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 masks, sketches and color as inputs. Our system consists of an end-to-end trainable convolutional network. In contrast to the existing methods, our system utilizes entirely free-form user input in terms of color and shape. This allows the system to respond to the user's sketch and color inputs, using them as guidelines to generate an image. In this work, we trained the network with an additional style loss, 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 for generating high-quality synthetic images using intuitive user inputs.

1. Introduction

Image completion with generative adversarial networks (GANs) is a topic of great interest in computer vision. With image exchange becoming a common medium of daily communication in the present day, there is an increasing demand for realism in images generated with minimal image completion features. This demand is reflected in social media statistics. However, most current image editing software requires considerable expertise, such as knowing which specific tools to use in certain scenarios to effectively modify the image in the desired way. Alternatively, an image completion method that could respond to user input would allow a novice to easily modify images as desired. To this end, our proposed system has the ability to easily produce high-quality face images provided given sketch and color inputs, even in the case that parts of the image have been erased.

In recent works, deep-learning-based image completion methods have been used to restore erased portions of an image. The most typical method is to use an ordinary (square) mask and then restore the masked region using an encoder-decoder generator. Then, global and local discriminators can be used to estimate whether the result is real or fake [5, 9]. However, the applicability of this system is limited to low-resolution images, and the generated images contain awkward edges around the masked regions. In addition, the synthesized images in the restored regions often fall short of user expectations since the generator is not given any user input to serve as a guide. Several works that have improved upon these limitations include DeepFillv2 [17], which utilizes user sketches as inputs, a work

Figure 1. Face image editing results generated by our system. It can take free-form inputs consisting of masks, sketches and color. These examples show that our system allows users to easily edit the shape and color of the face, even if the user wants to completely change the hairstyle and eyes (second row). Interestingly, the user can even edit earrings with our system (third row).

*Corresponding author

that utilized user's sketch as an input, and GuidedInpainting [21], in which part of another image is used as an input to restore the missing parts. However, since DeepFillv2 does not use color input, the color in the synthesized image is generated via inference from a prior distribution learned from the training dataset. By contrast, GuidedInpainting uses parts of other images to restore deleted regions. However, it is difficult to achieve detailed restoration because such a process requires inferring the user's preferred reference image. Another recent work proposed the iDeepColor system [20], which accepts color input from the user to use as a reference in creating a color image from a black-and-white image. However, the iDeepColor system does not allow the editing of object structures or the restoration of deleted parts of an image. In another work, the face editing system FaceShop [12], which accepts sketch and color inputs from the user, was introduced. However, FaceShop has some limitations as an interactive system for synthetic image generation. First, it utilizes random rectangular rotatable masks to erase the regions that are used in the local and global discriminators. This means that the local discriminator must resize a restored local patch to match the fitting input dimensions, and this resizing process will distort the information in both the erased and remaining portions of the image. As a result, the produced image will contain awkward edges around the restored portion. Second, FaceShop will produce an unreasonable synthetic image if an excessively large area is erased. For example, when given an image with the entire hair region erased, the system typically restores it with a distorted shape.

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

- We propose a network architecture similar to U-net [13] with gated convolutional layers [17]. With this architecture, both the training and inference stages are easier and faster. In our case, the proposed network produces superior and more detailed results compared to those of a network with the coarse-refined structure.

- We create free-form domain data consisting of masks, color and sketches. These data are used to create incomplete image data for training instead of input of a fixed form.

- We apply an SN-PatchGAN [17] discriminator to train our network with an additional style loss. This approach enables the consideration of cases in which large portions of an image are erased and is shown to be robust in managing the edges of masks. It also enables the creation of newly introduced details in the produced images, such as high-quality synthetic hairstyles and earrings.

2. Related Work

Interactive image modification has an extensive history, predominantly in regards to techniques that use handcrafted features rather than deep learning techniques. This predominance is reflected in commercial image editing software and its typical usage in practice. Because most commercial image editing software relies on defined operations, a typical image modification task requires expert knowledge to guide the strategic application of a combination of specific transformations to an image. In addition to requiring expert knowledge, users must devote many hours to producing a delicately crafted product. Therefore, the traditional approach is disadvantageous for nonexperts, and producing high-quality results is a tedious task. In addition to these conventional modeling methods, recent breakthroughs in GAN research have led to the development of several methods for the completion, modification, and transformation of images by using generative models trained on large datasets.

In this section, we discuss several works in the fields of image completion and image translation that are representative of the prevalent image editing methods based on deep learning.

2.1. Image Translation

The use of GANs for image translation was first proposed for learning image-domain transforms between two datasets [22, 6]. Pix2Pix [6] was proposed as a system using a dataset consisting of pairs of images that can be used to create models for converting segmentation labels into the original image, a sketch into an image, or a black-and-white image into a color image. However, this system requires that the initial and target images must exist as pairs in the training dataset in order to learn the transform between the domains. CycleGAN [22] was proposed as an improvement over such requirements. Given a target domain without a target image, there exists a virtual result in the target domain when an image is converted from the original domain. If that virtual result is inverted again, the inverted result must be the original image. Therefore, this system uses two generators for the conversion task.

Recently, building on the previous work on domain-to-domain transformation, several studies have demonstrated systems that can take user input as a basis for adding certain desired modifications to the generated results. StarGAN [3] uses a single generator and a discriminator to flexibly translate an input image into any desired target domain based on training with domain labels. The IDeepColor [20] system converts a monochrome image into a color image by taking a user's desired color mask as input. These works on interactive image transformation based on user input have shown that user input can be learned by feeding it into a generator along with images.

2.2. Image Completion

Image completion tasks involve two main challenges: 1) filling in the deleted areas of an image and 2) properly reflecting the provided user input in the restored areas. A previous study explored the possibility of using a GAN system to generate complete images from images with erased areas [5]. The presented system uses a generator with the U-net [13] structure and utilizes local and global discriminators. These discriminators decide whether the generated image is real or fake based on the newly filled-in parts and the fully reconstructed image, respectively. Deepfillv1 [18] also uses rectangular masks and global and local discriminator models but additionally includes a contextual attention layer, which greatly improves its performance. However, the global and local discriminators still produce awkward regions at the borders of the restored parts.

In the subsequent version, DeepFillv2 [17], free-form masks and a single SN-PatchGAN discriminator replace the rectangular masks and global and local discriminators used in the previous version. Besides, a gated convolutional layer that learns the features of the masked regions is introduced. This layer can be trained to automatically generate masks from data, giving the network the ability to utilize user sketch input to guide the results.

Our proposed network described in the next section allows the usage of not only sketch but also color data as inputs for editing an image. Although we utilize a U-net structure instead of a coarse-refined network structure such as that in Deepfillv1,2 [5, 17], our network generates high-quality results without requiring either a complex training schedule or other complex layers.

3. Approach

In this section, we describe the proposed SC-FEGAN, a neural-network-based face image editing system, and present methods for creating the input batch data. This network can be trained in an end-to-end manner and generates high-quality synthetic images with realistic texture details.

In Section 3.1, we discuss our method of creating training data. In Section 3.2, we describe our network structure and loss functions, which permit the extraction of features from sketch and color inputs while simultaneously achieving stability in training.

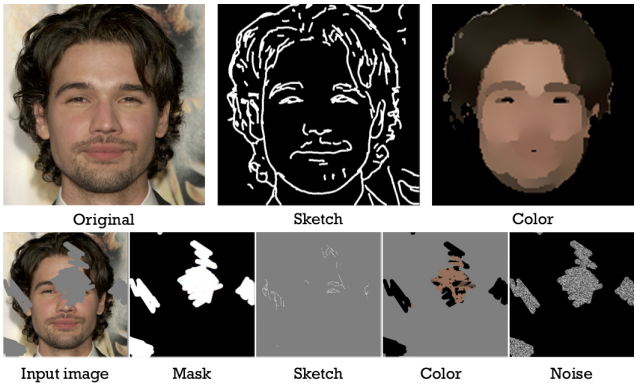

Figure 2. Sketch and color domain dataset and batch inputs. We extract sketches using the HED edge detector [16]. The color maps are generated from the median color of each area segmented using GFC [9]. The inputs to the network consist of the incomplete image, the mask, and the corresponding sketch,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

3.1. Training Data

Suitable training data are very important for enhancing 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 the application of several preprocessing steps, as described below. We first randomly selected a set of 29,000 images for training and a set of 1,000 images for testing. We resized the images to 512×512 pixels before generating the sketch and color dataset.

To better capture the complexity of the eyes in face images, we used a free-from masking approach based on the eye positions to train the network. Additionally, we created appropriate sketch and color domain data using a free-form masking and face segmentation process based on GFC [9]. This was a crucial step for enabling our system to produce persuasive results in the case of hand-drawn user input. We randomly applied masks to the hair regions in the input data because the hair has different properties compared with other parts of the face. We provide further details below.

Free-form masking with eye-positions We used a masking method similar to that presented in DeepFillv2 [17] to generate incomplete images. However, when training on the face images, we randomly applied a freely drawn mask with the eye positions as a starting point to capture the complex eye region. We also randomly added hair masks using GFC [9]. The details are described in Algorithm 1.

Sketch & **Color domain** For this process, we used a method similar to that used in FaceShop [12]. However, we excluded AutoTrace [15] which converts bitmaps to vector

Algorithm 1 Free-form masking with eye positions

maxDraw, maxLine, maxAngle, maxLength are hyperparameters
GFCHair is the GFC operation for obtaining the hair mask of the 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 the point (startX, startY) with the angle and length generated above.
    startX = startX + length * sin(angle)
    startY = stateY + length * cos(angle)
  end for
  Randomly draw a line on Mask from the eye position.
end for
Mask = Mask + HairMask (randomly)

graphics to generate sketch data. We used the HED [16] edge detector to generate sketch data corresponding to the user's input to enable the modification of the face image. After that, we smoothed the curves and erased small edges. To create color domain data, we first created blurred images by applying a median filter of size 3 followed by 20 applications of a bilateral filter. Then, GFC [9] was used to segment the face, and each segmented part was replaced with the median color of the corresponding parts (except for hair and skin). When creating the data for the color domain, histogram equalization was not applied to avoid color contamination from light reflection and shadowing. However, because it is more intuitive for users to express all parts of the face in the sketch domain regardless of blur caused by light interference, histogram equalization was used when creating the sketch domain data. More specifically, after histogram equalization, we applied HED to extract the edges from the image. Then, we smoothed the curves and erased small edges. Finally, we multiplied a mask (obtained through a process similar to the previous free-form masking process) with the color image to obtain the color brushed image. 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 an encoder-decoder architecture similar to that of U-net [13]], and our discrimination network is based on SN-PatchGAN [17]. Our network structure produces high-quality synthesis results with an image size of 512×512 while achieving stable and fast training. The generator and discriminator in our network are also trained simultaneously, similar to the other networks on which it is based. The generator receives an incomplete image along with user input, creates an output image in the RGB channel, and inserts the generated output image into the masked area of the incomplete input image to create a complete image. The discriminator receives either such a completed image or an original image (without masking) and determines whether the given input is real or fake. During adversarial training, additional user input is provided to the discriminator to help improve performance. We have also found that considering an additional loss that is different from the general GAN loss is effective in helping to restore large erased portions. The 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 convolutional layers use gated convolution [17] with a 3x3 kernel. Local signal normalization (LRN) [8] is applied after each feature map convolutional layer, excluding other soft gates. LRN is applied to all convolutional layers except the input and output layers. The encoder of our generator receives an input tensor with dimensions of 512×512×9: an incomplete RGB-channel image with one or more removed regions to be edited, a binary sketch that describes the structure of the removed parts, an RGB color stroke map, a binary mask and noise (see Figure 2). The encoder downsamples the input 7 times via kernel convolutions with a stride of 2, followed by dilated convolutions before upsampling. The decoder uses transposed convolutions for upsampling. Then, skip connections are added to allow concatenation with the previous layer with the same spatial resolution. We use the leaky ReLU activation function after each layer except the output layer, which uses a tanh function. Overall, our generator consists of 16 convolutional layers, and the output of the network is an RGB image of the same size as the input (512×512). We replace the remaining parts of the image outside the mask with the input image before applying the loss functions to it. This replacement allows the generator to be trained exclusively on the edited region. Our generator is trained with the losses that 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 We use the SN-PatchGAN [17] structure for the discriminator. Unlike Deepfillv2 [17], we do not ap-

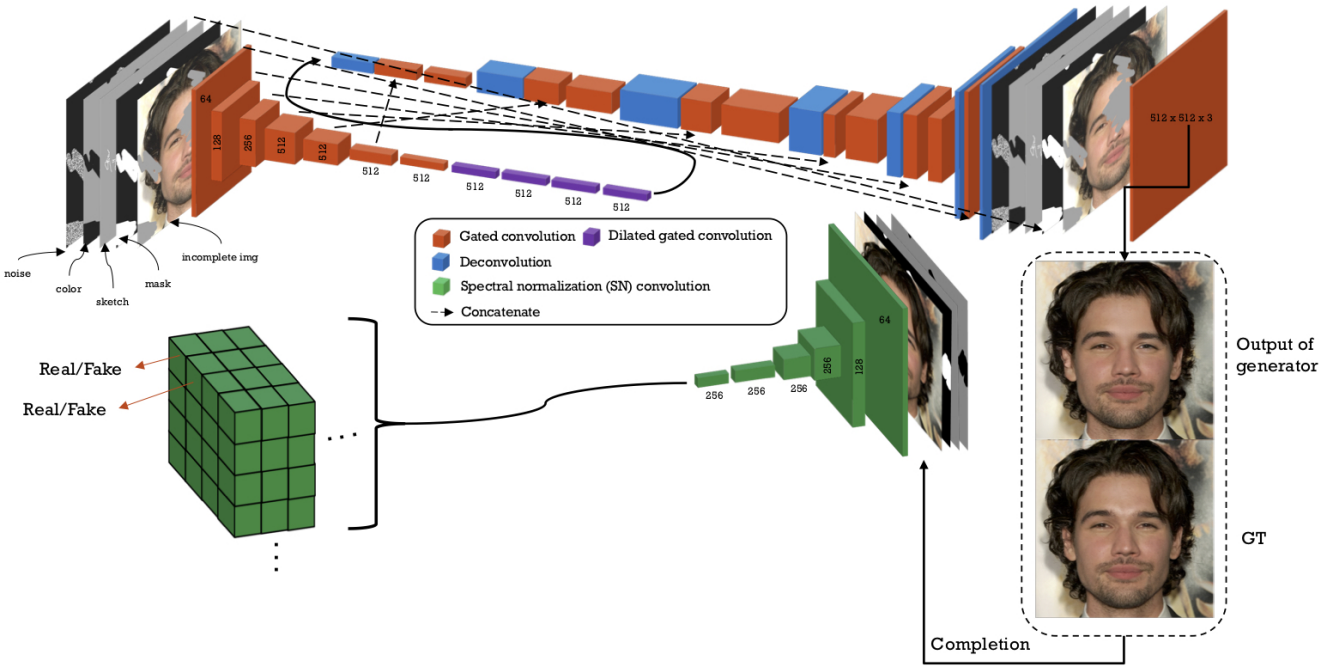

Figure 3. Network architecture of SC-FEGAN. LRN is applied after each convolutional layers except the input and output layers. We use tanh as the activation function for the output of generator. We use a SN convolutional layer [11] for the discriminator.

ply the ReLu function to the GAN loss. Additionally, we use a 3×3 convolution kernel and apply a gradient penalty loss term. We also add an extra term to prevent the discriminator output patch from reaching a value close to zero. Our overall loss functions are shown as below:

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

$$L_D = \mathbb{E}\left[1 - D(I_{gt})\right] + \mathbb{E}\left[1 + D(I_{comp})\right] + \theta L_{GP} + \epsilon \mathbb{E}\left[D(I_{gt})^2\right], \tag{2}$$

$$L_G = L_{per-pixel} + \sigma L_{percept} + \beta L_{G\_SN} + \gamma(L_{style}\left(I_{gen}\right) + L_{style}\left(I_{comp}\right)) + \upsilon L_{tv}. \tag{3}$$

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

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

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

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

Here, $\Theta_q(x)$ is the feature map of the q-th layer of VGG-16 [14] given the input x is and I_{comp}, which is the completed image obtained from I_{gen} with the nonerased parts of the image directly set equal to the ground truth. q is the selected layer of VGG-16: we use the $pool1, pool2$ and $pool3$ layers. The style loss compares the contents of two images on the basis of the Gram matrix. We compute the style loss as

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

where the $G_q(x) = (\Theta_q(x))^T(\Theta_q(x))$ is the Gram matrix used to perform autocorrelation on each feature map of VGG-16. When the feature has dimensions of $H_q \times$

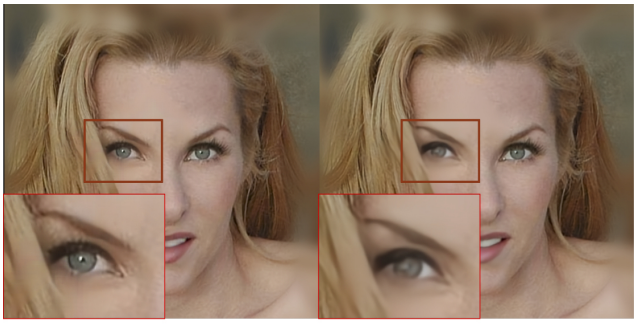

Figure 4. Our results with the U-net structure (Left) and the coarse-refined network structure (Right) when the eye region is removed.

$W_q \times C_q$, the output of the Gram matrix has dimensions of $C_q \times C_q$.

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

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

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

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

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

Here, **U** is a data point uniformly sampled along the straight line between the discriminator inputs from I_{comp} and I_{gt}. This term is critical to the quality of the 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 and comparisons to recent related works, followed by face editing results. All experiments were executed on an NVIDIA(R) Tesla(R) V100 GPU and a Power9 @ 2.3 GHz CPU with TensorFlow [1] v1.12, CUDA v10, cudnn v7 and Python 3. During testing, processing required an average of **44 ms** on the GPU and **53 ms** on the CPU for an image resolution of 512×512, regardless of the size and shape of the user inputs. The interactive code and additional results are available at Github.

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

Figure 6. Face image editing results from our system. These results shown that our system can appropriately change the shape and color of the face. They also show that it can be used to change the color of the eyes or erase unnecessary parts of the image. In particular, our system can also be used for hairstyle modification.

4.1. Ablation Studies and Comparisons

We first compare our results obtained using the coarse-refined network structure and the U-net structure. The authors of Deepfillv2 [17] have reported that the coarse-refined structure and a contextual attention module enable effective generation. However, we tested the coarse-refined network structure and noticed that the refinement stage blurred the output. We discovered that the reason for this is because the L^1 loss related to the output of the refined network is generally smaller than that for the coarse network. The coarse network generates a coarse estimate of the re-

Figure 7. Qualitative comparisons with DeepFillv1 [18] on the CelebA-HQ validation sets.

Figure 8. Our results regarding face restoration. Our system can satisfactorily restore the face if given sufficient input information even if many regions are erased.

covered region using incomplete input. This coarse image is then passed to the refined network. This setup allows the refined network to learn the transform between the ground truth and the coarsely recovered estimate. To achieve such an effect with convolution operations, blurring on the input data is used as a workaround for an otherwise much more complicated training method. This approach can ameliorate *checkerboard artifacts*, but it requires considerable memory and time for training. Figure 4 shows the results of our system with the coarse-refined network structure.

FaceShop [12] has shown difficulty in modifying images with very large erased regions, such as whole hair regions. Our system performs better in that regard due to the use of the perceptual and style losses. Figure 5 shows results obtained with and without VGG loss. We also conducted a comparison with the recently developed DeepFillv1 system [18]. Figure 7 shows that our system produces better results in terms of the quality of the image structure and shape with free-form masks.

We also conducted a quantitative comparison of various metrics (PSNR, SSIM, L2 Loss and LPIPS [19]) and the following results, as shown as Table 1 were obtained. We compared the Deepfillv1, PM [2] and our SC-FEGAN on our CelebA-HQ test dataset which has 1000 test images. A quantitative experiment on image inpainting task was performed using 10 randomly generated mask for each image. The base model SC-FEGAN does not take sketch and color as input even though it was trained with sketch and color data for fair evaluation. Nonetheless, SC-FEGAN outperformed PM and Deepfillv1 in every input configurations and metrics. As shown in Table 1, SC-FEGAN with mask and sketch show the best performance. That is, sketches are more informative to inpainting tasks. Though the performance for inpainting is slightly lower when color input is applied, it should be noted that additional color input enables more flexible and complete image editing.

Algorithm	Input[1]	PSNR(↑)	SSIM(↑)	L2(%)(↓)	LPIPS(↓)
PM	M	21.6649	0.9157	0.95	0.2068
Deepfillv1[2]	M	23.7554	0.9145	0.52	0.1835
SC-FEGAN	M	**29.4799**	**0.9618**	**0.14**	**0.0641**
SC-FEGAN	M,S	**31.1687**	**0.9671**	**0.0937**	**0.0552**
SC-FEGAN	M,C	**27.9131**	**0.9543**	**0.2**	**0.0795**
SC-FEGAN	M,S,C	**29.4912**	**0.9606**	**0.14**	**0.0682**

Table 1. Quantitative results on the CelebA-HQ dataset

4.2. Face Image Editing and Restoration

Figure 6 shows various results obtained with sketch and color inputs. It shows that our system allows users to intuitively edit face image features such as the hairstyle, face shape, eyes, and mouth. Even if the entire hair region is erased, our system is capable of generating an appropriate result when provided with a user sketch. Users can intuitively edit images with sketch and color inputs, and the network can tolerate a small drawing error. Thus, a user can intuitively modify a face image and obtain a realistic synthetic image that reflects shadows and shapes in detail. Figure 8 presents some results, which show that even when the user makes many modifications, a high-quality composite image can be obtained with sufficient user input. In addition, to check the reliance on the dataset on which the network was trained, we tested input with all image areas erased and compared the results with those of DeepFillv1 [18]. Whereas DeepFillv1 generates a faint image of the face, our SC-FEGAN generates only a faint image of

[1]M,S,C: denotes mask, sketch and color inputs, respectively
[2]Deepfillv1: we used publically released pre-trained model.

Figure 9. Comparison with DeepFillv1 [18] regarding totally erased input.

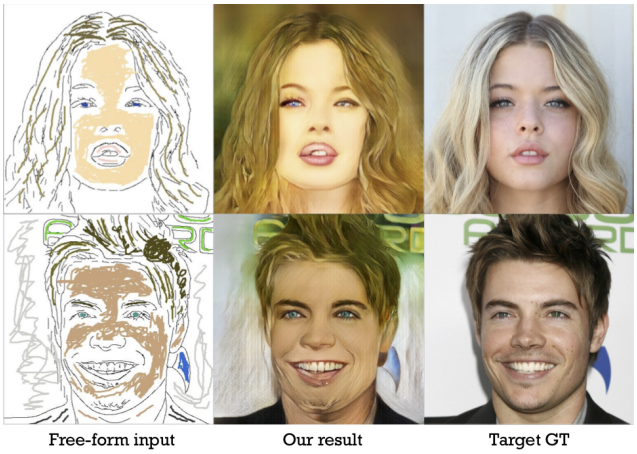

Figure 10. Our results regarding total restoration. These results shown that SC-FEGAN can function as an image translator: it can generate face images from only sketch and color input.

the hair (see Figure 9). This finding suggests that without additional information, such as sketch and color input, the shapes and positions of elements of the face show a certain dependency. Therefore, it is necessary to provide additional information to restore the image in the desired direction. In addition, our SC-FEGAN can generate a face image from only free-form sketch and color input even when the input image is completely erased (see Figure 10).

4.3. Interesting results

The image results generated by a GAN often show a strong dependency on the training dataset. The authors of DeepFillv2 [17] used the same dataset, CelebA-HQ, but used only landmarks to create the sketch dataset. In Faceshop [12], AutoTrace [15] erases small details from the images in the dataset. In our study, we applied HED to all areas, and by using this approach to extend the masking area, we were able to obtain specialized results; specifically, we could produce face images with newly added or modified earrings. Figure 11 shows a selection of such interesting results. These examples demonstrate that our network is capable of learning small details and generating reasonable results even for small inputs.

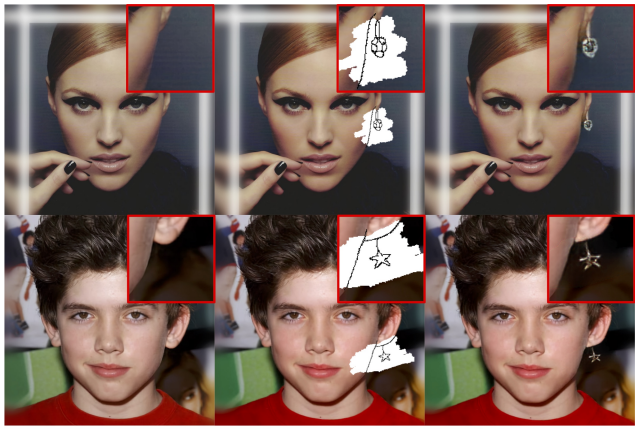

Figure 11. Our specialized results demonstrating the ability to edit earrings.

5. Conclusions

In this paper, we present a novel image editing system for free-form mask, sketch, and color inputs that is based on an end-to-end trainable generative network with a novel GAN loss. We show that our network architecture and loss functions lead to significantly improved inpainting results in comparison with other systems. We trained our system on high-resolution imagery based on the CelebA-HQ dataset and obtained 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 effort on the part of the user.

Acknowledgements We thank Kimin Yun and Albert Jin Chung for helpful discussions. This work was supported by Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.B0101-15-0266, Development of High Performance Visual BigData Discovery Platform for Large-Scale Realtime Data Analysis)

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspon-

dence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), Aug. 2009.

[3] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8789–8797, 2018.

[4] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[5] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4):107, 2017.

[6] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1125–1134, 2017.

[7] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 694–711. Springer, 2016.

[8] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. 2018.

[9] Yijun Li, Sifei Liu, Jimei Yang, and Ming-Hsuan Yang. Generative face completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3911–3919, 2017.

[10] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100, 2018.

[11] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[12] Tiziano Portenier, Qiyang Hu, Attila Szabo, Siavash Arjomand Bigdeli, Paolo Favaro, and Matthias Zwicker. Faceshop: Deep sketch-based face image editing. *ACM Transactions on Graphics (TOG)*, 37(4):99, 2018.

[13] Olaf Ronneberger, Philipp Fischer, and Thomas 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.

[14] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[15] Martin Weber. Autotrace, 2018. http://autotrace.sourceforge.net.

[16] Saining "Xie and Zhuowen" Tu. Holistically-nested edge detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1395–1403, 2015.

[17] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589*, 2018.

[18] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5505–5514, 2018.

[19] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[20] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics (TOG)*, 9(4), 2017.

[21] Yinan Zhao, Brian Price, Scott Cohen, and Danna Gurari. Guided image inpainting: Replacing an image region by pulling content from another image. *arXiv preprint arXiv:1803.08435*, 2018.

[22] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2223–2232, 2017.