SC-FEGAN: Face Editing Generative Adversarial Network with User's Sketch and Color

Youngjoo Jo Jongyoul Park
ETRI
South Korea
{run.youngjoo,jongyoul}@etri.re.kr

Abstract

We present a novel image editing system that generates images as the user provides free-form mask, sketch and color as an input. Our system consist of a end-to-end trainable convolutional network. Contrary to the existing methods, our system wholly utilizes free-form user input with color and shape. This allows the system to respond to the user's sketch and color input, using it as a guideline to generate an image. In our particular work, we trained network with additional style loss [3] which made it possible to generate realistic results, despite large portions of the image being removed. Our proposed network architecture SC-FEGAN is well suited to generate high quality synthetic image using intuitive user inputs.

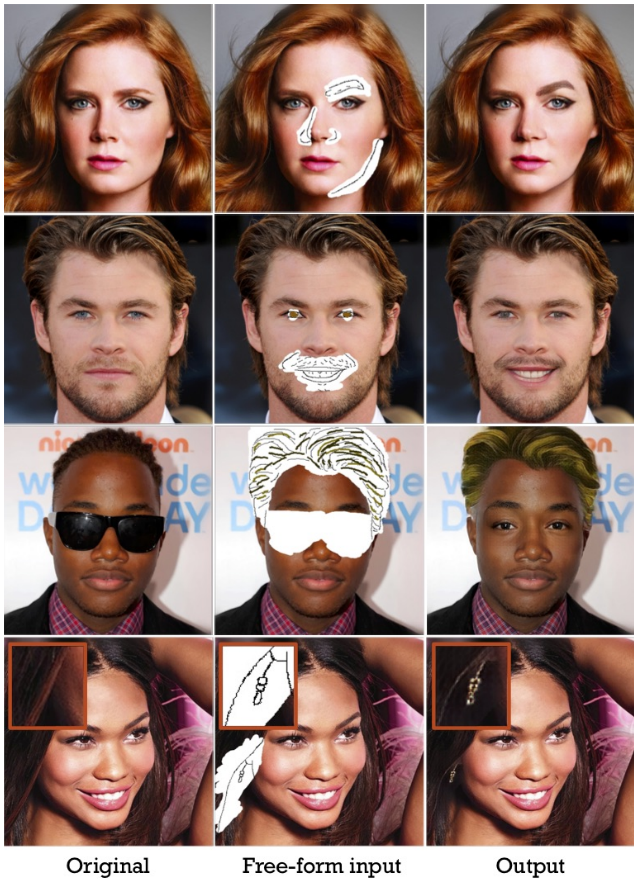


Figure 1. Face image editing results by our system. It can takes free-form input consist of mask, sketch and color. For each example, it shows that our system make users can easily edit shape and color of face even if user wants completely change hairstyle and eye (third row). Interestingly, user can edits earring by our system (fourth row).

1. Introduction

Image completion with generative adversarial networks (GANs) is a highly recognized topic in computer vision. With image exchange becoming a common medium in daily communication in the present day, there is an increase of demand for realism in generated image over a minimal image completion feature. Such demand is reflected on social media statistics. However, most image editing softwares require expertise such as knowing which specific tools to use at certain circumstances to effectively modify the image the way we want. Instead, an image completion method which responds to user input would allow novice to easily modify images as desired. Similarly, our proposed system has an ability to easily produce high quality face images, provided a sketch and color input is given even with the presence of erased parts in the image.

In recent works, deep learning based image completion methods have been used to restore erased portion of an image. The most typical method used an ordinary (square) mask, then restored the masked region with an encoder-decoder generator. Then global and local discriminator was used to make an estimation on whether the result was real or fake [5, 9]. However, this system is limited to low resolution images, and the generated image had awkward edges on of the masked regions. In addition, the synthesized images on the restored region often fell short of the users expectation as the generator was never given any user input to utilize as a guide. Several works that improved on this limitation include Deepfillv2 [17], a work that utilized user's sketch as an input, and GuidedInpating [20], which took a part of another image as an input to restore the missing parts. However, since Deepfillv2 does not use color input, the color in the synthesized image is gerated by inference from the prior distribution learned from the training data set. The Guided-Inpating used parts of other images to restore deleted regions. However it was difficult to restore in detail because such process required inferring the user's preferred reference image. Another recent work Ideepcolor [19] proposed a system that accepts color of user input as reference to create a color image for black and white images. However, the system in Ideepcolor does not allow editing of the object structures or restore deleted parts on image. In another work, a face editing system FaceShop [12] which accepts sketch and color as user input was introduced. However, FaceShop has some limitations to be used as an interactive system for synthetic image generation. Firstly, it utilized random rectangular rotable masks to erase the regions that are used in local and global discriminator. This means that local discriminator must resize the restored local patch to accept fitting input dimensions, and the resizing process would distort the information in both the erased and remaining portions of the image. As a result, the produced image would have awkward edges on the restored portion. Secondly, FaceShop would produce an unreasonable synthetic image if too much area is wiped out. Typically, when given an image with the entire hair image erased, system restores it with distorted shape.

To deal with the aforementioned limitations, we propose a SC-FEGAN with a fully-convolutional network that is capable of end-to-end training. Our proposed network uses a SN-patchGAN [17] discriminator to address and improve on the awkward edges. This system is trained with not only general GAN loss but also concurrently with style loss to edit the parts of the face image even if a large area is missing. Our system creates high quality realistic composite images with the user's free-form input. The free-form domain input of sketch and color also has an interesting additive effects, as shown in Figure 1. In summary, we make the following contributions:

- We suggest a network architecture similar to Unet [13] with gated convolutional layers [17]. Such architecture is easier and faster for both training and inference stages. It produced superior and detailed result compared to the Coarse-Refined network in our case.

- We created a free-form domain data of masks, color and sketch. This data is used for making incomplete image data for training instead of stereotyped form input.

- We applied SN-patchGAN [17] discriminator and trained our network with additional style loss. This application covers cases with large portions erased and has shown robustness at managing the edges of the masks. It also allowed production of details on the produced image such as high quality synthetic hair style and earring.

2. Related Work

Interactive image modification has an extensive history, predominantly in regards with the techniques that use hand-crafted features rather than deep learning techniques. Such predominance is reflected in commercial image editing software and our practice of its usage. Because most commercial image editing softwares use defined operations, a typical image modification task requires expert knowledge to strategically apply a combination of transformations for an image. In addition to expert knowledge, users are required to devote long working hours to produce a delicate product. Therefore traditional approach is disadvantageous for the non-experts and is tedious to use for producing high quality results. In addition to these conventional modeling methods, recent breakthroughs in GAN research have developed several methods for completion, modification, and transformation of images by training generative models with large data sets.

In this section, we discuss several works in the field of image completion and image translation among prevalent image editing methods that use deep learning.

2.1. Image Translation

GAN for image translation was first proposed for learning image domain transforms between two datasets [21, 6]. Pix2Pix [6] proposed a system used dataset which consists of pair of images that can be used to create models that convert segmentation labels to the original image, or convert a sketch to an image, or a black and white image to color image. But this system requires that images and target images must exist as a pair in the training dataset in order to learn the transform between domains. CycleGAN [21] suggests an improvement over such requirement. Given a target domain without a target image, there exists a virtual result in the target domain when converting the image in original domain. If the virtual result is inverted again, the inverted result must be the original image. So, it takes two generators for converting task.

Recently, following the domain to domain change, several researches efforts have demonstrated systems that can take user input for adding the desired directions to generated results. The StarGAN [2] used a single generator and a discriminator to flexibly translate an input image to any desired target domain by training with domain label. The Ideepcolor [19] is introduced as a system that convert a monochrome image into a color image by taking a user's desired color as a mask. In these works, image transformation that interacts with user input has shown that user input can be learned by feeding it to the generator with images.

2.2. Image Completion

Image completion field has two main challenges: 1) Filling deleted area of the image, 2) proper reflection of users input in the restored area. In a previous study, a GAN system explores the possibility of generating complete images with erased area [5]. It uses generator from the U-net [13] structure and utilize local and global discriminator. The discriminator decides whether the generated image is real or fake on both the newly filled parts and full reconstructed image respectively. The Deepfillv1 [18] also used rectangle mask and global and local discriminator model to suggest that contextual attention layer extensively improves the performance. However, the global and local discriminator still produces awkward region at the border of restored parts.

In the follow-up study Deepfillv2 [17], a free-form mask and SN-patchGAN were introduced to replace the existing rectangular mask and global and local discriminator with single discriminator. In addition, the gated convolutional layer that learns the features of the masked region was also suggested. This layer can present masks automatically from data by training, which gives the network ability to reflect user sketch input on the results.

Our proposed network described in the next section allows the usage of not only sketch but also color data as an input for editing the image. Even though we utilized a U-net structure instead of a Coarse-Refined net structure such as in Deepfillv1,2 [5, 17], our network generate high-quality results without complex training schedule nor with requirement of other complex layers.

3. Approach

In this paper, we describe the proposed SC-FEGAN, a neural network based face image editing system and also describe methods for making the input batch data. This network can be trained end-to-end and generates high quality synthetic image with realistic texture details.

In Section 3.1, we discuss our method for making training data. In Section 3.2, we describe our network structure and loss functions that allow extraction of features from sketch and color input, while simultaneously achieving stability in training.

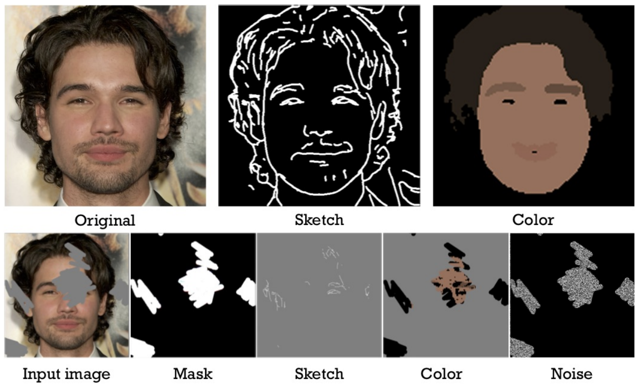


Figure 2. Sketch and color domain dataset and inputs for batch. We extract sketches using HED edge detector [16]. The color maps are generated by median color of segmented areas from using GFC [9]. The inputs of network consist of incomplete image, mask, sketch, color and noise.

3.1. Training Data

Suitable training data is a very important factor for increasing the training performance of the network and increasing the responsiveness to user input. To train our model, we used the CelebA-HQ [8] dataset after several pre-processing steps as described as following. We first randomly select 2 sets of 29,000 images for training and 1,000 images for testing. We resize the images to 512×512 pixels before attaining the sketch and color dataset.

To better express the complexity of the eye in the face image, we used a free-from mask based on the eye position to train network. Also, we created appropriate sketch domains and color domains by using free-form mask and face segmentation GFC [9]. This was a crucial step which enabled our system to produce persuasive results for the hand-drawn user input case. We randomly applied masks to hair region in input data because it has different properties compared other parts of face. We discuss more details below.

Free-form masking with eye-positions We used a similar masking method as that which was presented in Deepfillv2 [17] to make incomplete images. However, when training on the facial images, we randomly applied a free draw mask with eye-positions as a starting point in order to express complex parts of the eyes. We also randomly added hair mask using GFC [9]. Details are described in Algorithm 1.

Sketch & Color domain For this part, we used a method similar to that used in FaceShop [12]. However, we excluded AutoTrace [15] which converts bitmap to vector graphics for sketch data. We used the HED [16] edge detector to generate the sketch data whcih corresponds to the user's input to modify the facial image. After that, we smoothed the curves and erased the small edges. To cre-

Algorithm 1 Free-form masking with eye-positions

```
maxDraw, maxLine, maxAngle, maxLength are hyper parameters
GFCHair is the GFC for get hair mask of input image
Mask=zeros(inputSize,inputSize)
HairMask=GFCHair(IntputImage)
numLine=random.range(maxDraw)
for $i$=0 to numLine do
    startX = random.range(inputSize)
    startY = random.range(inputSize)
    startAngle = random.range(360)
    numV = random.range(maxLine)
    for $j$=0 to numV do
        angleP = random.range(-maxAngle,maxAngle)
        if $j$ is even then
            angle = startAngle+angleP
        else
            angle = startAngle+angleP+180
        end if
        length = random.range(maxLength)
        Draw a line on Mask from point (startX, startY) with angle and length.
        startX = startX + length * sin(angle)
        startY = stateY + length * cos(angle)
    end for
    Draw a line on Mask from eye postion randomly.
end for
Mask = Mask + HairMask (randomly)
```

ate color domain data, we first created blurred images by applying a median filtering with size 3 followed by 20 application of bilateral filter. After that, GFC [9] was used to segment the face, and each segmented parts were replaced with the median color of the corresponding parts. When creating data for the color domain, histogram equalization was not applied to avoid color contamination from light reflection and shadowing. However, because it was more resonant for users to express all part of face in sketch domain regardless of blur caused by interference of light, histogram equalization was used when creating the sketch from the domain data. More specifically, after histogram equalization, we applied HED to get the edges from the images. Then, we smoothed the curve and erased the small objects. Finally, we multiplied the mask, adopting a process similar to the previous free-form mask, and color images and get color brushed images. See Figure 2 for an example of our data.

3.2. Network Architecture

Inspired by recent image completion studies [5, 17, 12], our completion network (*i.e.* Generator) is based on encoder-decoder architecture like the U-net [13] and our discrimination network is based on SN-patchGAN [17]. Our network structure produces high-quality synthesis results with image size of 512×512 while achie ving stable and fast training. Our network also trains generator and discriminator simultaneously like the other networks. The generator receives incomplete images with user input to create an output image in the RGB channel, and inserts the masked area of the output image into the incomplete input image to create a complete image. The discriminator receives either a completed image or an original image (without masking) to determine whether the given input is real or fake. In adversarial training, additional user input to the discriminator also helps to improve performance. Also, we found that additional loss that is different to general GAN loss is effective to restore large erased portions. Details of our network are shown below.

Generator Figure 3 shows our network architecture in detail. Our generator is based on U-net [10] and all convolution layers use gated convolution [17] using 3x3 size kernel. Local signal normalization (LRN) [8] is applied after feature map convolution layers excluding other soft gates. LRN is applied to all convolution layers except input and output layers. The encoder of our generator receives input tensor of size 512×512×9: an incomplete RGB channel image with a removed region to be edited, a binary sketch that describes the structure of removed parts, a RGB color stroke map, a binary mask and a noise (see Figure 2). The encoder downsamples input 7 times using 2 stride kernel convolutions, followed by dilated convolutions before upsampling. The decoder uses transposed convolutions for upsampling. Then, skip connections were added to allow concatenation with previous layer with the same spatial resolution. We used the leaky ReLU activation function after each layer except for the output layer, which uses a tanh function. Overall, our generator consists of 16 convolution layers and the output of the network is an RGB image of same size of input (512×512). We replaced the remaining parts of image outside the mask with the input image before applying the loss functions to it. This replacement allows the generator to be trained on the edited region exclusively. Our generator is trained with losses which were introduced in PartialConv [10]: per-pixel losses, perceptual loss, style loss and total variance loss. The generic GAN loss function is also used.

Discriminator Our discriminator has SN-PatchGAN [17] structure. Unlike Deepfillv2 [17], we did not apply ReLu function on the GAN loss. Also we used 3×3 size convolution kernel and applied gradient penalty loss term. We added an extra term, to avoid the discriminator output patch reaching value close to zero. Our overall loss functions are shown as below:

$$L_{G\_SN} = -\mathbb{E}\left[D\left(I_{comp}\right)\right], \qquad (1)$$

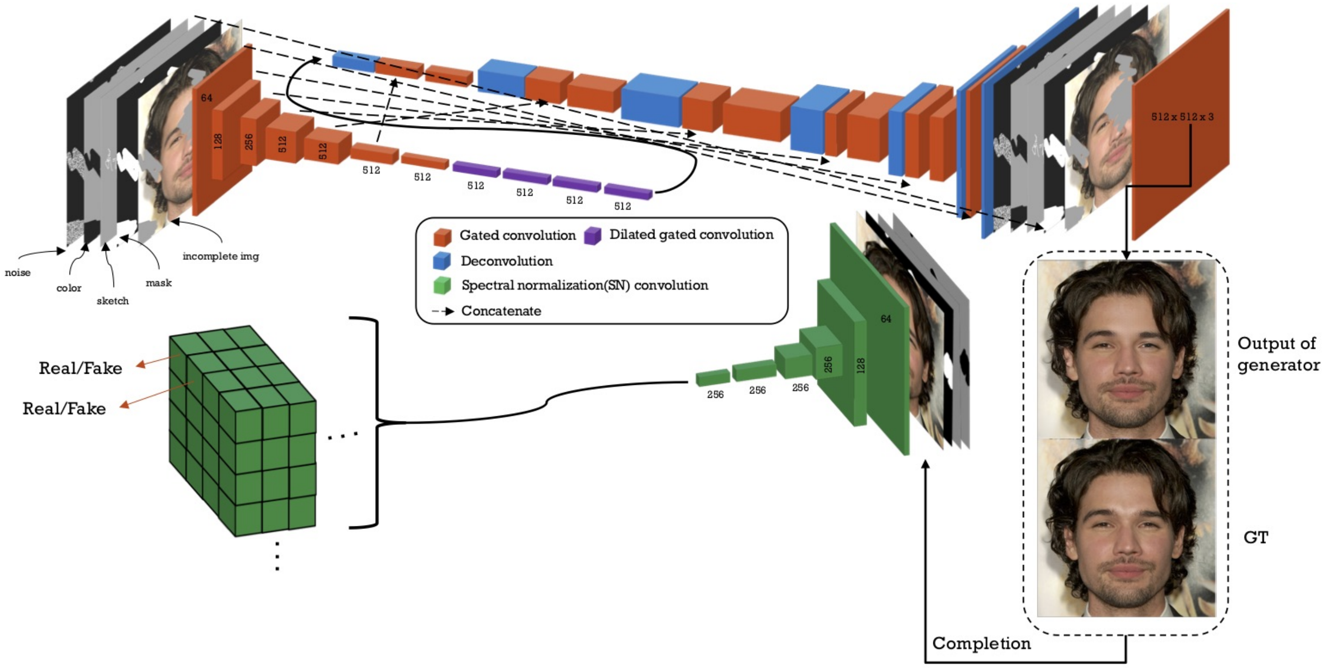


Figure 3. Network architecture of SC-FEGAN. LRN is applied after all convolution layers except input and output. We used tanh as activation function for output of generator. We used SN convolution layer [11] for discriminator.

$$
\begin{aligned}
L_G = L_{per-pixel} + \sigma L_{percept} + \beta L_{G\_SN} \\
+\gamma(L_{sytle}(I_{gen}) + L_{sytle}(I_{comp})) \\
+\upsilon L_{tv} + \epsilon \mathbb{E}\left[D(I_{gt})^2\right],
\end{aligned}
\tag{2}
$$

$$
L_D = \mathbb{E}\left[1 - D(I_{gt})\right] + \mathbb{E}\left[1 + D(I_{comp})\right] + \theta L_{GP}.
\tag{3}
$$

Our generator was trained with L_G and discriminator was trained with L_D. $D(I)$ is the output of the discriminator given input I. Additional losses, L_{sytle} and $L_{percept}$ are critical when editing large region such as hairstyle. The details of each loss are described below. The $L_{per-pixel}$ of L^1 distances between ground truth image I_{gt} and the output of generator I_{gen} is computed as

$$
\begin{aligned}
L_{per-pixel} = \frac{1}{N_{I_{gt}}} \left\| M \odot (I_{gen} - I_{gt}) \right\|_1 \\
+\alpha \frac{1}{N_{I_{gt}}} \left\| (1 - M) \odot (I_{gen} - I_{gt}) \right\|_1,
\end{aligned}
\tag{4}
$$

where, N_a is the number elements of feature a, M is the binary mask map and I_{gen} is the output of generator. We used the factor $\alpha > 1$ to give more weight the loss on the erased part. The perceptual loss, $L_{percept}$, also computes L^1 distances, but after projecting images into feature spaces using VGG-16 [14] which are pre-trained on ImageNet. It is computed as

$$
\begin{aligned}
L_{percept} = \sum_q \frac{\left\| \Theta_q(I_{gen}) - \Theta_q(I_{gt}) \right\|_1}{N_{\Theta_q(I_{gt})}} + \\
\sum_q \frac{\left\| \Theta_q(I_{comp}) - \Theta_q(I_{gt}) \right\|_1}{N_{\Theta_q(I_{gt})}}.
\end{aligned}
\tag{5}
$$

Here, $\Theta_q(x)$ is the feature map of the q-th layer of VGG-16 [14] given that input x is given and I_{comp} is the completion image of I_{gen} with the non-erased parts directly set to the ground truth. q is the selected layers from VGG-16, and we used layers of $pool1, pool2$ and $pool3$. Style loss compare the content of two images by using Gram matrix. We compute the style loss as

$$
L_{sytle}(I) = \sum_q \frac{1}{C_q C_q} \left\| \frac{(G_q(I) - G_q(I_{gt}))}{N_q} \right\|_1,
\tag{6}
$$

where the $G_q(x) = (\Theta_q(x))^T(\Theta - q(x))$ is the Gram matrix to perform an autocorrelation on each feature maps of VGG-16. When the feature is of shape $H_q \times W_q \times C_q$, the output of Gram matrix is of shape $C_q \times C_q$.

$L_{tv} = L_{tv-col} + L_{tv-row}$ is total variation loss suggested by fast-neural-style [7] to improve the *checkerboard artifacts* from perceptual loss term. It is computed as

$$
L_{tv-col} = \sum_{(i,j)\in R} \frac{\left\| I^{i,j+1}_{comp} - I^{i,j}_{comp} \right\|_1}{N_{comp}},
\tag{7}
$$

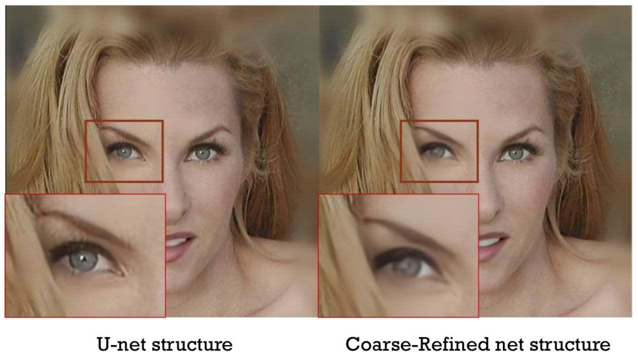


Figure 4. Our results with U-net(Left) and Coarse-Refined net(Right) when eye regions are removed.

$$L_{tv-row} = \sum_{(i,j)\in R} \frac{\left\| I_{comp}^{i+1,j} - I_{comp}^{i,j} \right\|_1}{N_{comp}}, \tag{8}$$

where R is the region of the erased parts. The WGAN-GP [4] loss is used for improving training and is computed as

$$L_{GP} = \mathbb{E}\left[\left(\|\nabla_U D(U) \odot M\|_2 - 1 \right)^2 \right]. \tag{9}$$

Here, **U** is a data point uniformly sampled along the straight line between discriminator inputs from I_{comp} and I_{gt}. This term is critical to quality of synthetic image in our case. We used $\sigma = 0.05, \beta = 0.001, \gamma = 120, \upsilon = 0.1, \epsilon = 0.001$ and $\theta = 10$.

4. Results

In this section, we present ablation studies with comparisons to recent related works, followed by face editing results. All experiments were executed on NVIDIA(R) Tesla(R) V100 GPU and Power9 @ 2.3GHz CPU with Tensorflow [1] v1.12, CUDA v10, Cudnn v7 and Python 3. For testing, it takes **44ms** on GPU, **53ms** on CPU for resolution 512×512 on average regardless of size and shape of inputs. The source code and more results are displayed in https://github.com/JoYoungjoo/SC-FEGAN.

4.1. Ablation Studies and Comparisons

We first compared our results to Coarse-Refined structure network and U-net structure network. In Deepfillv2 [17], it presents that Coarse-Refined structure and contextual attention module is effective for generation. However, we tested Coarse-Refined structure network and notice that the refining stage has blurred the output. We discovered that the reason for this is because L^1 loss about output of refined network is always smaller than of coarse network. The coarse network generates a coarse estimate of the recovered region by using incomplete input. This coarse image is then passed to the refined network. Such setup allows refined network to learn the transform between ground truth and the coarsely recovered about incomplete input. To achieve such effect with convolution operation, blurring on the input data is used as a workaround for an otherwise much complicated training method. It can ameliorate the *checkerboard artifacts* but it needs a lot of memories and time to training. Figure 4 shows the result of our system about Coarse-Refined structure network.

The system in FaceShop [12] has shown difficulty in modifying the huge erased image like whole hair regions.

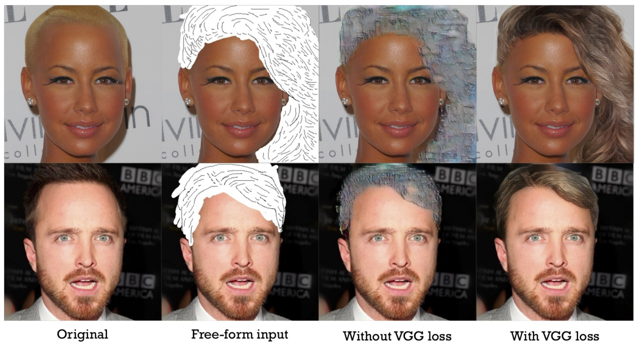


Figure 5. Our results from trained networks with and without VGG loss. When the network is trained without VGG loss, we encountered the similar problems as the FaceShop [12].

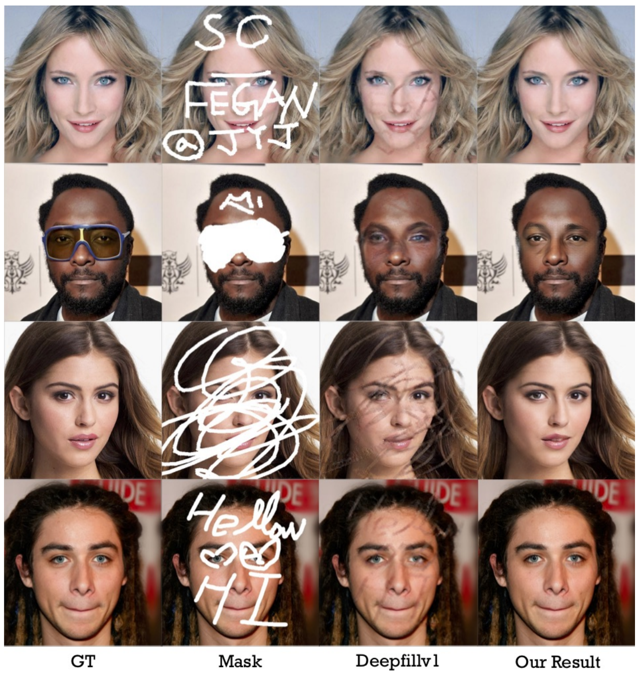


Figure 6. Qualitative comparisons with Deepfillv1 [18] on the CelebA-HQ validation sets.

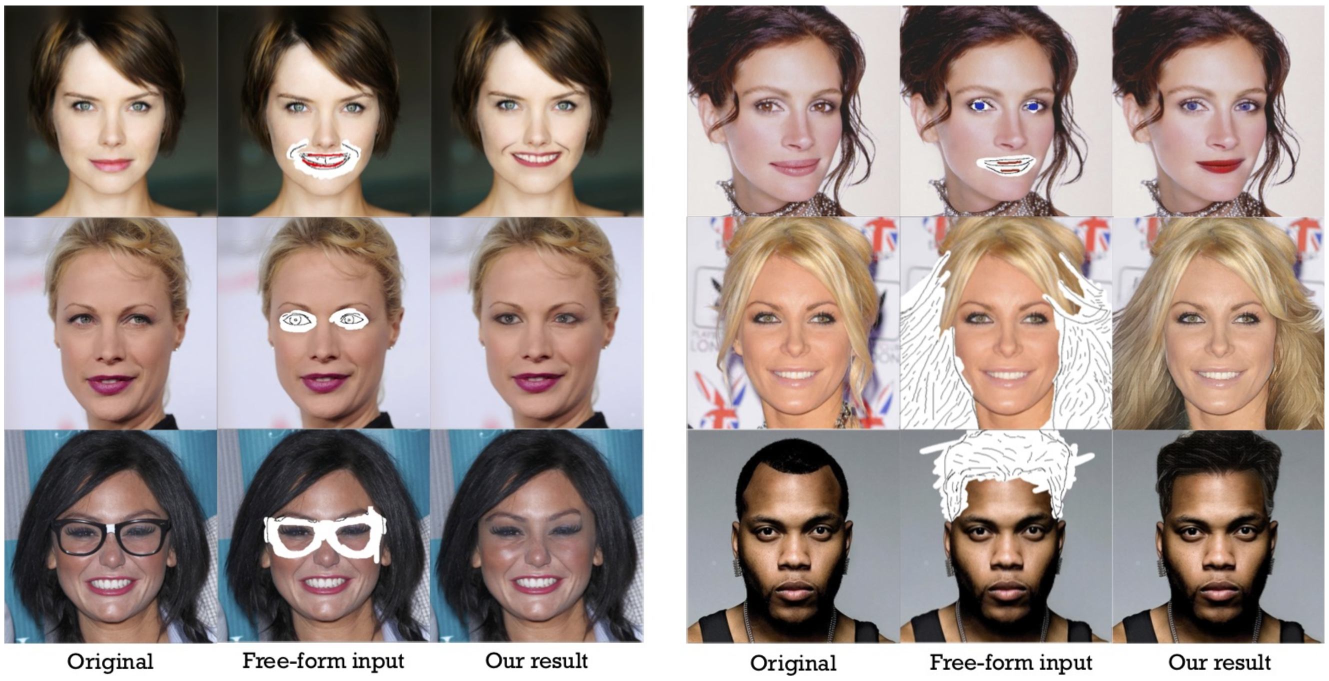


Figure 7. Face image editing results from our system. It shows that our system can change the shape and color of face properly. It also shows that it can be used in changing the color of eyes or erasing unnecessary parts. In particular, the right bottom two results show that our system can also be used for new hairstyle modification.

Our system performs better in that regard due to perceptual and style losses. Figure 5 shows the result of with and without VGG loss. We also conducted a comparison with the recent research Deepfillv1 [18] in which the test system was published. Figure 6 shows that our system produces better results in terms of quality for structure and shape with free-form masks.

4.2. Face Image Editing and Restoration

Figure 7 shows various results with sketch and color inputs. It shows that our system allows users to intuitively edit the face image features such as hair style, face shape, eyes, mouth *etc*. Even if the entire hair region is erased, our system is capable of generating an appropriate result once it is provided with a user sketch. Users can intuitively edit the images with sketches and colors, while the network tolerates small drawing error. The user can modify the face image intuitively through sketch and color input to obtain a realistic synthetic image that reflects shadows and shapes in detail. Figure 9 shows some results for the validation dataset, which shows that even though the user has a lot of modifications, the user can get a high quality composite image with enough user input. In addition, in order to check the reliance on the dataset the network learned, we tried to input the erased image of all areas. Compared with Deepfillv1 [18], Deepfillv1 generates faint image of face but our SC-FEGAN generates faint image of hair (see Figure 10). It means that without additional information, like sketch and color, the shape and position of elements of face have a certain dependent value. Therefore, it is only necessary to provide additional information to restore the image in the desired direction. In addition, our SC-FEGAN can generates face image with only sketch and color free-form input even if the input image is erased totally (see Figure 10).

4.3. Interesting results

The image results generated by the GAN often show high dependencies on the training data sets. Deepfillv2 [17] used same dataset CelebA-HQ, but used only landmark to make sketch dataset. In Faceshop [12], the AutoTrace [15] erased small details in the images of the dataset. In our study, we applied HED to all area, and by scheduling it to extend the masking area, we were able to obtain special results that produces facial image along with earrings. Figure 8 shows selection of such interesting results. These examples demonstrate that our network is capable of learning small details and generate reasonable results even for small input.

5. Conclusions

In this paper we present a novel image editing system for free-form masks, sketches, colors inputs which is based on an end-to-end trainable generative network with a novel

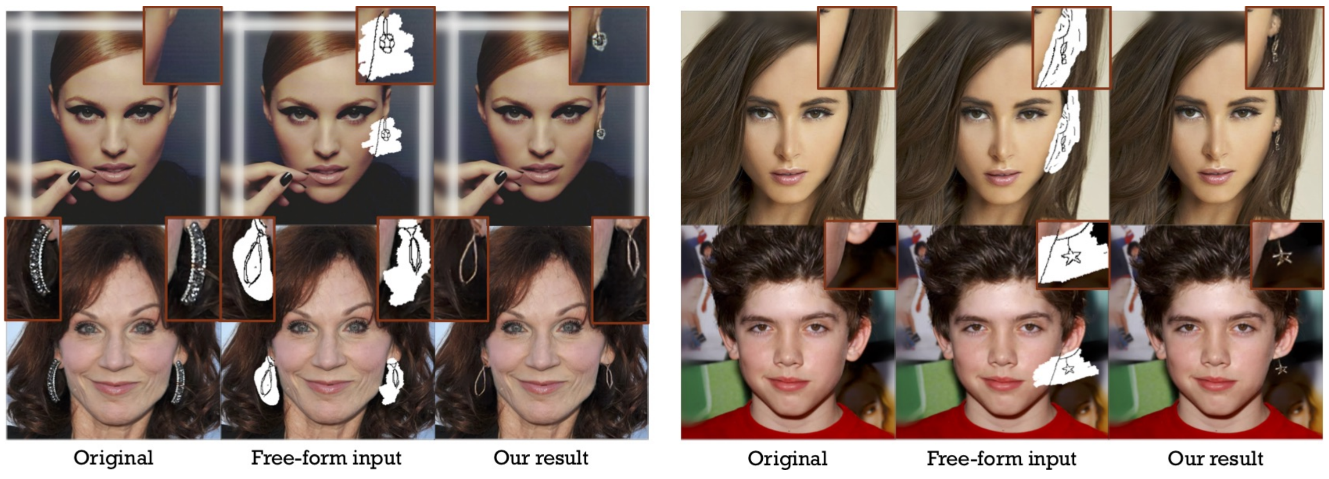


Figure 8. Our special results about edit earrings.

GAN loss. We showed that our network architecture and loss functions significantly improve inpainting results in comparison with other studies. We trained our system on high resolution imagery based on the celebA-HQ dataset and show a variety of successful and realistic editing results in many cases. We have shown that our system is excellent at modifying and restoring large regions in one pass, and it produces high quality and realistic results while requiring minimal efforts from the users.

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016. 6

[2] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[3] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016. 1

[4] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017. 6

[5] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4):107, 2017. 1, 3, 4

[6] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017. 2

[7] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016. 5

[8] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 3, 4

[9] Y. Li, S. Liu, J. Yang, and M.-H. Yang. Generative face completion. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 3, 2017. 1, 3, 4

[10] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro. Image inpainting for irregular holes using partial convolutions. *arXiv preprint arXiv:1804.07723*, 2018. 4

[11] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 5

[12] T. Portenier, Q. Hu, A. Szabo, S. Bigdeli, P. Favaro, and M. Zwicker. Faceshop: Deep sketch-based face image editing. *arXiv preprint arXiv:1804.08972*, 2018. 2, 3, 4, 6, 7

[13] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2, 3, 4

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 5

[15] M. Weber. Autotrace, 2018. http://autotrace.sourceforge.net. 3, 7

[16] S. "Xie and Z. Tu. Holistically-nested edge detection. In *Proceedings of IEEE International Conference on Computer Vision*, 2015. 3

[17] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589*, 2018. 2, 3, 4, 6, 7

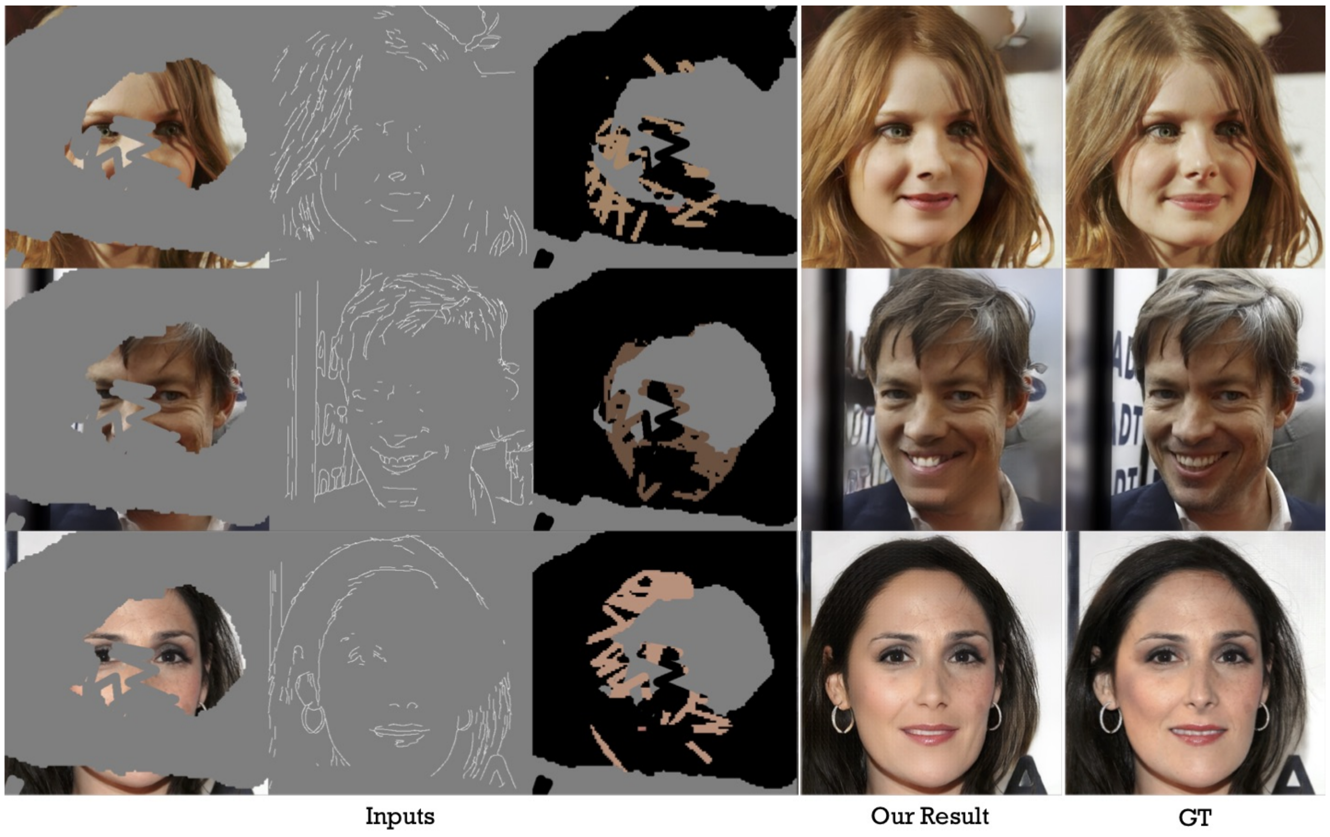
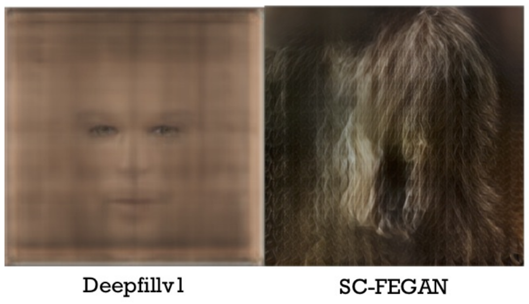


Figure 9. Our results about face restoration. Our system can restore the face satisfactorily if given enough input information even if a lot of regions are erased.



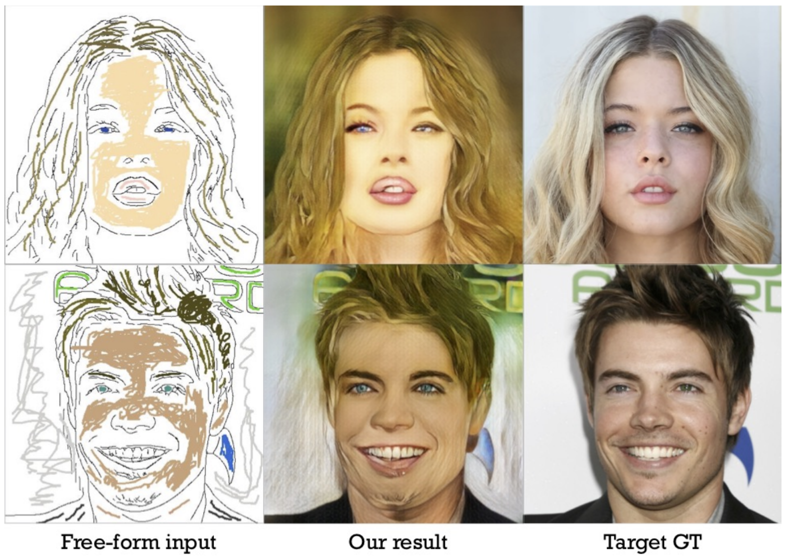


Figure 10. Our results about total restoration. On the left, it shows that results of Deepfillv1 [18] and SC-FEGAN about totally erased image. On the right, it shows that SC-FEGAN can be works like translation. It can generates face image only with sketch and color input.

[18] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Generative image inpainting with contextual attention. *arXiv preprint arXiv:1801.07892*, 2018. 3, 6, 7, 9

[19] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros. Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999*, 2017. 2, 3

[20] Y. Zhao, B. Price, S. Cohen, and D. Gurari. Guided image inpainting: Replacing an image region by pulling content from another image. *arXiv preprint arXiv:1803.08435*, 2018. 2

[21] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017. 2