The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

r-BTN: Cross-Domain Face Composite and Synthesis from Limited Facial Patches

Yang Song, Zhifei Zhang, Hairong Qi
Department of Electrical Engineering and Computer Science
University of Tennessee, Knoxville, TN 37996, USA
{ysong18, zzhang61, hqi@utk.edu}

Abstract

Recent face composite and synthesis related works have shown promising results in generating realistic face images from deep convolutional networks. However, these works either do not generate consistent results when the constituent patches contain large domain variations (i.e., from face and sketch domains) or cannot generate high-resolution images with limited facial patches (e.g., the inpainting approach tends to blur the generated region when the missing area is more than 50%). Motivated by the mental imagery and simulation in human cognition, 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 domain variations make it difficult to generate satisfactory results using a unidirectional cross-domain learning structure. We explore that the bidirectional transformation network can lead to the consistent result by minimizing the forward and backward errors in the cross-domain scenario. 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 to 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.

Introduction

We start by asking an interesting yet challenging question, "If provided with limited facial patches from sketch/face domains where human beings may be able to generate a real face image in brain (Kosslyn, Thompson, and Ganis 2006) as shown in Fig. 1, can advanced computer vision techniques generate the whole face image?" Recently, several face synthesis methods built on neural networks have emerged (Zhang, Song, and Qi 2017a; Sangkloy et al. 2016). These methods can generate face/sketch images based on whole face information from one domain. However, how to generate realistic faces/sketches that are consistent to the given sketch/face patches is still a challenging task because large missing area could lead to blurry generated images. In addition, some existing methods (e.g., Photofit (Photofit)) synthesize faces by stitching patches from cross domains which deteriorates the consistency and photo-reality. It is still unclear how to preserve the color/domain consistency between patches with large domain variations.

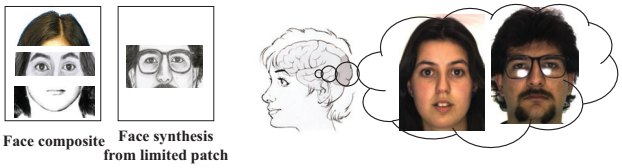

Figure 1: Illustration of face composite based on cross-domain patches and face synthesis from limited facial patch.

In this paper, we study the above-mentioned problems which would play a key role in many applications, such as face image stitching, face blending, face editing, etc. To the best of our knowledge, 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 synthesis/transformation and image inpainting.

The face/sketch synthesis works (Wang and Tang 2009; Tang and Wang 2003; Zhou, Kuang, and Wong 2012; Song et al. 2014) 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) (Goodfellow et al. 2014) has shown impressive performance in face generation (Radford, Metz, and Chintala 2015; Zhang, Song, and Qi 2017a), domain transformation (Zhu et al. 2016; Isola et al. 2016), and inpainting (Yeh et al. 2016; Pathak et al. 2016). 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.

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 (i.e., face composite) to output a realistic and consistent face in a generative manner.

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 (Wang and Tang 2009; Zhang, Wang, and Tang 2010; Zhou, Kuang, and Wong 2012) are matching-based, which synthesize faces from best matched patches by searching from the training dataset. For example, (Wang and Tang 2009) 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 (Taigman, Polyak, and Wolf 2016; Isola et al. 2016) are mainly developed from encoder-decoder networks and adversarial generative networks. For example, (Isola et al. 2016; Zhu et al. 2017) 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 part of an image. Most inpainting methods (Efros and Leung 1999; Shen and Chan 2002; Criminisi, Pérez, and Toyama 2004) estimate the missing part based on surrounding pixels, and therefore are not suitable for filling in large missing areas. Although some recent works (Yeh et al. 2016; Pathak et al. 2016) 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 (Zhang, Song, and Qi 2017a; Yan et al. 2016) 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.

The Bidirectional Transformation Network

In this section, we first elaborate on the benefit of the proposed BTN through a comparison with unidirectional transformations. 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 whole face/sketch pairs. The testing stage recursively generates the whole face/sketch from given small sketch/face patches.

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., (Isola et al. 2016), 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.

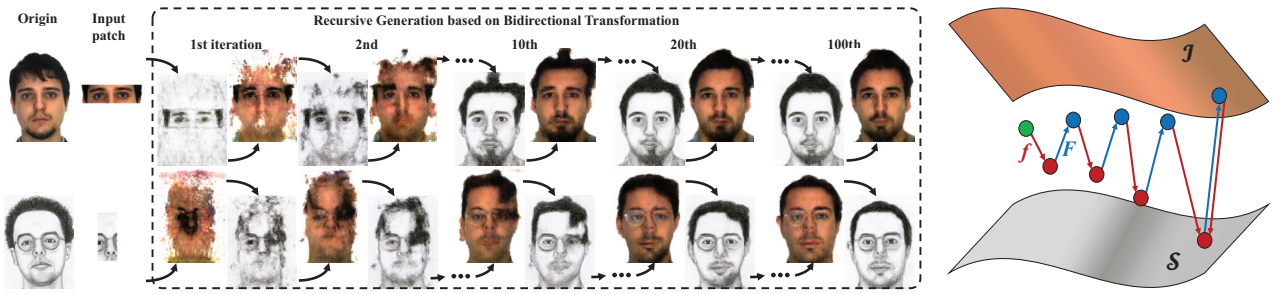

Figure 2: Examples of recursive generation from small patches by the bidirectional transformation network. Left: Original face/sketch and the corresponding input patches extracted from them. Inside of the dashed box demonstrates the generated face/sketch at different iteration steps. Right: Illustration 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.

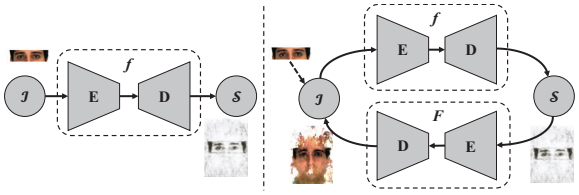

Figure 3: Comparison of unidirectional and bidirectional transformations between $\mathcal{I}$ and $\mathcal{S}$ domains. E and D are the encoder-decoder networks. The patch (eyes) generates the sketch, and then the sketch is transformed back where facial outline has been estimated.

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.

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

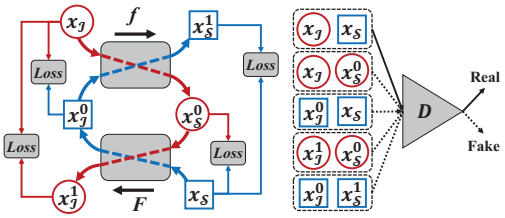

Figure 4: Training flow of the bidirectional transformation network. $x_{\mathcal{I}}$ and $x_{\mathcal{S}}$ are the real face/sketch pair. Red and blue arrows denote the transformation paths of $x_{\mathcal{I}}$ and $x_{\mathcal{S}}$, respectively. The transformation functions f and F could be encoder-decoder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

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 generated faces or sketches as Eq. 1.

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

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

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

where

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

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

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

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

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

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

where

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

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

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 demonstrates the

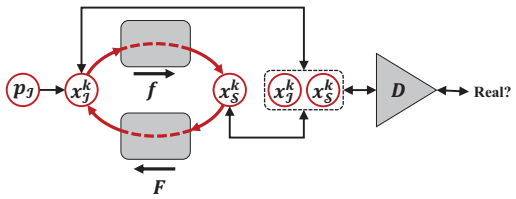

Figure 5: Testing flow of r-BTN, assuming a face patch $p_{\mathcal{I}}$ as the input. At step k, the generated face is $x_{\mathcal{I}}^k$. Replacing the corresponding area of $x_{\mathcal{I}}^k$ by the patch $p_{\mathcal{I}}$ and transforming $x_{\mathcal{I}}^k$ to $x_{\mathcal{S}}^k$, we get a face/sketch pair $\left(x_{\mathcal{I}}^k, x_{\mathcal{S}}^k\right)$. 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 $x_{\mathcal{I}}^{k+1}$.

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

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

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

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

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

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.

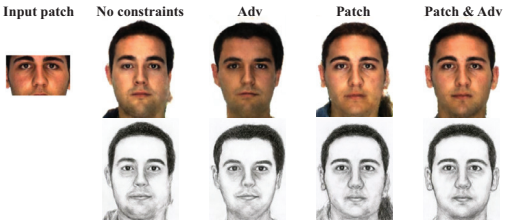

Figure 6: Comparison of generated results with/without the given patch (Patch) and adversarial (Adv) constraints.

Experiment and Results

Data Collection

We collect 1,577 face/sketch pairs from the datasets CUHK (Wang and Tang 2009), CUFSF (Zhang, Wang, and Tang 2011), AR (Martinez and Benavente 2007), FERET (Phillips et al. 2000), and IIIT-D (Bhatt et al. 2012). Because the dataset with face/sketch pairs is limited, we train a face to sketch transformation network based on Pix2Pix (Isola et al. 2016) to generate sketches from faces. We collect frontal face images with uniform background and controlled illumination from datasets CFD (Ma, Correll, and Wittenbrink 2015), SiblingsDB (Vieira et al. 2014), and PUT (Kasinski, Florek, and Schmidt 2008), 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.

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. 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 (Isola et al. 2016), each Conv layer is concatenated to its symmetrically corresponding Deconv layer, thus more details bypass the bottleneck. In the training, we adopt ADAM (Kingma and Ba 2014) ($\alpha = 0.0002,\ \beta = 0.5$). 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. Details are shown in supplementary materials. 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 testing stage. 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.

Qualitative Evaluation

Face Composite 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. 7, demonstrating the great versatility of r-BTN. We again observe the strong consistency and fidelity between the generated face/sketch pairs.

Face Synthesis from Limited Facial Patches The results generated by proposed r-BTN with respect to different missing percentage are shown in Fig. 8. From the result, it demonstrated that the proposed method cannot preserve the identity when the missing percentage is more than 70%. This phenomenon is consistent with human cognitive. For human beginnings, if only providing limited information, it is still hard to imagine a unique result. We also compare the proposed r-BTN with Pix2Pix (Isola et al. 2016) and image inpainting (Pathak et al. 2016). The inpainting method compared in this paper is modified from (Pathak et al. 2016) to achieve cross-domain inpainting. Specifically, the inputs are faces/sketches with random mask (20%∼80% masked), and the outputs are the 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. 9. 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. More results are shown in supplementary materials.

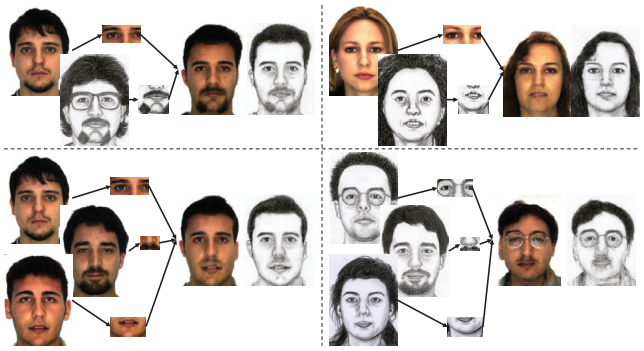

Figure 7: Examples of generated faces/sketches from multiple patches, which are from different people and/or different domains. Four examples are displayed in a 2-by-2 matrix. In each cell, the original faces and sketches are given on the left. The patches are extracted from where indicated by the arrows. The right are generated face/sketch pairs.

Quantitative Analysis

Evaluation Metrics To numerically evaluate the quality of generated faces, we design the metric named "face recognition rate (FRR)". It evaluates whether the generated images present facial elements and geometric structure, i.e., reasonable position of eyebrows, eyes, nose, lips, and chin. We adopt the off-the-shelf face landmark detection method (Kazemi and Sullivan 2014) 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 differ-

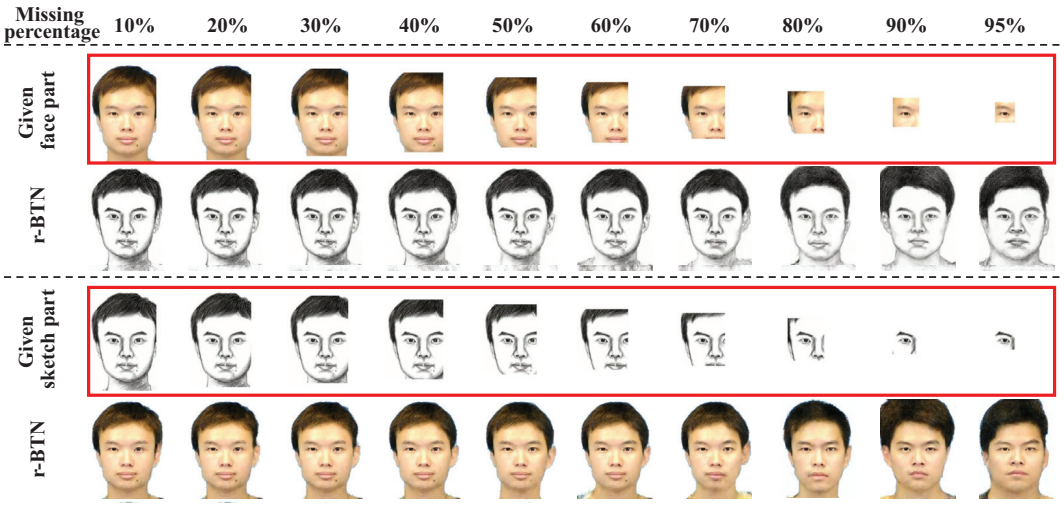

Figure 8: Comparison in generating faces/sketches from patches with different missing percentage. The red boxes indicate the given face/sketch patches. The rest rows are correspondingly generated sketches/faces.

ent 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. 9 that inpainting results are not photo-realistic as r-BTN although they are both capable of preserving the facial structure.

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 with different missing percentage) with respect to k as shown in Fig. 10 (middle). 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. 10 (right).

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., Fig. 2 and 8), 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.

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 (Parkhi, Vedaldi, and Zisserman 2015) 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-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.

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

Figure 9: 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.

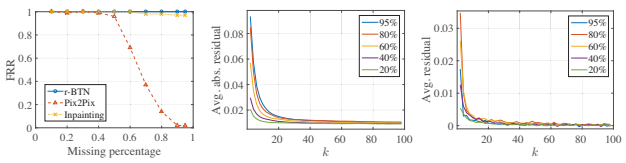

Figure 10: Left:Comparison of different methods on the proposed metrics: FRR. Middle and Right: Convergence evaluation of the proposed r-BTN. Averaged absolute (middle) and average (right) of residual with respect to iteration k are shown at missing percentage of 95%, 80%, 60%, 40%, and 20%, respectively.

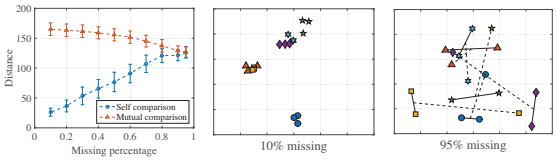

Figure 11: Left: Evaluation of similarity/diversity with increasing missing percentage. The bars indicate corresponding standard deviation. Middle and Right: High-level feature of generated faces at missing percentage of 10% and 95%, respectively. There are three same markers for type (person), denoting the generated faces from patches around left eye, right eye, and mouth. Solid lines connect the faces generated from eyes, and the dashed lines connect to the faces generated from mouth.

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 four perspectives: 1) concatenating a face calibration mechanism to r-BTN to battle against the alignment problem, 2) extending this work to be unsupervised like (Du, Abdalmageed, and Doermann 2013; Taigman, Polyak, and Wolf 2016) to alleviate the requirement for paired dataset, 3) generalizing r-BTN as a framework for cross-domain transformation, especially with large missing area, and further evaluating the performance on other datasets (Zhang, Song, and Qi 2017b), and 4) adpting this for mobile network application (Li et al. 2016).

References

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

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

Du, X.; Abdalmageed, W.; and Doermann, D. 2013. Large-scale signature matching using multi-stage hashing. In *IEEE Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, 976–980.

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

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

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2016. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*.

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

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

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

Kosslyn, S. M.; Thompson, W. L.; and Ganis, G. 2006. *The case for mental imagery*. Oxford University Press.

Li, D.; Salonidis, T.; Desai, N. V.; and Chuah, M. C. 2016. Deepcham: Collaborative edge-mediated adaptive deep learning for mobile object recognition. In *IEEE Edge Computing (SEC), IEEE/ACM Symposium on*, 64–76.

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

Martinez, A., and Benavente, R. 2007. The AR face database. *Computer Vision Center, Technical Report* 3.

Parkhi, O. M.; Vedaldi, A.; and Zisserman, A. 2015. Deep face recognition. In *British Machine Vision Conference*, 6.

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

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

Photofit. http://www.open.edu/openlearn/body-mind/photofit-me. [Online].

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

Sangkloy, P.; Lu, J.; Fang, C.; Yu, F.; and Hays, J. 2016. Scribbler: Controlling deep image synthesis with sketch and color. *arXiv preprint arXiv:1612.00835*.

Shen, J., and Chan, T. F. 2002. Mathematical models for local nontexture inpaintings. *SIAM Journal on Applied Mathematics* 62(3):1019–1043.

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

Taigman, Y.; Polyak, A.; and Wolf, L. 2016. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*.

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

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

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

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

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

Zhang, Z.; Song, Y.; and Qi, H. 2017a. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Zhang, Z.; Song, Y.; and Qi, H. 2017b. Stabilizing the conditional adversarial network by decoupled learning. In *ICML Workshop on Implicit Models*.

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

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

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

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

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*.