

FaceOff: A Video-to-Video Face Swapping System

Anonymous WACV 2023 Algorithms Track submission

Paper ID 615

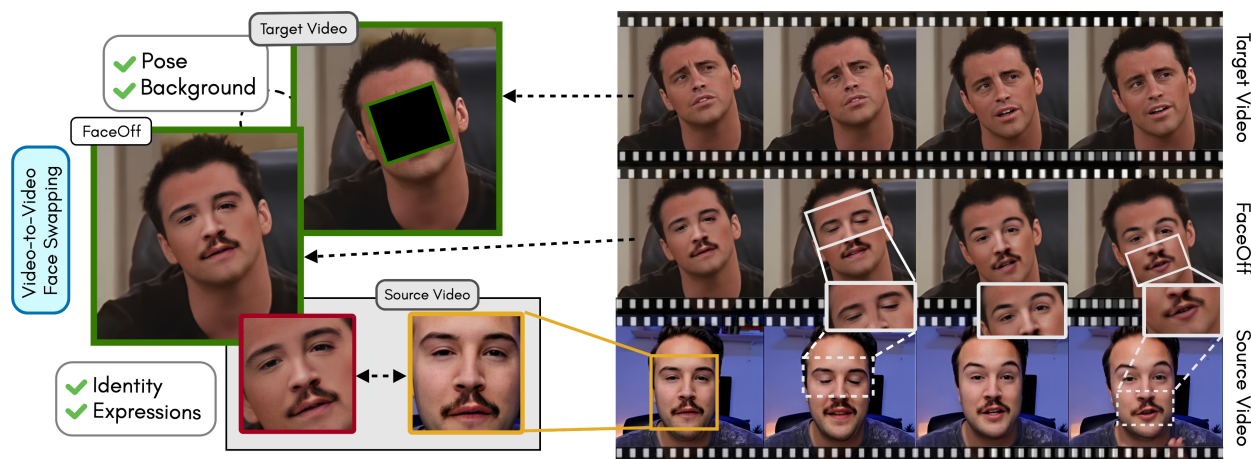


Figure 1: We introduce video-to-video (V2V) face-swapping, a novel task of face-swapping that aims to swap the **identity** and **expressions** from a source face video to a target face video. This differs from the face-swapping task that aims to swap only an identity. There are many downstream applications of V2V face-swapping, such as automating the process of an actor replacing their double in movie scenes, which today, is handled manually using expensive CGI technology. In this example, Nolan, an actor (source video), is recording his dialogues and expressions at the convenience of his home. Joey Tribiani (target video) is acting as his double in a scene of the famous sitcom FRIENDS. FaceOff face-swaps Nolan into the scene. Please note the zoomed-in source (yellow box) and face-swapped (red box) output. In this output, although the source face pose and skin complexion has changed and blended with the background, identity and expressions are preserved.

Abstract

Doubles play an indispensable role in the movie industry. They take the place of the actors in dangerous stunt scenes or in scenes where the same actor plays multiple characters. The double’s face is later replaced with the actor’s face and expressions manually using expensive CGI technology, costing millions of dollars and taking months to complete. An automated, inexpensive, and fast way can be to use face-swapping techniques that aim to swap an identity from a source face video (or an image) to a target face video. However, such methods can not preserve the source expressions of the actor important for the scene’s context. To tackle this challenge, we introduce video-to-video (V2V) face-swapping, a novel task of face-swapping that can preserve (1) the identity and expressions of the source (actor) face video and (2) the background and pose of the target (double) video. We propose FaceOff, a V2V face-swapping

system that operates by learning a robust blending operation to merge two face videos following the constraints above. It first reduces the videos to a quantized latent space and then blends them in the reduced space. FaceOff is trained in a self-supervised manner and robustly tackles the non-trivial challenges of V2V face-swapping. As shown in the experimental section, FaceOff significantly outperforms alternate approaches qualitatively and quantitatively.

1. Introduction

Having doubles¹ for the starring actors in movies is an indispensable component of movie-making. A double may take the actor’s place during stunt scenes involving difficult and dangerous life-risking acts. They may even stand-in for the actor during regular fill scenes or multiple retakes. For

¹[https://en.wikipedia.org/wiki/Double_\(filmmaking\)](https://en.wikipedia.org/wiki/Double_(filmmaking))

instance, ‘The Social Network’ extensively used body doubles as a stand-in for actor Armie Hammer who played multiple roles of twin brothers²³⁴. In such scenes, the double’s face is later replaced by the actor’s face and expressions using CGI technology requiring hundreds of hours of manual multimedia edits on heavy graphical units costing millions of dollars and taking months to complete. Thus, the production team is generally forced to avoid such scenes by changing the mechanics of the scene such that only the double’s body is captured to provide an illusion of the actor. This may act as a constraint to the director’s creativity. However, such adjustments are not always possible.

A different scenario is post-production scene modifications. If a dialogue is discovered in post-production that suits a scene better than the original, the entire scene is reset and re-shot. We propose that the actor could instead record in a studio and get their face superimposed on the previous recording. In fact, like other industries, the movie industry is also headed in this direction where actors can work from home. In today’s era, CGI technologies can produce incredible human structures, scenes, and realistic graphics. However, it is known that they struggle to create realistic-looking skin⁵. As shown in Fig. 1, an actor could lend their identity and expressions from the comfort of their home or studio while leaving the heavy-duty to graphics or a double. Today’s CGI technologies needed for such tasks are, however, manually operated, expensive and time-consuming.

To automate such tasks, fast and inexpensive computer vision based face-swapping [17, 23, 15, 14, 11, 1] techniques that aim to swap an identity between a source (actor) video and target (double) video can be considered. However, such techniques cannot be directly used. Face-swapping swaps only the source identity whilst retaining the rest of the target video characteristics. In this case, the expressions of the actor (source) are not captured in the output. To tackle this, we introduce “video-to-video (V2V) face-swapping” as a novel task of face-swapping that aims to (1) swap the identity and expressions of a source face video and (2) retain the pose and background of the target face video. The target pose is essential as it depends on the scene’s context. E.g., a stunt man performs at an outdoor location dealing with machines or talking to a fellow double; the actor acts in front of a green screen at a studio. Here, the double’s pose is context-aware, and the actor only improvises.

How is the proposed task a video-to-video face-swapping task? Unlike the face-swapping task that swaps a fixed identity component from one video to another video, V2V face-swapping swaps expressions changing over time

²Captain America - Skinny Steve Rogers Behind the Scenes

³How CGI made Cody and Caleb as PAUL WALKER — VFX

⁴Armie Hammer Didn’t Play Both Winklevoss Twins Social Network

⁵Why It’s SO HARD To Do CGI Skin!

(a video) with another video with changing pose and background (another video), making our task video-to-video.

Approach: Swapping faces across videos is non-trivial as it involves merging two different motions - the actor’s finer face motion (such as eye, cheek, or lip movements) and the double’s head motion (such as pose and jaw motion). This needs a network that can take two different motions as input and produce a third coherent motion. We propose **FaceOff**, a video-to-video face swapping system that operates by reducing the face videos to a quantized latent space and blending them in the reduced space. A fundamental challenge in training such a network is the absence of ground truth. Face-swapping approaches [23, 15, 17] use a discriminator-generator setup for training the networks. The discriminator is responsible for monitoring the desired characteristic of the swapped output. However, using a discriminator leads to hallucinating components of the output different from the input. For instance, modified identity or novel expressions. Thus, we devise a self-supervised training strategy for training our network: We use a single video as the source and target. We then introduce pseudo motion errors on the source video. Finally, we train a network to ‘fix’ these pseudo errors to regenerate the source video.

FaceOff can face-swap unseen cross-identities directly at inference without any finetuning. Moreover, unlike most of the face-swapping methods that need inference time optimization ranging from 5 minutes to 24 hours on high-end GPUs, FaceOff face-swaps videos in just one forward pass taking less than a second. A key feature of FaceOff is that it preserves at least one of the input expressions (source in our case), whereas, as we show later, existing methods fail to preserve either of the expressions (source or target expressions). Lastly, we curate and benchmark V2VFaceSwap, a V2V face-swapping test dataset made of instances from unconstrained YouTube videos on unseen identities, background, and lighting conditions.

Our contributions in this work are as follows: (1) We introduce V2V face-swapping, a novel task of face-swapping that aims to swap source face identity and expressions whilst retaining the target background and pose. (2) We propose FaceOff: a V2V face-swapping system trained in a self-supervised manner. FaceOff generates coherent videos by merging two different face videos. (3) Our approach works on unseen identities directly at the inference time without any finetuning. (4) Our approach does not need any inference time optimization taking less than a second for inference. (5) We release V2VFaceSwap test dataset and establish a benchmark for V2V face-swapping task.

2. Related Work

Table 1 provides a comparison between the existing tasks and FaceOff. FaceOff aims to solve a unique challenge of V2V face-swapping that has not been tackled before.

162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215

Method	Source		Target	
	Identity	Expression	Pose	Background
Face Swapping	✓	×	✓	✓
Face Reenactment	×	✓	×	✓
Face Editing	×	×	✓	✓
FaceOff (Ours)	✓	✓	✓	✓

Table 1: Comparison of FaceOff with existing tasks. ✓ and × indicate the characteristic is preserved and lost respectively. FaceOff solves a unique task of preserving source identity and expressions that has not been tackled before.

Face Swapping: Swapping faces across images and videos have been well-studied [17, 15, 23, 2, 10, 11, 14, 1, 3] over the years. These works aim to swap an identity obtained from a source video (or an image) with a target video of a different identity such that all the other target characteristics are preserved in the swapped output. DeepFakes⁶, DeepFaceLabs [17], and FSGAN [15] swap the entire identity of the source; Motion-coseg [23] specifically swaps the identity of single/multiple segments of a given source image (either hair or lips or nose, etc.) to a target video. Unlike these approaches that swap only the identity or a specific part of an image, we swap temporally changing expressions along with the identity of the source. Moreover, FSGAN takes 5 minutes of inference time optimization, DeepFaceLabs and DeepFakes take up to 24 hours of inference time optimization on high-end GPUs. FaceOff takes less than a second to face swap in-the-wild videos of unseen identities.

Face Manipulation: Face manipulation animates the pose and expressions of a target image/video according to a given prior [30, 24, 22, 31, 17, 33, 25, 35]. In audio-driven talking face generation [18, 19, 12, 34, 25, 21, 7], the expressions, pose, and lip-sync in the target video are conditioned on a given input speech audio. Unlike such works, we do not assume an audio prior for our approach. A different direction of **face reenactment** animates the source face movements according to the driving video [26, 21, 27, 9, 22, 24]. The identity is not exchanged in these works. This can tackle a special case of our task – when the target and source have the same identity. Here, a target image can be re-enacted according to the source video expressions. As we show in Section 4.2, FaceOff captures the micro-expression of the driving video, unlike the existing approaches. This is because we rely on a blending mechanism - allowing a perfect transfer of the driving expressions. Another direction that handles this special case is **face editing** that involves editing the expressions of a face video. Using this approach, one can directly edit the target video according to the source expressions. Image-based face editing works such as [8, 4, 5, 13] have gained considerable attention. However, realizing these edits on a

⁶<https://github.com/deepfakes/faceswap>

sequence of frames without modeling the temporal dynamics often results in temporally incoherent videos. Recently, STIT [28] was proposed that can coherently edit a given video to different expressions by applying careful edits in the video’s latent space. Despite the success, these techniques allow limited control over the types and variations in expressions. Moreover, obtaining a correct target expression that matches the source expressions is a manual hit and trial. FaceOff can add micro-expressions undefined in the label space simply by blending the emotion from a different video of the same identity with the desired expressions.

3. FaceOff: Face Swapping in videos

We aim to swap a source face video with a target face video such that (1) the identity and the expression of the source video is preserved and (2) the pose and background of the target video is retained. To do this, we learn to blend the foreground of the source face video with the background and pose of the target face video (as shown in Fig. 3) such that the blended output is coherent and meaningful. This is non-trivial as it involves merging the two separate motions (finer foreground expression motion of the source; head and background motion of the target). Please note that we only aim to blend the two motions, thus, the desired input characteristics – identity, expressions, pose, and background – are naturally retained from the inputs without any additional supervision. The main challenge of our blending approach is to align the foreground and background videos in a way that the output forms a coherent identity and has a single coherent pose. All the other characteristics are simply reconstructed from the inputs. Our core idea is to use a special temporal autoencoding model that merges these motions using a quantized latent space. Overall, our approach relies on (1) Encoding the two input motions to a quantized latent space and learning a robust blending operation in the reduced space. (2) A temporally and spatially coherent decoding. (3) In the absence of ground truth, a self-supervised training scheme.

3.1. Merging Videos using Quantized Latents

We pose face-swapping in videos as a blending problem: given two videos as input, blend the videos into a coherent and meaningful output. To do so, we rely on an encoder to encode the input videos to a meaningful latent space. Our overall network is a special autoencoder that can then learn to blend the reduced videos in the latent space robustly and generate a blended output. We select our encoder model carefully, focusing on “blending” rather than learning an overall data distribution. Encoder networks with a continuous latent space reduce the dimension of a given input, often down to a single vector that can be considered a part of an underlying distribution. This latent vector is highly stochastic; a very different latent is generated for each new

270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323

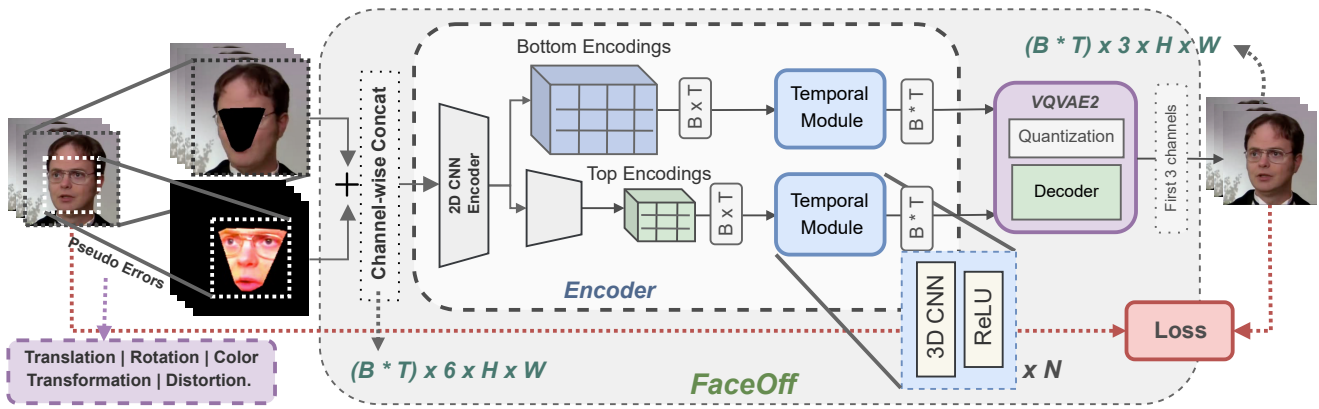


Figure 2: FaceOff is a temporal autoencoder operating in a hierarchical quantized latent space. We use a self-supervised training scheme to train FaceOff using a distance loss on the exact output-ground truth pairs. In the scheme, we first extract the face, f , and background, b , from a single video, s . We then apply “pseudo errors” made of random rotation, translation, scaling, colors, and non-linear distortions to modify f . Next, modified f (acting as a source) and b (acting as a target) are concatenated at each corresponding frame channel-wise to form a single video input. This video input is then reduced, blended, and a coherent and meaningful output is generated. This output is expected to match the original source video, s .

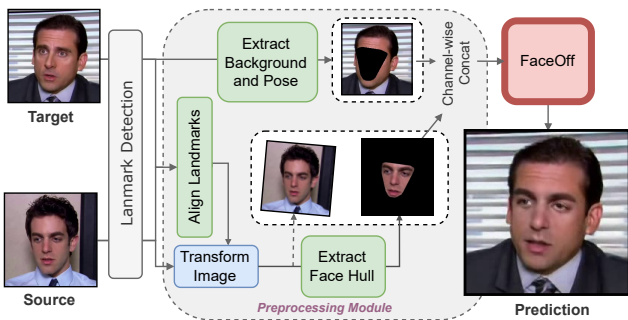


Figure 3: Inference pipeline: FaceOff can be directly inferred on any unseen identity without any finetuning. At inference, the source video is first aligned frame-by-frame using the target face landmarks. FaceOff then takes (1) foreground of aligned source video, and (2) the background and pose of target video as input and generates the output.

input, introducing high variations that a decoder needs to handle. Recently, “vector quantization” was proposed in [16, 6, 20]. Quantization reduces the variation in latents by fixing the number of possible latent codes. However, retaining the input properties using a single quantized latent vector is impossible. Thus the inputs are reduced to a higher dimensional quantized space (such as 64×64) such that properties of the input needed for a full reconstruction is preserved. We adopt such an encoder in our proposed autoencoder for encoding our videos. As shown in Fig. 2, our encoder is a modified VQVAE2 [20] encoder that encodes videos instead of images. To do so, we introduce temporal modules made of non-linear 3D convolution operations.

The input to our encoder is a single video made by concatenating the source foreground and target background frames channel-wise as shown in Fig. 3. Like VQVAE2, our encoder first encodes the concatenated video input frame-wise into 32×32 and 64×64 dimensional top and bottom hierarchies respectively. Before the quantization step at each of these hierarchies, our temporal modules are further added that process the reduced video frames. This step allows the network to backpropagate with temporal connections between the frames. The further processing is then again done in a framewise manner using a standard VQVAE2 decoder. In practice, we observed that this temporal module plays an important role in generating temporally coherent outputs as we show through ablations in Sec. 5. Our special autoencoder differs from standard autoencoders in the loss computation step. Instead of reconstructing the inputs, a six-channel video input – first three channels belonging to the source foreground and the last three channels belonging to the target pose and background – FaceOff aims to generate a three channel blended video output. Therefore, the loss computation is between a ground truth three-channel video and the three-channel video output.

3.2. Self-supervised Training Approach

Existing face-swapping approaches employ generators and discriminators to train their networks. These discriminators are classifiers that indicate a relationship between the generator’s outputs and underlying data distribution, such as an identity distribution or an expression distribution. In such a setup, the generators are encouraged to hallucinate some aspects of the outputs to match the discriminator’s data distribution causing it to output novel identities or ex-



Figure 4: Existing face-swapping methods [17, 23, 15] use a generator-discriminator training strategy. This results in outputs with novel expressions as explained in Sec. 3.2. We show this phenomenon on DeepFaceLabs [17]. The expressions in the output (red boxes) does not match either of the inputs, source or target. E.g., direction of the eye gaze (second row), or overall laugh expression (first row). FaceOff successfully preserves the source expressions (green boxes).

pressions. We show this phenomenon in Fig. 4. A hard distance loss (e.g. Euclidean distance) indicating the exact output-ground truth relationship instead of a stochastic discriminator loss can be used to overcome this issue. In V2V face-swapping, an important aspect is to retain the exact source expressions. Thus, we train our network using a distance loss by devising a self-supervised training scheme that forces the network to reconstruct a denoised version of a given input video allowing us to use a distance loss.

To understand the training scheme, we first look at the challenges we encounter when trying to blend two motions naively. First, there is a global and local pose difference between the faces in the source and target videos. We fix the global pose difference by aligning (rotating, translating, and scaling) the source poses according to the target poses using face landmarks, as shown in Fig. 3. However, the local pose difference is not overcome this way, and we observe temporal incoherence across the frames. Next, we observe a difference in the foreground and background color (illumination, hue, saturation, and contrast). Thus, we train our network to solve these known issues by reproducing these errors during training. As illustrated in Fig. 2, we train our model in the following manner: (1) Take a video, say s . (2) From s , extract the face region, say f ; and the background region, say b . (3) Introduce pseudo errors (rotation, color, scale, etc.) on f . (4) Construct the input v by concatenating f and b channel-wise at every corresponding frame. (5) Train the network to construct s from v . Although we train the network using the same identity in the self-supervised scheme, it can face-swap unseen identities directly at infer-

ence without any finetuning. We encourage our readers to view the supplementary video for results.

3.3. Reproducing Inference Errors at Training

Given two talking-head videos, source and target denoted by S and T respectively, our aim is to generate an output that preserves (1) the identity and the emotions from S and (2) the pose and background from T . We assume the number of frames, denoted by N , in S and T are equal. Given two frames, $s_i \in S$ and $t_i \in T$ such that $i = 1 \dots N$, we denote $f_{s_i} \in F_s$ and $b_{t_i} \in B_t$ as the foreground and background of s_i and t_i respectively. Given F_s and B_t as input, the network fixes the following issues:

First, the network encounters a local pose difference between f_{s_i} and b_{t_i} . This pose difference can be fixed using an affine transformation function: $\delta(f_{s_i}, b_{t_i}) = m(rf_{s_i} + d) + m(rb_{t_i} + d)$ where m , r , and d denote scaling, rotation, and translation. Face being a non-rigid body, this affine transformation only results in the two faces with a perfect match in pose but a mismatch in shape. One can imagine trying to fit a square in a circle. One would need a non-linear function to first transform the square to a shape similar to the circle so that they fit. We denote this non-linear transformation as a learnable function $\omega(f_{s_i}, b_{t_i})$. Being non-linear in nature, a network can perform any one of many such transformations on the input frames as long as both faces fit. These transformations can be constrained using a distance loss to encourage spatially-consistent transformations that generate a coherent and meaningful frame. However, these spatially-consistent transformations may be temporally-inconsistent across the video. This would result in a video with a face that wobbles as shown in the ablation Sec. 5. Thus, we constrain the transformations as $\omega(f_{s_i}, b_{t_i}, f_{s_k}, b_{t_k})$ where $k = 1 \dots N$ such that $k \neq i$. Here, the transformation on the current frame is constrained by the transformations on all the other frames in the video. This is enabled by the temporal module as explained in Sec. 3.1. Lastly, the network encounters a difference in color (contrast, hue, saturation, etc.) between f_{s_i} and b_{t_i} that is fixed as $c(f_{s_i}, b_{t_i})$.

As shown in Fig. 2, at the time of training $S = T$. For each frame $s_i \in S$, we first extract the foreground, $f_{s_i} \in F_s$ (acting as the source) and the background, $b_{t_i} \in B_t$ (acting as the target) from s_i . Next, we apply random rotation, translation, scaling, color, and distortion (Barrel, Mustache) errors on f_{s_i} . The training setting is then formulated as:

$$\Phi : \Omega(\delta, \omega, c) \quad (1)$$

$$J = \frac{1}{N} \sum_{i=1}^N [s_i - \Phi(f_{s_i}, b_{t_i}, f_{s_k}, b_{t_k})] + P(F_s, B_t) \quad (2)$$

where Ω is a learnable function, J is the overall cost of the network to be minimized, and P is a perceptual metric (LPIPS [32] in our case), and $k = 1 \dots N$ such that $k \neq i$.

486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539

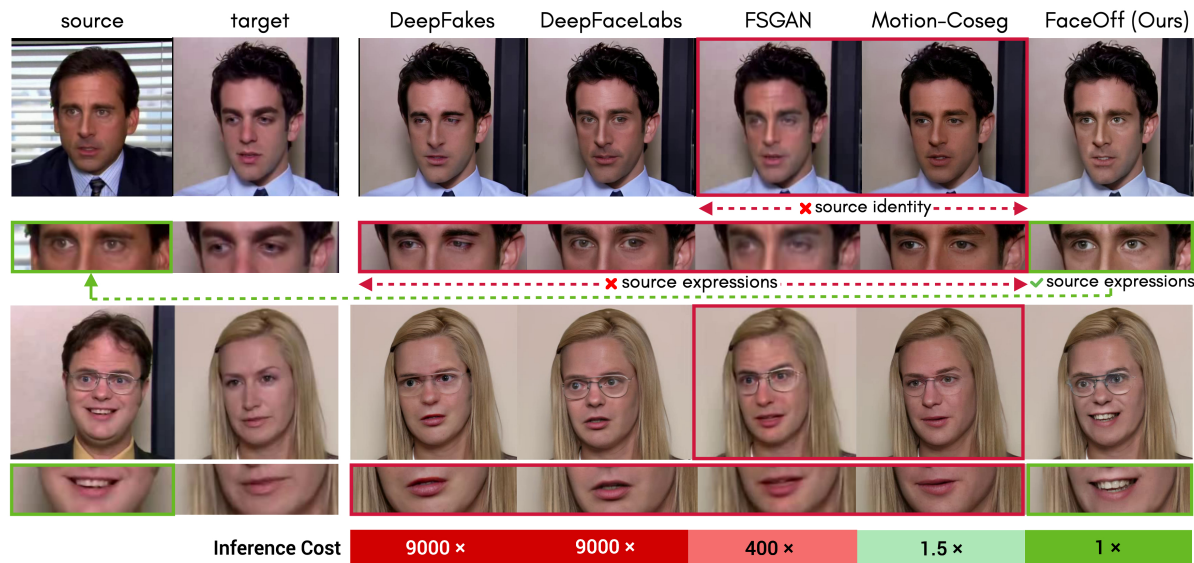


Figure 5: “Inference Cost” denotes the time taken for a single face-swap. FSGAN, with $400\times$ FaceOff’s inference cost, fails to swap the identities fully. DeepFakes and DeepFaceLabs swap the identities successfully but are $9000\times$ less efficient than FaceOff. FaceOff perfectly swaps source identity and expressions. None of the other methods can swap source expressions.

Method	Quantitative Evaluation					Human Evaluation		
	SPIDis ↓	LMD ↓	TL-ID ↑	TG-ID ↑	FVD ↓	Identity ↑	Exps. ↑	Ntrl. ↑
Motion-coseg [23]	0.48	0.59	0.872	0.893	293.652	6.82	5.81	7.44
FSGAN [15]	0.49	0.57	0.914	0.923	242.691	7.84	6.83	8.31
FaceOff (Ours)	0.38	0.41	0.925	0.915	255.980	9.64	9.86	8.18

Table 2: Quantitative metrics on V2VFaceSwap dataset. DeepFakes and DeepFaceLabs take upto 24 hours for best inference on a single face-swap [17], thus we do not compare with them. The metrics used for comparisons is explained in Sec. 4. For fair comparisons, FSGAN scores are reported without any inference time optimization. Although FSGAN has a slightly higher FVD and Naturalness (Ntrl.) score, it fails to swap the identity fully as can be clearly seen from SPIDis, LMD, and Identity metric. Moreover, the difference in the FVD of FSGAN and FaceOff is not statistically significant perceptually [29].

4. Experiments and Results

In this section, we try to answer the following questions: (1) How well can we preserve the source identity compared to the alternate approaches? (2) How well do we preserve the expressions of the input videos? (3) How efficient is FaceOff when compared to other techniques?

We compare FaceOff against different tasks: “face-swapping”, “face reenactment”, and “face editing”. Please note that none of these methods can fully solve the task of V2V face-swapping that we aim to solve. Specifically, V2V face-swapping aims to (1) swap source identity and expressions and (2) retain the target pose and background.

Quantitative Metrics: (1) **Source-Prediction Identity Distance (SPIDis)**: computes the difference in identity between face images. It is computed as the euclidean distance between the face embeddings generated using dlib’s face detection module. (2) **Fréchet Video Distance (FVD)**, as

proposed in [29], computes the temporal coherence in the generated video output. (3) **Landmark Distance (LMD)**: evaluates the overall face-structure and expressions of the source and swapped-output. To compute LMD, the source, and the swapped face landmarks are normalized: faces are first centered and then rotated about the x-axis such that the centroid and angle between the eye coordinates, respectively align a mean image. Next, the faces are scaled with respect to the mean image. Euclidean distance between the normalized swapped and source video landmarks gives the LMD. We compute LMD between the source and the output face expressions (excluding the landmarks of the face perimeter). (4) **Temporally Locally (TL-ID) and Temporally Globally (TG-ID) Identity Preservation**: proposed in [28]. They evaluate a video’s identity consistency at a local and global level. For both metrics, a score of 1 would indicate that the method successfully maintains the identity consistency of the original video.



Figure 6: Qualitative results of FaceOff. Note that there is a significant difference in the source and target expressions in all the cases. FaceOff swaps the source expressions (mouth, eyes, etc.) and identity; and retains the target pose and background.

Qualitative Metrics: A mean absolute opinion score on a scale of 1 – 10 is reported for (1) **Identity:** How similar is the swapped-output identity with the source identity? (2) **Expressions (Exps.):** How similar is the swapped-output expression with the source expression?, and (3) **Naturalness (Ntrl.):** Is the generated output natural?

Experimental Dataset: We benchmark V2VFaceSwap dataset made of unconstrained YouTube videos, with many unseen identities, backgrounds, and lighting conditions.

We strongly encourage our readers to view the supplementary video for best experience. Subjective human opinion, further details about the dataset, and evaluation setup are reported in the supplementary paper.

4.1. Face-Swapping Results

Fig. 5 and Table 2 present a qualitative and quantitative comparison respectively between the existing methods and FaceOff. Fig. 6 demonstrates FaceOff’s face-swapping results on videos. As shown in Fig. 5, FaceOff successfully swaps the identity and expressions of the source face video. Existing methods cannot swap the source expressions which shows that FaceOff solves a unique challenge of V2V face-swapping. An interesting finding of our experiments is that the existing methods do not preserve any of the input expressions – source or target – at the output and generates novel expressions, e.g., novel gaze direction or mouth movements. This phenomenon is also demonstrated in Fig. 4. FSGAN and Motion-Coseg fail to swap the identity entirely. This is further corroborated through quantitative metrics in Table 2. As shown, FaceOff has an improvement of $\sim 22\%$ and $\sim 28\%$ on SPIDis and LMD



Figure 7: Qualitative demonstration of Face Manipulation. As can be seen, none of the methods, except FaceOff, preserve the source expressions or pose information perfectly.

over FSGAN, clearly indicating FaceOff’s superiority.

FSGAN achieves a slightly higher FVD and is voted more natural in human evaluation. This is expected as FSGAN does not change the target identity much and thus retains the original target video making it more natural to observe. FaceOff swaps identity near-perfectly. Moreover, existing methods only have a single target motion to follow. FaceOff tackles an additional challenge of motion-to-motion swapping that needs source-target pose alignment at every frame in a temporally coherent manner. This requires FaceOff to generate a novel motion such that the identity, expressions, and pose in the motion look natural and match the inputs. Despite this challenge, the difference in FSGAN and FaceOff’s FVD is not perceptually significant, as stated

in [29]. DeepFaceLabs and DeepFakes swap identity well but are $9000\times$ more computationally expensive than FaceOff, making FaceOff much more scalable and applicable in the real world.

4.2. Target Face Manipulation Results

Given that the source and target have the same identity, the problem reduces to the following - transfer expressions from a source video to a target video. This is fundamentally the setting of “face reenactment”. One could also modify the expression of the target by identifying and quantifying the source expressions and using a “face-editing” network to edit the target expressions. Fig. 7 presents a qualitative comparison between FaceOff, “face reenactment” (FaceVid2Vid) and “face-editing” (STIT).

Face Reenactment: We compare against FaceVid2Vid [30], a SOTA face reenactment network that reenacts the pose and expression of a target image using source (driving) video. As shown in Fig. 7, FaceOff preserves source’s micro-expression such as, exact mouth opening, eye-frown. As FaceOff relies on a deterministic distance loss, it can retain the exact input expressions in the output. Moreover, FaceOff retains the temporal target pose and background whereas Face-Vid2Vid modifies a static frame.

Face Editing: Using a powerful neural network, one can simply introduce the desired expressions in a video by performing edits. We compare our method against STIT [28]. STIT modifies the expressions of a face video based on an input label. We observe the source expression and manually try out the various intensity of the “smile” emotion ranging from the negative to positive direction. As seen in Fig. 7, although STIT can change the overall expression, it needs a significant manual hit-and-trial to pinpoint the exact expression. It also lacks personalized expression (amount of mouth opening, subtle brow changes). Also, each and every expression cannot be defined using a single label, and introducing variations in emotion along the temporal dimension is hard. With our proposed method, one can incorporate any emotion in the video (as long as we have access to a source video).

5. Ablation Study

We investigate the contribution of different modules and errors in achieving FaceOff. Fig. 8 demonstrates the performance of FaceOff without the proposed temporal module. As shown, although at a frame level, the output is spatially-coherent, as we look across the frames, we can notice the temporal incoherence. The face seems to ‘wobble’ across the frames - squishing up and down. In fact, without the temporal module, the network does not understand an overall face structure and generates unnatural frames (marked in red). Jumping from one red box to another, we can see that the face structure has completely changed. This

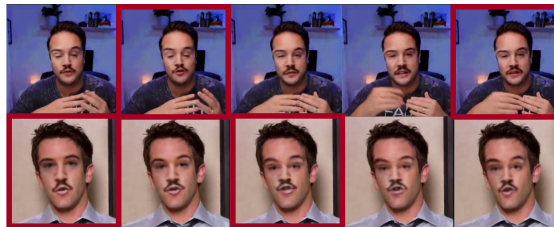


Figure 8: FaceOff without Temporal Module. As we jump from one frame to another (red boxes), we can observe a “wobble effect”: significant change in the facial structure (elongated and then squeezed). This occurs as the model does not have an understanding of the neighboring frames while generating the current frame.

Component	SPIDis ↓	LMD ↓	FVD ↓
FaceOff	0.38	0.41	255.980
w/o Temporal.	0.71	0.49	350.60
w/o Rotation	0.65	0.44	292.76
w/o Color	0.74	0.42	303.35
w/o Translation	0.58	0.47	271.82
w/o Distortion	0.55	0.45	285.54

Table 3: We remove different components and errors and evaluate their contributions in achieving FaceOff.

suggests that constraining the network by the neighboring frames using the temporal module enables the network to learn a global shape fitting problem, consequently generating a temporally coherent output.

Table 3 presents the quantitative contribution of the temporal module and each of the errors used for self-supervised training. The metrics indicate that each of them contributes significantly to achieving FaceOff.

6. Conclusion

We introduce “video-to-video (V2V) face-swapping”, a novel task of face-swapping. Unlike face-swapping that aims to swap an identity from a source face video (or an image) to a target face video, V2V face-swapping aims to swap the source expressions along with the identity. To tackle this, we propose FaceOff, a self-supervised temporal autoencoding network that takes two face videos as input and produces a single coherent blended output. As shown in the experimental section, FaceOff swaps the source identity much better than the existing approaches while also being $400\times$ computationally efficient. It also swaps the exact source identity that none of the methods can do. V2V face-swapping has many applications, a significant application can be automating the task of replacing the double’s face with the actor’s identity and expressions in movies. We believe our work adds a whole new dimension to movie editing that can potentially save months of tedious manual effort and millions of dollars.

810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863

References

- [1] Dmitri Bitouk, Neeraj Kumar, Samreen Dhillon, Peter Belhumeur, and Shree K. Nayar. Face swapping: Automatically replacing faces in photographs. *ACM Trans. Graph.*, 27(3):1–8, aug 2008. 2, 3
- [2] Renwang Chen, Xuanhong Chen, Bingbing Ni, and Yanhao Ge. SimSwap. In *Proceedings of the 28th ACM International Conference on Multimedia*. ACM, oct 2020. 3
- [3] Yi-Ting Cheng, Virginia Tzeng, Yu Liang, Chuan-Chang Wang, Bing-Yu Chen, Yung-Yu Chuang, and Ming Ouhyoung. 3d-model-based face replacement in video. 01 2009. 3
- [4] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. 2017. 3
- [5] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains, 2019. 3
- [6] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis, 2020. 4
- [7] P. Garrido, L. Valgaerts, H. Sarmadi, I. Steiner, K. Varanasi, P. Pérez, and C. Theobalt. Vdub: Modifying face video of actors for plausible visual alignment to a dubbed audio track. *Comput. Graph. Forum*, 34(2):193–204, may 2015. 3
- [8] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2016. 3
- [9] Hyeonwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *ACM Transactions on Graphics (TOG)*, 37(4):163, 2018. 3
- [10] Iryna Korshunova, Wenzhe Shi, Joni Dambre, and Lucas Theis. Fast face-swap using convolutional neural networks, 2016. 3
- [11] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. Faceshifter: Towards high fidelity and occlusion aware face swapping, 2019. 2, 3
- [12] Ian Magnusson, Aruna Sankaranarayanan, and Andrew Lippman. Invertible frowns: Video-to-video facial emotion translation, 2021. 3
- [13] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014. 3
- [14] J. Naruniec, L. Helminger, C. Schroers, and R.M. Weber. High-resolution neural face swapping for visual effects. *Computer Graphics Forum*, 39:173–184, 07 2020. 2, 3
- [15] Yuval Nirkin, Yosi Keller, and Tal Hassner. Fsgan: Subject agnostic face swapping and reenactment, 2019. 2, 3, 5, 6
- [16] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2017. 4
- [17] Ivan Perov, Daiheng Gao, Nikolay Chervoniy, Kunlin Liu, Sugasa Marangonda, Chris Umé, Mr. Dpfks, Carl Shift Facenheim, Luis RP, Jian Jiang, Sheng Zhang, Pingyu Wu, Bo Zhou, and Weiming Zhang. Deepfacelab: Integrated, flexible and extensible face-swapping framework, 2020. 2, 3, 5, 6
- [18] K R Prajwal, Rudrabha Mukhopadhyay, Vinay P. Namboodiri, and C.V. Jawahar. A lip sync expert is all you need for speech to lip generation in the wild. In *Proceedings of the 28th ACM International Conference on Multimedia*. ACM, oct 2020. 3
- [19] Prajwal K R, Rudrabha Mukhopadhyay, Jerin Philip, Abhishek Jha, Vinay Namboodiri, and C V Jawahar. Towards automatic face-to-face translation. In *Proceedings of the 27th ACM International Conference on Multimedia*. ACM, oct 2019. 3
- [20] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2, 2019. 4
- [21] Yurui Ren, Ge Li, Yuanqi Chen, Thomas H. Li, and Shan Liu. Pirenderer: Controllable portrait image generation via semantic neural rendering, 2021. 3
- [22] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 3
- [23] Aliaksandr Siarohin, Subhankar Roy, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Motion-supervised co-part segmentation. 2020. 2, 3, 5, 6
- [24] Aliaksandr Siarohin, Oliver J. Woodford, Jian Ren, Menglei Chai, and Sergey Tulyakov. Motion representations for articulated animation. 2021. 3
- [25] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Nießner. Neural voice puppetry: Audio-driven facial reenactment. 2019. 3
- [26] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures, 2019. 3
- [27] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. 2020. 3
- [28] Rotem Tzaban, Ron Mokady, Rinon Gal, Amit H. Bermano, and Daniel Cohen-Or. Stitch it in time: Gan-based facial editing of real videos, 2022. 3, 6, 8
- [29] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric challenges, 2018. 6, 8
- [30] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing, 2020. 3, 8
- [31] Wayne Wu, Yunxuan Zhang, Cheng Li, Chen Qian, and Chen Change Loy. Reenactgan: Learning to reenact faces via boundary transfer, 2018. 3
- [32] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. 5
- [33] Zhimeng Zhang, Lincheng Li, Yu Ding, and Changjie Fan. Flow-guided one-shot talking face generation with a high-resolution audio-visual dataset. In *2021 IEEE/CVF Confer-*

918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971

972		1026
973		1027
974		1028
975	[34]	1029
976		1030
977		1031
978	[35]	1032
979		1033
980		1034
981		1035
982		1036
983		1037
984		1038
985		1039
986		1040
987		1041
988		1042
989		1043
990		1044
991		1045
992		1046
993		1047
994		1048
995		1049
996		1050
997		1051
998		1052
999		1053
1000		1054
1001		1055
1002		1056
1003		1057
1004		1058
1005		1059
1006		1060
1007		1061
1008		1062
1009		1063
1010		1064
1011		1065
1012		1066
1013		1067
1014		1068
1015		1069
1016		1070
1017		1071
1018		1072
1019		1073
1020		1074
1021		1075
1022		1076
1023		1077
1024		1078
1025		1079