

FSGANv2: Improved Subject Agnostic Face Swapping and Reenactment

Yuval Nirkin¹, Yosi Keller², and Tal Hassner¹

Abstract—We present Face Swapping GAN (FSGAN) for face swapping and reenactment. Unlike previous work, we offer a subject agnostic swapping scheme that can be applied to pairs of faces without requiring training on those faces. We derive a novel iterative deep learning-based approach for face reenactment which adjusts significant pose and expression variations that can be applied to a single image or a video sequence. For video sequences, we introduce a continuous interpolation of the face views based on reenactment, Delaunay Triangulation, and barycentric coordinates. Occluded face regions are handled by a face completion network. Finally, we use a face blending network for seamless blending of the two faces while preserving the target skin color and lighting conditions. This network uses a novel Poisson blending loss combining Poisson optimization with a perceptual loss. We compare our approach to existing state-of-the-art systems and show our results to be both qualitatively and quantitatively superior. This work describes extensions of the FSGAN method, proposed in an earlier conference version of our work (Nirkin *et al.* 2019), as well as additional experiments and results.

Index Terms—Face swapping, face reenactment, deep learning

1 INTRODUCTION

FACE swapping is the visual transformation of a face from a source image to a target image, such that, the resulting image seamlessly replaces the face appearing in the target image as depicted in Fig. 1. *Face reenactment* (aka *face transfer* or *puppeteering*) utilizes the facial movements and expression of a control face in one video to guide the motions and deformations of a second face appearing in another video or image (Fig. 1). Both tasks have attracted significant research attention due to their applications in entertainment [2], [3], [4], privacy [5], [6], [7], and even training data generation and augmentation. Most contemporary works proposed methods for either swapping or reenactment, but rarely both, relying on underlying 3D face representations to transfer the face appearance. Face shapes were estimated from the input image [8], [9], [10] or kept fixed [10]. The 3D shape was then aligned with the input images and used as a proxy to transfer the image appearance (swapping) or controlling the facial expression and viewpoint (reenactment).

Deep networks were also applied to face swapping and reenactment. In particular, Generative Adversarial Networks (GANs) [11] were shown to successfully generate realistic fake faces images. Conditional GANs (cGANs) [12], [13], [14] were used to transform an image depicting real data from one domain to the other and inspired multiple face reenactment

schemes [15], [16], [17]. The DeepFakes project [18] applied cGANs to face swapping in videos, making swapping widely accessible to nonexperts and receiving substantial public attention. Such methods are able to generate more realistic face images by replacing the classical graphics pipeline by utilizing implicit 3D face representations.

Some methods applied domain separation in latent feature spaces [19], [20], [21], to decompose the identity component of a face from the other traits, such as pose and expression. The identity is encoded as the manifestation of latent feature vectors, resulting in significant information loss and limiting the quality of the synthesized images. Subject-specific approaches [9], [16], [18], [22] are particularly trained for each subject or pair of subjects to be swapped or reenacted. Thus, requiring significant training sets per subject, to achieve reasonable results, limiting their potential usage. A major concern shared by previous face synthesis schemes, particularly the 3D-based methods, is that they all require particular care to handle partially occluded faces.

In this work, we improve upon our previous work [1] in multiple ways. We provide a means for interpolating between face landmarks without relying on 3D information using a face landmarks transformer network. We improve the inpainting generator by adding symmetry and face landmarks cues. We completely revise the preprocessing pipeline and add a postprocessing step, to reduce the jittering and saturation artifacts of our previous method. Finally, we show additional qualitative and quantitative experiments with a new metric for comparing expressions.

To summarize, our contributions are:

- A face landmarks transformer network for interpolating between face landmarks without 3D information.
- Improved inpainting generator that utilizes symmetry and face landmarks cues.

- Yuval Nirkin and Yosi Keller are with the Faculty of Engineering, Bar Ilan University, Ramat Gan 5290002, Israel. E-mail: {yuval.nirkin, yosi.keller}@gmail.com.
- Tal Hassner is with the Computer Science, The Open University of Israel, Ra'anana 4353701, Israel. E-mail: talhassner@gmail.com.

Manuscript received 8 Dec. 2020; revised 24 Jan. 2022; accepted 15 Feb. 2022.
Date of publication 26 Apr. 2022; date of current version 5 Dec. 2022.
(Corresponding author: Yosi Keller.)
Recommended for acceptance by C. Theobalt.
Digital Object Identifier no. 10.1109/TPAMI.2022.3155571



Fig. 1. *Face swapping and reenactment.* Left: Source face swapped onto target. Right: Target video used to control the expressions of the face appearing in the source image. In both cases, our results appear in the middle.

- A demonstration of an additional use case for the new face reenactment method for pose-only face reenactment.
- Completely revised preprocessing and an additional postprocessing step for reducing jittering and saturation artifacts.
- Introduction of a new metric for facial expression comparison.
- Additional quantitative and qualitative experiments and ablation studies using new metrics.

2 RELATED WORK

Methods for manipulating the appearance of face images, particularly for face swapping and reenactment, were originally proposed to address privacy concerns [5], [6], [7], though they are increasingly being used for recreational [3] or entertainment tasks (e.g., [2], [4]).

2.1 3D-Based Methods

One of the earliest face swapping methods [5] required manual involvement, where a fully automatic method was later proposed [23]. The Face2Face approach transferred expressions from the source to target faces [8], by fitting a 3D morphable face model (3DMM) [24], [25] to both faces. The expression attributes of one face were then applied to the other, where particular attention was given to the interior mouth regions. The reenactment method of Suwajanakorn *et al.* [9] synthesized the mouth area using a reconstructed 3D model of (former president) Barack Obama, guided by face landmarks, and using a similar approach for filling the face interior as in Face2Face. The expression of frontal faces was manipulated by Averbuch-Elor *et al.* [26] by transferring the mouth interior from source to the target image using 2D wraps and face landmarks.

Finally, Nirkin *et al.* [10] proposed a face swapping method, demonstrating that 3D face shape estimation is unnecessary for realistic face swaps. Instead, they used a fixed 3D face shape as the proxy. This approach, similar to ours, proposed a face segmentation method, though the scheme was not end-to-end trainable and required particular attention to occluded regions. We show our current results to be superior.

2.2 GAN-Based Methods

GANs [11] were shown to generate fake images having the same distribution as the target domain. Although successful in generating realistic appearances, their training might be unstable, thus limiting their applicability to generating low-

resolution images. Subsequent methods, however, improved the stability of the training process [27], [28]. Karras *et al.* [29] train GANs using a progressive coarse-to-fine multiscale scheme. CycleGAN [30] proposed a cycle consistency loss, allowing to train unsupervised generic transformations between different domains. A cGAN with L_1 loss was applied by Isola *et al.* [13] to derive the pix2pix method, and was shown to produce realistic image synthesis results for applications such as transforming edges to faces.

Finally, attention-based upsampling was proposed to improve generation and dense prediction, to reduce the generation artifacts and improve the prediction accuracy [31], [32].

2.3 Facial Manipulation Using GANs

GANs were used for high resolution image-to-image translation by pix2pixHD [14], that applied a multiscale cGAN architecture, and a perceptual loss [33] using a pre-trained VGG network that improved the results. GANimation [15] proposed a dual generator cGAN architecture conditioned on emotion action units that generate an additional attention map. This map was used to interpolate between the reenacted and original images to preserve the background. The GANotation [17] deep facial reenactment scheme was driven by face landmarks, where the images were generated progressively using a triple consistency loss. The images were first frontalized using landmarks before further processing.

A hybrid 3D/deep method was proposed by Kim *et al.* [22], that render a reconstructed 3DMM of a specific subject using a classical graphic pipeline. The rendered image is then processed by a generator network, trained to map synthetic rendered views of each subject to photo-realistic images. Finally, feature disentanglement was proposed for face manipulation. RSGAN [20] disentangles the latent face and hair representations, while FSNet [21] proposed a latent space which separates the identity and geometric attributes, such as facial pose and expression.

Zakharov *et al.* [34] derived a few-shot face reenactment approach comprised of two parts: a meta-learning stage where the networks are trained using the face images of multiple subjects, and a task-specific phase, where the network is fine-tuned using the images of a particular person. In both stages, an encoder network learns to extract identity cues, and a generator CNN learns to map this information alongside the face landmarks to generate novel views.

Video dubbing [35], [36] was applied using visual and audio information, respectively. The first used a recurrent GAN to preserve the expression style of the target. The second, extracted deep audio features which were encoded as

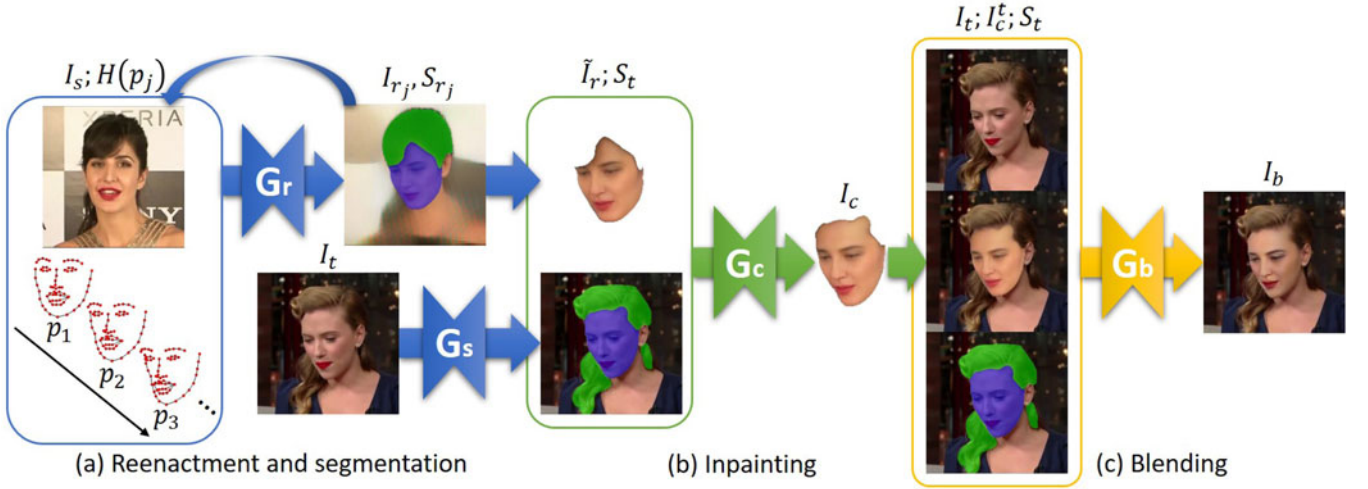


Fig. 2. Overview of the proposed FSGAN approach. (a) The recurrent reenactment generator, G_r , and the segmentation generator, G_s . G_r estimates the reenacted face, F_r , while G_s estimates the face and hair segmentation mask, S_t , of the target image, I_t . (b) The inpainting generator, G_c , inpaints the missing parts of F_r based on S_t to estimate the completed reenacted face, F_c . (c) The blending generator, G_b , blends F_c and I_t , using the segmentation mask, S_t .

expression coefficients. To generate the final rendering, both methods used neural rendering to translate classic renderings into realistic ones using an additional person-specific generator. Another kind of video dubbing was performed by text editing [36], [37], [38], Fried *et al.* [37] by mapping input text to 3DMM pose and expression which was used to recurrently render the lower part of the face, and Yao *et al.* [38] suggested an interactive text editing using fast phoneme matching and neural rendering.

Ha *et al.* [39] suggested a face reenactment framework using a few-shot setting with a landmark transformer that adjusts the shape and blending components using an image attention block. Li *et al.* [40] introduced a face swapping method that preserves the attributes of the source face by integrating them with the generator in a multi-scale scheme using attention blocks. In addition, occlusions are handled in an unsupervised way by comparing the reconstructed and target images.

Recently, Naruniec *et al.* [41] proposed a face swapping method at megapixel resolution using an encoder-decoder with contrast and light-preserving blending and landmark stabilization. Zakharov *et al.* [42] introduced a one-shot neural rendering-based face reenactment method that can perform in real-time. They achieve this by first rendering a coarse image and then combining it with a warped texture image that is generated offline.

3 FACE SWAPPING GAN

We introduce the Face Swapping GAN (FSGAN) whose overview is illustrated in Fig. 2. Let I_s and I_t be the source and target face images, respectively, representing the faces $F_s \in I_s$ and $F_t \in I_t$. We aim to synthesize a new image based on I_t , such that F_t is seamlessly replaced by F_s while retaining the same pose and expression.

The FSGAN consists of three major components. The first, detailed in Section 3.5 and shown in Fig. 2a, consists of the reenactment generator, G_r , and the segmentation CNN, G_s . The input to G_r are the heatmaps encoding the facial landmarks of F_t , and generates the reenacted image, I_r .

such that F_r denotes F_s having the same pose and expression as F_t . It also computes S_r ; that is, the segmentation mask of F_r . The G_s component computes the segmentations mask of the face and hair of F_t .

Given the reenacted image, I_r , there might be missing face parts, as demonstrated in Figs. 2a and 2b. Hence, we apply the face inpainting network, G_c , detailed in Section 3.7 using the segmentation, S_t , to inpaint the missing parts. The final phase of the FSGAN, shown in Fig. 2c and detailed in Section 3.8, is the blending of the completed face, F_c , into the target image, I_t , to derive the final face swapping result.

The architecture of our face segmentation network, G_s , is based on the U-Net [43], with bilinear interpolation for upsampling, while all other generators — G_r , G_c , and G_b — are based on the pix2pixHD architecture [14], utilizing a coarse-to-fine generator, and a multi-scale discriminator.

For the global generator, we use a U-Net with bottleneck blocks [44] instead of simple convolutions, and summation instead of concatenation. The summation allows for smaller channel dimensions than by using concatenation, while bottleneck blocks improve the convergence of large generator models [45]. As with the segmentation network, we use bilinear interpolation for upsampling in both the global generator and the enhancers. The actual number of layers differs per each generator, and the architectures are detailed in Section 3.2.

Following previous results [16], training subject-agnostic face reenactment is non-trivial and might fail when applied to face images related by large poses. For that, we propose to break-down large pose changes into smaller reenactment steps and interpolate between the closest available source images corresponding to the target's pose. These steps are detailed in the following sections.

3.1 Detection and Tracking

Each video is first processed by the dual-shot face detector (DSFD) [46] that is more accurate than SFD [47], which was used in our previous work. The detections are grouped into sequences by computing the IoUs of the detections in successive frames, where $\text{IoUs} > 0.75$ are grouped together in

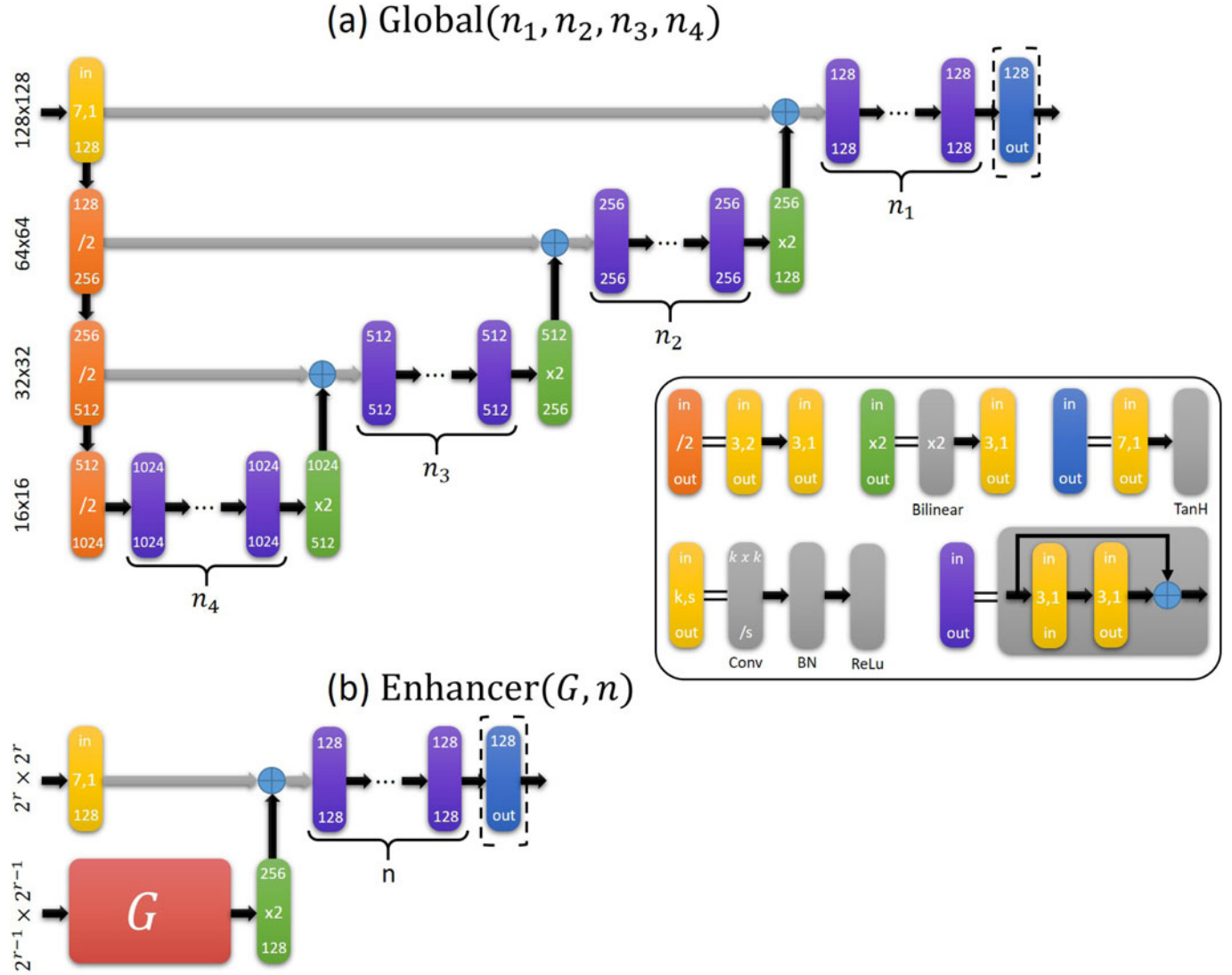


Fig. 3. *Generator architectures.* (a) The global generator is based on a residual variant of the U-Net [43] CNN, using a number of bottleneck layers per resolution. We replace the simple convolutions with bottleneck blocks (in purple), the concatenation with summation (plus sign), and the deconvolutions with bilinear upsampling following by a convolution. (b) The enhancer utilizes a submodule and a number of bottleneck layers. The last output block (in blue) is only used in the enhancer of the finest resolution.

the same sequence. The facial expressions were tracked using 2D face landmarks [48], trained on the WLFV dataset, consisting of 98 points per face. In our previous work, we used 2D and 3D landmarks of 68 points [49].

Even when using state-of-the-art face detection and landmarks extraction schemes, the bounding boxes and landmarks point are not sub-pixel accurate, causing inconsistencies between subsequent frames, resulting in noticeable jittering, an apparent issue of our previous method. Temporal averaging might help reduce the jittering but introduces a noticeable lag. For that, we extend the 1 ϵ filter [50], based on the fact that humans are more sensitive to jittering added to small motions, and more sensitive to lags, when there is a large motion.

We estimate the motion following Casiez *et al.* [50], but use it as per-frame weights for a 1D temporal averaging filter. The larger the motion, the less averaging is applied. For the bounding boxes, we apply the smoothing to the center and box dimensions separately, and for the face landmarks we apply the smoothing to each face part separately.

3.2 Generator Architecture

The generators G_r , G_c , and G_b follow the pix2pixHD design, as depicted in Fig. 3. The global generator, Global(n_1, n_2, n_3), is defined with respect to the number of bottleneck blocks n_1, n_2, n_3 (shown in purple), corresponding to each of the three scales. The enhancer, Enhancer(G, n), is defined with respect to its submodule and n bottleneck blocks. Our reenactment generator and completion generators are

$$G_r = G_c = \text{Enhancer}(\text{Global}(2, 2, 3), 2), \quad (1)$$

and the blending generator is

$$G_b = \text{Enhancer}(\text{Global}(1, 1, 1), 1). \quad (2)$$

3.3 Training Losses

3.3.1 Domain Specific Perceptual Loss

The perceptual loss [33], that has been widely used in recent works in face synthesis [17], outdoor scenes [14], and super resolution [51], is applied to capture the fine details of the

face texture. A feature maps of a pretrained VGG network is used to compare the high frequency details using a euclidean distance. Following others [52], [53], who show that the effectiveness of perceptual loss can be improved by fine-tuning on specific domains, we propose to train the VGG-19 CNN [54] used for perceptual loss using multiple face recognition and attribute classification datasets, instead of a generic ImageNet-based CNN. This allows to fully capture the distinct details inherent in face images. Let $F_i \in \mathbb{R}^{C_i \times H_i \times W_i}$ be the feature map of the i th layer of the VGG-19 network, the perceptual loss is thus given by

$$\mathcal{L}_{perc}(x, y) = \sum_{i=1}^n \frac{1}{C_i H_i W_i} \|F_i(x) - F_i(y)\|_1, \quad (3)$$

where C_i is the number of channels, and H_i and W_i are the height and width dimensions.

3.3.2 Reconstruction Loss

While the perceptual loss, as in Section 3.3.1, captures the high frequencies well, generators trained using only that loss, often produce images having inaccurate colors, corresponding to the erroneous reconstruction of the low-frequency image content. Hence, we also applied a pixelwise L_1 loss to the generator

$$\mathcal{L}_{pixel}(x, y) = \|x - y\|_1, \quad (4)$$

and the overall loss is given by

$$\mathcal{L}_{rec}(x, y) = \lambda_{perc} \mathcal{L}_{perc}(x, y) + \lambda_{pixel} \mathcal{L}_{pixel}(x, y). \quad (5)$$

The loss in Eq. (5) was applied to all image generators.

3.3.3 Adversarial Loss

To further improve the realism of the synthesized images, we applied the pix2pixHD adversarial objective [14], which utilizes a multi-scale discriminator comprising of multiple discriminators, D_1, D_2, \dots, D_n , each operating on a different image resolution. For a generator, G , and a multi-scale discriminator, D , our adversarial loss is

$$\mathcal{L}_{adv}(G, D) = \min_G \max_{D_1, \dots, D_n} \sum_{i=1}^n \mathcal{L}_{GAN}(G, D_i), \quad (6)$$

where

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}_{(x,y)} [\log D(x, y)] + \mathbb{E}_x [\log (1 - D(x, G(x)))]. \quad (7)$$

3.4 Face Segmentation

The goal of our face segmentation is to distinguish between the internal part of the face and the background, including the rest of the head. In practice, we found that adding information regarding the hair improves the segmentation accuracy. Given an image, $I \in \mathbb{R}^{3 \times H \times W}$, we define the face segmentation generator, $G_s : \mathbb{R}^{3 \times H \times W} \rightarrow \mathbb{R}^{3 \times H \times W}$. The output segmentation mask predicts three classes: background, face, and hair. G_s was trained using the cross-entropy loss.

To improve stability and accuracy, the training images are augmented by random rotations of $[-30^\circ, 30^\circ]$, random

color jittering (brightness, contrast, saturation, and hue), random horizontal flip, and random gaussian blur with a kernel radius of 5 and standard deviation of 1.1.

3.5 Face Reenactment

As in [1], the purpose of the face reenactment generator, G_r , is to align a *source* face according to the pose and expression of a *target* face, without subject-specific training. Similarly to [1], significant poses are decomposed into smaller, more manageable steps, in which G_r is applied iteratively, but instead of relying on 3D landmarks, we propose a landmarks transformer network to perform the interpolation between the landmarks in each step, with more details on this in Section 3.5.1. A single reenactment iteration is depicted in Fig. 4.

Additionally, the newly proposed G_r benefit from improved preprocessing, which allows for extracting higher quality cues from the same training videos, a more accurate landmarks extraction method [48], and improved heatmap encoding method that is discussed in Section 3.5.2.

We use an appearance map of the source face to identify and utilize the nearest available views, as detailed in Section 3.6. Given an image, $I \in \mathbb{R}^{3 \times H \times W}$, and a heatmap representation, $H(p) \in \mathbb{R}^{98 \times H \times W}$, of the facial landmarks, $p \in \mathbb{R}^{98 \times 2}$, we define the face reenactment generator

$$G_r : \{\mathbb{R}^{3 \times H \times W}, \mathbb{R}^{98 \times H \times W}\} \rightarrow \mathbb{R}^{3 \times H \times W}. \quad (8)$$

Let $p_s, p_t \in \mathbb{R}^{98 \times 2}$ and $\theta_s, \theta_t \in \mathbb{R}^3$, be the 2D landmarks and Euler angles of F_s and F_t , respectively. Utilizing n reenactment iterations, we generate $n - 1$ intermediate 2D landmarks, p_1, \dots, p_{n-1}

$$p_i = T\left(p_s, \left(1 - \frac{i}{n}\right)\theta_s + \frac{i}{n}\theta_t\right) \quad \text{for } i = 1, \dots, n - 1, \\ p_n = p_t. \quad (9)$$

where T is the landmark transformer discussed in the next section. The recursive reenactment output of each iteration is thus given by

$$I_{r_i} = G_r(I_{r_{i-1}}; H(p_i)), 1 \leq i \leq n \\ I_{r_0} = I_s. \quad (10)$$

3.5.1 Landmarks Transformer

To generate the intermediate face landmarks we need to rotate the source landmarks by the delta angle. In our previous work [1], 3D landmarks were used with geometric transformations, in this work we rely on 2D landmarks and pose estimation, and perform the transformation using a CNN. The 2D landmarks are detected as in Sun *et al.* [48], while the pose is estimated using landmark free pose estimation method [55], [56].

Given the landmarks, $p_s \in \mathbb{R}^{98 \times 2}$, in the source image, and the head pose, $\theta_t \in \mathbb{R}^3$, in the target image, we aim to estimate the 2D landmarks, p_t , in the target image. Thus, we propose the landmarks transformer, $T : \{p_1, \theta_1\} \rightarrow p_2$. The network is comprised of 12 linear layers, each followed by batch normalization and ReLU activation layers except for the final layer. It is trained using sequences of faces, where we randomly select a pair of face images, F_s and F_t , and

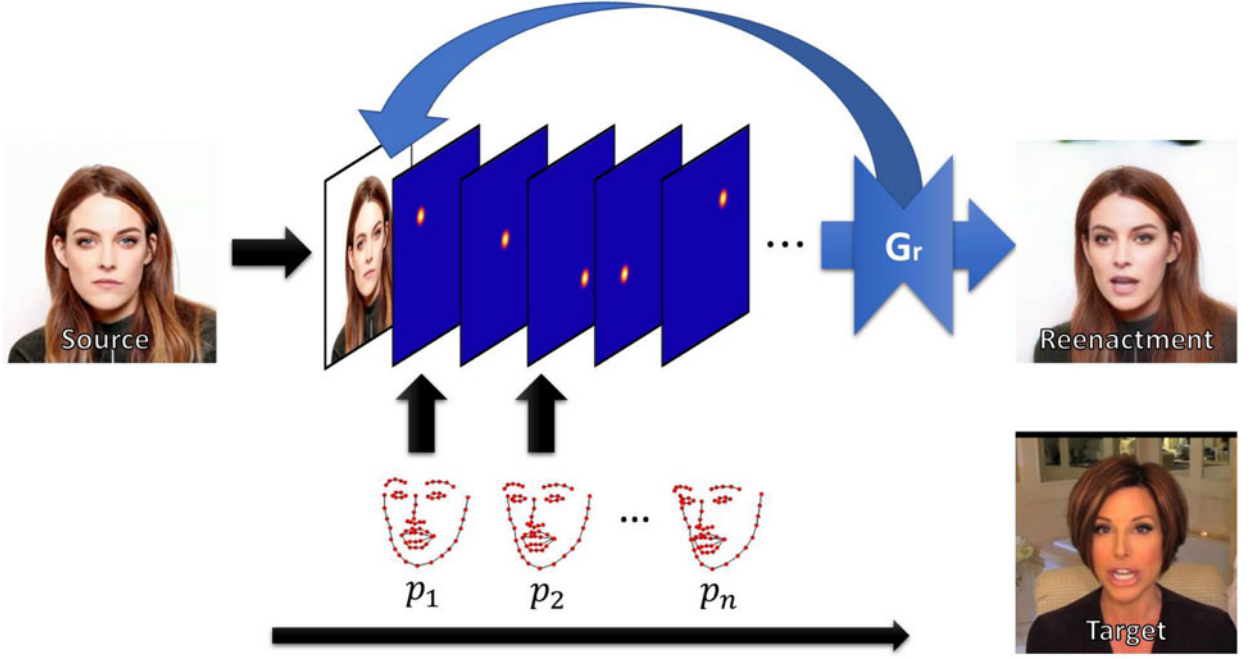


Fig. 4. *Face reenactment*. A single iteration of our proposed face reenactment method. The source image is concatenated with the heatmap corresponding to the first face landmarks, p_1 , and fed to the reenactment generator, G_r , producing the first reenactment iteration result. The next iteration uses the result of the previous iteration instead of the source image with the next face landmarks.

minimize the mean square error between the predicted landmarks, $T(p_{s_i}, \theta_{t_i})$, and the ground truth target landmarks, p_{t_i}

$$\mathcal{L}(T) = \frac{1}{n} \sum_{i=1}^n \|T(p_{s_i}, \theta_{t_i}) - p_{t_i}\|_2^2. \quad (11)$$

3.5.2 Landmarks Heatmaps

Let $\mathbf{A} \in \mathbb{R}^{C \times H \times W}$ be the output layer of the face landmarks network [48], where C is number of landmark points. In our previous work, the face landmarks were encoded by taking the points of maximum activation in each channel in \mathbf{A} , and were decoded by splatting the points on the heatmap followed by a Gaussian blur filter. Both operation are not sub-pixel accurate. To further reduce jittering and improve temporal consistency, we replace those two operations by sub-pixel accurate encoding and decoding.

We filter-out the lower values in \mathbf{A} corresponding to non-landmarks by normalizing \mathbf{A} to $[0,1]$ and zeroing $\mathbf{A} < 0.5$. A landmark point, p_c , encoded by the channel, C , in \mathbf{A} , is given by

$$p_c = \sum_{x,y} \mathbf{A}(C, y, x) \cdot (x + 0.5, y + 0.5), \quad (12)$$

After encoding the face landmarks we perform temporal smoothing as in Section 3.1. The heatmap layer corresponding to p is given by

$$\tilde{H}(p)_{c,y,x} = 1 - \frac{1}{\sqrt{2}} \|p_c - (x + 0.5, y + 0.5)\|_2, \quad (13)$$

and the support of a heatmap point is given by

$$H(p) = \max\left(\frac{\tilde{H}(p)^8 - t}{1 - t}, 0\right), \quad (14)$$

where $t = 0.8$ was used in all our experiments.

3.5.3 Training

Denote by I_s, I_t , the source and target images denoting the same subject, where both images are randomly selected from the same video sequence. Indicated by I_{r_n} , the reenactment result after n iterations, and \tilde{I}_r , the reenactment results without intermediate iterations. \tilde{I} is image I , but with its background removed. Thus, the loss of the generator, G_r , is

$$\mathcal{L}(G_r) = \lambda_{\text{stepwise}} \mathcal{L}_{\text{rec}}(\tilde{I}_{r_n}, \tilde{I}_t) + \lambda_{\text{rec}} \mathcal{L}_{\text{rec}}(\tilde{I}_r, \tilde{I}_t) + \lambda_{\text{adv}} \mathcal{L}_{\text{adv}}, \quad (15)$$

where L_{rec} is given in Eq. (5).

3.6 Face View Interpolation

Following the common computer graphics pipeline [57], in which textured mesh polygons are projected onto a plane and seamlessly rendered, we propose a novel scheme for iterative interpolation between face views. This step is an essential phase of our method, as it enables us to utilize the entire source video sequence, without training our model on a particular video frame, thus allowing our approach to be subject agnostic.

Given a set of source subject images, $\{I_{s_1}, \dots, I_{s_n}\}$, and Euler angles, $\{\mathbf{e}_1, \dots, \mathbf{e}_n\}$, of the corresponding faces, $\{\mathbf{F}_{s_1}, \dots, \mathbf{F}_{s_n}\}$, we construct the appearance map of the source subject illustrated in Fig. 5a. The appearance map embeds the head poses in a triangulated plane, allowing to follow continuous paths of head poses.

We start by projecting the Euler angles, $\{\mathbf{e}_1, \dots, \mathbf{e}_n\}$, to a plane by dropping the roll angle. Using a k-d tree data structure [57], we prune points in the angular domain that are too close to each other. We retain points for which the

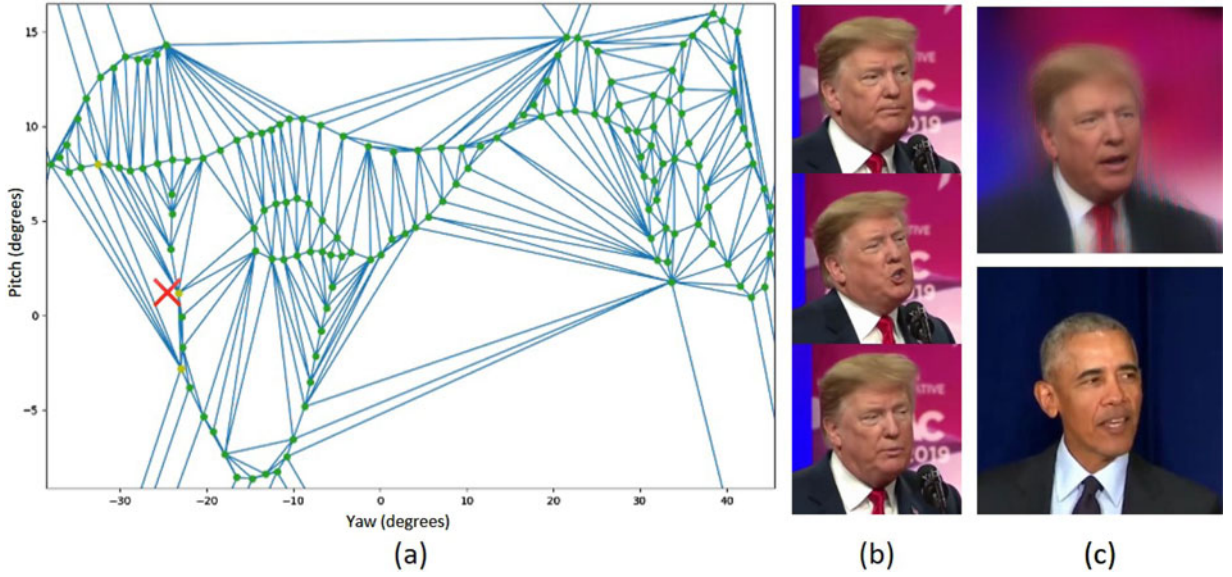


Fig. 5. *Face view interpolation*. (a) Shows an example of an appearance map of the source subject (Donald Trump). The green dots represent different views of the source subject, the blue lines represent the Delaunay Triangulation of those views, and the red X marks the location of the current target's pose. (b) The interpolated views associated with the vertices of the selected triangle (represented by the yellow dots). (c) The reenactment result and the current target image.

corresponding Euler angles have a roll angle closer to zero, and remove motion blurred images. Using the remaining points, $\{x_1, \dots, x_m\}$, and the four boundary points, $y_i \in [-75, 75] \times [-75, 75]$, we build a mesh M in the angular domain using a Delaunay Triangulation.

For a query Euler angle e_t , of a face, F_t , and its corresponding projected point, x_t , we find the triangle, $T \in M$, that contains x_t . Let $x_{i_1}, x_{i_2}, x_{i_3}$ be the vertices of T and $I_{s_{i_1}}, I_{s_{i_2}}, I_{s_{i_3}}$ be the corresponding face views. We calculate the barycentric coordinates, $\lambda_1, \lambda_2, \lambda_3$, of x_t , with respect to $x_{i_1}, x_{i_2}, x_{i_3}$. The interpolation result, I_r , is then given by

$$I_r = \sum_{k=1}^3 \lambda_k G_r(I_{s_{i_k}}; H(\mathbf{p}_t)), \quad (16)$$

where \mathbf{p}_t are the 2D landmarks of F_t . If any vertices of the triangle are boundary points, we exclude them from the interpolation and normalize the weights, λ_i , to sum to one.

A face view query is illustrated in Figs. 5b and 5c. To improve the interpolation accuracy, we used a horizontal flip to fill in views when the appearance map is one-sided with respect to the yaw dimension, and generate artificial views using G_r when the appearance map is too sparse.

3.7 Face Inpainting

A common challenge in face swapping is partial occlusions of the input face image, F_s . Such parts of F_s might be missing and thus cannot be rendered on the target face, F_t . Nirkin *et al.* [10] used the segmentation of F_s and F_t to remove occluded parts and only rendered (and swapped) the face parts in the intersection of the masks. Pixels in F_t were used to render the facial regions occluded in F_s .

For significant occlusions and when the textures of F_s and F_t differ notably, the resulting face swap might have noticeable artifacts. To overcome this, we propose to apply the face inpainting generator, G_c , shown in Fig. 2b, to render F_s such that the resulting face rendering, I_c , will cover the

entire segmentation mask, S_t (of F_t), thus, resolving any occlusion issues.

This is a particular inpainting task in which the expression of the occluded face is given, and the inpainted object is symmetric. These significant cues were not utilized in our previous work [1]. To this end, in addition to the reenactment image and segmentation mask, we also feed the inpainting generator with the target face landmarks and the horizontal flip of the reenactment image.

Given an image, $I \in \mathbb{R}^{3 \times H \times W}$, landmarks points, $p \in \mathbb{R}^{98 \times 2}$, and a segmentation mask, $S \in \mathbb{R}^{3 \times H \times W}$, let \tilde{I} be the input image with its background removed, \tilde{I}' the same image horizontally flipped, $H(p) \in \mathbb{R}^{98 \times H \times W}$ the heatmap representation of p , and $\tilde{S} \in \mathbb{R}^{H \times W}$ a binary face mask derived from S . We define the inpainting generator

$$G_c : \{\tilde{I}, \tilde{I}', H(p), \tilde{S}\} \rightarrow \mathbb{R}^{3 \times H \times W}. \quad (17)$$

3.7.1 Training

Given the reenactment result, I_r , its corresponding segmentation, S_r , the target image with its background removed, \tilde{I}_t , and its corresponding face landmarks, p_t , we augment S_r by simulating common face occlusions due to hair, by randomly removing ellipse-shaped parts, in various sizes and aspect ratios from the border of S_r . Let \tilde{I}_r be I_r with its background removed using the augmented version of S_r , and I_c the result of applying G_c to \tilde{I}_r with the additional input as described in the previous section. We define the inpainting generator loss

$$\mathcal{L}(G_c) = \lambda_{rec} \mathcal{L}_{rec}(I_c, \tilde{I}_t) + \lambda_{adv} \mathcal{L}_{adv}, \quad (18)$$

where L_{rec} and L_{adv} are the reconstruction and adversarial losses as in Sections 3.3.2 and 3.3.3, respectively.

3.8 Face Blending

The final phase of our face swapping scheme is the blending of the completed face, F_c , with the target face, F_t , as

depicted in Fig. 2c. The blending accounts for different skin tones and lighting conditions, and is inspired by previous works that applied the Poisson equation to inpainting [58] and blending [59]. For that, we propose a novel Poisson blending loss.

Let I_t be the target image, I_r^t the image of the reenacted face transferred onto the target image, and S_t the segmentation mask marking the transferred pixels. Following Perez *et al.* [60] we define the Poisson optimization

$$P(I_t; I_r^t; S_t) = \arg \min_f \|\nabla f - \nabla I_r^t\|_2^2$$

$$s.t. f(i, j) = I_t(i, j), \forall S_t(i, j) = 0, \quad (19)$$

where $\nabla(\cdot)$ is the gradient operator. We combine the Poisson optimization above with the perceptual loss, thus deriving the Poisson blending loss $\mathcal{L}(G_b)$

$$\mathcal{L}(G_b) = \lambda_{rec} \mathcal{L}_{rec}(G_b(I_t; I_r^t; S_t), P(I_t; I_r^t; S_t)) + \lambda_{adv} \mathcal{L}_{adv}. \quad (20)$$

3.9 Subject-Specific Finetuning

The face reenactment generator, G_r , pretrained on a large amount of subjects, can be finetuned to a specific subject using as a little as 400 iterations and the same training configuration as in Section 4.2. After finetuning G_r , is used to render a specific subject in any pose and expression. Finetuning allows to improve the quality of the rendering and reproducing subtle face details such as the appearance of the teeth. This formulation of our method is quantitatively explored in the Experiments results section. Unless explicitly stated, we do not finetune the reenactment generator in our experiments.

3.10 Post-Processing

The raw result of our face swapping pipeline, which was the final result in our previous method, exhibits a slight saturation relative to the original target image, therefore directly pasting the generated image into the full target image might result in noticeable bounding box borders. Hence, we apply soft erosion to the target segmentation mask, S_t , resulting in a seamless transition from "face" to "background". The final cropped face swapping result, I_b' , is then given by

$$I_b' = I_b \cdot S_t + I_t \cdot (1 - S_t).$$

I_b' is then seamlessly integrated into the full target image by resizing it to the appropriate size and copying it to the location of the detected bounding box.

4 DATASETS AND TRAINING

4.1 Datasets and Processing

We used the video sequences of the IJB-C dataset [61] to train the generator, G_r , from which we automatically extracted the frames containing particular subjects. IJB-C contains 11,000 face videos of which we used 5,500 that were in high definition. Similar to the frame pruning approach described in Section 3.6, we prune the face views that are too close together as well as motion-blurred frames.

We apply the segmentation CNN G_s , to the frames and prune the frames for which less than 15% of the pixels in the face bounding box were classified as face pixels. Dlib's

face verification¹ was applied to group frames according to the depicted subjects' identity, and limit the number of frames per subject to 100, by choosing the frames whose 2D landmarks have the maximal variance. In each training iteration, we choose the frames, I_s and I_t , from two randomly chosen subjects.

The VGG-19 CNNs were trained using perceptual losses on the VGGFace2 dataset [62] for face recognition and the CelebA [63] dataset for face attribute classification. The VGGFace2 dataset contains 3.3M images of 9,131 identities, whereas CelebA contains 202,599 images. The segmentation CNN, G_s , was trained using the dataset from [10], consisting of $\sim 10K$ face images labeled with face segmentations, and the LFW Parts Labels Dataset [64] with $\sim 3K$ images labeled for both face and hair segmentations, from which we removed the neck part using facial landmarks.

We used additional 1K images and corresponding hair segmentations from the Figaro dataset [65]. Finally, FaceForensics++ [66] provides 1000 videos, from which they generated 1,000 synthetic videos on random pairs using DeepFakes [18] and Face2Face [8].

4.2 Training Details

The proposed generators were trained from scratch, where the weights were initialized randomly using a normal distribution. We used the Adam optimizer [67] ($\beta_1 = 0.5, \beta_2 = 0.999$) and a learning rate of 0.0002 that was reduced by half every ten epochs. The following parameters were used for all the generators: $\lambda_{perc} = 1, \lambda_{pixel} = 0.1, \lambda_{adv} = 0.001, \lambda_{seg} = 0.1, \lambda_{rec} = 1, \lambda_{stepwise} = 1$, where $\lambda_{reenactment}$ is linearly increased from 0 to 1 during the training process. All of our networks were trained using eight NVIDIA Tesla V100 GPUs, where the training of the G_s generator required six hours to converge, while the rest of the networks converged in two days. All our networks except for G_s , were trained using a progressive multi-scale approach, starting with a resolution of 128×128 up to 256×256 . The reenactment inference rate is ~ 30 fps, and ~ 10 fps for face swapping, both using a single reenactment iteration on a single NVIDIA Tesla V100 GPU.

5 EXPERIMENTAL RESULTS

We performed extensive qualitative and quantitative trials to experimentally verify the proposed scheme and compare our method to contemporary face swapping methods: DeepFakes [18] and Nirkin *et al.* [10], and the Face2Face reenactment scheme [8]. We conduct all our experiment using FaceForensics++ [66] videos, running our method on the same pairs they used. Moreover, we provide an ablation study to exemplify the importance of each component in our pipeline.

5.1 Quantitative Experiments Metrics

We applied multiple metrics and corresponding quantitative tests to experimentally evaluate our face swapping scheme in general, and the identity preservation, in particular, while retaining the same pose and expression as the target subject. For that, we used the following metrics:

1. Available: <http://dlib.net/>



Fig. 6. Qualitative face reenactment results. Row 1: The source face for reenactment. Row 2: Our reenactment results (without background removal). Row 3: The target face from which to transfer the pose and expression.

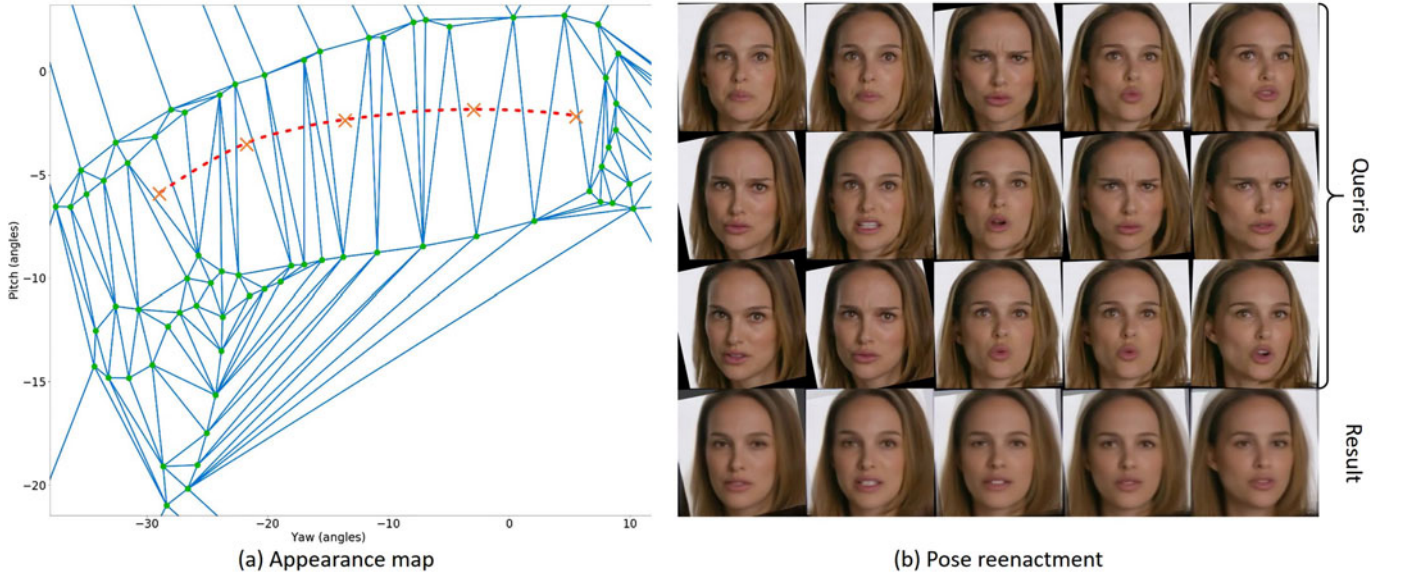


Fig. 7. Pose reenactment. (a) Example appearance map of an input subject (Natalie Portman). The red dashes represent the interpolation path and the orange Xs mark the pose of the views presented on the right. (b) The first three rows are input queries from the appearance map. The reenactment result is displayed in the last row.

Identity Verification (id). To measure the identity preservation of our method, we first extract face embeddings using Arcface [68], and then compare them using cosine similarity

$$\text{similarity} = \left\langle \frac{a}{\|a\|_2}, \frac{b}{\|b\|_2} \right\rangle, \quad (21)$$

where a and b are the face embeddings. This face verification network is a different network than the one used for face-specific perceptual loss, and it was trained on a different dataset.

Fréchet Inception Distance (FID). To assess the visual quality of our generated images we adopt the FID metric [69].

which is more stable than previous quality metrics such as the structural similarity index method (SSIM) or the Inception score [70].

\mathcal{L}_1 Distance. A quality measurement that focuses on the low frequencies of the images, validating the color preservation in the generated image. For this metric we compare images with color values in the range [0,1].

Euler Angles. To compare the face poses we calculate the euclidean distance between the Euler angles of the generated face and those of the target face, using the method from [55]. This distance is quantified in degrees.

Face Landmarks. Similarly to the Euler angles metric, we measure the accuracy of the facial expression and shape by calculating the euclidean distance between the face

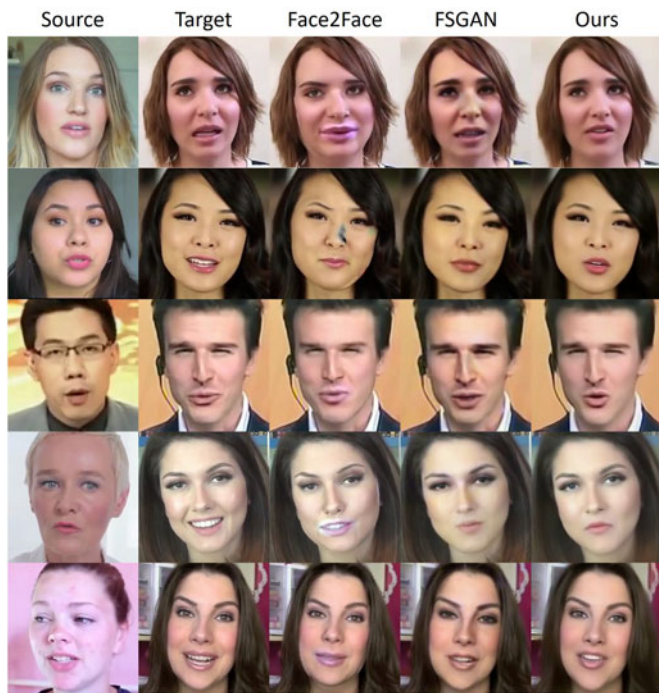


Fig. 8. *Qualitative expression-only reenactment comparison.* The expression of the face in the left column is transferred to the face in the second column using Face2Face [8] (third column), FSGAN [74] (4th column), and our method (last column).

landmarks of the generated face and the target face. This distance is measured in pixels.

Facial Expression Comparison (FEC). To further validate the accuracy of the generated facial expressions, we add the FEC metric. This is an alternative to the expression classification evaluation used for the same purpose by

Chang *et al.* [71], that used a limited number of expressions. The FEC metric is directly trained to compare facial expressions and mimics the way humans perceive expressions.

To this end, we train an expression embedding network using the FEC dataset [72], consisting of 156K images and 500K human-annotated triplets, such that each triplet contains a pair of expressions that are mutually more similar to the third expression. Following Vemulapalli *et al.* [72] we use the triplet loss [73] to train an embedding network to compute a 16-dimensional expression embedding.

5.2 Face Reenactment Results

5.2.1 Qualitative Face Reenactment Results

We start by depicting qualitative face reenactment results in Fig. 6. We show raw face reenactment results, of varying ethnicities, poses, and expressions without background removal. In the left-most column we demonstrate reenacting a face that differs significantly from the training set, exemplifying that our method performs well when applied to faces from completely different domains. The second column from the left demonstrates our method's ability to cope with rare facial expressions that are not present in the training set, and extreme iris position differences.

To exemplify the importance of the iterative reenactment method, we illustrate in Fig. 12 the reenactment of the same subject for both small and large angular differences. For large angular differences, the identity and texture are better preserved using multiple iterations than by using a single one.

5.2.2 Pose Only Reenactment

Using the reenactment generator, G_r , and the landmarks transformer, T , we can reenact a source face without a target



Fig. 9. *Face inpainting qualitative results.* Top row: Input real images with random ellipses removed from the faces. Second row: Inpainting results without facial landmarks. Third row: Inpainting results with facial landmarks. Bottom row: The full input image.

TABLE 1
Inpainting Ablation

Method	FID ↓	id ↑	\mathcal{L}_1 ↓	euler ↓	landmarks ↓	FEC ↓
FSGAN [1]	2.64	0.36 ± 0.07	0.46 ± 0.01	1.77 ± 0.92	34.8 ± 15.0	0.15 ± 0.09
G_c without landmarks	0.74	0.64 ± 0.11	0.31 ± 0.02	1.20 ± 0.75	19.4 ± 7.5	0.11 ± 0.09
G_c with landmarks	0.62	0.64 ± 0.10	0.16 ± 0.01	1.04 ± 0.52	17.9 ± 3.8	0.12 ± 0.07

We compare our previous inpainting results [1] with two variations of the proposed face inpainting method, the first (middle row) without using face landmarks and the second (bottom row) with face landmarks alongside a horizontally flipped replica of the source image.

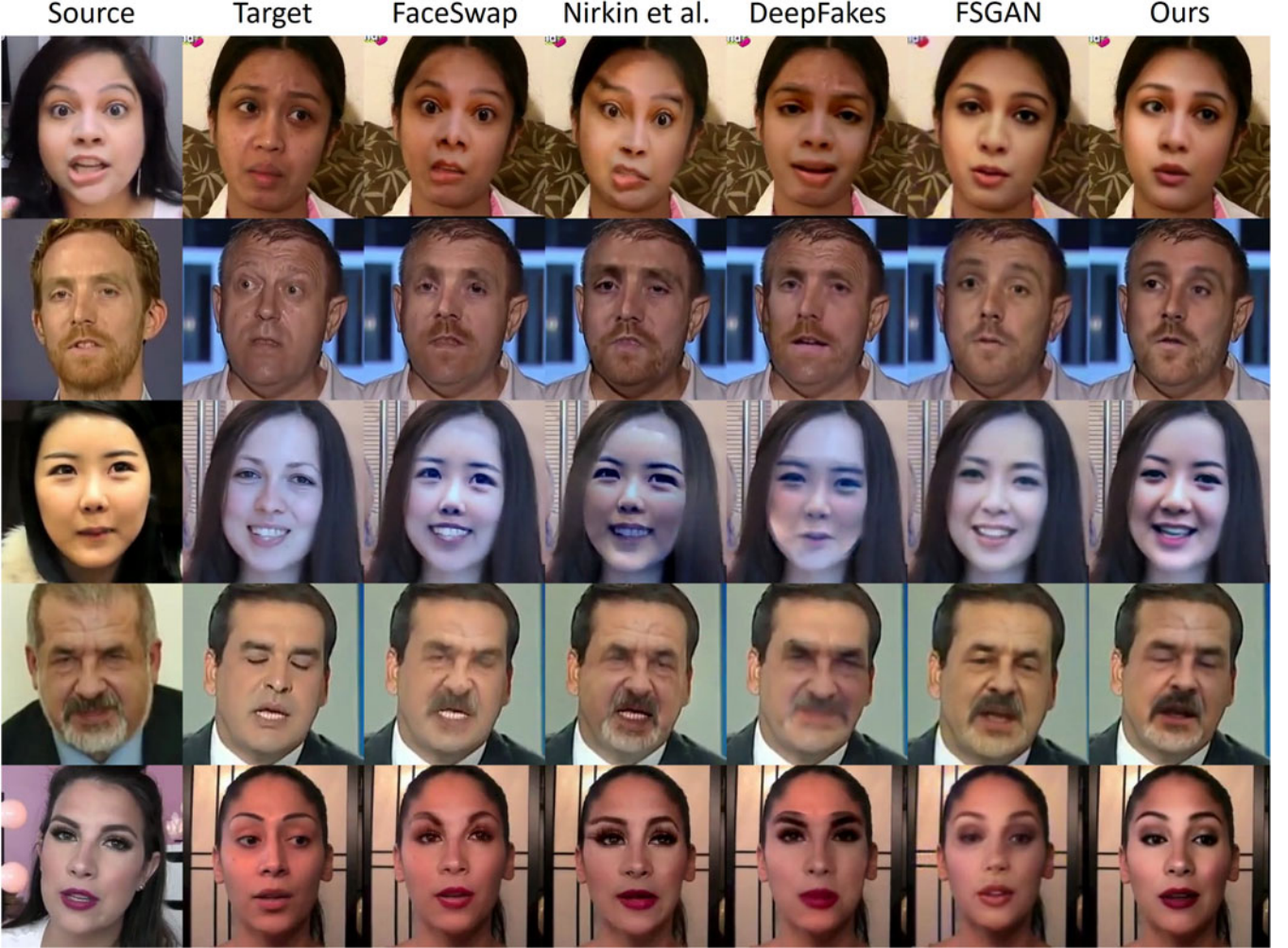


Fig. 10. Qualitative face swapping results on [66]. The 3rd and 4th columns show the results of the 3DMM-based methods FaceSwap [74] and Nirkin et al. [10], the 5th column shows DeepFakes [18] results, a direct mapping method, which was trained specifically on each of those videos. The 6th column shows FSGAN [1] results and the last column shows the results of our method, both on images of faces they have not seen before.

face, using only Euler angles to guide the pose. For that, we apply the face view interpolation in Section 3.6, where the Euler angles were varied smoothly by a user using a mouse or keyboard. For each Euler angle, the appearance map (as in Section 3.6) of the input subject is queried for three face views, which are then rotated to the same tilt angle as the current Euler angle. The views are then interpolated the same way as in Section 3.6. Such a reenactment is shown in Fig. 7.

5.2.3 Expression-Only Reenactment Comparison

We compare our method to the contemporary Face2Face [8] and FSGAN [1] methods in the expression-only reenactment problem, and the results are shown in Figs. 8 and 14. For this

problem we use a slightly different formulation: Given a pair of faces, $F_s \in I_s$ and $F_t \in I_t$, we aim to transfer the expression from I_s to I_t . Hence, we modify the corresponding 2D landmarks of F_t by swapping in the mouth points of the 2D landmarks of F_s , similar to the generation of the intermediate landmarks in Section 3.5. The reenactment result is then given by $G_r(I_t; H(\hat{p}_t))$ where \hat{p}_t are the modified landmarks.

5.3 Face Inpainting Ablation Results

To exemplify the effectiveness of our proposed improvements of G_c , we perform an ablation study. In this study we remove random ellipses from faces in the first 100 frames of the first 500 videos in FaceForensics++, similar to Section 3.7.1, inpaint

TABLE 2
Quantitative Face Swapping Comparison

Method	FID ↓	id ↑	\mathcal{L}_1 ↓	euler ↓	landmarks ↓	FEC ↓
Nirkin <i>et al.</i> [10]	0.67	0.45 ± 0.06	0.38 ± 0.03	3.28 ± 1.34	42.1 ± 9.19	0.37 ± 0.18
FaceSwap [74]	0.37	0.38 ± 0.05	0.44 ± 0.02	2.43 ± 0.85	36.9 ± 7.35	0.34 ± 0.17
DeepFakes [18]	0.38	0.45 ± 0.04	0.44 ± 0.02	4.05 ± 1.13	53.1 ± 8.32	0.45 ± 0.14
FSGAN [1]	0.46	0.36 ± 0.06	0.25 ± 0.01	2.37 ± 1.29	30.8 ± 11.2	0.34 ± 0.14
FaceShifter [40]	0.40	0.60 ± 0.03	0.42 ± 0.11	2.33 ± 0.8	45.51 ± 11.36	0.29 ± 0.11
Ours (fine-tuned)	0.50	0.44 ± 0.04	0.22 ± 0.01	2.40 ± 0.77	28.7 ± 5.76	0.30 ± 0.11
Ours	0.53	0.37 ± 0.05	0.21 ± 0.0	2.07 ± 0.68	21.6 ± 5.25	0.28 ± 0.10

We compare to previous face swapping methods and to a fine-tuned formulation of our method trained using the FaceForensics++ [66].

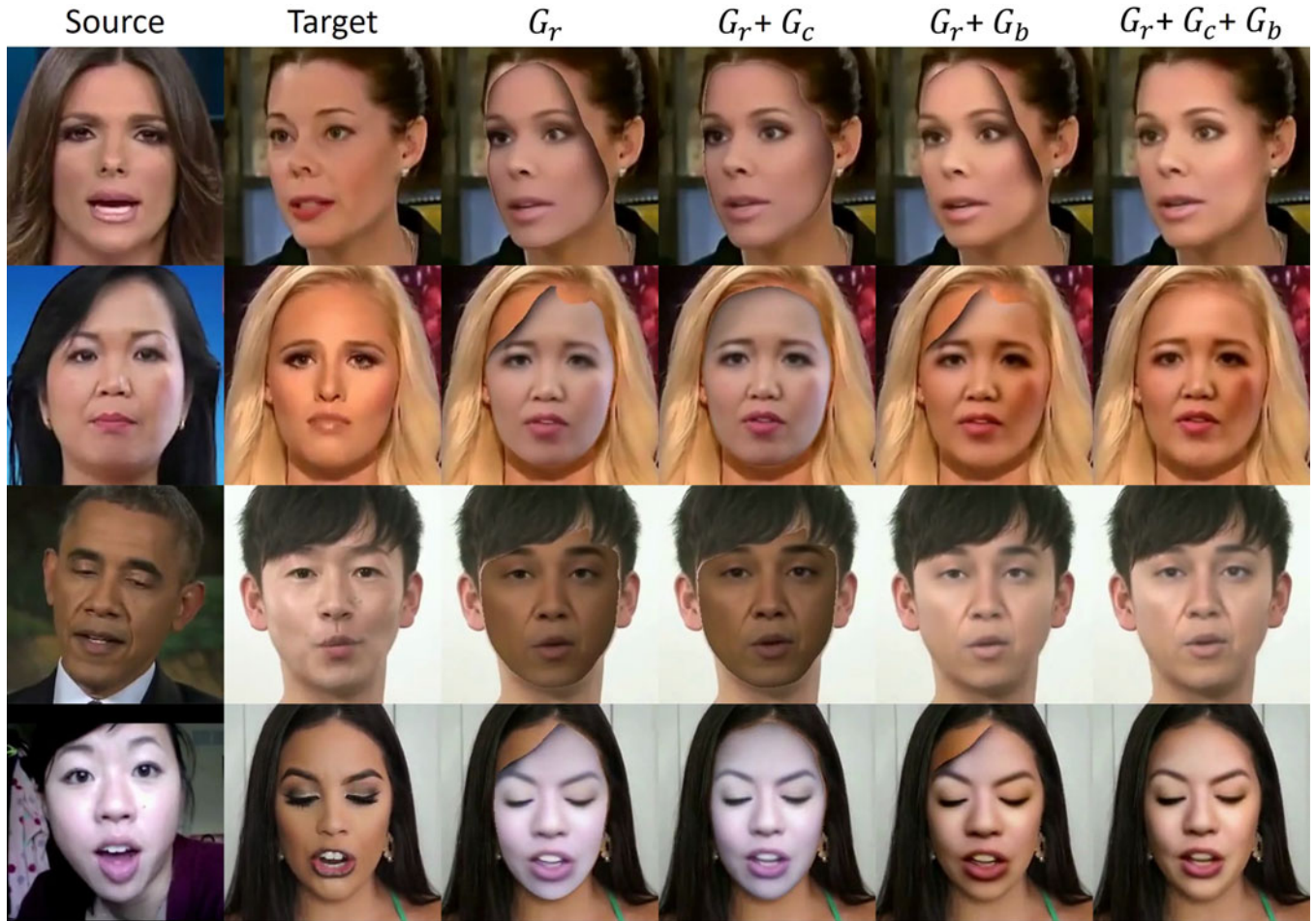


Fig. 11. Ablation study on the pipeline components. As shown in columns 3 and 5, without the completion network, G_c , the transferred face won't cover the entire target face leaving obvious artifacts. In columns 3 and 4, we can see that without the blending network, G_b , the skin color and lighting conditions of the transferred face are inconsistent with its new context.

TABLE 3
Quantitative Face Swapping Results on FaceForensics++ [66] for Our Ablation Study on the Components of FSGAN

Method	FID ↓	id ↑	\mathcal{L}_1 ↓	euler ↓	landmarks ↓	FEC ↓
FSGAN (G_r)	0.72	0.44 ± 0.06	0.26 ± 0.01	2.93 ± 1.11	32.9 ± 8.46	0.36 ± 0.13
Ours (G_r)	0.71	0.45 ± 0.07	0.22 ± 0.01	2.93 ± 0.91	31.8 ± 5.66	0.30 ± 0.11
Ours ($G_r + G_c$)	0.67	0.41 ± 0.06	0.21 ± 0.01	2.57 ± 0.83	23.2 ± 4.95	0.29 ± 0.10
Ours ($G_r + G_b$)	0.54	0.41 ± 0.06	0.20 ± 0.01	2.40 ± 0.79	28.0 ± 5.17	0.29 ± 0.11
Ours ($G_r + G_c + G_b$)	0.53	0.37 ± 0.06	0.21 ± 0.0	2.07 ± 0.68	21.6 ± 5.25	0.28 ± 0.10

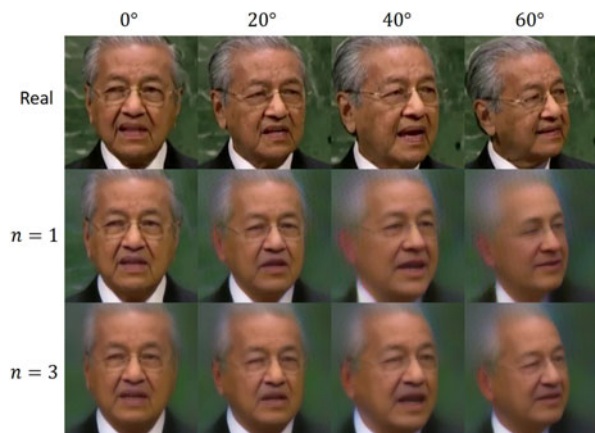


Fig. 12. Reenactment limitations. Top left image transformed into each of the images in Row 1 (using the same subject for clarity). Row 2: Reenactment with one iteration. Row3: Three iterations.

the face using the different inpainting methods and compare the inpainting results to the original faces.

Qualitative examples of the ablation study are shown in Fig. 9. It follows that our *new* FSGAN architecture for G_c

produces higher quality results than the *previous* FSGAN[1]. Moreover, the FSGAN formulation that receives the face landmarks as an additional input better reconstructs the facial expressions and shapes of the input faces. Quantitative results are shown in Table 1 using the metrics detailed in Section 5.1.

As apparent from the table, both new FSGAN formulations significantly outperform the previous FSGAN [1] in all metrics. Our G_c formulation that also receives the face landmarks and the horizontal flip of the image, outperforms the variant without those additions on all metrics, but for the id metric, for which both variants achieve similar scores.

5.4 Face Swapping Results

5.4.1 Qualitative Face Swapping Results

Qualitative results of our face swapping method applied to the FaceForensics++ videos *without training our model on any of them* are shown in Fig. 10. We purposefully chose examples showing different poses and expressions, face shapes, and hair occlusions. Nirkin *et al.* [10] is an image-to-image face swapping method and therefore to be applied to videos it

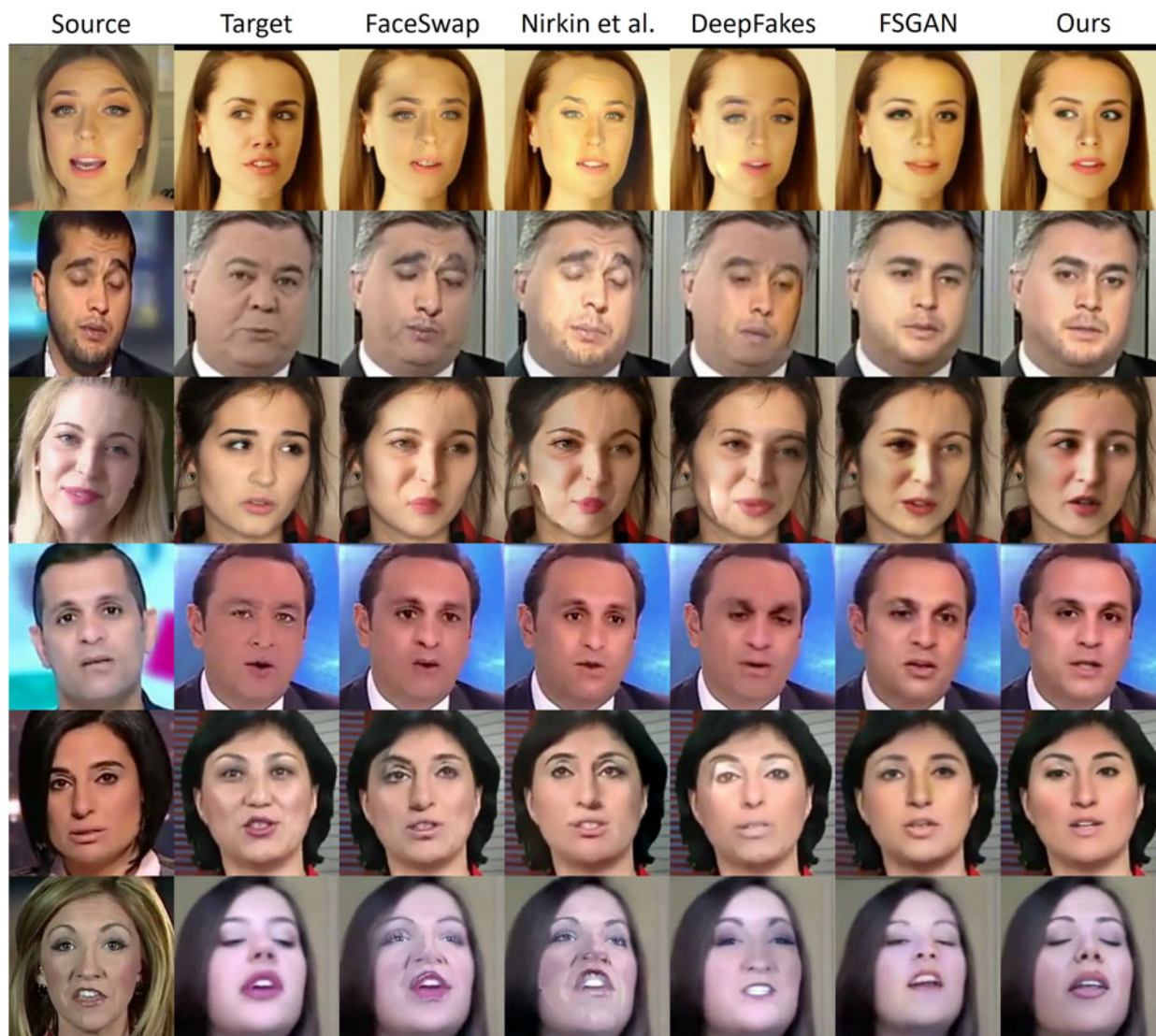


Fig. 13. Additional qualitative face swapping comparison to previous methods on FaceForensics++ [66].
Authorized licensed use limited to: Hochschule Aalen (FH) - Technik und Wirtschaft. Downloaded on July 06,2024 at 13:46:53 UTC from IEEE Xplore. Restrictions apply.

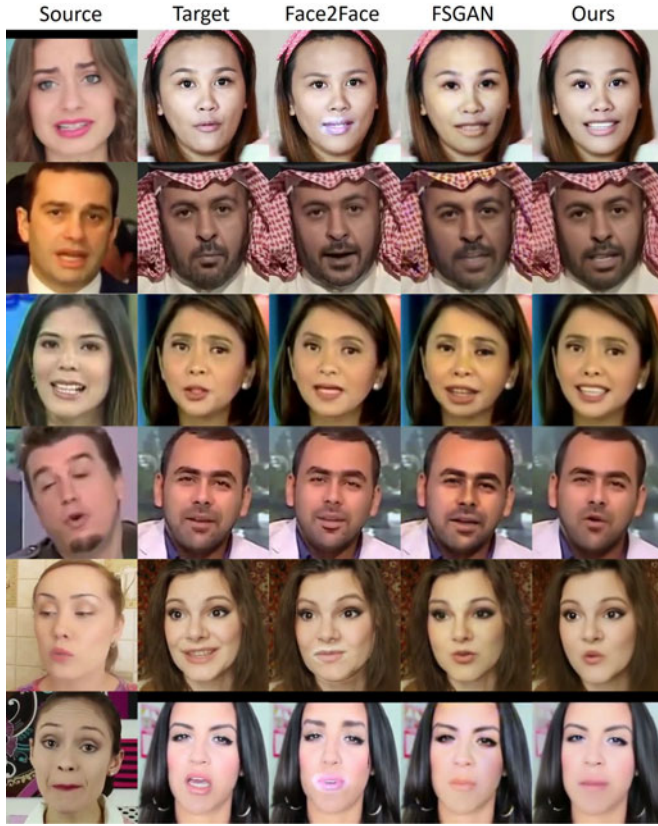


Fig. 14. Additional qualitative expression-only reenactment comparison on FaceForensics++ [66].

requires a selection strategy for the source frames. Thus, to be fair in our comparison, for each frame in the target video, we select the source frame having the most similar pose, as face swapping becomes more difficult the larger the pose discrepancy. To compare FSGAN in a video-to-video scenario such as this, we use our face view interpolation, as described in Section 3.6. Additional qualitative comparison is presented in Fig. 13.

5.4.2 Quantitative Face Swapping Results

We also applied quantitative tests to compare our entire face swapping pipeline with previous face swapping schemes. For that, we used the same videos as in Section 5.3, and the metrics detailed in Section 5.1. We compute the mean and standard deviation, averaged across the videos, for all the metrics except for the FID metric that is computed globally. The metrics L_1 , Euler, landmarks, and FEC, for each frame, are compared to the corresponding frame in the target video, to test color, shape, pose, and expression preservation of the target face. The id metric for each frame, is compared against random source frames to quantify how well the identity is retained. Finally, the FID metric is used to compare the texture distributions of the source and generated faces.

We compare with five previous face swapping methods: Nirkin *et al.* [10], FaceSwap [74], DeepFakes [18], FSGAN [1], and FaceShifter [40]. The first column of Table 2 shows that the FaceSwap approach achieves the best FID score as it directly warps the texture from the source image. Our method achieves a slightly better FID score as explained in

Section 6.1. Our fine-tuned version attains an improved FID score as it is further trained using the source images. The second column shows that FaceShifter, a recent state-of-the-art method, best preserves the identity of the source subject. Our method achieves a lower identity score, yet retains color, pose, and expression significantly better than previous methods, as shown the four rightmost metrics in Table 2. As can be seen from the results, there is a trade-off between how much a method can preserve the source identity and faithfully maintain other important attributes such as: color, pose, and expression.

5.5 Ablation Study on Pipeline Components

To demonstrate the necessity of each component in our pipeline, we conducted an ablation study on four different configurations of our method: G_r only, G_r with G_c , G_r with G_b , and the entire pipeline. The segmentation network, G_s , is used in all configurations. Qualitative examples of our ablation study are depicted in Fig. 11.

The quantitative results of our ablation study are shown in Table 3. Comparing our G_r to our previous reenactment generator (first row), demonstrates an improvement in all metrics, indicating that the landmarks method described in Section 3.5.1 is superior to the one in FSGAN [1]. Our entire pipeline achieves the best FID score as shown in the first column, indicating that the texture of the source face is best preserved when none of the steps are omitted. The id scores (in the second column) show that the source subject identity is preserved across all pipeline networks. The L_1 scores (in the third column) confirms that the appearance of the target face are best reconstructed when the blending generator, G_b , is applied, indicating the importance of the final blending step. Finally, as demonstrated by the last three columns, the pose and expression of the target face are best preserved by our entire pipeline.

6 CONCLUSION AND FUTURE WORK

6.1 Limitations

Fig. 12 shows our reenactment results for different facial yaw angles. Evidently, the larger the angular differences, the more the identity and texture quality degrade. Moreover, applying too many iterations of the face reenactment generator also blurs the texture. As opposed to 3DMM based methods such as Face2Face [8], which warp the texture directly from the image, our method is limited to the image resolution of the training data. Another limitation arises from using a sparse landmark tracking method that does not fully capture the complexity of the facial expressions.

In the video-to-video scenario, the proposed face view interpolation method assume an homogeneous image quality and facial expressions. If those assumptions do not hold, it might lead to artifacts in the interpolation results. Finally, there is a trade-off between how much our method can preserve the source identity while maintaining the attributes of the target subject such as: skin color, pose, and expression.

6.2 Discussion

Our method eliminates laborious, subject-specific data collection and model training, making face swapping and reenactment accessible to non-experts. We feel strongly that it is

of *paramount importance* to publish such technologies, to drive the development of technical counter measures for detecting such forgeries, as well as compel law makers to set clear policies for addressing their implications. Suppressing the publication of such methods would not stop their development, but rather make them available to select a few and potentially blindside policy makers if it is misused.

6.3 Future Work

We have identified that the human factor is a major limitation of our method. It is imposed as we rely on face and face landmarks detection, which are trained using human labeling. Face landmarks in particular, are an abstraction originally derived for humans, which filter-out significant important information that could be otherwise utilized by the CNN. Future research should avoid using methods that were trained using human labeling. This would allow to achieve subpixel accurate face tracking and segmentation, and improve image quality and resolution. Previous reenactment methods mainly focused on specific domains such as faces or bodies, reenactment methods could potentially be generalized to multiple domains.

Utilizing only one frame at a time is suboptimal as it does not make use of all the information available for the specific subject. Additionally, temporal information can help improve the tracking of the face pose and expression. Future methods should be able to take all this available information into consideration.

REFERENCES

- [1] Y. Nirkin, Y. Keller, and T. Hassner, "FSGAN: Subject agnostic face swapping and reenactment," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 7184–7193.
- [2] O. Alexander, M. Rogers, W. Lambeth, M. Chiang, and P. Debevec, "Creating a photoreal digital actor: The digital emily project," in *Proc. IEEE Conf. Vis. Media Prod.*, 2009, pp. 176–187.
- [3] I. Kemelmacher-Shlizerman, "Transfiguring portraits," *ACM Trans. Graph.*, vol. 35, no. 4, 2016, Art. no. 94.
- [4] L. Wolf, Z. Freund, and S. Avidan, "An eye for an eye: A single camera gaze-replacement method," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 817–824.
- [5] V. Blanz, K. Scherbaum, T. Vetter, and H.-P. Seidel, "Exchanging faces in images," *Comput. Graph. Forum*, vol. 23, no. 3, pp. 669–676, 2004.
- [6] Y. Lin, S. Wang, Q. Lin, and F. Tang, "Face swapping under large pose variations: A 3D model based approach," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2012, pp. 333–338.
- [7] S. Mosaddegh, L. Simon, and F. Jurie, "Photorealistic face de-identification by aggregating donors face components," in *Proc. Asian Conf. Comput. Vis.*, 2014, pp. 159–174.
- [8] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2Face: Real-time face capture and reenactment of RGB videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2387–2395.
- [9] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing obama: Learning lip sync from audio," *ACM Trans. Graph.*, vol. 36, no. 4, 2017, Art. no. 95.
- [10] Y. Nirkin, I. Masi, A. T. Tuan, T. Hassner, and G. Medioni, "On face segmentation, face swapping, and face perception," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2018, pp. 98–105.
- [11] I. Goodfellow et al., "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
- [12] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.
- [13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1125–1134.
- [14] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8798–8807.
- [15] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer, "GANimation: Anatomically-aware facial animation from a single image," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 818–833.
- [16] W. Wu, Y. Zhang, C. Li, C. Qian, and C. C. Loy, "ReenactGAN: Learning to reenact faces via boundary transfer," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 622–638.
- [17] E. Sanchez and M. Valstar, "Triple consistency loss for pairing distributions in GAN-based face synthesis," 2018, *arXiv:1811.03492*.
- [18] DeepFakes, Accessed: Feb. 06, 2019. [Online]. Available: <https://github.com/deepfakes/faceswap>
- [19] Y. Tian, X. Peng, L. Zhao, S. Zhang, and D. N. Metaxas, "CR-GAN: Learning complete representations for multi-view generation," 2018, *arXiv:1806.11191*.
- [20] R. Natsume, T. Yatagawa, and S. Morishima, "RSGAN: Face swapping and editing using face and hair representation in latent spaces," 2018, *arXiv:1804.03447*.
- [21] R. Natsume, T. Yatagawa, and S. Morishima, "FSNet: An identity-aware generative model for image-based face swapping," in *Proc. Asian Conf. Comput. Vis.*, 2018, pp. 117–132.
- [22] H. Kim et al., "Deep video portraits," *ACM Trans. Graph.*, vol. 37, no. 4, 2018, Art. no. 163.
- [23] D. Bitouk, N. Kumar, S. Dhillon, P. Belhumeur, and S. K. Nayar, "Face swapping: Automatically replacing faces in photographs," *ACM Trans. Graph.*, vol. 27, no. 3, 2008, Art. no. 39.
- [24] V. Blanz, S. Romdhani, and T. Vetter, "Face identification across different poses and illuminations with a 3D morphable model," in *Proc. Int. Conf. Autom. Face Gesture Recognit.*, 2002, pp. 192–197.
- [25] V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1063–1074, Sep. 2003.
- [26] H. Averbuch-Elor, D. Cohen-Or, J. Kopf, and M. F. Cohen, "Bringing portraits to life," *ACM Trans. Graph.*, vol. 36, no. 6, 2017, Art. no. 196.
- [27] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2794–2802.
- [28] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*.
- [29] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," 2017, *arXiv:1710.10196*.
- [30] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2223–2232.
- [31] J. Wang, K. Chen, R. Xu, Z. Liu, C. C. Loy, and D. Lin, "CARAFE: Content-aware reassembly of features," 2019, *arXiv:1905.02188*.
- [32] H. Lu, Y. Dai, C. Shen, and S. Xu, "Indices matter: Learning to index for deep image matting," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 3266–3275.
- [33] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 694–711.
- [34] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky, "Few-shot adversarial learning of realistic neural talking head models," 2019, *arXiv:1905.08233*.
- [35] H. Kim et al., "Neural style-preserving visual dubbing," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–13, 2019.
- [36] J. Thies, M. Elgharib, A. Tewari, C. Theobalt, and M. Nießner, "Neural voice puppetry: Audio-driven facial reenactment," 2019, *arXiv:1912.05566*.
- [37] O. Fried et al., "Text-based editing of talking-head video," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–14, 2019.
- [38] X. Yao, O. Fried, K. Fatahalian, and M. Agrawala, "Iterative text-based editing of talking-heads using neural retargeting," 2020, *arXiv:2011.10688*.
- [39] S. Ha, M. Kersner, B. Kim, S. Seo, and D. Kim, "MarioNETte: Few-shot face reenactment preserving identity of unseen targets," 2019, *arXiv:1911.08139*.
- [40] L. Li, J. Bao, H. Yang, D. Chen, and F. Wen, "FaceShifter: Towards high fidelity and occlusion aware face swapping," 2019, *arXiv:1912.13457*.
- [41] J. Naruniec, L. Helming, C. Schroers, and R. M. Weber, "High-resolution neural face swapping for visual effects," *Comput. Graph. Forum*, vol. 39, no. 4, pp. 173–184, 2020.

- [42] E. Zakharov, A. Ivakhnenko, A. Shysheya, and V. Lempitsky, "Fast bi-layer neural synthesis of one-shot realistic head avatars," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 524–540.
- [43] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, 2015, pp. 234–241.
- [44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [45] S. Ha, M. Kersner, B. Kim, S. Seo, and D. Kim, "MarioNETte: Few-shot face reenactment preserving identity of unseen targets," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 10893–10900.
- [46] J. Li et al., "DSFD: Dual shot face detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5060–5069.
- [47] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "S3FD: Single shot scale-invariant face detector," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 192–201.
- [48] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," 2019, *arXiv:1902.09212*.
- [49] A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3D facial landmarks)," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1021–1030.
- [50] G. Casiez, N. Roussel, and D. Vogel, "1f filter: A simple speed-based low-pass filter for noisy input in interactive systems," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2012, pp. 2527–2530.
- [51] C. Ledig et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4681–4690.
- [52] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 586–595.
- [53] A. R. Tej, S. S. Halder, A. P. Shandeelya, and V. Pankajakshan, "Enhancing perceptual loss with adversarial feature matching for super-resolution," in *Proc. Int. Joint Conf. Neural Netw.*, 2020, pp. 1–8.
- [54] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
- [55] N. Ruiz, E. Chong, and J. M. Rehg, "Fine-grained head pose estimation without keypoints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 2074–2083.
- [56] F.-J. Chang, A. Tuan Tran, T. Hassner, I. Masi, R. Nevatia, and G. Medioni, "FacePoseNet: Making a case for landmark-free face alignment," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2017, pp. 1599–1608.
- [57] J. F. Hughes, A. Van Dam, J. D. Foley, and S. K. Feiner, *Computer Graphics: Principles and Practice*. London, U.K.: Pearson Education, 2014.
- [58] R. A. Yeh, C. Chen, T. Yian Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5485–5493.
- [59] H. Wu, S. Zheng, J. Zhang, and K. Huang, "GP-GAN: Towards realistic high-resolution image blending," 2017, *arXiv:1703.07195*.
- [60] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 313–318, 2003.
- [61] B. Maze et al., "IARPA janus benchmark-C: Face dataset and protocol," in *Proc. Int. Conf. Biometrics*, 2018, pp. 158–165.
- [62] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "VGGFace2: A dataset for recognising faces across pose and age," in *Proc. Int. Conf. Autom. Face Gesture Recognit.*, 2018, pp. 67–74.
- [63] Z. Liu, P. Luo, X. Wang, and X. Tang, "Large-scale celebfaces attributes (celeba) dataset," *Retrieved August*, vol. 15, 2018, Art. no. 2018.
- [64] A. Kae, K. Sohn, H. Lee, and E. Learned-Miller, "Augmenting CRFs with boltzmann machine shape priors for image labeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2019–2026.
- [65] M. Svanera, U. R. Muhammad, R. Leonardi, and S. Benini, "Figaro, hair detection and segmentation in the wild," in *Proc. Int. Conf. Image Process.*, 2016, pp. 933–937.
- [66] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to detect manipulated facial images," 2019, *arXiv:1901.08971*.
- [67] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [68] J. Deng, J. Guo, X. Niannan, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4685–4694.
- [69] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6626–6637.
- [70] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.
- [71] F.-J. Chang, A. T. Tran, T. Hassner, I. Masi, R. Nevatia, and G. Medioni, "ExpNet: Landmark-free, deep, 3D facial expressions," in *Proc. Int. Conf. Autom. Face Gesture Recognit.*, 2018, pp. 122–129.
- [72] R. Vemulapalli and A. Agarwala, "A compact embedding for facial expression similarity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5683–5692.
- [73] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 815–823.
- [74] FaceSwap, "FaceSwap," Accessed: Nov. 15, 2019. [Online]. Available: <https://github.com/MarekKowalski/FaceSwap/>



Yuval Nirkin received the BSc degree in computer engineering from the Technion Israel Institute of Technology, Haifa, Israel, in 2011, the MSc degree in computer science from The Open University of Israel, Ra'anana, Israel, in 2017, and the PhD degree in electrical engineering from Bar-Ilan University, Ramat Gan, Israel, in 2022. His research focuses on deep learning, computer vision, and computer graphics. He has been a reviewer in ECCV, ICCV, and CVPR, and was recognized as a high quality reviewer in ECCV'20.



Yosi Keller received the BSc degree in electrical engineering from the Technion Israel Institute of Technology, Haifa, Israel, in 1994, and the MSc and PhD degrees in electrical engineering from Tel Aviv University, summa cum laude, in 1998 and 2003, respectively. From 2003 to 2006 he was a Gibbs assistant professor with the Department of Mathematics, Yale University, New Haven, Connecticut. He is currently an associate professor with the Faculty of Engineering, Bar Ilan University, Ramat-Gan, Israel. His research interests are in computer vision, machine and deep learning, and biometrics.



Tal Hassner received the MSc and PhD degrees in applied mathematics and computer science from the Weizmann Institute of Science, Israel, in 2002 and 2006, respectively. In 2008, he joined the Department of Mathematics and Computer Science, The Open University of Israel where he was an associate professor until 2018. From 2015 to 2018, he was a senior computer scientist with the Information Sciences Institute (ISI) and a visiting associate professor with the Institute for Robotics and Intelligent Systems, Viterbi School of Engineering, both with University of Southern California, Los Angeles, California, working on the IARPA Janus face recognition project. From 2018 to 2019, he was a principal applied scientist with AWS, where he led the design and development of the latest AWS face recognition pipelines. Since June 2019, he is an Applied Research Lead with Facebook AI, where he supported the text (OCR) and people (faces) photo understanding teams. He is an associate editor for both *IEEE Transactions on Pattern Analysis and Machine Intelligence* and *IEEE Transactions on Biometrics, Behavior, and Identity Science*. Some of his recent organizational roles include program chair for WACV'17, ICCV'21, and ECCV'22.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.