be quite similar. The sketch however, should have some more minor edges due to pencil shading. This can be dealt with by setting the weight of the edge loss lower than the weight of the adversarial loss so as to make sure alignment is secondary to producing a realistic sketch. We detect edges using Laplacian edge detection for the input face/sketch and the corresponding sketch/face generated, and calculate the L1 loss for both generators.

$$L_{edge} = \frac{1}{N} \sum_{p \in P} |\omega * (G(x)(p)) - \omega * x(p)|$$

where $N = hw$ is the number of pixels, P are pixels, x is the input (face or sketch depending on the generator) and $G(x)$ is the generator output (sketch or face), and ω is the Laplacian edge detection filter.

2.4 Dataset

For face images, we used the Chicago Face Dataset (Ma, 2015) and the IMM Face dataset (Nordstrøm u.a., 2004). For sketches, we used the Chinese University of Hong Kong's Face Sketch Database (Wang u.a., 2009). Our training set consisted of roughly 500 unpaired face images and 500 sketches. Our test set used 150 face images.

2.5 Training

The model was trained for 400 epochs, and took approximately 48 hours to train on a GTX 1080Ti. The parameters λ_1, λ_2, λ_3 and λ_4 were set to 2, 10, 10, and 1000 respectively throughout the training. The value of λ_5 was set to 0 for the first 100 epochs and then 0.5 for the remainder of the training.

3 Experiments

We compiled our own test set, using in total roughly 150 face images across the Chicago Face dataset, IMM Face dataset, and CUHK's dataset.We compared our method with CycleGAN, using the 150 face image test set described above, to generate face sketches.

3.1 Quantitative results

To quantitatively evaluate our results, we use Structural Similarity Index (SSIM) to calculate the similarity of our output images with ground truth images. These paired images were obtained from the CUHK Face Sketch database. The formulae of SSIM is as follow:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{7}$$

In the above equation, μ_x and μ_y are the mean values of two images x and y. σ_x and σ_y are the variances of x and y, respectively; c_1 and c_2 are two variables used to stabilize the division with weak denominator. And σ_{xy} is the covariance value of two images x and y. The range of $SSIM(x, y)$ is between 0 and 1. The closer the value is to 1, the closer the generated image is to ground truth image. We calculated the average SSIM values of 150 ground truth images and generated images from Generative Attentional Net trained with proposed loss function. Our comparison of SSIM with CycleGAN and the current state-of-the-art GAN-based method (Chen u.a., 2018) is summarized in the below table.

	Generative Attentional Net (with proposed loss function)	CycleGAN	Residual net + Pseudo Sketch Feature Loss + LSGAN (Chen u.a., 2018)
SSIM	0.62	0.55	0.55

Figure 3: Structural Similarity Index comparison between Generative Attentional Net and CycleGAN

It could be observed that U-GAT-IT with our proposed loss function improvement which reinforces edge alignments outperforms CycleGAN and the method claiming to be state-of-the-art in this problem setting. Hence, we acheive state-of-the-art performance for the CUHK Face Sketch Database.

3.2 Qualitative results

3.2.1 Our model vs CycleGAN

Figure 4: Comparison of face sketch generation: (a) Test images, (b) Face sketches generated by our method, (c) Face sketches generated by CycleGAN

Visually, both CycleGAN and Generative Attentional Network output are all very similar to the ground truths. However, there seems to be a tradeoff between the two results. While CycleGAN generates a very detailed sketch, its outputs are not as realistic as that of Generative Attentional Network. On the other hand, U-GAT-IT seems to find a good balance and hence, give a more visually pleasing result.

3.2.2 Without edge loss

Figure 5: Faces generated with no edge loss function

We also implemented our model without our loss function, with results for the same 6 test faces as above. However, there seems to be a convergence issue, as the training loss for both discriminators and generators stagnated for many epochs.

4 Conclusion

Our proposed loss function with the recently released architecture U-GAT-IT produces good results and outperforms results obtained from CycleGAN. One drawback was that our network was only trained with data images that were taken under the same lighting conditions or plain background. This has caused our previous attempts to generate sketches from our face shots, which were taken with neck clothes or different lightning conditions to produce blurry sketches. For future directions, we could include more diverse set of training images to allow real-time face sketch generating application that could handle various backgrounds.

Furthermore, we wanted to introduce facial landmark alignment loss. This would be getting the 68 facial landmarks (Bulat u.a., 2017) from the input and output and encouraging the network to align them so as to preserve the shape and faacial expressions. This would be a much more reliable method than edge loss. However, due to the face and landmark detector not working on all outputs, many samples had to be discarded which made training very slow. Given more resources and time, it would be interesting to see if the landmark alignment loss can give better results.

References

Nordstrøm, M. M. / Larsen, M. / Sierakowski, J. / Stegmann, M. B.(2004): *The IMM Face Database - An Annotated Dataset of 240 Face Images*.

Wang, X. / Tang, X.(2009): *Face Photo-Sketch Synthesis and Recognition*In: IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI).

Goodfellow, Ian / Pouget Abadie, Jean / Mirza, Mehdi / Xu, Bing / Warde Farley, David / Ozair,
Sherjil / Courville, Aaron / Bengio, Yoshua(2014): *Generative adversarial nets*In: Advances in neural information processing systems2672–2680.

Ma, Wittenbrink(2015): *The Chicago Face Database: A Free Stimulus Set of Faces and Norming Data*In: Behavior Research Methods1122–1135.

Bulat, Adrian / Tzimiropoulos, Georgios(2017): *How far are we from solving the 2D & 3D Face Alignment problem? (and a dataset of 230,000 3D facial landmarks)*In: International Conference on Computer Vision.

Zhu, Jun Yan / Park, Taesung / Isola, Phillip / Efros, Alexei A(2017): *Unpaired image-to-image translation using cycle-consistent adversarial networks*In: Proceedings of the IEEE international conference on computer vision2223–2232.

Chen, Chaofeng / Liu, Wei / Tan, Xiao / Wong, Kwan-Yee K.(2018): *Semi-Supervised Learning for Face Sketch Synthesis in the Wild*In: Asian Conference on Computer Vision (ACCV).

Kim, Junho / Kim, Minjae / Kang, Hyeonwoo / Lee, Kwanghee(2019): *U-GAT-IT: Unsupervised Generative Attentional Networks with Adaptive Layer-Instance Normalization for Image-to-Image Translation*.