Recursive Cross-Domain Face/Sketch Generation from Limited Facial Parts

Yang Song, Zhifei Zhang, and Hairong Qi
The University of Tennessee, Knoxville, TN, USA
{ysong18, zzhang61, hqi}@utk.edu

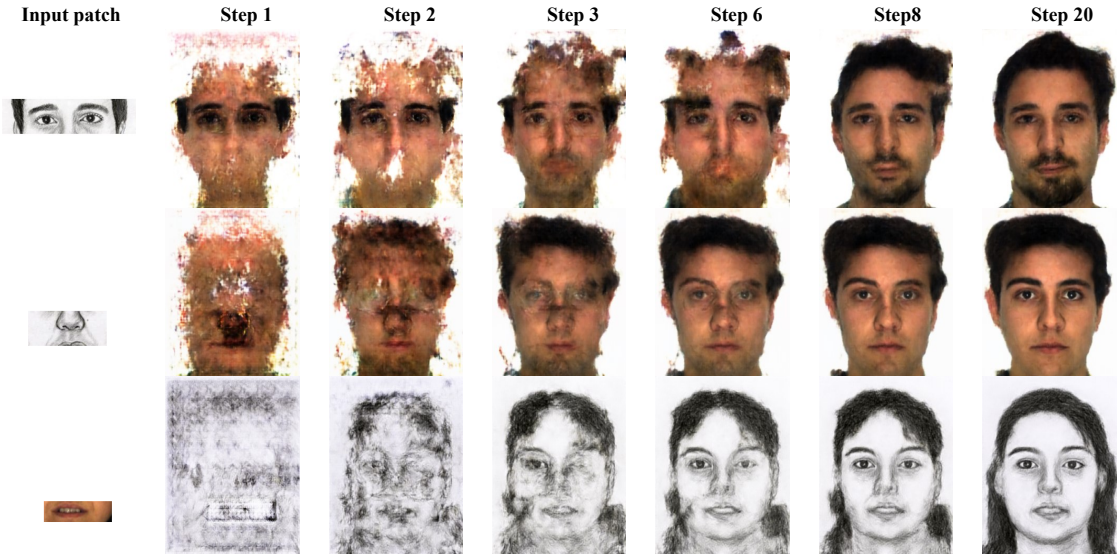

Figure 1: A demo of generating face/sketch from small patches using the proposed recursive generation by bidirectional transformation networks (r-BTN). The left column shows the given sketch or face patches. The right columns show the results of recursive generation at different iteration steps in the correspondingly transferred domain. The proposed r-BTN gradually generates realistic faces/sketches consistent to the given small patches.

Abstract

We start by asking an interesting yet challenging question, "If a large proportion (e.g., *more than 90% as shown in Fig. 1) of the face/sketch is missing, can a realistic whole face sketch/image still be estimated?" Existing face completion and generation methods either do not conduct domain transfer learning or can not handle large missing area. For example, the inpainting approach tends to blur the generated region when the missing area is large* (i.e., *more than 50%). In this paper, we exploit the potential of deep learning networks in filling large missing region* (e.g., *as high as 95% missing) and generating realistic faces with high-fidelity in cross domains. We propose the recursive generation by bidirectional transformation networks (r-BTN) that recursively generates a whole face/sketch from a small sketch/face patch. The large missing area and the cross domain challenge make it difficult to generate satisfactory results using a unidirectional cross-domain learning structure. On the other hand, a forward and backward bidirectional learning between the face and sketch domains would enable recursive estimation of the missing region in an incremental manner (Fig. 1) and yield appealing results. r-BTN also adopts an adversarial constraint to encourage the generation of realistic faces/sketches. Extensive experiments have been conducted to demonstrate the superior performance from r-BTN as compared to existing potential solutions.*

1. Introduction

This work represents the first attempt to cross-filling large missing area in both face and sketch domains. Existing works that may potentially address this problem are mainly in the perspectives of face/sketch synthe-

sis/transformation and image inpainting. The face/sketch synthesis works [24, 22, 30, 20] synthesize target faces from the source domain through patch-wise searching of similar patches in the training set. Without the generative capability, these methods fail to render reasonable pixels for large missing areas. The rapid development of generative adversarial networks (GANs) [5] has shown impressive performance in face generation [18, 29], domain transformation [31, 6], and inpainting [26, 16]. However, generating faces from small patches in either single or cross domains has not been explored. Intuitively, combining domain transformation and inpainting works could be a potential solution. However, with large missing area, the generated results tend to be blurred and may look unrealistic.

In this paper, we investigate the problem of cross-domain face/sketch generation conditioned on a given small patch of sketch/face. We assume that faces and sketches lie on high-dimensional manifolds $\mathcal{I}$ and $\mathcal{S}$, respectively, as shown in Fig. 2 (right). The given small sketch/face patch will initially deviate from the corresponding manifold due to large amount of missing data. With the learned bidirectional transformation network (BTN), *i.e.*, f and F, the given patch will be recursively mapped forward and backward between $\mathcal{I}$ and $\mathcal{S}$. Each mapping will yield a result progressively closing in onto either the face or sketch manifold, and eventually approaching the real whole face/sketch images as shown in Fig. 2 (middle). An adversarial network is imposed on both f and F, forcing more photo-realistic faces/sketches. The rationale and benefit of the proposed r-BTN will be further discussed in section 3.

This paper makes the following contributions: 1) We tackle the challenging problem of face/sketch generation from small patches, estimating large missing area based on limited information while alleviating the blur effect suffered by existing works. 2) We propose the recursive generation by bidirectional transformation networks (r-BTN), which learns both a forward and backward mapping function between cross domains to enable a recursive update of the generated faces/sketches for more consistent and high-fidelity results even with large portions of missing data. 3) We further exploit the capacity of r-BTN in fusing multiple patches from multiple domains and multiple people to output a realistic and consistent face in a generative manner. 4) In the area of generative imaging, there is a lack of quantitative evaluation of image reality. We design two metrics – face recognition rate (FRR) and relative discrimination score (RDS), for effective evaluation of image reality.

2. Related Works

We will discuss related works from three closely related areas, namely, face/sketch synthesis/transformation, image inpainting, and face manipulation.

Face/Sketch Synthesis/Transformation related works mainly fall into two categories: matching-based and generation-based methods. Most face/sketch synthesis works [24, 27, 30] are matching-based, which synthesize faces from best matched patches by searching from the training dataset. For example, [24] divided a given face/sketch image into patches, each of which was matched to a series of similar patches from the training dataset. Then, the patches in the target domain corresponding to the matched patches were stitched via Markov random field to synthesize a transformed face. The matching-based methods have two drawbacks: 1) The matching procedure is time-consuming for a large training dataset, and 2) they cannot effectively estimate the patch content from missing area. The generation-based methods [21, 6] are mainly developed from encoder-decoder networks and adversarial generative networks. For example, [6] proposed a general domain transformation method through conditional generative adversarial network. It could also be utilized for face/sketch transformation. However, it is not trained for the purpose of estimating missing areas. Moreover, to achieve bidirectional face/sketch transformation, two transformation networks (*i.e.*, face to sketch and sketch to face) need to be learned independently.

Image Inpainting aims to fill in unwanted or missing parts of an image. Most inpainting methods [4, 19, 2] estimate the missing part based on surrounding pixels, and therefore are not suitable for filling in large missing areas. Although some recent works [26, 16] claimed the ability of filling in up to 80% missing regions, they tend to generate blurred results, which may be with visible inconsistency between the given and estimated areas. In addition, inpainting related methods train on randomly masked inputs and perform filling in a single domain, while the proposed work uses the whole face/sketch pairs in training and perform cross-domain filling.

Face Manipulation works [29, 25] could be a potential solution to the proposed task because they can generate faces by manipulating the latent variables. Given a small patch, they may search the latent space for a best matched face. Thus, the generative model performs like matching-based methods which may be time-consuming. A more efficient way is to minimize the error between the generated face and the given patch. However, it cannot ensure consistent results because only the patch location (where the error comes from) will be updated regardless of its surroundings.

3. The Bidirectional Transformation Network

In this section, we first elaborate on the benefit of the proposed BTN through a comparison with unidirectional transformations (Sec. 3.1). This is followed by a detailed description of the training and testing stages of the proposed r-BTN. The training stage learns the bidirectional transformation between the face and sketch domains using

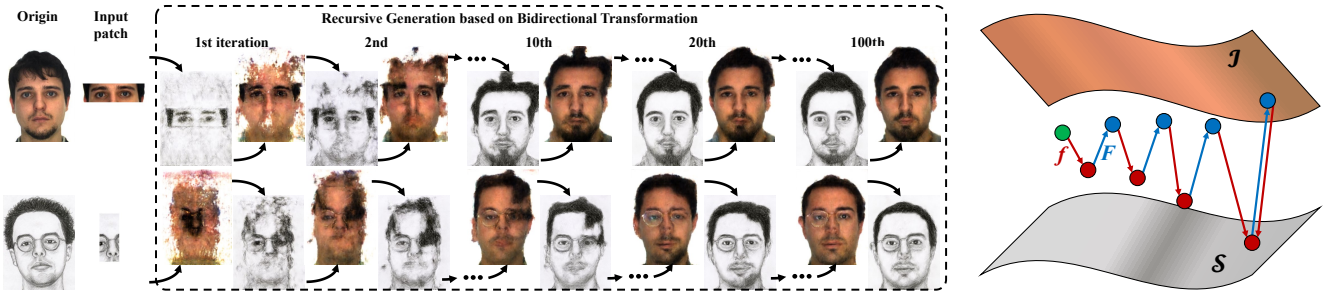

Figure 2: Examples of recursive generation from small patches by the bidirectional transformation network. Left: Original face/sketch and the corresponding input patches extracted from them. Inside of the dashed box demonstrates the generated face/sketch at different iteration steps. Right: Illustration of transformation between the face and sketch manifolds $\mathcal{I}$ and $\mathcal{S}$, respectively. The green dot denotes a given face patch. The red and blue arrows are the learned mapping f and F, respectively. The red and blue dots are generated sketches and faces through corresponding mapping.

whole face/sketch pairs (Sec. 3.2). The testing stage recursively generates the whole face/sketch from given small sketch/face patches (Sec. 3.3).

3.1. The Bidirectional Network Structure

Assume a training set in $\mathcal{I} \times \mathcal{S}$, where $\mathcal{I}$ and $\mathcal{S}$ denote the face and sketch domains, respectively. The unidirectional transformation, *e.g.*, [6], learns a mapping $f : \mathcal{I} \rightarrow \mathcal{S}$ which could be implemented by encoder-decoder networks, as shown in Fig. 3 (left). The BTN, on the other hand, simultaneously involves the forward mapping f and backward mapping $F : \mathcal{S} \rightarrow \mathcal{I}$, as shown in Fig. 3 (right). The bidirectional transformation forms a closed loop where the output of f serves as the input to F, and the output of F serves as the input to f in the next iteration. The forward transformation f may discard information in general due to the domain difference (*e.g.*, color information will be discarded from $\mathcal{I}$ to $\mathcal{S}$), but the backward transformation F closes the loop by connecting the output from f in the $\mathcal{S}$ domain and the original input in the $\mathcal{I}$ domain and generates an intermediate result in $\mathcal{I}$ where additional face information (*e.g.*, facial outline) has been estimated and the discarded information (*e.g.*, color) restored. The bidirectional network structure enables the recursive update of the face (from F) and sketch (from f), taking advantage of the progressively learned knowledge in both domains and generate full face/sketch with high fidelity.

The effectiveness of the recursive bidirectional transformation between face and sketch domains is well demonstrated in Fig. 2. In general, the missing area is roughly filled at the beginning (iteration 1 and 2) although it is blurred. Then, facial details are progressively enhanced (iteration 10) and sharpened (iteration 20). Finally, a realistic face/sketch, including reasonable hair style, is generated. Because of the very limited information provided in the input patch, it is difficult to generate a face/sketch exactly the same as the original. However, the generated face/sketch still preserves the pixel-level content of the given patch.

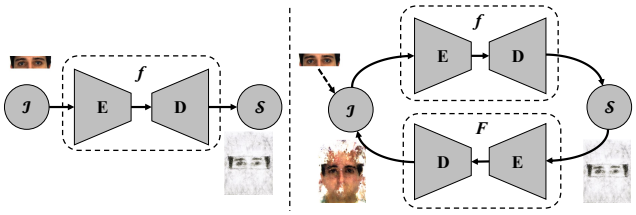

Figure 3: Comparison of unidirectional and bidirectional transformations between $\mathcal{I}$ and $\mathcal{S}$ domains. E and D are the encoder-decoder networks. The patch (eyes) generates the sketch, and then the sketch is transformed back where facial outline has been estimated.

3.2. Training Stage

Fig. 4 illustrates the details of the BTN structure where the mapping functions, f and F, are learned in a bidirectional fashion instead of the commonly used unidirectional mapping.

Given the original face/sketch pair $x_{\mathcal{I}}$ and $x_{\mathcal{S}}$, the following transformations are performed,

$$x^0_{\mathcal{S}} = f(x_{\mathcal{I}}),\ x^1_{\mathcal{I}} = F(x^0_{\mathcal{S}}) = F(f(x_{\mathcal{I}})),$$
$$x^0_{\mathcal{I}} = F(x_{\mathcal{S}}),\ x^1_{\mathcal{S}} = f(x^0_{\mathcal{I}}) = f(F(x_{\mathcal{S}})).$$

The objective is to learn the bidirectional transformations between $\mathcal{I}$ and $\mathcal{S}$, so that any face/sketch pair could be uniquely mapped forward and backward into another domain. To achieve invertible transformation, *i.e.*, preserving the identity of face and sketch during transformations, we minimize the reconstruction error $\mathcal{L}_{rec}$ between real and

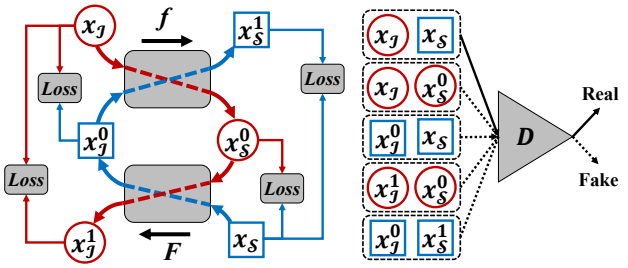

Figure 4: Training flow of the bidirectional transformation network. $x_{\mathcal{I}}$ and $x_{\mathcal{S}}$ are the real face/sketch pair. Red and blue arrows denote the transformation paths of $x_{\mathcal{I}}$ and $x_{\mathcal{S}}$, respectively. The transformation functions f and F could be encoder-decoder networks. *Loss* denotes the ℓ_1-norm. The discriminator D is trained on real and generated (fake) face/sketch pairs.

generated faces or sketches as Eq. 1.

$$\mathcal{L}_{rec} = \sum_{i=0}^{1} \left(\|x_{\mathcal{I}} - x_{\mathcal{I}}^i\|_1 + \|x_{\mathcal{S}} - x_{\mathcal{S}}^i\|_1 \right), \quad (1)$$

where the ℓ_1-norm instead of the ℓ_2-norm is used to avoid blurry results. Besides $\mathcal{L}_{rec}$, an adversarial constraint is employed to encourage photo-realistic face/sketch pairs. The discrimination loss can be written as

$$\mathcal{L}_{adv} = \mathbb{E}_{\omega \in \Omega} [\log D(\omega)] - \mathbb{E}_{\substack{x_{\mathcal{I}} \in \mathcal{I} \\ x_{\mathcal{S}} \in \mathcal{S}}} [\log D(x_{\mathcal{I}}, x_{\mathcal{S}})], \quad (2)$$

where

$$\begin{aligned} \Omega &= \{(x_{\mathcal{I}}, x_{\mathcal{S}}^0)_j, (x_{\mathcal{I}}^1, x_{\mathcal{S}}^0)_j, (x_{\mathcal{I}}^0, x_{\mathcal{S}})_j, (x_{\mathcal{I}}^0, x_{\mathcal{S}}^1)_j, \cdots\} \\ &= \{(x_{\mathcal{I}}, f(x_{\mathcal{I}}))_j, (F(f(x_{\mathcal{I}}), f(x_{\mathcal{I}})))_j, \\ &\quad (F(x_{\mathcal{S}}), x_{\mathcal{S}})_j, (F(x_{\mathcal{S}}), f(F(x_{\mathcal{S}})))_j, \cdots\} \end{aligned}$$

indicates the fake face/sketch pairs, and j indexes the fake pairs generated from the jth real pair in a mini-batch. Note that only $(x_{\mathcal{I}}, x_{\mathcal{S}})$ is the real pair. Combining Eqs. 1 and 2, the objective function is

$$\min_{f,F,D} \ \mathcal{L}_{adv} + \lambda \mathcal{L}_{rec}, \quad (3)$$

where λ balances the adversarial loss and reconstruction loss. In optimization, f, F, and D are updated alternatively. The discriminator D is updated by minimizing $\mathcal{L}_{adv}$. The update of f and F is performed by

$$\min_{f} \mathbb{E}_{\omega \in \Omega_f} [\log D(\omega)] + \lambda \sum_{i=0}^{1} \|x_{\mathcal{S}} - x_{\mathcal{S}}^i\|_1, \quad (4)$$

$$\min_{F} \mathbb{E}_{\omega \in \Omega_F} [\log D(\omega)] + \lambda \sum_{i=0}^{1} \|x_{\mathcal{I}} - x_{\mathcal{I}}^i\|_1, \quad (5)$$

where

$$\begin{aligned} \Omega_f &= \{(x_{\mathcal{I}}, x_{\mathcal{S}}^0)_j, (x_{\mathcal{I}}^0, x_{\mathcal{S}}^1)_j, \cdots\} \\ &= \{(x_{\mathcal{I}}, f(x_{\mathcal{I}}))_j, (x_{\mathcal{I}}^0, f(x_{\mathcal{I}}^0))_j \cdots\}, \\ \Omega_F &= \{(x_{\mathcal{I}}^0, x_{\mathcal{S}})_j, (x_{\mathcal{I}}^1, x_{\mathcal{S}}^0)_j, \cdots\} \\ &= \{(F(x_{\mathcal{S}}), x_{\mathcal{S}})_j, (F(x_{\mathcal{S}}^0), x_{\mathcal{S}}^0)_j, \cdots\}, \end{aligned}$$

and $\Omega = \Omega_f \cup \Omega_f$. Here, j is again the index of training samples in a mini-batch.

3.3. Testing Stage

During testing, given an arbitrary patch from either domain, a whole face from the other domain could be generated in a recursive manner through the bidirectional transformation. The testing flow is shown in Fig. 5, which

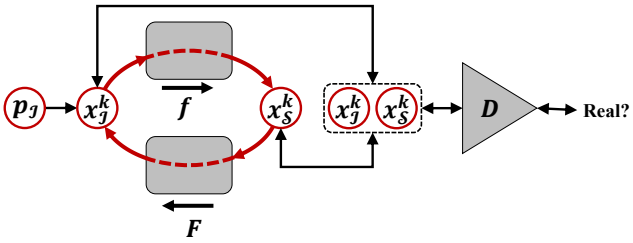

Figure 5: Testing flow of r-BTN, assuming a face patch $p_{\mathcal{I}}$ as the input. At step k, the generated face is $x_{\mathcal{I}}^k$. Replacing the corresponding area of $x_{\mathcal{I}}^k$ by the patch $p_{\mathcal{I}}$ and transforming $x_{\mathcal{I}}^k$ to $x_{\mathcal{S}}^k$, we get a face/sketch pair $(x_{\mathcal{I}}^k, x_{\mathcal{S}}^k)$. Then, this pair is adjusted by the error back propagated from D as comparing to the output of real pairs. Finally, $x_{\mathcal{S}}^k$ is transformed back to the face domain, generating $x_{\mathcal{I}}^{k+1}$.

demonstrates the case of given a face patch $p_{\mathcal{I}}$. Similarly, if a sketch patch $p_{\mathcal{S}}$ is given, it will be fed to $x_{\mathcal{S}}$ and similar testing flow can be carried out to generate a whole face image. In this paper, a patch is created through multiplying a whole face/sketch by a mask M, *e.g.*, $p_{\mathcal{I}} = x_{\mathcal{I}} \odot M$ where $\odot$ denotes the element-wise multiplication.

The bidirectional transformation network structure enables a recursive generation between sketches and faces. Given the current result $x_{\mathcal{I}}^k$, the next generation $x_{\mathcal{I}}^{k+1}$ can be obtained by

$$x_{\mathcal{I}}^k \leftarrow x_{\mathcal{I}}^k \odot (1 - M) + p_{\mathcal{I}}, \quad (6)$$

$$x_{\mathcal{S}}^k \leftarrow f\left(x_{\mathcal{I}}^k\right), \quad (7)$$

$$x_{\mathcal{S}}^k \leftarrow x_{\mathcal{S}}^k - \frac{\partial D(x_{\mathcal{I}}^k, x_{\mathcal{S}}^k)}{\partial x_{\mathcal{S}}^k}, \quad (8)$$

$$x_{\mathcal{I}}^{k+1} \leftarrow F\left(x_{\mathcal{S}}^k\right). \quad (9)$$

In order to generate photo-realistic faces/sketches such that the given patch and the estimated complement blend

together in a consistent fashion, we have applied two constraints during the recursive generation process. First, we keep the given patch, $p_{\mathcal{I}}$, as the anchor that remains the same across different iterations. In other words, $p_{\mathcal{I}}$ directly covers the corresponding area of the newly generated face to explicitly preserve the given content (Eq. 6). Then, $x_{\mathcal{I}}^k$ is transformed to the sketch domain by f (Eq. 7). Unlike most GANs related works which utilize D only in the training stage, we utilize D as a second constraint in the testing process to ensure realistic faces/sketches generation in each iteration such that small deviations get to be corrected instead of accumulated through iterations.

Given a small patch, the testing stage needs multiple iterations to gradually generate a whole face/sketch, as illustrated previously in Fig. 2. In each iteration, backpropagating the loss of D will enforce the photo-reality during the recursive generation. In the case of Fig. 5, the backpropagation error is used to adjust the generated sketch $x_{\mathcal{S}}^k$ as shown in Eq. 8. Finally, $x_{\mathcal{S}}^k$ is mapped back to the face domain (Eq. 9), generating $x_{\mathcal{I}}^{k+1}$ as an improved version of $x_{\mathcal{I}}^k$ with more details. Repeating this procedure, the large missing area can be filled up gradually.

To illustrate the effect of the two constraints, *i.e.*, the given patch and the adversarial constraints, applied during the testing stage, Fig. 6 shows the generated results with/without the constraints. The given patch and the adversarial constraints are denoted as "Patch" and "Adv", respectively. It is interesting to observe that the generated face/sketch without "Patch" (the second and third columns) cannot preserve the identity of the input patch, and those without "Adv" (the second and forth columns) tend to yield unrealistic face/sketch (*e.g.*, the left ear location) or hair style (*e.g.*, the extra hair below the left ear in the fourth column). The results with both constraints obviously outperform the others.

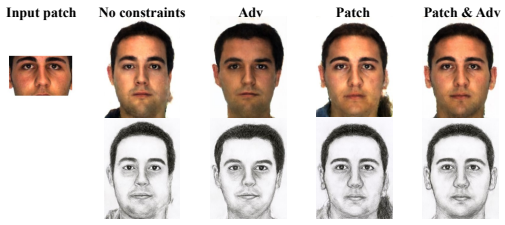

Figure 6: Comparison of generated results with/without the given patch (Patch) and adversarial (Adv) constraints.

4. Experiment and Results

4.1. Data Collection

We collect 1,577 face/sketch pairs from the datasets CUHK [24], CUFSF [28], AR [13], FERET [17], and IIIT-D [1]. Because the dataset with face/sketch pairs is limited, we train a face to sketch transformation network based on Pix2Pix [6] to generate sketches from faces. We collect frontal face images with uniform background and controlled illumination from datasets CFD [12], SiblingsDB [23], and PUT [7], as well as from searching engines by keywords like "XXX University faculty profile". Finally, we obtain 3,126 face/sketch pairs, from which 300 pairs are randomly selected as the testing dataset.

4.2. Implementation Details

All the face/sketch images are cropped and well-aligned based on the eye locations, and preprocessed to be uniform white background. The transformations f and F are implemented by the Conv-Deconv network as shown in Table 1.

Table 1: Network structure used for transformation

Conv. (LeakyReLU)	Deconv. (ReLU)
$256^2 \times 3$, $128^2 \times 64$,	$2^2 \times 1024$, $4^2 \times 1024$,
$64^2 \times 128$, $32^2 \times 256$,	$8^2 \times 1024$, $16^2 \times 512$,
$16^2 \times 512$, $8^2 \times 1024$,	$32^2 \times 256$, $64^2 \times 128$,
$4^2 \times 1024$, $2^2 \times 1024$	$128^2 \times 64$, $256^2 \times 3$

The kernel size is 5×5 for both Conv and Deconv networks, and batch normalization is adopted after each Conv/Deconv layer. The discriminator D is implemented by the Conv network but adding a fully-connected layer of single output with the sigmoid activation function. In addition, the input layer is modified to be $256^2 \times 6$ because the inputs to D are image pairs. Inspired by [6], each Conv layer is concatenated to its symmetrically corresponding Deconv layer, thus more details bypass the bottleneck. In the training, we adopt ADAM [9] ($\alpha = 0.0002$, $\beta = 0.5$). Because we utilize D to enforce realistic generations during testing, an approximately optimal D is preferred. Therefore, we update D three times for each update of f and F. The parameter λ in Eq. 3 is set to be 100. After 100 epochs, we could achieve the results as shown in this paper.

During testing, given a small patch from either the face or the sketch domain, it will be transformed recursively as discussed in Sec. 3.3. Empirically, the generated images will have most facial features filled quickly at the first five to ten iterations and then tend to converge after 50 iterations. The results shown in this paper are mostly obtained at the 100th iteration.

4.3. Qualitative Evaluation

4.3.1 Comparison with Other Methods

We compare the proposed r-BTN with Pix2Pix [6] and image inpainting [16]. The inpainting method compared in this paper is modified from [16] to achieve cross-domain inpainting. Specifically, the inputs are faces/sketches with random mask (20%∼80% masked), and the outputs are the

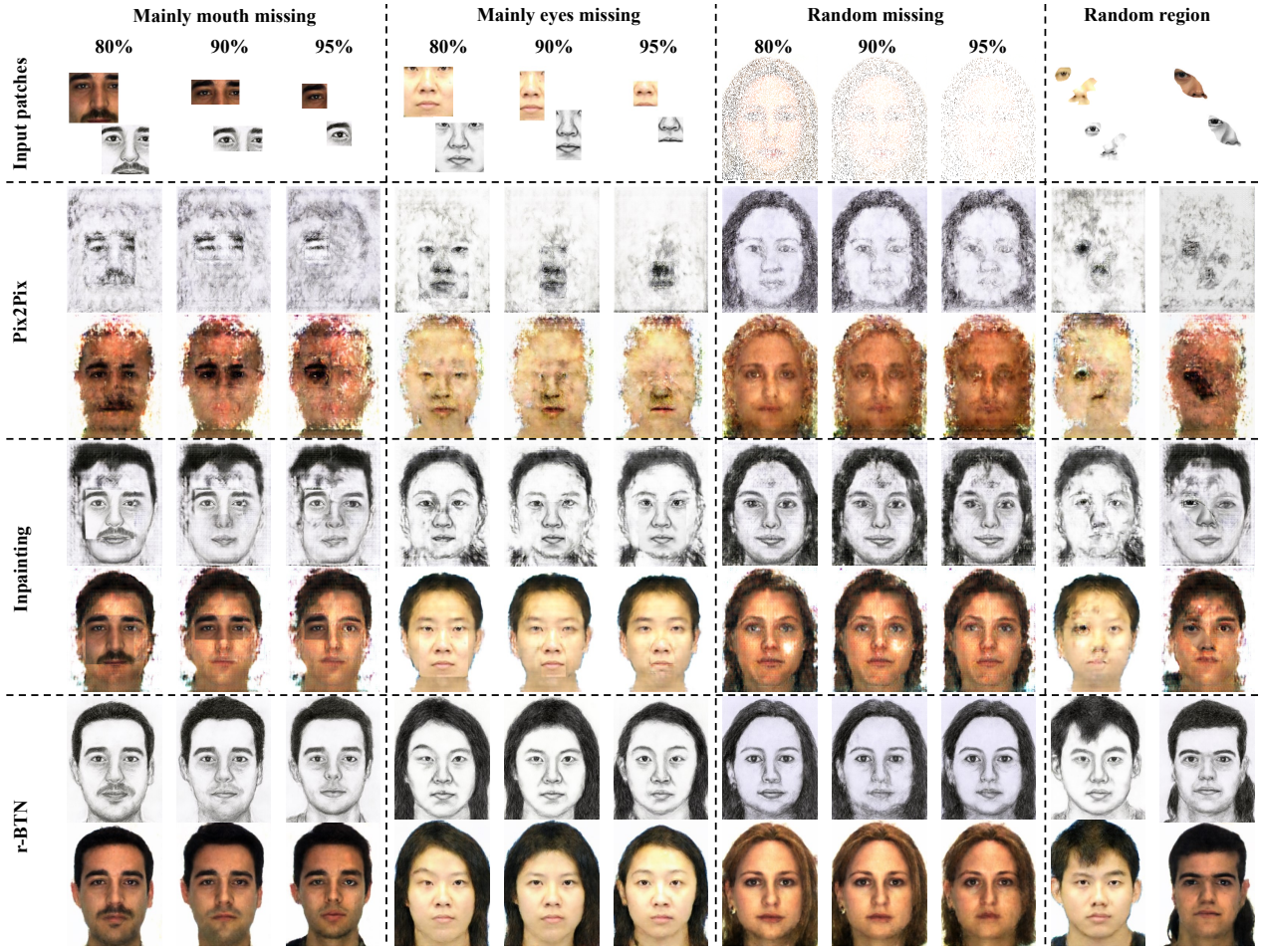

Figure 7: Comparison with other potential methods for filling large missing areas. The first row shows the input patches, and the rest rows display the results from different methods. The percentage indicates missing proportion (missing area over image area). Because Pix2Pix is for domain transfer rather than missing area filling, its results cannot compete with inpainting or r-BTN. We show them here to provide the baseline of domain transfer methods in filling large missing areas.

whole sketch/face. Pix2Pix and r-BTN are trained with the whole face/sketch pairs. All methods are trained on the same training dataset with the same parameter setting. The comparison results are shown in Fig. 7. The Pix2Pix and inpainting methods train face-sketch and sketch-face transformation networks independently, so the identity between generated sketches and faces cannot be preserved. For example, comparing the two rows labeled with "inpainting", especially the 4th-6th columns, the sketches seem female while the faces appear like male. In addition, the inpainting results present apparent discontinuity between the given patch and the estimated area. On the other hand, the results from r-BTN demonstrate higher fidelity, better consistency to given patches, and better identity preservation.

4.3.2 Generation from Multiple Patches

We explore the r-BTN to generate consistent and realistic faces from multiple patches that may be from two domains and multiple people. Examples generated from multiple patches are shown in Fig. 8, demonstrating the great versatility of r-BTN. We again observe the strong consistency and fidelity between the generated face/sketch pairs.

4.4. Quantitative Evaluation

4.4.1 Evaluation Metrics

To numerically evaluate the quality of generated faces, we design two metrics: 1) face recognition rate (FRR) and 2) relative discrimination score (RDS).

Face recognition rate (FRR) evaluates whether the generated images present facial elements and geometric structure, *i.e.*, reasonable position of eyebrows, eyes, nose, lips,

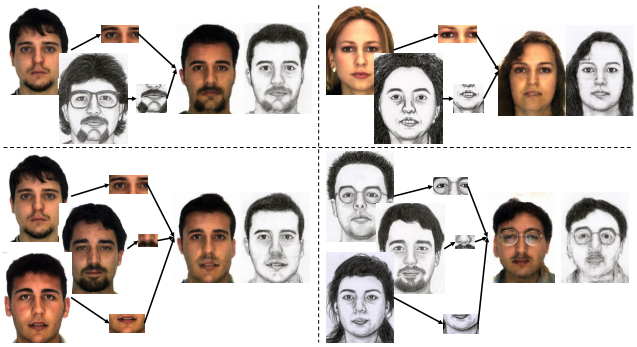

Figure 8: Examples of generated faces/sketches from multiple patches, which are from different people and/or different domains. Four examples are displayed in a 2-by-2 matrix. In each cell, the original faces and sketches are given on the left. The patches are extracted from where indicated by the arrows. The right are generated face/sketch pairs.

and chin. We adopt the off-the-shelf face landmark detection method [8, 3] to detect and localize those facial elements. An unsuccessful detection indicates a failure of face generation. Therefore, FRR is the ratio between the numbers of successfully detected and total generated faces. Fig. 10 (left) shows FRR of each method, computed from 300 generated faces using patches with different missing percentages. We observe that when the missing percentage is larger than 50%, Pix2Pix fails to generate reasonable faces while inpainting and r-BTN maintain high and similar FRR. However, we recall from Fig. 7 that inpainting results are not photo-realistic as r-BTN although they are both capable of preserving the facial structure. In this case, the relative discrimination score (RDS) is designed to further evaluate the aspect of photo-reality.

Relative discrimination score (RDS) aims to estimate the photo-reality of generated faces. We train a discriminator to distinguish real and generated faces, using the Conv network in Table 1 with a fully-connected layer using the sigmoid activation function. With more epochs, the discriminator output from real faces would be close to one, and that from generated faces should approach zero. If the generated faces are realistic, their discriminator output would be relatively higher and decrease slower with epochs as compared to that of unrealistic faces. Fig. 9 shows the discriminator output of each method during training the discriminator. RDS computes the ratio of area under the curves of generated faces from certain method and real faces. Higher RDS indicates more photo-realistic faces. During training the discriminator, 900 real faces are randomly selected from the training set, and 300 faces are generated from each method. Thus, we have balanced real and generated samples. The area under a curve is computed by trapezoidal numerical integration. Finally, RDS of

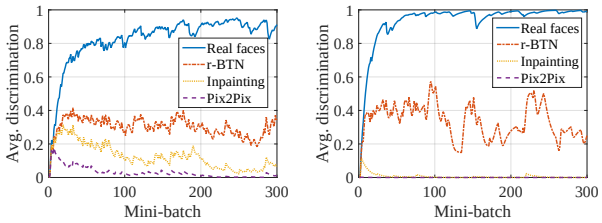

Figure 9: Averaged discriminator output at each mini-batch (30 samples) during training the discriminator that aims to distinguish real and generated faces from Pix2Pix, inpainting, and r-BTN, respectively. The generated faces are from random patches with 10% (left) and 95% (right) missing.

each method with respect to missing percentage is shown in Fig. 10 (right). We observe that although both inpainting and r-BTN demonstrate same level of FRR, the RDS level of r-BTN is much higher than that of inpainting, showing more photo-realistic outputs generated from r-BTN.

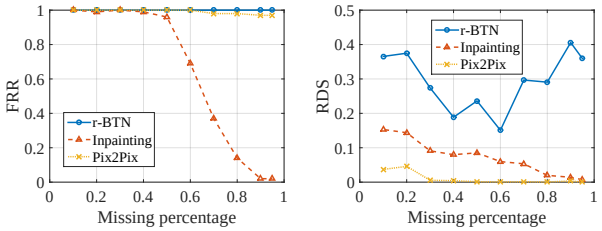

Figure 10: Comparison of different methods on the proposed metrics: FRR (left) and RDS (right).

4.4.2 Similarity/Diversity Evaluation

Intuitively speaking, the generated faces from the patches of the same person should be similar. By contrast, patches from different persons are supposed to yield diverse faces. To verify this property, we collect 50 faces and pick patches of different size around the eyes, the nose, and the mouth. The proposed r-BTN is then applied to generate full faces from those patches. To measure the similarity/diversity between generated faces, we utilize the pre-trained VGG-Face [15] model to extract high-level features and compute their Euclidean distance. We perform two comparisons: 1) self comparison (similarity) and 2) mutual comparison (diversity), conducting on faces generated from patches of the same and different persons, respectively.

Fig. 11 (left) shows the averaged distance and standard deviation with respect to missing percentage. The blue circles shows the results of self comparison, and the red triangles denote mutual comparison. With lower missing percentage, *e.g.*, 0.1 to 0.6, the generated faces preserve relatively high intra-class (same person) similarity and inter-

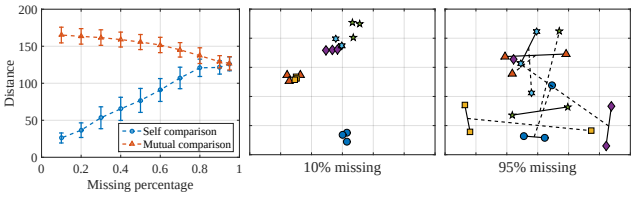

Figure 11: Left: Evaluation of similarity/diversity with increasing missing percentage. Circles/triangles are averaged distances of self comparison and mutual comparison, respectively. The bars indicate corresponding standard deviation. Middle and right: High-level feature (reduced to 2-D by PCA) of generated faces at missing percentage of 10% and 95%, respectively. Different marker types indicated different persons. There are three same markers for type (person), denoting the generated faces from patches around left eye, right eye, and mouth. In the right figure, solid lines connect the faces generated from eyes, and the dashed lines connect to the faces generated from mouth.

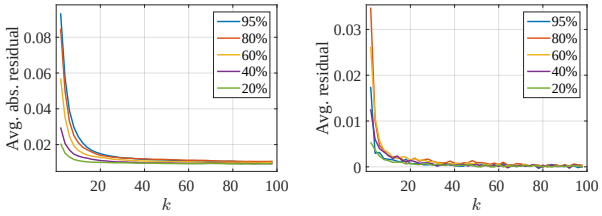

Figure 12: Convergence evaluation of the proposed r-BTN. Averaged absolute (left) and average (right) of residual with respect to iteration k are shown at missing percentage of 95%, 80%, 60%, 40%, and 20%, respectively.

class (different persons) diversity. As the missing percentage increases, the two curves eventually intersect, indicating the generated faces from very small patches (*e.g.*, 95% missing) have lost the identity of the original face. Interestingly, we discover that the generated faces from either the left or right eye of the same person still tend to be more similar as compared to those generated from nose/mouth as illustrated in Fig. 11 (right). This discovery is well in line with the quality of different biometrics where studies have shown eyes to carry more valuable cues than nose or mouth in face recognition tasks. This finding, from another perspective, demonstrates the high effectiveness of r-BTN in generating high-fidelity and realistic faces/sketches.

4.4.3 Convergence of Recursive Generation

Will the generated faces/sketches converge to a certain point? How many iterations are sufficient to achieve a photo-realistic result? This section mainly answers these two questions.

We first define the residual in the face domain between subsequent iterations as $r^{k+1} = \left(x_{\mathcal{I}}^{k+1} - x_{\mathcal{I}}^{k}\right)$, where $x_{\mathcal{I}}^{k}$ and $x_{\mathcal{I}}^{k+1}$ denote the kth and k+1th generated results. The convergence is mainly evaluated by calculating the averaged residual on testing samples (*i.e.*, 300 samples generated from Sec. 4.4.1 with different missing percentage) with respect to k as shown in Fig. 12 (right). However, the average residual is not sufficient to demonstrate the convergence because some pixels may significantly increase while the other decrease with the same level. In this case, we calculate the averaged absolute residual which illustrate the changing amplitude as shown in Fig. 12 (left).

With more iterations, the averaged residual approaches zero while the averaged absolute residual stabilizes at a small value. This well demonstrates that the generated faces are stable. In addition, from the experiments (*e.g.*, Figs. 1 and 2), the generated faces/sketches will not significantly change after 20 iterations. Therefore, we could empirically conclude that the recursive generation will converge to certain face/sketch for a given patch.

5. Discussion and Future works

In this paper, we proposed and solved the challenging task of cross-domain face generation with large missing area. A novel recursive generation method by bidirectional transformation networks (r-BTN) was proposed to achieve high-fidelity and consistent face/sketch even with as large as 95% missing area. We demonstrated the effectiveness of r-BTN by comparing to some potential solutions like pix2pix and inpainting. However, r-BTN requires well-aligned faces/sketches. Otherwise, the generated results may not be visually pleasing because the network would fail to localize facial components and thus missing their geometric structure.

In the future, we plan to improve the proposed r-BTN from three perspectives: 1) concatenating a face calibration mechanism to r-BTN to battle against the alignment problem, 2) extending this work to be unsupervised like [11, 21] to alleviate the requirement for paired dataset, and 3) generalizing r-BTN as a framework for cross-domain transformation, especially with large missing area, and further evaluating the performance on other datasets.

References

[1] H. S. Bhatt, S. Bharadwaj, R. Singh, and M. Vatsa. Memetically optimized MCWLD for matching sketches with digital face images. *IEEE Transactions on Information Forensics and Security*, 7(5):1522–1535, 2012. 5

[2] A. Criminisi, P. Pérez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing*, 13(9):1200–1212, 2004. 2

[3] Dlib C++ Library. http://dlib.net/. [Online]. 7

[4] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *International Conference on Computer Vision*, volume 2, pages 1033–1038. IEEE, 1999. 2

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 2

[6] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016. 2, 3, 5

[7] A. Kasinski, A. Florek, and A. Schmidt. The PUT face database. *Image Processing and Communications*, 13(3-4):59–64, 2008. 5

[8] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014. 7

[9] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[10] S. Klum, H. Han, A. K. Jain, and B. Klare. Sketch based face recognition: Forensic vs. composite sketches. In *International Conference on Biometrics*, pages 1–8. IEEE, 2013.

[11] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 469–477, 2016. 8

[12] D. S. Ma, J. Correll, and B. Wittenbrink. The Chicago face database: A free stimulus set of faces and norming data. *Behavior Research Methods*, 47(4):1122–1135, 2015. 5

[13] A. Martinez and R. Benavente. The AR face database, 1998. *Computer Vision Center, Technical Report*, 3, 2007. 5

[14] S. Ouyang, T. M. Hospedales, Y.-Z. Song, and X. Li. ForgetMeNot: memory-aware forensic facial sketch matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5571–5579, 2016.

[15] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, page 6, 2015. 7

[16] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016. 2, 5

[17] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000. 5

[18] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2

[19] J. Shen and T. F. Chan. Mathematical models for local non-texture inpaintings. *SIAM Journal on Applied Mathematics*, 62(3):1019–1043, 2002. 2

[20] Y. Song, L. Bao, Q. Yang, and M.-H. Yang. Real-time exemplar-based face sketch synthesis. In *European Conference on Computer Vision*, pages 800–813. Springer, 2014. 2

[21] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016. 2, 8

[22] X. Tang and X. Wang. Face sketch synthesis and recognition. In *IEEE International Conference on Computer Vision*, pages 687–694. IEEE, 2003. 2

[23] T. F. Vieira, A. Bottino, A. Laurentini, and M. De Simone. Detecting siblings in image pairs. *The Visual Computer*, 30(12):1333–1345, 2014. 5

[24] X. Wang and X. Tang. Face photo-sketch synthesis and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):1955–1967, 2009. 2, 5

[25] X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision*, pages 776–791. Springer, 2016. 2

[26] R. Yeh, C. Chen, T. Y. Lim, M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with perceptual and contextual losses. *arXiv preprint arXiv:1607.07539*, 2016. 2

[27] W. Zhang, X. Wang, and X. Tang. Lighting and pose robust face sketch synthesis. In *European Conference on Computer Vision*, pages 420–433. Springer, 2010. 2

[28] W. Zhang, X. Wang, and X. Tang. Coupled information-theoretic encoding for face photo-sketch recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 513–520. IEEE, 2011. 5

[29] Z. Zhang, Y. Song, and H. Qi. Age progression/regression by conditional adversarial autoencoder. *arXiv preprint arXiv:1702.08423*, 2017. 2

[30] H. Zhou, Z. Kuang, and K.-Y. K. Wong. Markov weight fields for face sketch synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1091–1097. IEEE, 2012. 2

[31] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016. 2

Supplementary

Sketch Generation

Because of limited face/sketch pairs, we collect frontal face images and generate sketches from them by Pix2Pix. Therefore, the limitation of available face/sketch pairs could be relaxed to certain extent. Fig. 13 shows the generated sketches from some collected face images. Although the resolution of generated sketches is not as hight as the artists' painting, they could preserve the identity of corresponding faces. These generated face/sketch pairs will be added to the training dataset.

Figure 13: The collected faces and their sketches generated through Pix2Pix.

More Results from r-BTN

Fig. 14 displays more results generated from eyes, nose, mouth, and random regions using the proposed r-BTN.

Visualizing the Quantitative Comparison

In the section 4.4.1 of the original paper, we provide the statistical analysis on the quality of generated faces from three methods — Pix2Pix, inpainting, and r-BTN. Fig. 15 visualizes the comparison through two examples. The proposed r-BTN generates higher fidelity and more smooth results.

Visualizing the Similarity

In the section 4.4.2, we show that the generated faces from either eye of the same person would present relatively high similarity. Fig. 17 illustrates such similarity.

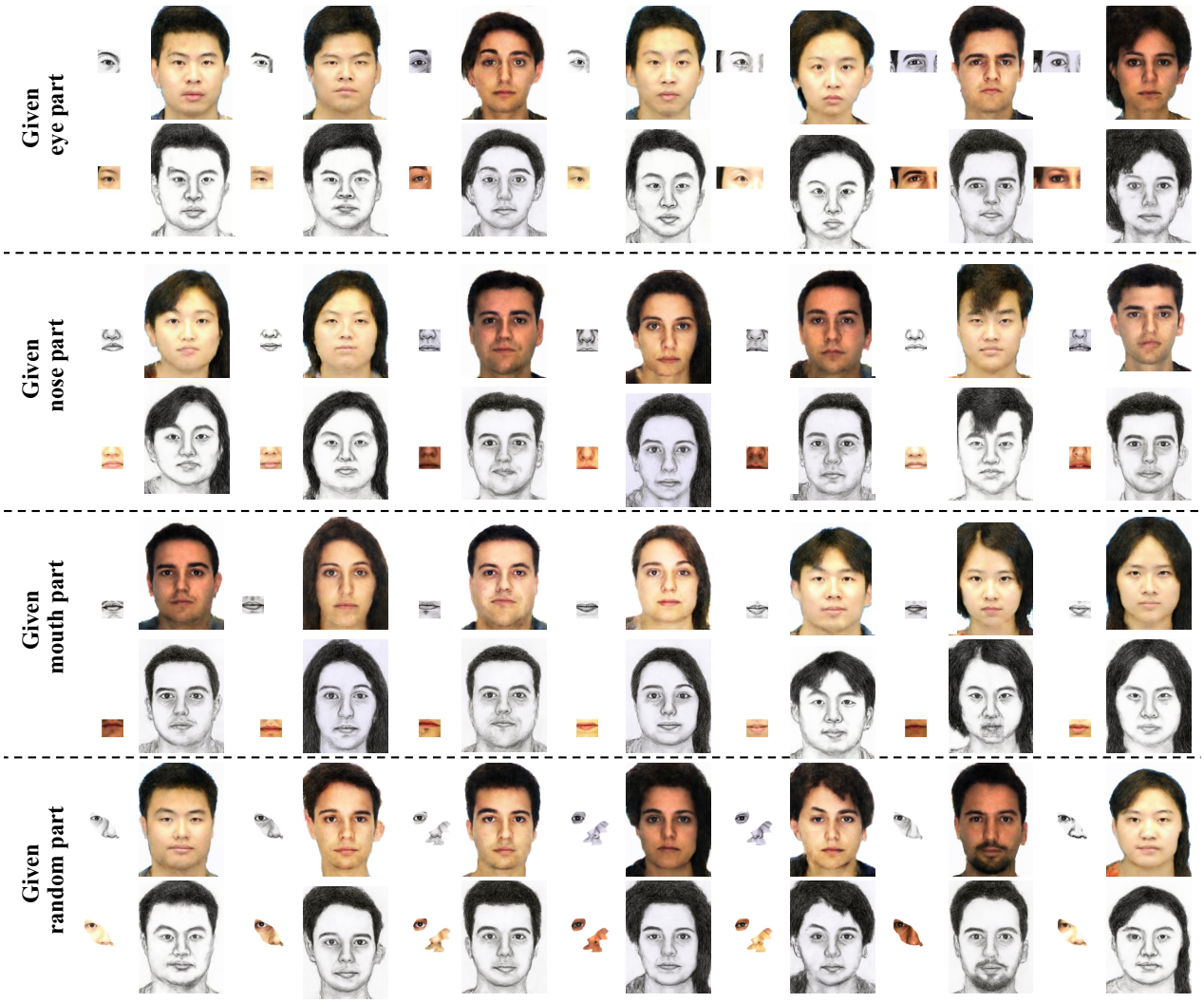

Figure 14: Generated faces/sketches from small patches of eyes, nose, mouth, and random regions by r-BTN.

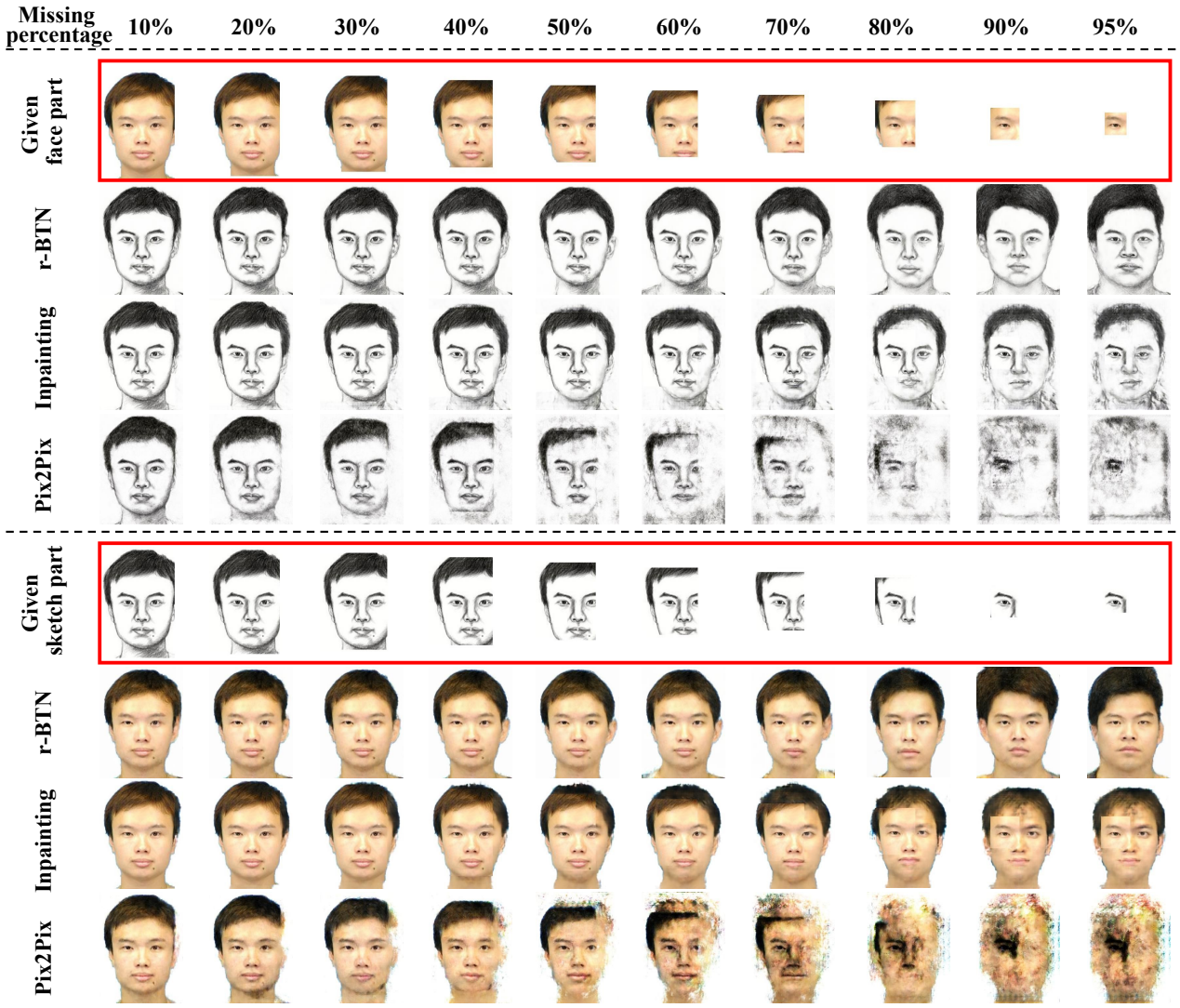

Figure 15: Example 1: Comparison of different methods in generating faces/sketches from patches with different missing percentage. The red boxes indicate the given face/sketch patches. The rest rows are correspondingly generated sketches/faces by the denoted methods. Please zoom in to see the details for small missing percentages.

Figure 16: Example 2: Comparison of different methods in generating faces/sketches from patches with different missing percentage. The red boxes indicate the given face/sketch patches. The rest rows are correspondingly generated sketches/faces by the denoted methods. Please zoom in to see the details for small missing percentages.

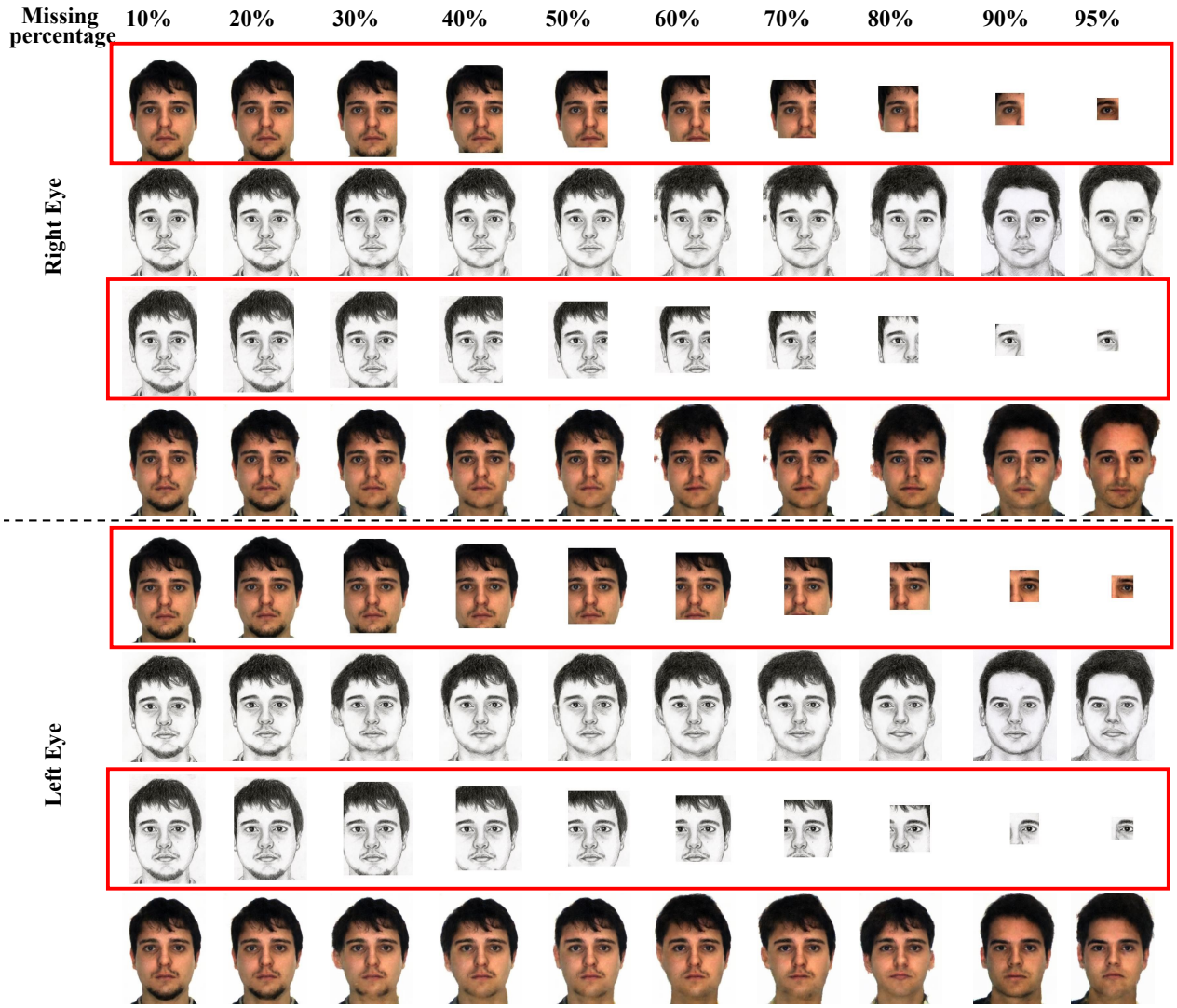

Figure 17: Given face/sketch patches (red boxes) from the same person but with different missing percentage, the proposed r-BTN generates similar sketches/faces. Please note that the given patches are mainly from one of the eyes.