Chapter 4

Toward the Creation and Obstruction of DeepFakes

Yuezun Li, Pu Sun, Honggang Qi, and Siwei Lyu

Abstract AI-synthesized face-swapping videos, commonly known as *DeepFakes*, is an emerging problem threatening the trustworthiness of online information. The need to develop and evaluate DeepFake detection algorithms calls for large-scale datasets. However, current DeepFake datasets suffer from low visual quality and do not resemble DeepFake videos circulated on the Internet. We present a new large-scale challenging DeepFake video dataset, *Celeb-DF*, which contains 5,639 high-quality DeepFake videos of celebrities generated using an improved synthesis process. We conduct a comprehensive evaluation of DeepFake detection methods and datasets to demonstrate the escalated level of challenges posed by Celeb-DF. Then we introduce *Landmark Breaker*, the first dedicated method to disrupt facial landmark extraction, and apply it to the obstruction of the generation of DeepFake videos. The experiments are conducted on three state-of-the-art facial landmark extractors using our Celeb-DF dataset.

Y. Li (✉)
Ocean University of China, Qingdao, China
e-mail: liyuezun@ouc.edu.cn

P. Sun · H. Qi
University of Chinese Academy of Sciences, Beijing, China
e-mail: sunpu18@mailsucas.ac.cn

H. Qi
e-mail: hgqi@ucas.ac.cn

S. Lyu
University at Buffalo, Buffalo, NY, USA
e-mail: siweilyu@buffalo.edu

State University of New York, Buffalo, NY, USA

4.1 Introduction

A recent twist to the disconcerting problem of online disinformation is falsified videos created by AI technologies, in particular, deep neural networks (DNNs). Although fabrication and manipulation of digital images and videos are not new [15], the use of DNNs has made the process to create convincing fake videos increasingly easier and faster.

One particular type of DNN-based fake video, commonly known as *DeepFakes*, has recently drawn much attention. In a DeepFake video, the faces of a *target* individual are replaced by the faces of a *donor* individual synthesized by DNN models, retaining the target's facial expressions and head poses. Since faces are intrinsically associated with identity, well-crafted DeepFakes can create illusions of a person's presence and activities that do not occur in reality, which can lead to serious political, social, financial, and legal consequences [10].

With the escalated concerns over DeepFakes, there is a surge of interest in developing DeepFake detection methods recently [1, 18, 29, 30, 37, 40–42, 47, 48, 61], with an upcoming dedicated global *DeepFake Detection Challenge*.¹ The availability of large-scale datasets of DeepFake videos is an enabling factor in the development of the DeepFake detection method. To date, we have the UADFV dataset [61], the DeepFake-TIMIT dataset (DF-TIMIT) [26], the FaceForensics++ dataset (FF-DF) [47]², the Google DeepFake detection dataset (DFD) [14], and the Facebook DeepFake detection challenge (DFDC) dataset [12].

However, a closer look at the DeepFake videos in existing datasets reveals stark contrasts in visual quality to the actual DeepFake videos circulated on the Internet. Several common visual artifacts that can be found in these datasets are highlighted in Fig. 4.1, including low-quality synthesized faces, visible splicing boundaries, color mismatch, visible parts of the original face, and inconsistent synthesized face orientations. These artifacts are likely the result of imperfect steps of the synthesis method and the lack of curating of the synthesized videos before included in the datasets. Moreover, DeepFake videos with such low visual qualities can hardly be convincing, and are unlikely to have a real impact. Correspondingly, high detection performance on these datasets may not bear strong relevance when the detection methods are deployed *in the wild*.

In the first section, we present a new large-scale and challenging DeepFake video dataset, *Celeb-DF*,³ for the development and evaluation of DeepFake detection algorithms. There are in total 5, 639 DeepFake videos, corresponding more than 2 million frames, in the Celeb-DF dataset. The real source videos are based on publicly available YouTube video clips of 59 celebrities of diverse genders, ages, and ethnic groups. The DeepFake videos are generated using an improved DeepFake synthesis method. As a result, the overall visual quality of the synthesized DeepFake videos in Celeb-DF is greatly improved when compared to existing datasets, with signifi-

¹ <https://deepfakedetectionchallenge.ai>.

² FaceForensics++ contains other types of fake videos. We consider only the DeepFake videos.

³ <http://www.cs.albany.edu/~lsw/celeb-deepfakeforensics.html>.

Fig. 4.1 Visual artifacts of DeepFake videos in existing datasets. Note some common types of visual artifacts in these video frames, including low-quality synthesized faces (row 1 col 1, row 3 col 2, row 5 col 3), visible splicing boundaries (row 3 col 1, row 4 col 2, row 5 col 2), color mismatch (row 5 col 1), visible parts of the original face (row 1 col 1, row 2 col 1, row 4 col 3), and inconsistent synthesized face orientations (row 3 col 3). This figure is best viewed in color

cantly fewer notable visual artifacts. Based on the Celeb-DF dataset and other existing datasets, we conduct an evaluation of current DeepFake detection methods. This is the most comprehensive performance evaluation of DeepFake detection methods to date. The results show that Celeb-DF is challenging to most of the existing detection methods, even though many DeepFake detection methods are shown to achieve high, sometimes near perfect, accuracy on previous datasets.

In the second section, we describe a white-box method to obstruct the creation of DeepFakes based on disrupting the facial landmark extraction, i.e., Landmark Breaker. The facial landmarks are key locations of important facial parts including tips and middle points of eyes, nose, mouth, eyebrows as well as contours; see Fig. 4.2. Landmark Breaker attacks the facial landmark extractors by adding adversarial perturbations [17, 54], which are image noises purposely designed to mislead DNN-based facial landmark extractors. Specifically, Landmark Breaker attacks facial landmark heat-map prediction, which is the common first step in many recent

Fig. 4.2 The overview of Landmark Breaker on obstructing DeepFake generation by disrupting the facial landmark extraction. The top row shows the original DeepFake generation, and the bottom row corresponds to the disruption after facial landmarks are disrupted. The landmark extractor we use is FAN [7] and the “Heat-maps” is visualized by summing all heat-maps. Note that training of the DeepFake generation model is also affected by disrupted facial landmarks, but is not shown here

DNN-based facial landmark extractors [7, 45, 50]. We introduce a new loss function to encourage errors between the predicted and original heat-maps to change the final locations of facial landmarks. Then we optimize this loss function using the momentum iterative fast gradient sign method (MI-FGSM) [13].

Training the DNN-based DeepFake generation model predicates on aligned input faces as training data, which are obtained by matching the facial landmarks of input face to a standard configuration. Also, in the synthesis process of DeepFakes, the facial landmarks are needed to align the input faces. As Landmark Breaker disrupts the essential face alignment step, it can effectively degrade the quality of the DeepFakes, Fig. 4.2.

We conduct experiments to test Landmark Breaker on attacking three state-of-the-art facial landmark extractors (FAN [7], HRNet [50], and AVS-SAN [45]) using the Celeb-DF dataset [31]. The experimental results demonstrate the effectiveness of Landmark Breaker in disrupting the facial landmark extraction as well as obstructing the DeepFake generation. Moreover, we perform ablation studies for different parameter settings and robustness with regards to image and video compression.

The contribution of this section is summarized as follows:

- We propose a new method to obstruct DeepFake generation by disrupting facial landmark extraction. To the best of our knowledge, this is the first study on the vulnerabilities of facial landmark extractors, as well as their application to the obstruction of DeepFake generation.
- Landmark Breaker is based on a new loss function to encourage the error between predicted and original heat-maps and optimize it using momentum iterative fast gradient sign method.
- We conduct experiments on three state-of-the-art facial landmark extractors and study the performance under different settings including video compression.

4.2 Backgrounds

4.2.1 DeepFake Video Generation

Although in recent years there have been many sophisticated algorithms for generating realistic synthetic face videos [6, 8, 11, 20, 21, 23, 27, 44, 52, 53, 55, 56], most of these have not been in the mainstream as open-source software tools that anyone can use. It is a much simpler method based on the work of neural image style transfer [32] that becomes the *tool of choice* to create DeepFake videos in scale, with several independent open-source implementations, e.g., FakeApp,⁴ DFaker,⁵ faceswap-GAN,⁶ faceswap,⁷ and DeepFaceLab.⁸ We refer to this method as the *basic DeepFake maker*, and it is underneath many DeepFake videos circulated on the Internet or in the existing datasets.

The overall pipeline of the basic DeepFake maker is shown in Fig. 4.3 (left). From an input video, faces of the target are detected, from which facial landmarks are further extracted. The landmarks are used to align the faces to a standard configuration [22]. The aligned faces are then cropped and fed to an auto-encoder [25] to synthesize faces of the donor with the same facial expressions as the original target's faces.

The auto-encoder is usually formed by two convolutional neural networks (CNNs), i.e., the *encoder* and the *decoder*. The encoder E converts the input target's face to a vector known as the *code*. To ensure the encoder capture identity-independent attributes such as facial expressions, there is one single encoder regardless of the identities of the subjects. On the other hand, each identity has a dedicated decoder D_i , which generates a face of the corresponding subject from the code. The encoder and decoder are trained in tandem using uncorresponded face sets of multiple subjects in an unsupervised manner, Fig. 4.3 (right). Specifically, an encoder-decoder pair is formed alternatively using E and D_i for the input face of each subject, and to optimize their parameters to minimize the reconstruction errors (ℓ_1 difference between the input and reconstructed faces). The parameter update is performed with the backpropagation until convergence.

The synthesized faces are then warped back to the configuration of the original target's faces and trimmed with a *mask* from the facial landmarks. The last step involves smoothing the boundaries between the synthesized regions and the original video frames. The whole process is automatic and runs with little manual intervention.

⁴ <https://www.malavida.com/en/soft/fakeapp/>.

⁵ <https://github.com/dfaker/df>.

⁶ <https://github.com/shaoanlu/faceswap-GAN>.

⁷ <https://github.com/deepfakes/faceswap>.

⁸ <https://github.com/iperov/DeepFaceLab>.

Fig. 4.3 Synthesis (left) and training (right) of the basic DeepFake maker algorithm. See texts for more details

4.2.2 DeepFake Detection Methods

Since DeepFakes become a global phenomenon, there has been an increasing interest in DeepFake detection methods. Most of the current DeepFake detection methods use data-driven deep neural networks (DNNs) as a backbone.

Since synthesized faces are spliced into the original video frames, state-of-the-art DNN splicing detection methods, e.g., [5, 33, 63, 64], can be applied. There have also been algorithms dedicated to the detection of DeepFake videos that fall into three categories. Methods in the first category are based on inconsistencies exhibited in the **physical/physiological** aspects in the DeepFake videos. The method in the work of [30] exploits the observation that many DeepFake videos lack reasonable eye blinking due to the use of online portraits as training data, which usually do not have closed eyes for aesthetic reasons. Incoherent head poses in DeepFake videos are utilized in [61] to expose DeepFake videos. In [2], the idiosyncratic behavioral patterns of a particular individual are captured by the time series of facial landmarks extracted from real videos are used to spot DeepFake videos. The second category of DeepFake detection algorithms (e.g., [29, 37]) use **signal-level** artifacts introduced during the synthesis process such as those described in the Introduction. The third category of DeepFake detection methods (e.g., [1, 18, 41, 42]) are **data-driven**, which directly employ various types of DNNs trained on real and DeepFake videos, not relying on any specific artifact.

4.2.3 Existing DeepFake Datasets

DeepFake detection methods require training data and need to be evaluated. As such, there is an increasing need for large-scale DeepFake video datasets. Table 4.1 lists the current DeepFake datasets.

Table 4.1 Basic information of various DeepFake video datasets

Dataset	# Real		# DeepFake		Release Date
	Video	Frame	Video	Frame	
UADFV	49	17.3k	49	17.3k	2018.11
DF-TIMIT-LQ	320 ^a	34.0k	320	34.0k	2018.12
DF-TIMIT-HQ			320	34.0k	
FF-DF	1,000	509.9k	1,000	509.9k	2019.01
DFD	363	315.4k	3,068	2,242.7k	2019.09
DFDC	1,131	488.4k	4,113	1,783.3k	2019.10
Celeb-DF	590	225.4k	5,639	2,116.8k	2019.11

^aThe original videos in DF-TIMIT are from Vid-TIMIT dataset

UADFV: The UADFV dataset [61] contains 49 real YouTube and 49 DeepFake videos. The DeepFake videos are generated using the DNN model with FakeAPP.

DF-TIMIT: The DeepFake-TIMIT dataset [26] includes 640 DeepFake videos generated with faceswap-GAN and is based on the Vid-TIMIT dataset [49]. The videos are divided into two equal-sized subsets: DF-TIMIT-LQ and DF-TIMIT-HQ, with synthesized faces of size 64×64 and 128×128 pixels, respectively.

FF-DF: The FaceForensics++ dataset [47] includes a subset of DeepFakes videos, which has 1,000 real YouTube videos and the same number of synthetic videos generated using faceswap.

DFD: The Google/Jigsaw DeepFake detection dataset [14] has 3,068 DeepFake videos generated based on 363 original videos of 28 consented individuals of various genders, ages, and ethnic groups. The details of the synthesis algorithm are not disclosed, but it is likely to be an improved implementation of the basic DeepFake maker algorithm.

DFDC: The Facebook DeepFake detection challenge dataset [12] is part of the DeepFake detection challenge, which has 4,113 DeepFake videos created based on 1,131 original videos of 66 consented individuals of various genders, ages, and ethnic groups.⁹ This dataset is created using two different synthesis algorithms, but the details of the synthesis algorithm are not disclosed.

Based on release time and synthesis algorithms, we categorize UADFV, DF-TIMIT, and FF-DF as the *first generation* of DeepFake datasets, while DFD, DFDC, and the proposed Celeb-DF datasets are of the *second generation*. In general, the second generation datasets improve in both quantity and quality over the first generation.

⁹ The full set of DFDC has not been released at the time of CVPR submission, and information is based on the first-round release in [12].

4.3 Celeb-DF: the Creation of DeepFakes

The Celeb-DF dataset is comprised of 590 real videos and 5,639 DeepFake videos (corresponding to over two million video frames). The average length of all videos is approximately 13 seconds with the standard frame rate of 30 frame-per-second. The real videos are chosen from publicly available YouTube videos, corresponding to interviews of 59 celebrities with diverse distribution in their genders, ages, and ethnic groups.¹⁰ 56.8% subjects in the real videos are male, and 43.2% are female. 8.5% are of age 60 and above, 30.5% are between 50 and 60, 26.6% are in their 40s, 28.0% are in their 30s, and 6.4% are younger than 30. 5.1% are Asians, 6.8% are African Americans, and 88.1% are Caucasians. In addition, the real videos exhibit a large range of changes in aspects such as the subjects' face sizes (in pixels), orientations, lighting conditions, and backgrounds. The DeepFake videos are generated by swapping faces for each pair of the 59 subjects. The final videos are in MPEG4.0 format.

4.3.1 Synthesis Method

The DeepFake videos in Celeb-DF are generated using an improved DeepFake synthesis algorithm, which is key to the improved visual quality as shown in Fig. 4.4. Specifically, the basic DeepFake maker algorithm is refined in several aspects targeting the following specific visual artifacts observed in existing datasets.

Low resolution of synthesized faces: The basic DeepFake maker algorithm generates low-resolution faces (typically 64×64 or 128×128 pixels). We improve the resolution of the synthesized face to 256×256 pixels. This is achieved by using encoder and decoder models with more layers and increased dimensions. We determine the structure empirically for a balance between increased training time and better synthesis result. The higher resolution of the synthesized faces is of better visual quality and less affected by resizing and rotation operations in accommodating the input target faces, Fig. 4.5.

Color mismatch: Color mismatch between the synthesized donor's face with the original target's face in Celeb-DF is significantly reduced by training data augmentation and post-processing. Specifically, in each training epoch, we randomly perturb the colors of the training faces, which forces the DNNs to synthesize an image containing the same color pattern with the input image. We also apply a color transfer algorithm [46] between the synthesized donor face and the input target face. Figure 4.6 shows an example of the synthesized face without (left) and with (right) color correction.

¹⁰ We choose celebrities' faces as they are more familiar to the viewers so that any visual artifacts can be more readily identified. Furthermore, celebrities are anecdotally the main targets of DeepFake videos.

Fig. 4.4 Example frames from the Celeb-DF dataset. The left column is the frame of real videos and the right five columns are corresponding DeepFake frames generated using different donor subject

Fig. 4.5 Comparison of DeepFake frames with different sizes of the synthesized faces. Note the improved smoothness of the 256×256 synthesized face, which is used in Celeb-DF. This figure is best viewed in color

Fig. 4.6 DeepFake frames using synthesized face without (left) and with (right) color correction. Note the reduced color mismatch between the synthesized face region and the other part of the face. Synthesis method with color correction is used to generate Celeb-DF. This figure is best viewed in color

Fig. 4.7 Mask generation in existing datasets (top two rows) and Celeb-DF (third row). **a** Warped synthesized face overlaying the target's face. **b** Mask generation. **c** Final synthesis result

Inaccurate face masks: In previous datasets, the face masks are either rectangular, which may not completely cover the facial parts in the original video frame, or the convex hull of landmarks on eyebrows and lower lip, which leaves the boundaries of the mask visible. We improve the mask generation step for Celeb-DF. We first synthesize a face with more surrounding context, so as to completely cover the original facial parts after warping. We then create a smoothness mask based on the landmarks on eyebrows and interpolated points on cheeks and between lower lip and chin. The difference in mask generation used in existing datasets and Celeb-DF is highlighted in Fig. 4.7 with an example.

Temporal flickering: We reduce temporal flickering of synthetic faces in the DeepFake videos by incorporating temporal correlations among the detected face landmarks. Specifically, the temporal sequence of the face landmarks are filtered using a Kalman smoothing algorithm to reduce imprecise variations of landmarks in each frame.

4.3.2 Visual Quality

The refinements to the synthesis algorithm improve the visual qualities of the DeepFake videos in the Celeb-DF dataset, as demonstrated in Fig. 4.4. We would like to have a more quantitative evaluation of the improvement in the visual quality of the DeepFake videos in Celeb-DF and compare with the previous DeepFake datasets. Ideally, a reference-free face image quality metric is the best choice for this purpose. However, unfortunately, to date there is no such metric that is agreed upon and widely adopted.

Instead, we follow the face in-painting work [51] and use the Mask-SSIM score [36] as a referenced quantitative metric of the visual quality of synthesized DeepFake video frames. Mask-SSIM corresponds to the SSIM score [57] between the head regions (including face and hair) of the DeepFake video frame and the corresponding original video frame, i.e., the head region of the original target is the reference for visual quality evaluation. As such, low Mask-SSIM score may be due to inferior visual quality as well as changes of the identity from the target to the donor. On the other hand, since we only compare frames from DeepFake videos, the errors caused by identity changes are biased in a similar fashion to all compared datasets. Therefore, the numerical values of Mask-SSIM may not be meaningful to evaluate the absolute visual quality of the synthesized faces, but the difference between Mask-SSIM reflects the difference in visual quality.

The Mask-SSIM score takes value in the range of $[0, 1]$ with higher value corresponding to better image quality. Table 4.2 shows the average Mask-SSIM scores for all compared datasets, with Celeb-DF having the highest scores. This confirms the visual observation that Celeb-DF has improved visual quality, as shown in Fig. 4.4.

Table 4.2 Average Mask-SSIM scores of different DeepFake datasets. Computing Mask-SSIM requires exact corresponding pairs of DeepFake-synthesized frames and original video frames, which is not the case for DFD and DFDC. For these two datasets, we calculate the Mask-SSIM on videos that we have exact correspondences for, i.e., 311 videos in DFD and 2,025 videos in DFDC

Datasets	UADFV	DF-TIMIT		FF-DF	DFD	DFDC	Celeb-DF
		LQ	HQ				
Mask-SSIM	0.82	0.80	0.80	0.81	0.88	0.84	0.92

4.3.3 Evaluations

In Table 4.3, we list individual frame-level AUC scores of all compared DeepFake detection methods over all datasets including Celeb-DF, and Fig. 4.10 shows the frame-level ROC curves of several top detection methods on several datasets.

Comparing different datasets, in Fig. 4.8, we show the average frame-level AUC scores of all compared detection methods on each dataset. Celeb-DF is in general the most challenging to the current detection methods, and their overall performance on Celeb-DF is lowest across all datasets. These results are consistent with the differences in visual quality. Note that many current detection methods predicate on visual artifacts such as low resolution and color mismatch, which are improved in the synthesis algorithm for the Celeb-DF dataset. Furthermore, the difficulty level for detection is clearly higher for the second generation datasets (DFD, DFDC, and Celeb-DF, with average AUC scores lower than 70%), while some detection methods achieve near-perfect detection on the first generation datasets (UADFV, DF-TIMIT, and FF-DF, with average AUC scores around 80%).

In terms of individual detection methods, Fig. 4.9 shows the comparison of average AUC score of each detection method on all DeepFake datasets. These results show that detection has also made progress with the most recent DSP-FWA method achieves the overall top performance (87.4%).

As online videos are usually recompressed to different formats (MPEG4.0 and H264) and in different qualities during the process of uploading and redistribution,

Table 4.3 Frame-level AUC scores (%) of various methods on compared datasets. Bold faces correspond to the top performance

Methods↓ Datasets→	UADFV [61]	DF-TIMIT [26]		FF-DF [47]	DFD [14]	DFDC [12]	Celeb- DF
		LQ	HQ				
Two-stream [63]	85.1	83.5	73.5	70.1	52.8	61.4	53.8
Meso4 [1]	84.3	87.8	68.4	84.7	76.0	75.3	54.8
MesoInception4	82.1	80.4	62.7	83.0	75.9	73.2	53.6
HeadPose [61]	89.0	55.1	53.2	47.3	56.1	55.9	54.6
FWA [29]	97.4	99.9	93.2	80.1	74.3	72.7	56.9
VA-MLP [37]	70.2	61.4	62.1	66.4	69.1	61.9	55.0
VA-LogReg	54.0	77.0	77.3	78.0	77.2	66.2	55.1
Xception-raw [47]	80.4	56.7	54.0	99.7	53.9	49.9	48.2
Xception-c23	91.2	95.9	94.4	99.7	85.9	72.2	65.3
Xception-c40	83.6	75.8	70.5	95.5	65.8	69.7	65.5
Multi-task [40]	65.8	62.2	55.3	76.3	54.1	53.6	54.3
Capsule [42]	61.3	78.4	74.4	96.6	64.0	53.3	57.5
DSP-FWA	97.7	99.9	99.7	93.0	81.1	75.5	64.6

Fig. 4.8 Average AUC performance of all detection methods on each dataset

Fig. 4.9 Average AUC performance of each detection method on all evaluated datasets

Table 4.4 AUC performance of four top detection methods on original, medium (23), and high (40) degrees of H.264 compressed Celeb-DF, respectively

	Original	c23	c40
FWA	56.9	54.6	52.2
Xception-c23	65.3	65.5	52.5
Xception-c40	65.5	65.4	59.4
DSP-FWA	64.6	57.7	47.2

it is also important to evaluate the robustness of detection performance with regards to video compression. Table 4.4 shows the average frame-level AUC scores of four state-of-the-art DeepFake detection methods on original MPEG4.0 videos, medium (23), and high (40) degrees of H.264 compressed videos of Celeb-DF, respectively. The results show that the performance of each method is reduced along with the compression degree increased. In particular, the performance of FWA and DSP-FWA degrades significantly on recompressed video, while the performance of Xception-c23 and Xception-c40 is not significantly affected. This is expected because the latter methods were trained on compressed H.264 videos such that they are more robust in this setting (Fig. 4.10).

Fig. 4.10 ROC curves of six state-of-the-art detection methods (FWA, Meso4, MesoInception4, Xception-c23, Xception-c40, and DSP-FWA) on four largest datasets (FF-DF, DFD, DFDC, and Celeb-DF)

4.4 Landmark Breaker: the Obstruction of DeepFakes

4.4.1 Facial Landmark Extractors

The facial landmark extractors detect and locate key points of important facial parts such as the tips of the nose, eyes, eyebrows, mouth, and jaw outline. Earlier facial landmark extractors are based on simple machine learning methods such as the ensemble of regression trees (ERT) [22] as in the Dlib package [24]. The more recent ones are based on CNN models, which have achieved significantly improved performance over the traditional methods, e.g., [7, 19, 45, 50, 58, 65]. The current CNN-based facial landmark extractors typically contain two stages of operations. In the first stage, a set of heat-maps (feature maps) are obtained to represent the spatial probability of each landmark. In the second stage, the final locations of facial landmarks are extracted based on the peaks of the heat-maps. In this work, we mainly focus on attacking the CNN-based facial landmark extractors because of their better performance.

4.4.2 Adversarial Perturbations

CNNs have been proven vulnerable against adversarial perturbations, which are intentionally crafted imperceptible noises aiming to mislead the CNN-based image classifiers [4, 17, 28, 34, 38, 39, 43, 54, 60, 62], object detectors [9, 59], and semantic segmentation [3, 16]. There are two attack settings: white-box attack, where the attackers can access the details of CNNs, and black-box attack, where the attackers

do not know the details of CNNs. However, to date, there is no existing work to attack CNN-based facial landmark extractors using adversarial perturbations. Compared to the attack to image CNN-based classifiers, which aims to change the prediction of a single label, disturbing facial landmark extractors are more challenging as we need to simultaneously perturb the spatial probabilities of multiple facial landmarks to make the attack effective.

4.4.3 Notation and Formulation

Let $\mathbf{F}$ denote the mapping function of a CNN-based landmark extractor of which the parameters we have access to, and $\{h_1, \dots, h_k\} = \mathbf{F}(\mathbf{I})$ be the set of heat-maps of running $\mathbf{F}$ on input image $\mathbf{I}$. Our goal is to find an image $\mathbf{I}^{adv}$, which can lead the prediction of landmark locations to a large error, while visually similar to as original image $\mathbf{I}$. The difference $\mathbf{I}^{adv} - \mathbf{I}$ is the adversarial perturbation. We denote the heat-maps from the perturbed image as $\{\hat{h}_1, \dots, \hat{h}_k\} = \mathbf{F}(\mathbf{I}^{adv})$.

To this end, we introduce a loss function that aims to enlarge the error between predicted heat-maps and original heat-maps while constraining the pixel distortion in a certain budget as

$$\begin{aligned} \operatorname{argmin}_{\mathbf{I}^{adv}} L(\mathbf{I}^{adv}, \mathbf{I}) &= \sum_{i=1}^k \frac{h_i^\top \hat{h}_i}{\|h_i\| \|\hat{h}_i\|}, \\ \text{s.t. } \|\mathbf{I}^{adv} - \mathbf{I}\|_\infty &\leq \epsilon, \end{aligned} \quad (4.1)$$

where ϵ is a constant. We use cosine distance to measure the error as it can naturally normalize the loss range in $[-1, 1]$. Minimizing this loss function increases the error between predicted and original heat-maps, which will disrupt the facial landmark locations.

4.4.4 Optimization

We use the gradient MI-FGSM [13] method to optimize problem Eq.(4.1). Specifically, let t denote the iteration number and $\mathbf{I}_t^{adv}$ denote the adversarial image obtained at iteration t . The start image is initialized as $\mathbf{I}_0^{adv} = \mathbf{I}$. $\mathbf{I}_{t+1}^{adv}$ is obtained by considering the momentum and gradient as

$$\begin{aligned} m_{t+1} &= \lambda \cdot m_t + \frac{\nabla_{\mathbf{I}^{adv}}(L(\mathbf{I}_t^{adv}, \mathbf{I}))}{\|\nabla_{\mathbf{I}^{adv}}(L(\mathbf{I}_t^{adv}, \mathbf{I}))\|_1}, \\ \mathbf{I}_{t+1}^{adv} &= \operatorname{clip}\{\mathbf{I}_t^{adv} - \alpha \cdot \operatorname{sign}(m_{t+1})\}, \end{aligned} \quad (4.2)$$

where $\nabla_{\mathbf{I}^{adv}}(L(\mathbf{I}_t^{adv}, \mathbf{I}))$ is the gradient of L with respect to the input image $\mathbf{I}_t^{adv}$ at iteration t ; m_t is the accumulated gradient and λ is the decay factor of momentum; α is the step size and sign returns the signs of each component of the input vector;

`clip` is the truncation function to ensure the pixel value of the resulting image is in $[0, 255]$. The algorithm stops when the maximum number of iterations T is reached or the distortion threshold ϵ is reached. The overall algorithm is given in Algorithm 1.

Algorithm 1 *Overview of Landmark Breaker*

Require: landmark extractor $\mathbf{F}$; input image $\mathbf{I}$; perturbed image $\mathbf{I}^{adv}$; maximal iteration number T

- 1: $\mathbf{I}_0^{adv} = \mathbf{I}$, $t = 0$, $m_0 = 0$
- 2: **while** $t \leq T$ and $\|\mathbf{I}_{t+1}^{adv} - \mathbf{I}\|_\infty \leq \epsilon$ **do**
- 3: $m_{t+1} = \lambda \cdot m_t + \frac{\nabla_{\mathbf{I}^{adv}}(L(\mathbf{I}_t^{adv}, \mathbf{I}))}{\|\nabla_{\mathbf{I}^{adv}}(L(\mathbf{I}_t^{adv}, \mathbf{I}))\|_1}$,
- 4: $\mathbf{I}_{t+1}^{adv} = \text{clip}\{\mathbf{I}_t^{adv} - \alpha \cdot \text{sign}(m_{t+1})\}$
- 5: $t = t + 1$

Ensure: Adversarial perturbed image $\mathbf{I}_t^{adv}$

4.4.5 Experimental Settings

Landmark Extractors. Landmark Breaker is validated on three state-of-the-art CNN-based facial landmark extractors, namely FAN [7], HRNet [50], and AVS-SAN [45]. FAN¹¹ is constructed by multiple stacked hourglass structures, where we use one hourglass structure for simplicity. HRNet¹² is composed by parallel high-to-low resolution sub-networks and repeats the information exchange across multi-resolution sub-networks. AVS-SAN¹³ first disentangles face images to style and structure space, which is then used as augmentation to train the network. We use implementations of all three methods trained on WLFW dataset [58].

Datasets. To demonstrate the effectiveness of Landmark Breaker on obstructing DeepFake generation, we conduct experiments on the Celeb-DF dataset [31], which contains high-quality DeepFake videos of 59 celebrities. Each video contains one subject with various head poses and facial expressions. We choose this dataset as the pretrained DeepFake models are available to us, which can be used to test our method.

In our experiment, we utilize the DeepFake method described in [31] to synthesize fake videos using original and adversarial images, respectively. We randomly select 6 identities, corresponding to 36 videos in total. Since the adjacent frames in a video show little variations, we apply Landmark Breaker to the key frames of each video, i.e., 600 frames in total, for evaluation. Since the Celeb-DF dataset does not have the ground truth of facial landmarks, we use the results of HRNet as the ground truth due to its superior performance.

¹¹ <https://github.com/hzh8311/pyhowfar>.

¹² <https://github.com/HRNet/HRNet-Facial-Landmark-Detection>.

¹³ <https://github.com/TheSouthFrog/stylealign>.

Fig. 4.11 Evaluation pipeline. $SSIM_I$ denotes the image quality of the adversarial image referred to as the original image, while $SSIM_W$ denotes the image quality of the corresponding synthesized image. NME denotes the distance of facial landmarks on adversarial image and ground truth

Evaluations. We use two metrics to evaluate Landmark Breaker, namely Normalized Mean Error (NME) [50] and Structural Similarity (SSIM) [57]. The relation of these metrics are shown in Fig. 4.11.

- NME is the average Euclidean distance between landmarks on adversarial image and the ground truth, which is then normalized by the distance between the leftmost key point in the left eye and the rightmost key point in the right eye. Higher NME score indicates less accurate landmark detection, which is the objective of Landmark Breaker.
- The SSIM metric simulates perceptual image quality. We use this indicator to demonstrate that Landmark Breaker can affect the visual quality of DeepFake. As shown in Fig. 4.11, we compute SSIM of original and adversarial input images ($SSIM_I$)¹⁴ and then compute the SSIM of the synthesized results ($SSIM_W$). The lower score indicates the image quality is degraded. Ideally, the attacking method should have large $SSIM_I$ such that the adversarial perturbation does not affect the quality of input image, and small $SSIM_W$ such that the synthesis quality is degraded.

Baselines. To better analyze Landmark Breaker, we adapt other two methods FGSM [54] and I-FGSM [17] from attacking image classifiers to our task. Specifically, the FGSM is a single-step optimization method as $\mathbf{I}_1^{adv} = \text{clip}(\mathbf{I}_0^{adv} - \alpha \cdot \text{sign}(\nabla_{\mathbf{I}_0^{adv}}(L(\mathbf{I}_0^{adv}, \mathbf{I})))$, while I-FGSM is an iterative optimization method without considering momentum as $\mathbf{I}_{t+1}^{adv} = \text{clip}(\mathbf{I}_t^{adv} - \alpha \cdot \text{sign}(\nabla_{\mathbf{I}_t^{adv}}(L(\mathbf{I}_t^{adv}, \mathbf{I})))$. The step size α and iteration number T of I-FGSM are set as the same in Landmark

¹⁴ We employ mask-SSIM [36] to measure the quality inside a region of interest determined by face detection.

Breaker. We use these two adapted methods as our baseline methods, which are denoted as *Base1* and *Base2*, respectively.

Implementation Details. Following the previous works [35, 60], we set the maximum perturbation budget $\epsilon = 15$. The other parameters in Landmark Breaker are set as follows: The maximum iteration number $T = 20$; the step size $\alpha = 1$; the decay factor is set as $\lambda = 0.5$.

4.4.6 Results

Table 4.5 shows the NME and SSIM performance of Landmark Breaker. The landmark extractors shown in the leftmost column denote where the adversarial perturbation is from and the ones shown in the top row denotes which landmark extractor is attacked. “None” denotes no perturbations are added to the image. Landmark Breaker can notably increase the NME score and decrease the $SSIM_w$ score in white-box attack (e.g., the value in the row of “FAN” and the column of “FAN”), which indicates Landmark Breaker can effectively disrupt facial landmark extraction and subsequently affect the visual quality of the synthesized faces. We also compare Landmark Breaker with two baselines Base1 and Base2 in Table 4.6. We can observe the Base1 method merely has any effect on the NME performance but can largely degrade the quality of adversarial images compared to Base2 and Landmark Breaker (LB). The Base2 method can also achieve the competitive performance with Landmark Breaker in NME but is slightly degraded in SSIM.

Following existing works attacking image classifiers, [13, 54], which achieves the black-box attack by transferring the adversarial perturbations from a known model to an unknown model (transferability), we also test the black-box attack using the adversarial perturbation generated from one landmark extractor to attack other extractors. However, the results show that the adversarial perturbations have merely any effect on different extractors.

As shown in Table 4.5, the transferability of Landmark Breaker is weak. To improve the transferability, we employ the strategies commonly used in black-box attacks on image classifiers: (1) Input transformation [60]: we randomly resize the input image and then pad around with zero at each iteration (denoted as LB_{trans}); (2) Attacking mixture [60]: we alternatively use Base2 and Landmark Breaker to increase the diversity in optimization (denoted as LB_{mix}). Table 4.7 shows the results of a black-box attack, which reveals that the strategies effective in attacking image classifiers do not work on attacking landmark extractors. This is probably due to the mechanism of landmark extractors being more complex than image classifiers, as the landmark extractors need to output a series of points instead of labels, and only a minority of points shifted do not affect the overall prediction.

Table 4.5 The NME and SSIM scores of Landmark Breaker on different landmark extractors. The landmark extractors shown in leftmost column denote where the adversarial perturbation is from and the ones shown in the top row denote which landmark extractors are attacked

NME↑			
Attacks	FAN	HRNet	AVS-SAN
None	0.03	0.00	0.09
FAN	0.87	0.05	0.09
HRNet	0.04	0.87	0.09
AVS-SAN	0.06	0.04	0.92

SSIM				
Attacks	SSIM _I ↑	SSIM _W ↓		
		FAN	HRNet	AVS-SAN
FAN	0.81	0.68	0.89	0.89
HRNet	0.78	0.89	0.67	0.88
AVS-SAN	0.78	0.87	0.87	0.69

Table 4.6 The NME and SSIM performance of different attacking methods

NME↑			
Attacks	FAN	HRNet	AVS-SAN
None	0.03	0.00	0.09
Base1	0.05	0.04	0.10
Base2	0.85	0.88	0.92
LB	0.87	0.87	0.92

SSIM _I ↑ / SSIM _W ↓			
Attacks	FAN	HRNet	AVS-SAN
Base1	0.52/0.73	0.46/0.71	0.49/0.69
Base2	0.88/0.71	0.88/0.70	0.86/0.73
LB	0.81/0.68	0.78/0.67	0.78/0.69

4.4.7 Robustness Analysis

We study the robustness of Landmark Breaker toward three extractors under image and video compression. Note that image compression considers the spatial correlation, while video compression also considers the temporal correlation.

Image compression. We compress the adversarial images to quality 75% (Q75) and 50% (Q50) using OpenCV and then observe the variations in the performance of each method. Table 4.8 shows the NME and SSIM performance of each method under

Table 4.7 The NME and SSIM performance of black-box attack. See text for details

		NME↑		
Attacks		FAN	HRNet	AVS-SAN
None		0.03	0.00	0.09
FAN	LB _{trans}	0.22	0.03	0.09
	LB _{mix}	0.24	0.04	0.09
HRNet	LB _{trans}	0.04	0.10	0.09
	LB _{mix}	0.04	0.14	0.09
AVS-SAN	LB _{trans}	0.04	0.03	0.55
	LB _{mix}	0.05	0.03	0.56

		SSIM			
Attacks		SSIM _I ↑	SSIM _W ↓		
			FAN	HRNet	AVS-SAN
FAN	LB _{trans}	0.91	0.88	0.94	0.94
	LB _{mix}	0.90	0.86	0.94	0.93
HRNet	LB _{trans}	0.92	0.95	0.94	0.95
	LB _{mix}	0.90	0.95	0.91	0.94
AVS-SAN	LB _{trans}	0.89	0.94	0.93	0.82
	LB _{mix}	0.88	0.93	0.93	0.81

different compression levels. Compared to the two baseline methods, Landmark Breaker is more robust against image compression. Another observation is that the attacks on AVS-SAN exhibit high robustness, where the performance of NME and SSIM is only slightly degraded. In contrast, the attacking performance on HRNet drops quickly with compression. Figure 4.12 (left) plots the trend of each method.

Video compression. As the videos are widespread on the Internet, we also investigate the robustness against video compression. We create a video using the adversarial images using the codec in MPEG4 (denoted as C) and then separate the videos into frames to test the performance. We also perform double compression to the MPEG4 videos using the codec in H264 (denoted as C²). Table 4.8 also shows the performance against video compression, which has the same trend as in image compression. Compared to the baseline methods, Landmark Breaker is more robust. Also, the attacks on AVS-SAN exhibit strong robustness even after double compression C², on the other hand, the attacks on HRNet are vulnerable against video compression; see Fig. 4.12 (right). Note the curve of Base1 and LB are fully overlapped in the last plot.

Table 4.8 The NME and SSIM performance of different attacking methods under different image compression (IC) and video compression (VC) levels

		NME↑			
		Attacks	FAN	HRNet	AVS-SAN
None		Base1	0.05	0.04	0.10
		Base2	0.85	0.88	0.92
		LB	0.87	0.87	0.92
IC	Q75	Base1	0.05	0.04	0.10
		Base2	0.64	0.10	0.90
		LB	0.77	0.24	0.91
	Q50	Base1	0.05	0.03	0.10
		Base2	0.50	0.05	0.88
		LB	0.70	0.10	0.89
VC	C	Base1	0.05	0.03	0.10
		Base2	0.44	0.10	0.88
		LB	0.72	0.41	0.90
	C ²	Base1	0.05	0.03	0.10
		Base2	0.26	0.06	0.83
		LB	0.60	0.20	0.89

		SSIM _I ↑ / SSIM _W ↓			
		Attacks	FAN	HRNet	AVS-SAN
None		Base1	0.52/0.73	0.46/0.71	0.49/0.69
		Base2	0.88/0.71	0.88/0.70	0.86/0.73
		LB	0.81/0.68	0.78/0.67	0.78/0.69
IC	Q75	Base1	0.56/0.75	0.47/0.72	0.51/0.71
		Base2	0.90/0.74	0.93/0.94	0.87/0.74
		LB	0.84/0.70	0.85/0.85	0.80/0.70
	Q50	Base1	0.59/0.76	0.55/0.75	0.52/0.71
		Base2	0.89/0.76	0.93/0.95	0.87/0.74
		LB	0.84/0.71	0.88/0.92	0.80/0.70
VC	C	Base1	0.57/0.76	0.52/0.74	0.52/0.71
		Base2	0.90/0.78	0.92/0.94	0.87/0.74
		LB	0.84/0.71	0.84/0.79	0.80/0.71
	C ²	Base1	0.58/0.76	0.52/0.74	0.53/0.72
		Base2	0.91/0.85	0.93/0.95	0.88/0.76
		LB	0.85/0.73	0.84/0.87	0.82/0.72

Fig. 4.12 The performance of each method on different landmark extractors under image and video compression

4.4.8 Ablation Study

This section presents ablation studies on the impact of different parameters on Landmark Breaker.

Step size. We study the impact of step size α on the performance of NME and SSIM scores. We set the step size α from 0.5 to 1.5. The results are plotted in Fig. 4.13. We observe that the NME score increases first and then decreases, which is because the small step size does not disturb the image enough within the maximum iteration number and then the large step size may not precisely follow the gradient descent direction. Moreover, a larger step size can degrade the input image quality, which also leads to the degradation of the synthesized image.

Maximum iteration number. We then study the impact of the maximum iteration number T on the performance of NME and SSIM. We vary the maximum iteration number T from 14 to 28 and illustrate the results in Fig. 4.13. From the figure, we observe that the NME score is increased and SSIM is decreased with iteration number

Fig. 4.13 Ablation study of Landmark Breaker regarding the performance with different step sizes and iteration numbers

increasing. Since the distortion budget constraint, the curve becomes flat after about 17 iterations. Note that several curves are fully overlapped in the plot.

4.5 Conclusion

This chapter describes our recent efforts toward the creation and obstruction of DeepFakes. Section 4.1 describes a new challenging large-scale dataset for the development and evaluation of DeepFake detection methods. The Celeb-DF dataset reduces the gap in the visual quality of DeepFake datasets and the actual DeepFake videos circulated online. Based on the Celeb-DF dataset, we perform a comprehensive performance evaluation of current DeepFake detection methods, and show that there is still much room for improvement. Section 4.2 describes a new method, namely Landmark Breaker, to obstruct the DeepFake generation by breaking the prerequisite step—facial landmark extraction. To do so, we create the adversarial perturbations to disrupt the facial landmark extraction, such that the input faces to the DeepFake

model cannot be well aligned. Landmark Breaker is validated on Celeb-DF dataset, which demonstrates the efficacy of Landmark Breaker on disturbing facial landmark extraction. We also study the performance of Landmark Breaker under various parameter settings.

References

1. Afchar D, Nozick V, Yamagishi J, Echizen I (2018) Mesonet: a compact facial video forgery detection network. In: WIFS
2. Agarwal S, Farid H, Gu Y, He M, Nagano K, Li H (2019) Protecting world leaders against deep fakes. In: IEEE conference on computer vision and pattern recognition workshops (CVPRW)
3. Arnab A, Miksik O, Torr PH (2018) On the robustness of semantic segmentation models to adversarial attacks. In: CVPR
4. Baluja S, Fischer I (2018) Learning to attack: Adversarial transformation networks. In: AAAI
5. Bappy JH, Simons C, Nataraj L, Manjunath B, Roy-Chowdhury AK (2019) Hybrid lstm and encoder-decoder architecture for detection of image forgeries. *IEEE Trans Image Process (TIP)*
6. Bitouk D, Kumar N, Dhillon S, Belhumeur P, Nayar SK (2008) Face swapping: automatically replacing faces in photographs. *ACM Trans Graph (TOG)*
7. Bulat A, Tzimiropoulos G (2017) How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In: ICCV
8. Chan C, Ginosar S, Zhou T, Efros AA (2019) Everybody dance now. In: ICCV
9. Chen ST, Cornelius C, Martin J, Chau DH (2018) Robust physical adversarial attack on faster r-cnn object detector. [arXiv:1804.05810](https://arxiv.org/abs/1804.05810)
10. Chesney R, Citron DK (2018) Deep fakes: a looming challenge for privacy, democracy, and national security. 107 *California Law Review* (2019, Forthcoming); U of Texas Law, Public Law Research Paper No 692; U of Maryland Legal Studies Research Paper No 2018-21
11. Dale K, Sunkavalli K, Johnson MK, Vlasic D, Matusik W, Pfister H (2011) Video face replacement. *ACM Trans Graph (TOG)*
12. Dolhansky B, Howes R, Pflaum B, Baram N, Ferrer CC (2019) The deepfake detection challenge (DFDC) preview dataset. [arXiv:1910.08854](https://arxiv.org/abs/1910.08854)
13. Dong Y, Liao F, Pang T, Su H, Zhu J, Hu X, Li J (2018) Boosting adversarial attacks with momentum. In: CVPR
14. Dufour N, Gully A, Karlsson P, Vorbyov AV, Leung T, Childs J, Bregler C (2019-09) Deepfakes detection dataset by google & jigsaw
15. Farid H (2012) *Digital image forensics*. MIT Press
16. Fischer V, Kumar MC, Metzen JH, Brox T (2017) Adversarial examples for semantic image segmentation. [arXiv:1703.01101](https://arxiv.org/abs/1703.01101)
17. Goodfellow IJ, Shlens J, Szegedy C (2015) Explaining and harnessing adversarial examples. In: ICLR
18. Güera D, Delp EJ (2018) Deepfake video detection using recurrent neural networks. In: AVSS
19. Hu T, Qi H, Xu J, Huang Q (2018) Facial landmarks detection by self-iterative regression based landmarks-attention network. In: AAAI
20. Karras T, Aila T, Laine S, Lehtinen J (2018) Progressive growing of GANs for improved quality, stability, and variation. In: ICLR
21. Karras T, Laine S, Aila T (2019) A style-based generator architecture for generative adversarial networks. In: CVPR
22. Kazemi V, Sullivan J (2014) One millisecond face alignment with an ensemble of regression trees. In: CVPR
23. Kim H, Garrido P, Tewari A, Xu W, Thies J, Nießner N, Pérez P, Richardt C, Zollhöfer M, Theobalt C (2018) Deep video portraits. *ACM Trans Graph* 2018 (TOG)

24. King DE (2009) Dlib-ml: A machine learning toolkit. JMLR
25. Kingma DP, Welling M (2014) Auto-encoding variational bayes. In: ICLR
26. Korshunov P, Marcel S (2018) Deepfakes: a new threat to face recognition? assessment and detection. [arXiv:181208685](#)
27. Korshunova I, Shi W, Dambre J, Theis L (2017) Fast face-swap using convolutional neural networks. In: ICCV
28. Kurakin A, Goodfellow I, Bengio S (2017) Adversarial machine learning at scale. In: ICLR
29. Li Y, Lyu S (2019) Exposing deepfake videos by detecting face warping artifacts. In: CVPR Workshops
30. Li Y, Chang MC, Lyu S (2018) In ictu oculi: exposing AI generated fake face videos by detecting eye blinking. In: WIFS
31. Li Y, Yang X, Sun P, Qi H, Lyu S (2020) Celeb-df: a large-scale challenging dataset for deepfake forensics. In: CVPR
32. Liu MY, Breuel T, Kautz J (2017) Unsupervised image-to-image translation networks. In: NeurIPS
33. Liu Y, Guan Q, Zhao X, Cao Y (2018) Image forgery localization based on multi-scale convolutional neural networks. In: ACM workshop on information hiding and multimedia security (IHMMSec)
34. Luo B, Liu Y, Wei L, Xu Q (2018) Towards imperceptible and robust adversarial example attacks against neural networks. In: AAAI
35. Luo Y, Boix X, Roig G, Poggio T, Zhao Q (2015) Foveation-based mechanisms alleviate adversarial examples. [arXiv:151106292](#)
36. Ma L, Jia X, Sun Q, Schiele B, Tuytelaars T, Van Gool L (2017) Pose guided person image generation. In: NeurIPS
37. Matern F, Riess C, Stamminger M (2019) Exploiting visual artifacts to expose deepfakes and face manipulations. In: WACV Workshops
38. Moosavi-Dezfooli SM, Fawzi A, Frossard P (2016) Deepfool: a simple and accurate method to fool deep neural networks. In: CVPR
39. Moosavi-Dezfooli SM, Fawzi A, Fawzi O, Frossard P (2017) Universal adversarial perturbations. In: CVPR
40. Nguyen HH, Fang F, Yamagishi J, Echizen I (2019) Multi-task learning for detecting and segmenting manipulated facial images and videos. In: IEEE international conference on biometrics: theory, applications and systems (BTAS)
41. Nguyen HH, Yamagishi J, Echizen I (2019) Capsule-forensics: using capsule networks to detect forged images and videos. In: IEEE international conference on acoustics, speech and signal processing (ICASSP)
42. Nguyen HH, Yamagishi J, Echizen I (2019) Use of a capsule network to detect fake images and videos. [arXiv:191012467](#)
43. Papernot N, McDaniel P, Jha S, Fredrikson M, Celik ZB, Swami A (2016) The limitations of deep learning in adversarial settings. In: EuroS&P
44. Pham HX, Wang Y, Pavlovic V (2018) Generative adversarial talking head: bringing portraits to life with a weakly supervised neural network. [arXiv:180307716](#)
45. Qian S, Sun K, Wu W, Qian C, Jia J (2019) Aggregation via separation: boosting facial landmark detector with semi-supervised style translation. In: ICCV
46. Reinhard E, Adhikmin M, Gooch B, Shirley P (2001) Color transfer between images. IEEE Comput Graph Appl
47. Rössler A, Cozzolino D, Verdoliva L, Riess C, Thies J, Nießner M (2019) FaceForensics++: learning to detect manipulated facial images. In: ICCV
48. Sabir E, Cheng J, Jaiswal A, AbdAlmageed W, Masi I, Natarajan P (2019) Recurrent-convolution approach to deepfake detection-state-of-art results on faceforensics++. [arXiv:190500582](#)
49. Sanderson C, Lovell BC (2009) Multi-region probabilistic histograms for robust and scalable identity inference. In: International conference on biometrics

50. Sun K, Xiao B, Liu D, Wang J (2019) Deep high-resolution representation learning for human pose estimation. In: CVPR
51. Sun Q, Ma L, Joon Oh S, Van Gool L, Schiele B, Fritz M (2018) Natural and effective obfuscation by head inpainting. In: CVPR
52. Suwajanakorn S, Seitz SM, Kemelmacher-Shlizerman I (2015) What makes tom hanks look like tom hanks. In: ICCV
53. Suwajanakorn S, Seitz SM, Kemelmacher-Shlizerman I (2017) Synthesizing obama: learning lip sync from audio. *ACM Trans Graph (TOG)*
54. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R (2014) Intriguing properties of neural networks. In: ICLR
55. Thies J, Zollhofer M, Stamminger M, Theobalt C, Niessner M (2016) Face2face: real-time face capture and reenactment of rgb videos. In: CVPR
56. Thies J, Zollhöfer M, Nießner M (2019) Deferred neural rendering: image synthesis using neural textures. In: SIGGRAPH
57. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP, et al. (2004) Image quality assessment: from error visibility to structural similarity. *TIP*
58. Wu W, Qian C, Yang S, Wang Q, Cai Y, Zhou Q (2018) Look at boundary: a boundary-aware face alignment algorithm. In: CVPR
59. Xie C, Wang J, Zhang Z, Zhou Y, Xie L, Yuille A (2017) Adversarial examples for semantic segmentation and object detection. In: ICCV
60. Xie C, Zhang Z, Zhou Y, Bai S, Wang J, Ren Z, Yuille AL (2019) Improving transferability of adversarial examples with input diversity. In: CVPR
61. Yang X, Li Y, Lyu S (2019) Exposing deep fakes using inconsistent head poses. In: ICASSP
62. Zeng X, Liu C, Wang YS, Qiu W, Xie L, Tai YW, Tang CK, Yuille AL (2019) Adversarial attacks beyond the image space. In: CVPR
63. Zhou P, Han X, Morariu VI, Davis LS (2017) Two-stream neural networks for tampered face detection. In: IEEE conference on computer vision and pattern recognition workshops (CVPRW)
64. Zhou P, Han X, Morariu VI, Davis LS (2018) Learning rich features for image manipulation detection. In: CVPR
65. Zou X, Zhong S, Yan L, Zhao X, Zhou J, Wu Y (2019) Learning robust facial landmark detection via hierarchical structured ensemble. In: ICCV

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.