# Generator from Edges: Reconstruction of Facial Images

Nao Takano(✉) and Gita Alaghband

Computer Science and Engineering, University of Colorado Denver, Denver, USA
{nao.takano,gita.alaghband}@ucdenver.edu

**Abstract.** Applications that involve supervised training require paired images. Researchers of single image super-resolution (SISR) create such images by artificially generating blurry input images from the corresponding ground truth. Similarly we can create paired images with the canny edge. We propose Generator From Edges (GFE) [Fig. 1]. Our aim is to determine the best architecture for GFE, along with reviews of perceptual loss [1,2]. To this end, we conducted three experiments. First, we explored the effects of the adversarial loss often used in SISR. In particular, we uncovered that it is not an essential component to form a perceptual loss. Eliminating adversarial loss will lead to a more effective architecture from the perspective of hardware resource. It also means that considerations for the problems pertaining to generative adversarial network (GAN) [3], such as mode collapse, are not necessary. Second, we reexamined VGG loss and found that the mid-layers yield the best results. By extracting the full potential of VGG loss, the overall performance of perceptual loss improves significantly. Third, based on the findings of the first two experiments, we reevaluated the dense network to construct GFE. Using GFE as an intermediate process, reconstructing a facial image from a pencil sketch can become an easy task.

## 1 Introduction

While there have been quite a few methods and proposals for single image super-resolution (SISR), few applications exist for reconstructing an original face from the corresponding edge image. The techniques used in our Generator From Edges (GFE) are variations of those used in SISR. In SISR, there are roughly two categories. The first is by way of the perceptual loss that includes adversarial loss, which requires a generative adversarial network. The second omits adversarial loss. GFE belongs to the second category.

We focus on three perceptual losses that have large impact on the overall performance; adversarial loss, MSE loss, and VGG loss. Removing adversarial loss results in a simpler architecture, enabling us to eliminate the discriminator used in GAN. Figure 2a depicts the differences between SRGAN [2], one of the most influential works for SISR, and our proposed GFE.

In general, the larger the neural network model is, the better the outcome we expect. This is true in the accuracy of classification as well as in the synthesis of

G. Bebis et al. (Eds.): ISVC 2020, LNCS 12510, pp. 430–443, 2020.
https://doi.org/10.1007/978-3-030-64559-5_34

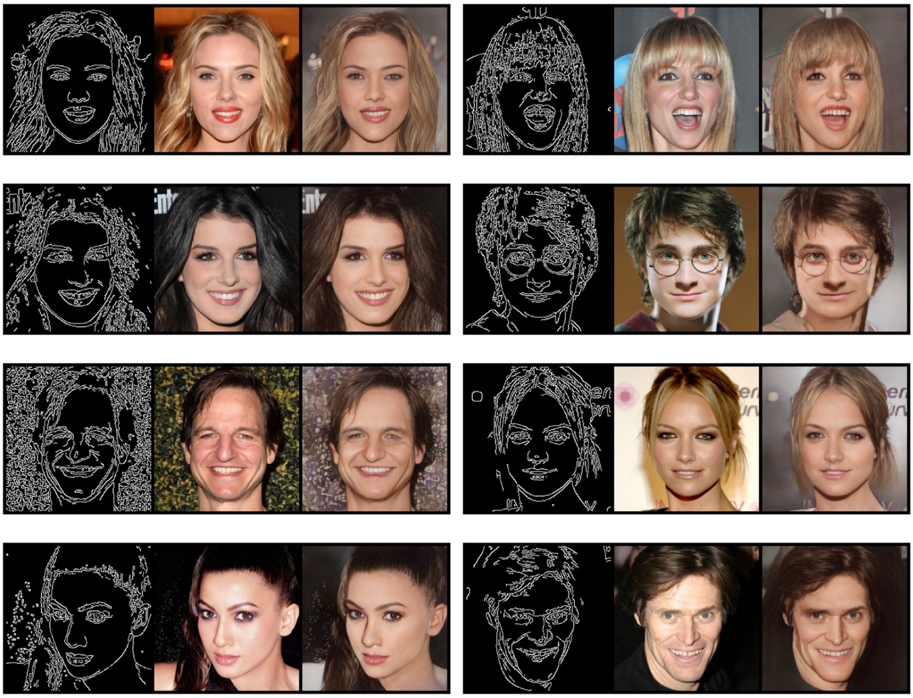
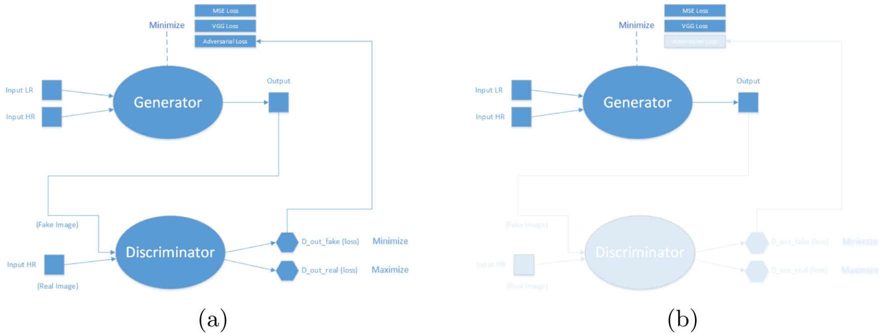


**Fig. 1.** From Left: Input, Ground Truth, Output; With CelebA-HQ [5], we used 28,998 for training and 1,000 for testing. These are sample outcomes from the testing set.



(a)

(b)

**Fig. 2.** (a): SRGAN diagram, (b): GFE, the proposed architecture – many of the elements go away images. But as the size of the input image and the complexity of the synthesis grow, we come across problems with “instabilities specific to large-scale GANs” [4] and the rate of contribution by the discriminator diminishes. In addition, the discriminator requires a large memory footprint. In order to use a large network for training with the limited amount of memory for hardware such as GPU, the question arises as to how much the discriminator contributes to the outcome of the synthesis. We observed very little, if any, positive effect using the adversarial loss for GFE. If we do away with the discriminator, we can free up its otherwise occupied GPU memory, making it possible to construct a larger generator. Moreover, using only two loss functions (image loss and VGG loss – Sect. 3) permits easier settings for hyper-parameters. We attempt to measure the effect of the discriminator and perform image synthesis without it.

This is all achieved without sacrificing the fidelity of the outputs. In the sections that follow, we present our contributions by describing details for the three experiments conducted in this study. We define Generator From Edges to be a generator (in the same sense as the one used in GAN, but without discriminator) for an application that restores images from their corresponding canny edge. All the experiments train with pairs of images; a ground truth and the corresponding single-channel (grayscale) edge image. The edge images are created by running the OpenCV Canny function from CelebA (Experiment 1) and CelebA-HQ [5] datasets (Experiments 2 and 3).

- **Experiment 1: Effect of Adversarial Loss.** Using CelebA dataset, we measured the effectiveness of adversarial loss for both SISR and the synthesis from canny images. The larger the images and more complex the task becomes, the less adversarial loss contributes to the outcome (Sect. 4).
- **Experiment 2: VGG loss.** VGG loss is another element of perceptual loss. The absence of adversarial loss leaves our perceptual loss more reliant on the VGG loss [2,6]. We used CelebA-HQ dataset with the image size 224 × 224. Using middle layers is more effective than using the last layers (Sect. 5).
- **Experiment 3: Dense Network.** In pursuit of the best quality, we investigate and propose the network architecture for GFE. The specific focus is on the effectiveness of dense connections in the network. Each residual block should have exactly one batch normalization layer, and skip connections are ineffective (Sect. 6).

## 2 Related Works

Reconstruction from edge images is a derivative of the SISR problem, thus it is imperative to consider SISR first. Since the inception of GAN [3], the technique to use a discriminator was adopted by SRGAN [2], whose influences and follow-up research [6–9] are the inspiration for our work. For human facial synthesis, whereas we generate a photo from edges (sketch), other research generates a sketch from an input photo [10–12]. Huang et al. [13] made a frontal view synthesis from profile images. Li et al. [14] as well as Jo and Park [15] generated facial images with a partial reconstruction from sketches. In terms of application, our work is related to Lu et al. [16] and Yu et al. [17]. The former uses Contextual GAN, where input photos and images are trained in semi-supervised fashion. The latter uses a Conditional CycleGAN to address the heterogeneous nature of photos and sketches. StarGAN [18] offers an impressive multi-domain image-to-image translation on human faces. Liu et al. [19] uses SR for human faces. In the general domains to generate images from sketches other than human faces, substantial advancements have been made [20–25]. More recently, Mask based image synthesis on human faces [26,27] is gaining attention. Also, Qian et al. [28] propose Additive Focal Variational Auto-encoder (AF-VAE) for facial manipulation.

Going back a few years, a wide applicability of image-to-image translation in the supervised setting was proposed by Isola et al. [29] with GAN. Our work follows this line of research; but instead of adversarial loss, we use VGG loss.

Chen et al. [30] demonstrated a similar applicability to [29] but with a Cascaded Refinement Network, which starts with a low resolution module and doubles its size for consecutive modules. For SISR, VDSR [31] and SRResNet [2] were notable architectures prior to SRGAN. Lim et al. [32] used a multi-scale model (EDSR) that enables flexible input image size, which also reduces the number of parameters. Tong et al. [33] applied dense skip connections in a very deep network to boost the SR reconstruction. Mei et al. [34] used a multi-frame network to generate more than one output and fused them into a single output. Ma, et al. [35] replaced simple skip connections with the connection nodes and proposed a multi-level aggregated network (MLAN). The research presented in these papers successfully synthesized images without using a GAN, which led us to ask ourselves: If we can create images without GAN, then how much does a GAN contribute to the outcome? If we drop it from our system altogether, what would be gained? We examine these topics in the context of the synthesis/reconstruction of an image of a human face.

## 3 Perceptual Loss Functions

Perceptual loss functions were first defined by Johnson, et al. [1] and adopted by Ledig, et al. [2]. They are per-pixel loss functions used in feed-forward image transformations. In SRGAN [2] and its variants, three loss functions are used. Empirically, none of the loss functions among the three can generate a convincing image alone. In our study, we use at least two losses in various combinations to determine if and how they contribute to the overall outcome.

- Image Loss (I): also referred to as per-pixel loss [1], or MSE loss [2]. This is a pixel-wise L2 loss between the output of the generator and the ground truth. We call it image loss in order to distinguish it from the mean squared error used in VGG loss. The resources required to calculate the image loss are the least expensive among the three. We used L2 in this paper, but it is also possible to use L1.

$$L_I(G) = \sum_{x=1}^{W}\sum_{y=1}^{H}(I_{x,y}^{target} - G(I^{Input})_{x,y})^2 \tag{1}$$

- VGG loss (V): Using a pre-trained VGG Network [36] (available in [37]) plays a crucial role for training the generator. The network has been trained with ImageNet and already knows what real-world images look like, delivering the results for object classification/identification as well as synthesis. Given $\phi$ is a VGG network, the loss function is defined to be

$$L_V(G) = \sum_{x=1}^{W}\sum_{y=1}^{H}[\phi_{i,j}(I^{target})_{x,y} - \phi_{i,j}(G(I^{Input}))_{x,y}]^2 \tag{2}$$

  where $\phi_{i,j}$ refers to the feature maps obtained from the j-th Convolution/ReLU pair before the i-th maxpooling layer within the VGG-19 network, the same notation used in [2,6].
- Adversarial Loss (A): which is calculated with the discriminator, is what makes the system a GAN. In other words, in the absence of this loss, there is no need for a discriminator, and the resulting framework is no longer designated as a generative adversarial network. The resources required for computing the adversarial loss and how impactful it is in our image reconstruction deserves attention.

$$L_A(G) = \sum_{n=1}^{N} -\log D(G(I^{Input})) \tag{3}$$

The total loss, $L$, is calculated as

$$L = \lambda_0 I + \lambda_1 V + \lambda_2 A \tag{4}$$

where $I, V, A$ represent image loss ($L_I$), VGG loss ($L_V$), and adversarial loss ($L_A$), respectively. In the actual calculation, we set $\lambda_0 = 1$, so that only two parameters $\lambda_1$ and $\lambda_2$ are considered to determine the portion of each loss influencing the computation. In Sects. 4 and 5, we examine these losses more closely.

## 4 Experiment 1. – Impact of Adversarial Loss

In the realm of supervised training, there are quite a few papers that report successful reconstruction of images without adversarial loss [30–35]. Prior to the architecture of GFE described in Sect. 6, this section analyzes the value of adversarial loss with the degree of its effect on both super-resolution (SR) and canny edge (Canny). In this experiment, we used smaller image sizes as well as a shallower network than those used in the experiments 2 and 3.

### 4.1 Architecture

The generator consists of 16 layers of residual blocks, each with 64 feature maps. This is the structure used in [2]. We used it for both SISR and image reconstruction from edges (Canny) for the experiment. The discriminator has eight convolutional layers with an increasing number of feature maps; 64-64-128-128-256-256-512-512, followed by two dense layers and a sigmoid activation function. In search of a suitable implementation we turn off VGG loss, if any, and run only in adversarial loss to see how the network converges. We selected a few implementations published in Github [38,39] among those that converge with adversarial loss only, and plugged them into our implementation so that fair comparisons can be made. The sizes of input and output images are the same; we experimented on 3 sizes – 96 × 96, 128 × 128, and 176 × 176 for both SR and Canny.

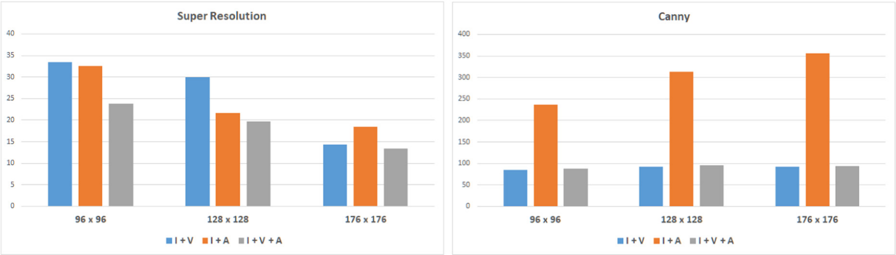
### 4.2 Methods

Since the image loss has the minimum overhead to calculate, we leave it in all three scenarios listed below. In all three cases, we set $\lambda_0 = 1$ in Eq. (4). We ran 20 epochs and took the best Fréchet Inception Distance (FID) [40,41]. FID uses a pre-trained Inception network and calculates the Fréchet distance between two multivariate Gaussian distributions with mean $\mu$ and covariance $\Sigma$,

$$\mathrm{FID}(x, g) = ||\mu_x - \mu_g||^2 + \mathrm{Tr}(\Sigma_x + \Sigma_g - 2(\Sigma_x \Sigma_g)^{1/2})$$

where $x$, $g$ are the activations of the pool_3 layer of the Inception-v3 net for real samples and generated samples, respectively.

- Image loss + VGG loss [I + V] ($\lambda_1 > 0$ and $\lambda_2 = 0$)
- Image loss + Adversarial loss [I + A] ($\lambda_1 = 0$ and $\lambda_2 > 0$)
- Image loss + VGG loss + Adv. loss [I + V + A] ($\lambda_1 > 0$ and $\lambda_2 > 0$)



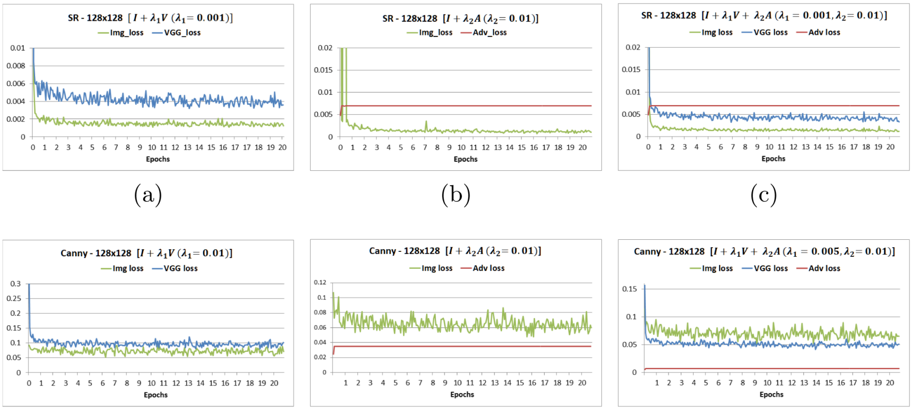
(a) (b)

**Fig. 3.** FID scores (lower is better) – combinations of losses for SR and Canny

### 4.3 Discussion

As commonly seen, the more complex the task is, the more difficult it is for the generative adversarial network to converge. For SISR, Fig. 3a clearly shows its contribution by adversarial loss to the image quality, especially in lower resolutions. However, for synthesis from canny images, a task more complex than SR, adversarial loss does not show any positive effect to the outcome. In fact, we could not successfully generate convincing images at all if VGG loss is not included (the case [I + A] in Fig. 3b). SISR is easier for image reconstruction, where adversarial loss can be incorporated into a part of the perceptual loss more naturally, than the reconstruction from canny edges.

We recorded the loss values as the training continued at each epoch. Figure 4 shows sample loss values over the course of training for the size of 128 × 128 of Figs. 3a and 3b. While image loss and VGG loss show a typical, oscillating yet steady decrease in values, adversarial loss converges rather quickly to a constant value after several hundred iterations. This raises a few interesting theoretical points: First, if we knew the constant value in advance, we could use it in lieu of the adversarial loss and save computer resources. Second, if we could come up with a method to decrease the adversarial loss throughout the training, we could take full advantage of the power of the generative adversarial network. For now, however, these are left for future research, and we conclude that adversarial loss does not contribute to the synthesis of images from canny edge, and that the resource is better used for a larger generator. Consequently at this point, GAN is not used in our study. Unless otherwise noted, the remainder of this paper uses only image loss and VGG loss.



(a) (b) (c)

(d) (e) (f)

**Fig. 4.** Loss values by epochs from Fig. 3a [128 × 128] (a) I+V (b) I+A (c) I+V+A, and Loss values from Fig. 3b [128 × 128] (d) I+V (e) I+A (f) I+V+A

# 5 Experiment 2. – Optimizing VGG Loss

VGG-19 consists of 16 Convolution layers, each followed by ReLU activation. Between the last (16th) ReLU layer and the output softmax layer, there are three fully connected layers, which are not used for perceptual loss. For perceptual loss, both aforementioned papers used the last pair (16th layer, $\phi_{5,4}$); while Ledig et al. [2] used the activation layer, Wang et al. [6] claims it is more effective to use the convolutional layer before the activation, which we confirm to be true. In this experiment, we further analyze using VGG loss computed from various layers and recommend an optimized VGG loss for our image reconstruction application.

## 5.1 Architecture and Dataset

We used VGG-19 along with image loss as part of the perceptual loss in the generator (GFE). As we established in Sect. 4.3, adversarial loss is not used and we can eliminate the discriminator. The architecture of the generator is the same as the one used in Sect. 4, but with the CelebA-HQ dataset—it consists of 30,000 high-resolution images with the size 1024 × 1024 (We resize them to 224 × 224). Removing 2 outliers (imgHQ00070 and imgHQ02815), and setting aside 1,000 images for validation/testing, we have 28,998 images for training.

## 5.2 Result

Contrary to common usage of how VGG loss is applied, our study shows using middle layers is more effective than using later layers, either the convolutional layer or the activation layer. Discarding the later layers also saves the memory space in the hardware. Fig. 5 shows that the convolutional layers of $\phi_{4,2}$ and $\phi_{4,3}$, (10th and 11th convolutional layers, respectively) show the best FID scores (lower is better – Yellow bars are convolutional, Blue is ReLU, and Black is Pool Layer).

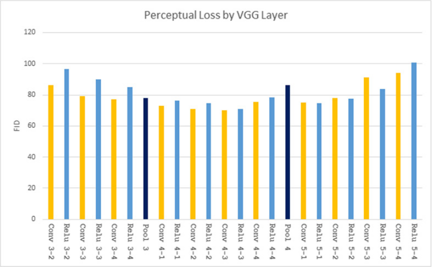


**Fig. 5.** VGG loss – layer by layer analysis

## 5.3 Multiple Layers of VGG Loss

More than one layer can be used as part of perceptual loss. Without the assistance of adversarial loss, we have $\lambda_2 = 0$, and assuming $\lambda_0 = 1$, Eq. (4) becomes

$$L = I + \lambda_{1_1} V + \lambda_{1_2} V + \cdots + \lambda_{1_n} V \tag{5}$$

where $n$ is the number of VGG layers to be used for perceptual loss. Starting with Conv $\phi_{4,3}$, we selected the best 4 layers and added new layers one by one (see Fig. 5 for reference to layers).

Figure 6 shows a sample of FID scores for $n = \{1, 2, 3, 4\}$ in our experiments with 16 block layers of generator ([A] Conv4-1; [B] Conv4-2; [C] Conv4-3; [D] ReLu4-3). Using two layers ($n = 2$) is better than using a single layer ($n = 1$), and $n = 3$ is better than $n = 2$. But for $n > 3$, the effect of adding extra layers diminishes. Most of our experiments in Sect. 6 use 2 layers of Conv $\phi_{4,2}$ and Conv $\phi_{4,3}$.

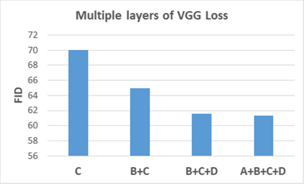


**Fig. 6.** VGG loss – using multiple layers

## 6 Experiment 3. – Generator from Edges

We form GFE based on the results obtained from Sects. 4 and 5. Increasing the size of the network is effective up to a certain point due to the vanishing gradient problem, and residual blocks along with skip connections are notable solutions for large networks [42]. In SISR, making the residual block denser (more connections within the block), as well as having more skip connections between blocks is reported to improve performance. For GFE, however, dense networks are not an effective solution when making the network larger.

In this section, we describe the experiments in pursuit of the best architecture for GFE with the network in a monolithic structure, which has a constant number of feature maps (64) throughout the generator, adapted by SRGAN [2] and other models for image generation.

### 6.1 Architecture

By using a fixed number of feature maps at every block layer, we can focus on the study of structures in the residual block and skip connections for a large network. The number of feature maps at each layer is 64, and the kernel sizes are all $3 \times 3$. Starting with 16, we increase the number of block layers at increments of 8. Without the discriminator, we have more memory available for the construction of GFE. All experiments were conducted in a single GPU with 11GB of memory, and it is worth noting that the image size we generate ($224 \times 224$) is mainly determined by the capacity of the GPU memory for training. The same dataset as Sect. 5 (Experiment 2) is used. Also, we used L2 (MSE) loss for perceptual loss calculation throughout our study. Despite certain claims that L1 loss gives a better result, in our experiments FID scores are consistently better using L2.

### 6.2 Sketch to Photo

Figure 7 shows some potential practical applications with GFE. We took some pencil sketches from CUHK [43] as well as from the internet. Note that since we trained with the canny transformation, we first have to convert the sketch image to canny, and then make an inference with the trained network. Not just as a simple coloring exercise, in the output images we can see the depth of textures of human face that commonly appear in every person, which shows a great potential for an image translation from a pencil sketch to a photo.

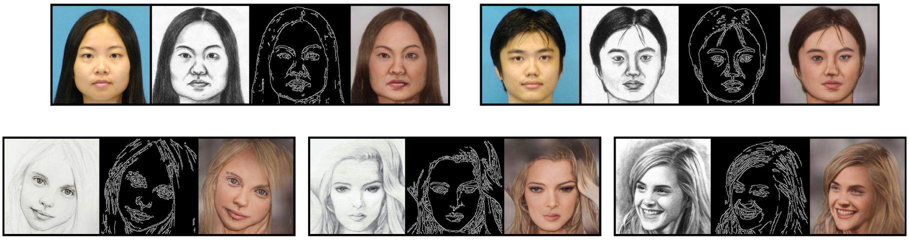


**Fig. 7.** From Left: (Top) GT, Sketch, Canny, Output (Bottom) Sketch, Canny, Output

### 6.3 Residual Block

The base unit of the construction, often called the residual block, is illustrated in Fig. 8a. The input is followed by a convolutional layer, followed by a ParametricReLU and another convolutional layer. Then a batch normalization (BN) is added before the output that is combined with the input as a single dense connection. This is very similar to SRGAN [2]; the difference being the omission of the first BN layer. This omission is crucial for reducing the memory footprint.

The batch normalization layers consume the same amount of memory as the preceding convolutional layers, and removing a BN layer from the unit block saves us approximately 20% of the memory space in our model. If we had removed both BN layers, we would have saved 40% of the memory usage [32], but our experiments show that leaving in one BN layer yields better results than none at all. Comparing a single BN layer with two BN layers, we found no noticeable differences.

Several studies in SISR propose dense residual blocks [6,33], but a generator with such dense residual units requires considerably more GPU memory, forcing us to train the network with smaller batch sizes (mini-batches). For GFE, dense residual blocks are inadequate; a large network with 32 or more block layers negatively impact the outcome.

Thus, we use one connection within the block, between the input and the BN layer. Even with just one connection at each block, when a generator is constructed by having residual blocks stacked multiple times, the entire network is connected in such a way that the gradient vanishing problem is dramatically reduced.

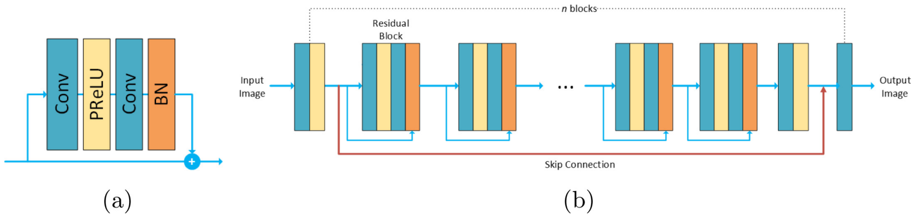


**Fig. 8.** Residual block (base unit) and skip connection type 1

## 6.4 Skip Connections and Large Networks

Let the number of basic blocks (residual blocks in the middle, and blocks for Conv + PReLU at the beginning and end of the network) be $n$ [Fig. 8b]. We define the skip connection type as follows:

- Type 0: No skip connection
- Type 1: Connect with layer 1 and layer $n - 1$
- Type 2: Connect with layer 1 and layer $n/2$, as well as $n/2$ and $n - 1$
- Type 3: Connect with layer 1 and layer $n - 1$, as well as layer $n/2$ and $n - 1$
- Type 4: Connect with Type 1 and Type 2 combined

Figure 8b shows skip connection Type 1. By going deeper in the generator, the output of synthesized images becomes better, and we found that forming 48 block layers (with each block consisting of 4 sub-layers [Fig. 8a]) achieves the best result. We tested with the above 5 skip connections to see which type is best using the residual block defined in Sect. 6.3. None of the connection types has a positive effect for our application [Fig. 9], thus, we conclude that no skip connection is necessary for our monolithic architecture of GFE.

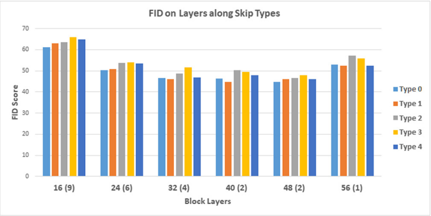


**Fig. 9.** Lower the better. (Numbers in parentheses indicate batch size)

## 6.5 The Limit of Depth in the Generator

Larger networks are not necessarily better than smaller ones. We started out our experiment with 16 layers of residual blocks, with a batch size of 9 (nine images are processed in the GPU at once in a single iteration). As we increased layers, we had to decrease the batch size due to the limitation of GPU memory. Initially the image quality improved but soon it saturated in improvement.

We observed some degradation for a network whose block size is greater than 48, where the mini-batch size needs to be 1 (one) to fit in our GPU [Fig. 9]. At this point we suspect that batch normalization is no longer in effect, and in fact the training is somewhat unstable (consecutive epochs have FID values in a wide swing). Although we attempted to tweak hyper parameters such as learning rate, we were unable to improve image quality.

## 7 Conclusion and Discussion

We demonstrated the Generator From Edges (GFE) for image translation on human faces, from edges to photo, without a generative adversarial network. This was led by the analysis of architectural features that unnecessarily consume GPU memory, such as a discriminator and extra batch normalization layers. We also reviewed a dense network and observed that skip connections are not effective if the basic unit is densely connected.

Although the trained network can restore facial images even when the edges are not drawn precisely in the input [44], the nature of supervised training commands deterministic outputs. For a practical application in mind, however, removing the GAN loses stochasticity in the inference mechanism, in which when an incomplete image is fed to the network, the outcome would also be less than ideal. This could be addressed with an unsupervised training in such a way that incomplete input leads to more convincing output. At the same time, as mentioned in Sect. 2, we are seeing rapid advancements in research—such as mask-guided (with GAN) or geometry-guided (with VAE) settings—to fill in the gap where nondeterministic outcomes are desired.

## References

1. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 694–711. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_43
2. Ledig, C., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: IEEE Conference on CVPR (2017)
3. Goodfellow, I.J.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, vol. 27 (2014)
4. Brock, A., Donahue, J., Simonyan, K.: Large Scale GAN Training for High Fidelity Natural Image Synthesis. ArXiv (2018)
5. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Stability, and Variation. ICLR, Progressive Growing of GANs for Improved Quality (2018)
6. Wang, X., et al.: Enhanced super-resolution generative adversarial networks. In: ECCV Workshops, ESRGAN (2018)
7. Vu, T., Luu, T.M., Yoo, C.D.: Perception-enhanced image super-resolution via relativistic generative adversarial networks. In: ECCV Workshops (2018)
8. Liu, Y., et al.: An attention-based approach for single image super resolution. In: 2018 24th International Conference on Pattern Recognition (ICPR) (2018)
9. Feng, R., Gu, J., Qiao, Y., Dong, C.: Suppressing model overfitting for image super-resolution networks (2019)
10. Yu, J., Shi, S., Gao, F., Tao, D., Huang, Q.: Towards realistic face photo-sketch synthesis via composition-aided GANs (2017)

11. Zhang, S., Ji, R., Hu, J., Gao, Y., Lin, C.-W.: Robust face sketch synthesis via generative adversarial fusion of priors and parametric sigmoid. In: IJCAI (2018)
12. Zhu, M., Wang, N., Gao, X., Li, J.: Deep graphical feature learning for face sketch synthesis. In: IJCAI (2017)
13. Huang, R., Zhang, S., Li, T., He, R.: Beyond face rotation: global and local perception GAN for photorealistic and identity preserving frontal view synthesis. In: 2017 IEEE International Conference on Computer Vision (ICCV) (2017)
14. Li, Y., Liu, S., Yang, J., Yang, M.-H.: Generative face completion. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
15. Jo, Y., Park, J.: SC-FEGAN: Face Editing Generative Adversarial Network with User's Sketch and Color. ArXiv (2019)
16. Lu, Y., Wu, S., Tai, Y.-W., Tang,C.-K.: Image generation from sketch constraint using contextual GAN. In: ECCV (2017)
17. Yu, S., Han, H., Shan, S., Dantcheva, A., Chen, X.: Improving face sketch recognition via adversarial sketch-photo transformation. In: 2019 14th IEEE International Conference on Automatic Face and Gesture Recognition (2019)
18. Choi, Y., Choi, M.-J., Kim, M., Ha, J.-W., Kim, S., Choo, J.: StarGAN: unified generative adversarial networks for multi-domain image-to-image translation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2017)
19. Liu, L., Wang, S., Wan, L.: Component semantic prior guided generative adversarial network for face super-resolution. IEEE Access **7**, 77027–77036 (2019)
20. Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: feature learning by inpainting. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
21. Chen, W., Hays, J.: SketchyGAN: towards diverse and realistic sketch to image synthesis. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018)
22. Wu, H., Zheng, S., Zhang, J., Huang, K.: GP-GAN: Towards realistic high-resolution image blending. In: MM 2019 (2017)
23. Zhao, Y., Price, B.L., Cohen, S., Gurari, D.: Guided image inpainting: replacing an image region by pulling content from another image. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV) (2018)
24. Yu, J., Lin, Z.L., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-Form Image Inpainting with Gated Convolution. ArXiv (2018)
25. Wang, Z., Wang, N., Shi, J., Li, J.-J., Yang, H.: Multi-instance sketch to image synthesis with progressive generative adversarial networks. IEEE Access **7**, 56683–56693 (2019)
26. Gu, S., Bao, J., Yang, H., Chen, D., Wen, F., Yuan, L.: Mask-guided portrait editing with conditional GANs. In: CVPR (2019)
27. Lee, C.-H., Liu, Z., Lingyun, W., Luo, P.: MaskGAN: Towards Diverse and Interactive Facial Image Manipulation. ArXiv (2019)
28. Qian, S., et al.: Make a face: towards arbitrary high fidelity face manipulation. In: ICCV (2019)
29. Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR (2017)
30. Chen, Q., Koltun, V.: Photographic image synthesis with cascaded refinement networks (2017)
31. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)

32. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single imawge super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2017)
33. Tong, T., Li, G., Liu, X., Gao, Q.: Image super-resolution using dense skip connections (2017)
34. Mei, K., Jiang, A., Li, J., Liu, B., Ye, J., Wang, M.: Deep residual refining based pseudo-multi-frame network for effective single image super-resolution. IET Image Process. **13**(4), 591–599 (2019)
35. Ma, C., Tan, W., Bare, B., Yan, B.: A multi-level aggregated network for image restoration. In: 2019 IEEE International Conference on Multimedia and Expo (ICME) (2019)
36. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2015)
37. https://pytorch.org/docs/stable/torchvision/models.html
38. https://github.com/david-gpu/srez
39. https://github.com/twhui/SRGAN-PyTorch
40. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In: Advances in Neural Information Processing Systems (2017)
41. https://github.com/bioinf-jku/TTUR
42. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
43. http://mmlab.ie.cuhk.edu.hk/archive/facesketch.html
44. Takano, N., Alaghband, G.: SRGAN: training dataset matters. In: International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV) (2019)