The Creation and Detection of Deepfakes: A Survey

YISROEL MIRSKY, Georgia Institute of Technology and Ben-Gurion University
WENKE LEE, Georgia Institute of Technology

Generative deep learning algorithms have progressed to a point where it is difficult to tell the difference between what is real and what is fake. In 2018, it was discovered how easy it is to use this technology for unethical and malicious applications, such as the spread of misinformation, impersonation of political leaders, and the defamation of innocent individuals. Since then, these "deepfakes" have advanced significantly.

In this article, we explore the creation and detection of deepfakes and provide an in-depth view as to how these architectures work. The purpose of this survey is to provide the reader with a deeper understanding of (1) how deepfakes are created and detected, (2) the current trends and advancements in this domain, (3) the shortcomings of the current defense solutions, and (4) the areas that require further research and attention.

CCS Concepts: • **Security and privacy → Social engineering attacks**; *Human and societal aspects of security and privacy*; • **Computing methodologies → Machine learning**;

Additional Key Words and Phrases: Deepfake, deep fake, reenactment, replacement, face swap, generative AI, social engineering, impersonation

ACM Reference format:
Yisroel Mirsky and Wenke Lee. 2020. The Creation and Detection of Deepfakes: A Survey. *ACM Comput. Surv.* 54, 1, Article 7 (December 2020), 41 pages.
https://doi.org/10.1145/3425780

1 INTRODUCTION

A deepfake is content, generated by an artificial intelligence, that is authentic in the eyes of a human being. The word *deepfake* is a combination of the words *"deep learning"* and *"fake,"* and primarily relates to content generated by an artificial neural network, a branch of machine learning.

The most common form of deepfakes involve the generation and manipulation of human imagery. This technology has creative and productive applications. For example, realistic video dubbing of foreign films,[1] education though the reanimation of historical figures [90], and virtually trying on clothes while shopping.[2] There are also numerous online communities devoted to

[1] https://variety.com/2019/biz/news/ai-dubbing-david-beckham-multilingual-1203309213/.
[2] https://www.forbes.com/sites/forbestechcouncil/2019/05/21/gans-and-deepfakes-could-revolutionize-the-fashion-industry/.

Authors' addresses: Y. Mirsky (corresponding author), Georgia Institute of Technology, 756 W. Peachtree St. NW, Atlanta, Georgia, 30308, Ben-Gurion University, P.O.B. 653, Beer-Sheva, Israel, 8410501; emails: yisroel@gatech.edu, yisroel@post.bgu.ac.il; W. Lee, Georgia Institute of Technology, 756 W Peachtree St NW, Atlanta, Georgia, 30308; email: wenke@cc.gatech.edu.

0360-0300/2020/12-ART7 $15.00
https://doi.org/10.1145/3425780

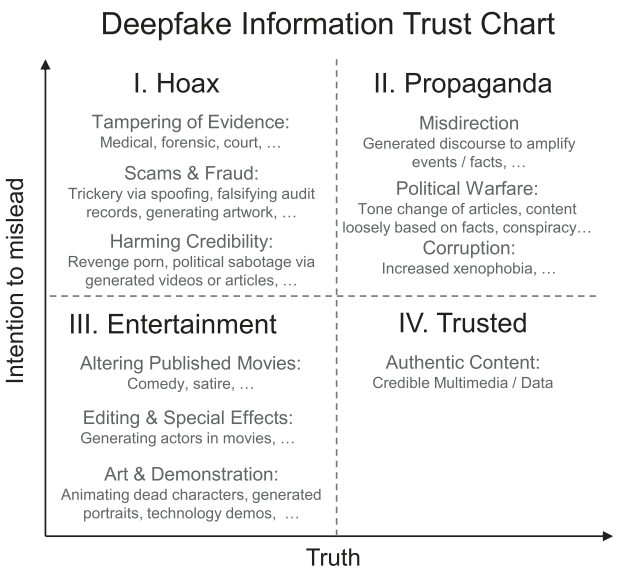
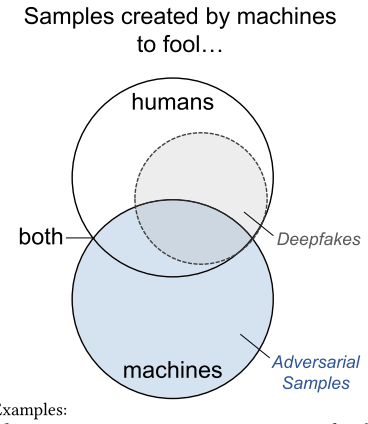


Fig. 1. A deepfake information trust chart.



Examples:
...**humans**: entertainment, impersonation, art fraud.
...**machines**: hiding a stop sign, evading face recog.
...**both**: tampering medical scans, malware evasion.

Fig. 2. The difference between *adversarial machine learning* and *deepfakes*.

creating deepfake memes for entertainment,[3] such as music videos portraying the face of actor Nicolas Cage.

However, despite the positive applications of deepfakes, the technology is infamous for its unethical and malicious aspects. At the end of 2017, a Reddit user by the name of "deepfakes" was using deep learning to swap faces of celebrities into pornographic videos and was posting them online.[4] The discovery caused a media frenzy and a large number of new deepfake videos began to emerge thereafter. In 2018, BuzzFeed released a deepfake video of former president Barak Obama giving a talk on the subject. The video was made using the Reddit user's software (FakeApp), and raised concerns over identity theft, impersonation, and the spread of misinformation on social media. Figure 1 presents an information trust chart for deepfakes, inspired by Ref. [49].

Following these events, the subject of deepfakes gained traction in the academic community, and the technology has been rapidly advancing over the last few years. Since 2017, the number of articles published on the subject rose from 3 to over 250 (2018–20).

To understand where the threats are moving and how to mitigate them, we need a clear view of the technology, challenges, limitations, capabilities, and trajectory. Unfortunately, to the best of our knowledge, there are no other works that present the techniques, advancements, and challenges in a technical and encompassing way. Therefore, the goals of this article are (1) to provide the reader with an understanding of how modern deepfakes are created and detected; (2) to inform the reader of the recent advances, trends, and challenges in deepfake research; (3) to serve as a guide to the design of deepfake architectures; and (4) to identify the current status of the attacker-defender game, the attacker's next move, and future work that may help give the defender a leading edge.

We achieve these goals through an overview of human visual deepfakes (Section 2), followed by a technical background that identifies technology's basic building blocks and challenges (Section 3). We then provide a chronological and systematic review for each category of deepfake,

[3] https://www.reddit.com/r/SFWdeepfakes/.
[4] https://www.vice.com/en_us/article/gydydm/gal-gadot-fake-ai-porn.

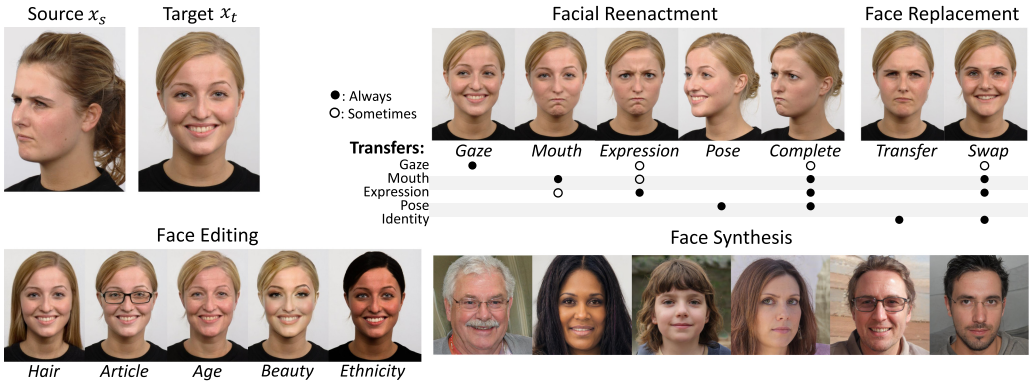


Fig. 3. Examples of reenactment, replacement, editing, and synthesis deepfakes of the human face.

and provide the networks' schematics to give the reader a deeper understanding of the various approaches (Sections 4 and 5). Finally, after reviewing the countermeasures (Section 6), we discuss their weaknesses, note the current limitations of deepfakes, suggest alternative research, consider the adversary's next steps, and raise awareness to the spread of deepfakes to other domains (Section 7).

Scope. In this survey, we will focus on deepfakes pertaining to the human face and body. We will not be discussing the synthesis of new faces or the editing of facial features because they do not have a clear attack goal associated with them. In Section 7.3, we will discuss deepfakes with a much broader scope, note the future trends, and exemplify how deepfakes have spread to other domains and media such as forensics, finance, and healthcare.

We note to the reader that deepfakes should not be confused with adversarial machine learning, which is the subject of fooling machine learning algorithms with maliciously crafted inputs (Figure 2). The difference being that for deepfakes, the objective of the generated content is to fool a human and not a machine.

2 OVERVIEW AND ATTACK MODELS

We define a deepfake as

"Believable media generated by a deep neural network."

In the context of human visuals, we identify four categories: reenactment, replacement, editing, and synthesis. Figure 3 illustrates some examples facial deepfakes in each of these categories and their sub-types. Throughout this article, we denote s and t as the source and the target identities. We also denote x_s and x_t as images of these identities and x_g as the deepfake generated from s and t.

2.1 Reenactment

A reenactment deepfake is where x_s is used to drive the expression, mouth, gaze, pose, or body of x_t:

Expression reenactment is where x_s drives the expression of x_t. It is the most common form of reenactment since these technologies often drive target's mouth and pose as well, providing a wide range of flexibility. Benign uses are found in the movie and video game industry where the performances of actors are tweaked in post, and in educational media where historical figures are reenacted.

Mouth reenactment, also known as "dubbing," is where the mouth of x_t is driven by that of x_s, or an audio input a_s containing speech. Benign uses of the technology include realistic voice dubbing into another language and editing.

Gaze reenactment is where direction of x_t's eyes, and the position of the eyelids, are driven by those of x_s. This is used to improve photographs or to automatically maintain eye contact during video interviews [45].

Pose reenactment is where the head position of x_t is driven by x_s. This technology has primarily been used for face frontalization of individuals in security footage and as a means for improving facial recognition software [158].

Body reenactment, a.k.a. pose transfer and human pose synthesis, is similar to the facial reenactments listed above except that it's the pose of x_t's body being driven.

The Attack Model. Reenactment deep fakes give attackers the ability to impersonate an identity, controlling what he or she says or does. This enables an attacker to perform acts of defamation, cause discredability, spread misinformation, and tamper with evidence. For example, an attacker can impersonate t to gain the trust of a colleague, friend, or family member as a means to gain access to money, network infrastructure, or some other asset. An attacker can also generate embarrassing content of t for blackmailing purposes or generate content to affect the public's opinion of an individual or political leader. The technology can also be used to tamper surveillance footage or some other archival imagery in an attempt to plant false evidence in a trial. Finally, the attack can either take place online (e.g., impersonating someone in a *real-time* conversation) or offline (e.g., fake media spread on the Internet).

2.2 Replacement

A replacement deepfake is where the content of x_t is replaced with that of x_s, preserving the identity of s.

Transfer is where the content of x_t is replaced with that of x_s. A common type of transfer is facial transfer, used in the fashion industry to visualize an individual in different outfits.

Swap is where the content transferred to x_t from x_s is driven by x_t. The most popular type of swap replacement is "face swap," often used to generate memes or satirical content by swapping the identity of an actor with that of a famous individual. Another benign use for face swapping includes the anonymization of one's identity in public content in-place of blurring or pixelation.

The Attack Model. Replacement deepfakes are well-known for their harmful applications. For example, revenge porn is where an attacker swaps a victim's face onto the body of a porn actress to humiliate, defame, and blackmail the victim. Face replacement can also be used as a short-cut to fully reenacting t by transferring t's face onto the body of a look-alike. This approach has been used as a tool for disseminating political opinions in the past [136].

2.3 Editing and Synthesis

An enchantment deepfake is where the attributes of x_t are added, altered, or removed. Some examples include changing a target's clothes, facial hair, age, weight, beauty, and ethnicity. Apps such as FaceApp enable users to alter their appearance for entertainment and easy editing of multimedia. The same process can be used by an attacker to build a false persona for misleading others. For example, a sick leader can be made to look healthy [67], and child or sex predators can change their age and gender to build dynamic profiles online. A known unethical use of editing deepfakes is the removal of a victim's clothes for humiliation or entertainment [133].

Synthesis is where the deepfake x_g is created with no target as a basis. Human face and body synthesis techniques such as those in Ref. [78] (used in Figure 3) can create royalty-free stock footage or generate characters for movies and games. However, similar to editing deepfakes, it can also be used to create fake personas online.

Although human image editing and synthesis are active research topics, reenactment and replacement deepfakes are the greatest concern because they give an attacker control over one's identity[12, 28, 66]. Therefore, in this survey, we will be focusing on reenactment and replacement deepfakes.

3 TECHNICAL BACKGROUND

Although there are a wide variety of neural networks, most deepfakes are created using variations or combinations of generative networks and encoder decoder networks. In this section, we provide a brief introduction to these networks, how they are trained, and the notations that we will be using throughout the article.

3.1 Neural Networks

Neural networks are non-linear models for predicting or generating content based on an input. They are made up of layers of neurons, where each layer is connected sequentially via synapses. The synapses have associated weights that collectively define the concepts learned by the model. To execute a network on an n-dimensional input x, a process known as *forward-propagation* is performed where x propagated through each layer and an activation function is used to summarize a neuron's output (e.g., the Sigmoid or Rectified Linear Unit (ReLU) function).

Concretely, let $l^{(i)}$ denote the i-th layer in the network M, and let $\|l^{(i)}\|$ denote the number of neurons in $l^{(i)}$. Finally, let the total number of layers in M be denoted as L. The weights that connect $l^{(i)}$ to $l^{(i+1)}$ are denoted as the $\|l^{(i)}\|$-by-$\|l^{(i+1)}\|$ matrix $W^{(i)}$ and $\|l^{(i+1)}\|$ dimensional bias vector $\vec{b}^{(i)}$. Finally, we denote the collection of all parameters θ as the tuple $\theta \equiv (W, b)$, where W and b are the weights of each layer, respectively. Let $a^{(i+1)}$ denote the output (activation) of layer $l^{(i)}$ obtained by computing $f(W^{(i)} \cdot \vec{a}^{(i)} + \vec{b^{(i)}})$, where f is often the Sigmoid or ReLU function. To execute a network on an n-dimensional input x, a process known as *forward-propagation* is performed where x is used to activate $l^{(1)}$, which activates $l^{(2)}$ and so on until the activation of $l^{(L)}$ produces the m-dimensional output y.

To summarize this process, we consider M a black box and denote its execution as $M(x) = y$. To train M in a supervised setting, a dataset of paired samples with the form (x_i, y_i) is obtained and an objective loss function $\mathcal{L}$ is defined. The loss function is used to generate a signal at the output of M, which is *back-propagated* through M to find the errors of each weight. An optimization algorithm, such as gradient descent (GD), is then used to update the weights for a number of epochs. The function $\mathcal{L}$ is often a measure of error between the input x and predicted output y'. As a result, the network learns the function $M(x_i) \approx y_i$ and can be used to make predictions on unseen data.

Some deepfake networks use a technique called one-shot or few-shot learning, which enables a pre-trained network to adapt to a new dataset X' similar to X on which it was trained. Two common approaches for this are (1) to pass information on $x' \in X'$ to the inner layers of M during the feed-forward process, and (2) to perform a few additional training iterations on a few samples from X'.

3.2 Loss Functions

In order to update the weights with an optimization algorithm, such as GD, the loss function must be differentiable. There are various types of loss functions that can be applied in different ways depending on the learning objective. For example, when training an M as an n-class classifier, the

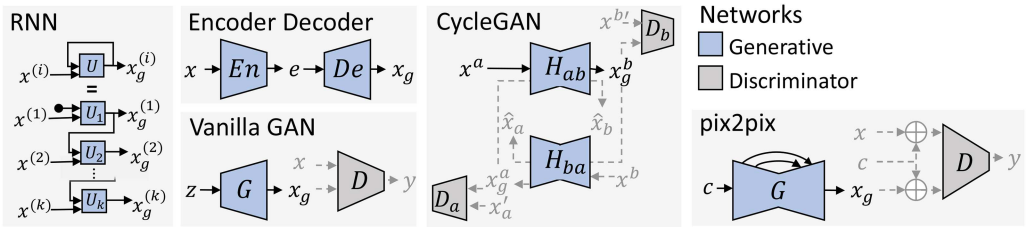


Fig. 4. Five basic neural network architectures used to create deepfakes. The lines indicate dataflows used during deployment (black) and training (gray).

output of M would be the probability vector $y \in \mathbb{R}^n$. To train M, we perform *forward-propagation* to obtain $y' = M(x)$, compute the cross-entropy loss ($\mathcal{L}_{CE}$) by comparing y' to the ground truth label y, and then perform *back-propagation* and to update the weights with the training signal. The loss $\mathcal{L}_{CE}$ over the entire training set X is calculated as

$$\mathcal{L}_{CE} = -\sum_{i=1}^{|X|}\sum_{c=1}^{n} y_i[c]\log(y_i'[c]), \tag{1}$$

where $y'[c]$ is the predicted probability of x_i belonging to the c-th class.

Other popular loss functions used in deepfake networks include the L1 and L2 norms $\mathcal{L}_1 = |x - x_g|^1$ and $\mathcal{L}_2 = |x - x_g|^2$. However, L1 and L2 require paired images (e.g., of s and t with same expression) and perform poorly when there are large offsets between the images such as different poses or facial features. This often occurs in reenactment when x_t has a different pose than x_s, which is reflected in x_g, and ultimately we'd like x_g to match the appearance of x_t.

One approach to compare two unaligned images is to pass them through another network (a perceptual model) and measure the difference between the layer's activations (feature maps). This loss is called the perceptual loss ($\mathcal{L}_{perc}$) and is described in Ref. [76] for image generation tasks. In the creation of deepfakes, $\mathcal{L}_{perc}$ is often computed using a face recognition network such as VGGFace. The intuition behind $\mathcal{L}_{perc}$ is that the feature maps (inner layer activations) of the perceptual model act as a normalized representation of x in the context of how the model was trained. Therefore, by measuring the distance between the feature maps of two different images, we are essentially measuring their semantic difference (e.g., how similar the noses are to each other and other finer details.) Similar to $\mathcal{L}_{perc}$, there is a feature matching loss ($\mathcal{L}_{FM}$) [132], which uses the last output of a network. The idea behind $\mathcal{L}_{FM}$ is to consider the high-level semantics captured by the last layer of the perceptual model (e.g., the general shape and textures of the head).

Another common loss is a type of content loss ($\mathcal{L}_C$) [59], which is used to help the generator create realistic features, based on the perspective of a perceptual model. In $\mathcal{L}_C$, only x_g is passed through the perceptual model and the difference between the network's feature maps are measured.

3.3 Generative Neural Networks (for Deepfakes)

Deep fakes are often created using combinations or variations of six different networks, five of which are illustrated in Figure 4.

Encoder-Decoder Networks (ED). An Encoder-Decoder (ED) consists of at least two networks, an encoder En and decoder De. The ED has narrower layers toward its center so that when it's trained as $De(En(x)) = x_g$, the network is forced to summarize the

observed concepts. The summary of x, given its distribution X, is $En(x) = e$, often referred to as an encoding or embedding and $E = En(X)$ is referred to as the "latent space." Deepfake technologies often use multiple encoders or decoders and manipulate the encodings to influence the output x_g. If an encoder and decoder are symmetrical, and the network is trained with the objective $De(En(x)) = x$, then the network is called an autoencoder, and the output is the reconstruction of x denoted $\hat{x}$. Another special kind of ED is the variational autorencoder (VAE), where the encoder learns the posterior distribution of the decoder given X. VAEs are better at generating content than autoencoders because the concepts in the latent space are disentangled, and, thus, encodings respond better to interpolation and modification.

Convolutional Neural Network (CNN). In contrast to a fully connected (dense) network, a convolutional neural network (CNN) learns pattern hierarchies in the data and is therefore much more efficient at handling imagery. A convolutional layer in a CNN learns filters that are shifted over the input forming an abstract feature map as the output. Pooling layers are used to reduce the dimensionality as the network gets deeper and up-sampling layers are used to increase it. With convolutional, pooling, and upsampling layers, it is possible to build ED CNNs for imagery.

Generative Adversarial Networks (GAN). The generative adversarial network (GAN) was first proposed in 2014 by Goodfellow et al. in Ref. [61]. A GAN consists of two neural networks that work against each other: the generator G and the discriminator D. G creates fake samples x_g with the aim of fooling D, and D learns to differentiate between real samples ($x \in X$) and fake samples ($x_g = G(z)$ where $z \sim N$). Concretely, there is an adversarial loss used to train D and G, respectively:

$$\mathcal{L}_{adv}(D) = \max \log D(x) + \log(1 - D(G(z))) \tag{2}$$

$$\mathcal{L}_{adv}(G) = \min \log(1 - D(G(z))) \tag{3}$$

This zero-sum game leads to G learning how to generate samples that are indistinguishable from the original distribution. After training, D is discarded and G is used to generate content. When applied to imagery, this approach produces photo realistic images.

Numerous variations and improvements of GANs have been proposed over the years. In the creation of deepfakes, there are two popular image translation frameworks that use the fundamental principles of GANs:

Image-to-Image Translation (pix2pix). The pix2pix framework enables paired translations from one image domain to another [72]. In pix2pix, G tries to generate the image x_g given a visual context x_c as an input, and D discriminates between (x, x_c) and (x_g, x_c). Moreover, G is an ED CNN with skip connections from En to De (called a U-Net), which enables G to produce high-fidelity imagery by bypassing the compression layers when needed. Later, pix2pixHD was proposed [169] for generating high-resolution imagery with better fidelity.

CycleGAN. A CycleGAN is an improvement of pix2pix that enables image translation through unpaired training [191]. The network forms a cycle consisting of two GANs used to convert images from one domain to another, and then back again to ensure consistency with a cycle consistency loss ($\mathcal{L}_{cyc}$).

Recurrent Neural Networks (RNN). A recurrent neural network (RNN) is type of neural network that can handle sequential and variable length data. The network remembers its internal state after processing $x^{(i-1)}$ and can use it to process $x^{(i)}$ and so on. In deepfake creation, RNNs are often used to handle audio and sometimes video. More advanced

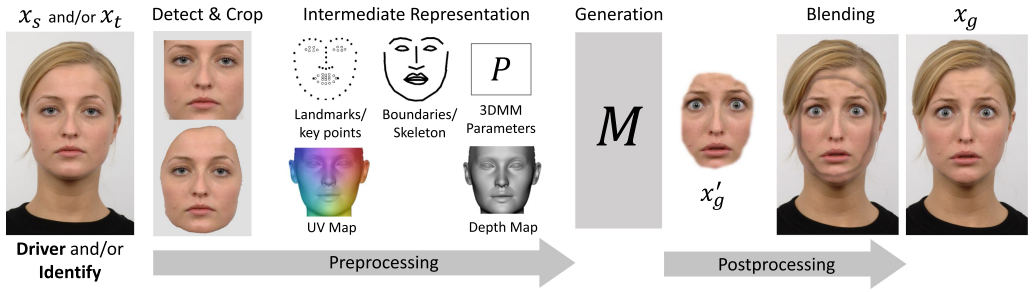


Fig. 5. The processing pipeline for making reenactment and face swap deepfakes. Usually only a subset of these steps are performed.

versions of RNNs include long short-term memory (LSTM) and gate reccurent units (GRU).

3.4 Feature Representations

Most deep fake architectures use some form of intermediate representation to capture and sometimes manipulate s and t's facial structure, pose, and expression. One way is to use the facial action coding system (FACS) and measure each of the face's taxonomized action units (AU) [43]. Another way is to use monocular reconstruction to obtain a 3D morphable model (3DMM) of the head from a 2D image, where the pose and expression are parameterized by a set of vectors and matrices. Then use the parameters or a 3D rendering of the head itself. Some use a UV map of the head or body to give the network a better understanding of the shape's orientation.

Another approach is to use image segmentation to help the network separate the different concepts (face, hair, etc.). The most common representation is landmarks (a.k.a. key-points), which are a set of defined positions on the face or body that can be efficiently tracked using open source computer vision (CV) libraries. The landmarks are often presented to the networks as a 2D image with Gaussian points at each landmark. Some works separate the landmarks by channel to make it easier for the network to identity and associate them. Similarly, facial boundaries and body skeletons can also be used.

For audio (speech), the most common approach is to split the audio into segments, and for each segment, measure the Mel-Cepstral Coefficients (MCC), which captures the dominant voice frequencies.

3.5 Deepfake Creation Basics

To generate x_g, reenactment and face swap networks follow some variation of this process (illustrated in Figure 5): Pass x through a pipeline that (1) detects and crops the face, (2) extracts intermediate representations, (3) generates a new face based on some driving signal (e.g., another face), and then (4) blends the generated face back into the target frame.

In general, there are six approaches to driving an image:

(1) Let a network work directly on the image and perform the mapping itself.
(2) Train an ED network to disentangle the identity from the expression, and then modify/swap the encodings of the target before passing it through the decoder.
(3) Add an additional encoding (e.g., AU or embedding) before passing it to the decoder.
(4) Convert the intermediate face/body representation to the desired identity/expression before generation (e.g., transform the boundaries with a secondary network or render a 3D model of the target with the desired expression).

(5) Use the optical flow field from subsequent frames in a source video to drive the generator.
(6) Create a composite of the original content (hair, scene, etc) with a combination of the 3D rendering, warped image, or generated content, and pass the composite through another network (such as pix2pix) to refine the realism.

3.6 Generalization

A deepfake network may be trained or designed to work with only a specific set of target and source identities. An identity agnostic model is sometimes hard to achieve due to correlations learned by the model between s and t during training.

Let E be some model or process for representing or extracting features from x, and let M be a *trained* model for performing replacement or reenactment. We identify three primary categories in regard to generalization:

one-to-one: A model that uses a specific identity to drive a specific identity: $x_g = M_t(E_s(x_s))$
many-to-one: A model that uses any identity to drive a specific identity: $x_g = M_t(E(x_s))$
many-to-many: A model that uses any identity to drive any identity: $x_g = M(E_1(x_s), E_2(x_t))$

3.7 Challenges

The following are some challenges in creating realistic deepfakes:

Generalization. Generative networks are data driven and, therefore, reflect the training data in their outputs. This means that high-quality images of a specific identity requires a large number of samples of that identity. Moreover, access to a large dataset of the driver is typically much easier to obtain than the victim. As a result, over the last few years, researchers have worked hard to minimize the amount of training data required, and to enable the execution of a trained model on new target and source identities (unseen during training).

Paired Training. One way to train a neural network is to present the desired output to the model for each given input. This process of *data pairing* is laborious and sometimes impractical when training on multiple identities and actions. To avoid this issue, many deepfake networks either (1) train in a self-supervised manner by using frames selected from the same video of t, (2) use unpaired networks such as Cycle-GAN, or (3) utilize the encodings of an ED network.

Identity Leakage. Sometimes the identity of the driver (e.g., s in reenactment) is partially transferred to x_g. This occurs when training on a single input identity, or when the network is trained on many identities, but data pairing is done with the same identity. Some solutions proposed by researchers include attention mechanisms, few-shot learning, disentanglement, boundary conversions, and AdaIN or skip connections to carry the relevant information to the generator.

Occlusions. Occlusions are where part of x_s or x_t is obstructed with a hand, hair, glasses, or any other item. Another type of obstruction is the eyes and mouth region that may be hidden or dynamically changing. As a result, artifacts appear such as cropped imagery or inconsistent facial features. To mitigate this, works such as Refs [120], [127], and [144] perform segmentation and in-painting on the obstructed areas.

Temporal Coherence. Deepfake videos often produce more obvious artifacts such as flickering and jitter [163]. This is because most deepfake networks process each frame individually with no context of the preceding frames. To mitigate this, some researchers

Table 1. Summary of Deep Learning Reenactment Models (Body and Face)

				Reenactment						Retraining for new...		Model				Representation					Model Training							Model Execution			Model Output	
				Mouth	Expr.	Pose	Gaze	Body		Source (*s*)	Target (*t*)	Identity Agnostic	Encoders	Decoders	Discriminators	Other Netw.	AU/AAM	3DMM/Rendering	UV Mapping	Segmentation	Landmark / Keypoint	Boundary / Skeleton	Labeling of: ID	Labeling of: Action	No Pairing	Paring within Same Video	Paring ID to Same ID	Paring ID to Diffr. Actions	Paring Action to Diffr. IDs	Requires Video	Source ($x_{s...}$)	Target ($x_{t...}$)	Image/Frame	Video	Resolution
One-to-One	[175]	2017	FT-GAN	●	●	●	○		>20 min. video	>20 min. video		2	2	2	0														●	portrait	portrait	●		128x128
	[15]	2018	Recycle-GAN	●	●	●			5-10 min. video	5-10 min. video		4	4	2	0							●		●					●	portrait	-	●		512x512
	[71]	2018	DeepFaceLab	●	●	●			1-3 hr. video	1-3 hr. video		1	2	1	1							●		●					●	portrait video	-	●		512x512
	[104]	2019	Liu et al. 2019	●	●	●		●	1-3 hr. video	1-3 hr. video		4	4	2	1	●					●	●				●			●	upperbody video	-	●		>256x256
Many-to-One	[151]	2017	Syth. Obama	●					None	17 hr. video		0	0	0	1					●					●			●	audio	portiat video		●	2048x1024
	[89]	2017	ObamaNet	●					None	17 hr. video		1	1	1	1						●	●			●			●	text	-		●	256x256
	[83]	2018	Deep Video Portr.	●	●	●	●		None	1-3 min. video		1	1	1	0		●	●				●			●			●	portrait video	neural texture	●		1024x1024
	[173]	2018	ReenactGAN	●	●	●			None	30 min. video		N	N	N	1						●	●		●				●	portrait	portriat	●		256x256
	[168]	2018	Vid2vid	●	●	●	○	or ●	None	3-8 min. video		3	3	2	1						●	●			●			●	portrait video	-	●		2048x1024
	[161]	2018	MocoGAN	●	●			or ●	None	1 min. video		2	1	2	N							●	●	●				●	expression label	identity label		●	64x64
	[73]	2018	SD-CGAN	●	○	○			None	2 hr. video		0	1	1	1					●		●			●			●	audio	-		●	128x128
	[180]	2019	GRN				●		None	3-10 images	o	3	1	0	2				●							●			gaze	3-10 eye images	●		64x128
	[55]	2019	TETH	●					None	1 hr. video		1	1	2	0	●						●			●			●	text	portiat video		●	512x512
	[153]	2019	N.V. Puppetry	●					None	2-3 min. video		3	2	2	4	●						●			●			●	audio	portiat video		●	512x512
	[102]	2019	NRR-HAV			●		●	None	8 min. video		1	1	1	0	●					●	●			●				body image	background	●		512x512
	[2]	2019	Deep Video P.C.			●		●	None	2 min. video		0	1	2	2					●		●			●			●	body image	-	●		256x256
	[25]	2019	Everybody D. N.		●	●		●	None	20 min. video		0	2	4	2						●	●			●			●	body image	-	●		256x256
	[190]	2019	D. D. Generation			●		●	None	3 min. video		2	2	2	2					●	●	●			●			●	body video	-	○	●	512x512
	[182]	2019	N. Talking Heads	●	●	●			None	1-3 portraits	o	1	2	1	1						●	●			●			●	portrait/landmarks	1-3 portraits	●		256x256
	[167]	2019	Few-shot Vid2Vid	●	●	●	○	or ●	None	1-10 portraits	o	3	3	2	4							●			●			●	portrait/body video	1-10 portr./bodies	●		2048x1024
Many-to-Many	[142]	2015	Shimba et al.	●	○				None	None		0	0	0	1	●						●			●			●	audio	face database		●	*
	[57]	2016	DeepWarp				●		None	None	●	0	0	0	2					●		●				●			gaze	eye image	●		>40x50
	[16]	2017	CVAE-GAN	●	●	●			None	None	●	1	1	1	1							●	●	●					latent variables	portrait	●		>128x128
	[123]	2017	RDFT	●	●				None	None	●	1	1	1	0			●				●					●		portrait	portrait	●		256x256
	[189]	2017	FE-CDAE	●	●				None	None	●	1	1	2	0	●						●	●	●					portrait	AU label	●		32x32
	[112]	2018	paGAN	●	●	●	●		None	None	●	1	1	1	1		●	●				●					●		portrait	portrait - neutral	●		512x512
	[171]	2018	X2Face	●	●	●			None	None	●	2	2	0	1							●			●			●	portrait	1-3 portraits	●		256x256
	[134]	2018	GANnotation	●	●	●			None	None	●	1	1	1	3					●		●			●			●	portrait/landmarks	portrait	●		128x128
	[126]	2018	GATH	●	●				None	None	●	1	1	1	2	●						●				●			portrait/AUs	portrait	●		100x100
	[140]	2018	FaceID-GAN	●	●	●			None	None	●	1	1	2	1		●				●	●	●	●					portrait	portrait	●		128x128
	[141]	2018	FaceFeat-GAN	●	●	●			None	None	●	1	1	3	4		●				●	●	●						latent variables	portrait	●		128x128
	[70]	2018	CAPG-GAN			●			None	None	●	1	1	2	1					●		●	●				●		portrait	portrait	●		128x128
	[158]	2018	DR-GAN			●			None	None	●	1	1	1	0						●	●	●						pose	1+ portraits	●		96x96
	[145]	2018	Deformable GAN			●		●	None	None	●	1	1	1	0					●	●	●				●			body image/landm.	body image	●		256x256
	[14]	2018	SHUP			●		●	None	None	●	3	3	1	1					●		●				●		●	body image	body image/pose	●		256x256
	[46]	2018	DPIG			●		●	None	None	●	4	2	1	0					●				●	●				body image	body image	●		128x64
	[116]	2018	Dense Pose Tr.			●		●	None	None	●	25	25	1	2			●				●					●		body image	body image	●		256x256
	[146]	2018	Song et al.	●	○				None	None	●	2	1	3	0										●			●	audio	portrait		●	128x128
	[60]	2019	wg-GAN	●	●				None	None	●	2	2	3	0					●					●			●	portrait	portrait	●		256x256
	[120]	2019	FSGAN	●	●	●			None	None	●	1	1	1	1		●			●		●			●			●	portrait/landmarks	portrait	●		256x256
	[127]	2019	GANimation	●	●				None	None	●	2	2	1	1	●						●	●						portrait/AUs	portrait	●		128x128
	[159]	2019	ICface	●	●	●			None	None	●	2	2	1	2	●						●			●			●	portrait/AUs	portrait	●		128x128
	[184]	2019	FaceSwapNet	●	●				None	None	●	4	2	1	0					●		●					●		portrait/landmarks	portrait/landmarks	●		256x256
	[143]	2019	Monkey-Net	●	●			or ●	None	None	●	3	3	1	0					●		●			●			●	portrait/body	portrait/body	●		64x64
	[144]	2019	First-Order-Model	●	●	●		or ●	None	None	●	3	3	1	1					●		●			●			●	portrait/body	portrait/body	●		256x256
	[124]	2019	M&T GAN	●	●				None	None	●	2	1	2	1					●		●			●			●	expression label	portrait		●	64x64
	[48]	2019	AF-VAE	●	●	●			None	None	●	2	1	0	1						●				●				portrait/boundaries	portrait	●		256x256
	[56]	2019	Fu et al. 2019	●	●	●			None	None	●	3	2	3	4						●		●		●				portrait/label	portrait	●		1024x1024
	[185]	2019	FusionNet	●	●	●	●		None	None	●	1	2	3	3				●						●				portriat/landmarks	portrait	●		256x256
	[23]	2019	AD-GAN			●			None	None	●	2	2	2	1			●						●	●				pose	portrait	●		128x128
	[163]	2019	Speech D. Anm. 1	●	○		○		None	None	●	5	1	2	3							●			●			●	audio	portrait		●	96x128
	[164]	2019	Speech D. Anm. 2	●	○		○		None	None	●	5	1	3	3							●			●			●	audio	portrait		●	96x128
	[79]	2019	Speech D. Anm. 3	●	○		○		None	None	●	5	1	3	3							●			●			●	audio	portrait		●	96x128
	[188]	2019	DAVS	●					None	None	●	3	1	1	4							●	●		●			●	audio/portrait video	portrait	○	●	256x256
	[27]	2019	ATVGnet	●					None	None	●	1	0	1	5					●		●			●			●	audio	portiat video		●	128x128
	[74]	2019	Speech2Vid	●					None	None	●	3	1	0	2							●			●			●	audio	portiat video		●	109x109
	[181]	2019	DwNet					●	None	None	●	2	1	1	3				●		●	●			●			●	body video	body image		●	256x256
	[62]	2019	LW-GAN					●	None	None	●	3	3	1	2				●			●			●			●	body image	body image	●		256x256
	[35]	2019	C-DGPose			●		●	None	None	●	2	1	1	0					●		●				●			body image	body/pose image	●		64x64
	[192]	2019	PPAT-PIG			●		●	None	None	●	2	1	2	1					●					●				body image	body/pose image	●		256x256
	[170]	2020	ImaGINator	●	●				None	None	●	1	1	2	0							●			●			●	expression label	portrait		●	64x64
	[65]	2020	MarioNETte	●	●	●			None	None	●	2	2	1	3					●		●			●			●	portrait	1-8 portraits	●		256x256
	[62]	2020	FLNet	●	●	●			None	None	●	1	5	1	1					●		●			●	●		●	portrait	16 portraits	●		224x224

either provide this context to *G* and *D*, implement temporal coherence losses, use RNNs, or perform a combination thereof.

4 REENACTMENT

In this section, we present a chronological review of deep learning based reenactment, organized according to their class of identity generalization. Table 1 provides a summary and systematization of all the works mentioned in this section. Later, in Section 7, we contrast the various methods and identify the most significant approaches.

4.1 Expression Reenactment

Expression reenactment turns an identity into a puppet, giving attackers the most flexibility to achieve their desired impact. Before we review the subject, we note that expression reenactment has been around long before deepfakes were popularized. In 2003, researchers morphed models of 3D scanned heads [19]. In 2005, it was shown how this can be done without a 3D model [26], and through warping with matching similar textures [58]. Later, between 2015 and 2018, Thies et al. demonstrated how 3D parametric models can be used to achieve high-quality and real-time results with depth sensing and ordinary cameras (see Refs [155]–[157]).

Regardless, today deep learning approaches are recognized as the simplest way to generate believable content. To help the reader understand the networks and follow the text, we provide the model's network schematics and loss functions in Figures 6–8.

4.1.1 One-to-One (Identity-to-Identity). In 2017, the authors of Ref. [175] proposed using a CycleGAN for facial reenactment without the need for data pairing. The two domains were video frames of s and t. However, to avoid artifacts in x_g, the authors note that both domains must share a similar distribution (e.g., poses and expressions).

In 2018, Bansal et al. proposed a generic translation network based on CycleGAN called Recycle-GAN [15]. Their framework improves temporal coherence and mitigates artifacts by including next-frame predictor networks for each domain. For facial reenactment, the authors train their network to translate the facial landmarks of x_s into portraits of x_t.

4.1.2 Many-to-One (Multiple Identities to a Single Identity). In 2017, the authors of Ref. [16] proposed a conditional VAE-GAN (CVAE-GAN) where the generator is conditioned on an attribute vector or class label. However, reenactment with CVAE-GAN requires manual attribute morphing by interpolating the latent variables (e.g., between target poses).

Later, in 2018, a large number of source-identity agnostic models were published, each proposing a different method to decoupling s from t:[5]

Facial Boundary Conversion. One approach was to first convert the structure of source's facial boundaries to that of the target's before passing them through the generator [173]. In their framework "ReenactGAN," the authors use a CycleGAN to transform the boundary b_s to the target's face shape as b_t before generating x_g with a pix2pix-like generator.

Temporal GANs. To improve the temporal coherence of deepfake videos, the authors of Ref. [161] proposed motion and content GAN (MoCoGAN): a temporal GAN that generates videos while disentangling the motion and content (objects) in the process. Each frame is generated using a target expression label z_c, and a motion embedding $z_M^{(i)}$ for the i-th frame, obtained from a noise seeded RNN. MoCoGAN uses two discriminators, one for realism (per frame) and one for temporal coherence (on the last T frames).

In Ref. [168], the authors proposed a framework called Vid2Vid, which is similar to pix2pix but for videos. Vid2Vid considers the temporal aspect by generating each frame based on the last L source and generated frames. The model also considers optical flow to perform next-frame occlusion prediction (due to moving objects). Similar to pix2pixHD, a progressive training strategy is to generate high-resolution imagery. In their evaluations, the authors demonstrate facial reenactment using the source's facial boundaries. In comparison to MoCoGAN, Vid2Vid is more practical since the deepfake is driven by x_s (e.g., an actor) instead of crafted labels.

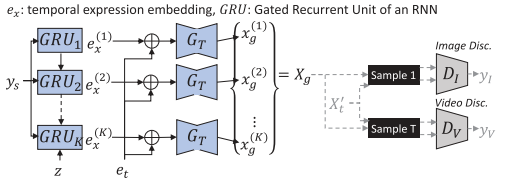
[5]Although works such as Refs [123] and [189] achieved fully agnostic models (many-to-many) in 2017, their works were on low resolution or partial faces.

x_s, x_t, x_g: The source, target, and generated images (e.g., portraits)
y: A label (e.g., fake vs real, one-hot encoding, ...)
x': Another sample from the same distribution, $\hat{x}$: reconstructed
m: Binary mask, s: Segmentation map, l: Landmark or Keypoint, z: Noise
⊕: Concatenate, ⊖: Subtract, ◉:Multiply, Σ: Add, (p): Paste content
: Crop out region a from image where $a \in$ {f:face, e:eye, m:mouth}
: Create mask using region a of the image where $a \in$ {f:face, e:eye, m:mouth}
$x^{(a)}$: Image x cropped to the region of $a \in \{f$:face, e:eye, m:mouth$\}$
Ⓢ: Spatial replication of a vector (channel-wise or dim-wise)
: Scale image down by factor of X
$LE, BE, AE, 3DE$: Landmark, Boundary, Action Unit (AU), and 3DMM facial model Extractors (open source CV library)
$LT, 3DT$: Landmark and 3D model transformers, from s to t
ME: MFCC audio feature extractor

Losses: $\mathcal{L}_1$: L1, $\mathcal{L}_2$: L2, $\mathcal{L}_{CE}$: Cross Entropy, $\mathcal{L}_{adv}$: Adversarial, $\mathcal{L}_{FM}$: Feature Matching, $\mathcal{L}_{perc}$: Perceptual, $\mathcal{L}_{cyc}$: Cycle Consistency, $\mathcal{L}_{att}$: Attention, $\mathcal{L}_{trip}$: Triplet, $\mathcal{L}_{tv}$: Total Variance, $\mathcal{L}_{KL}$: KL Divergence

[161] MocoGAN:

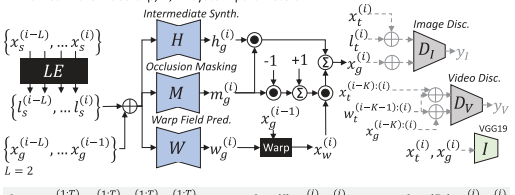
y_s: source expression label, e_t: one-hot encoding of target identity,
e_x: temporal expression embedding, GRU: Gated Recurrent Unit of an RNN



$\mathcal{L}_{adv(I)}$: $y_I, {x'^{(i)}_t}/{x^{(i)}_g}$ $\quad$ $\mathcal{L}_{adv(V)}$: $y_V, {x_t^{(i:i+T)}}/{x_g^{(i:i+T)}}$

[168] Vid2Vid:

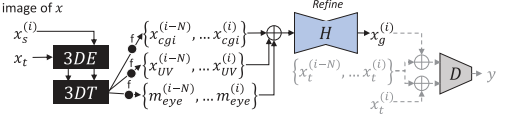
T: frames in the video clip, L, K: system parameters



$\mathcal{L}_{flow}$: $w_g^{(1:T)}, w_t^{(1:T)}, x_w^{(1:T)}, x_t^{(1:T)}$ $\quad$ $\mathcal{L}_{FM(I)}$: $x_t^{(i)}, x_g^{(i)}$ $\quad$ $\mathcal{L}_{FM(D_I)}$: $x_t^{(i)}, x_g^{(i)}$
$\mathcal{L}_{adv(D_I)}$: $y_I, l_t^{(i)}, {x_t^{(i)}}/{x_g^{(i)}}$ $\quad$ $\mathcal{L}_{advV(D_V)}$: $y_V, w_t^{(i-K-1):(i)}, {x_t^{(i-K):(i)}}/{x_g^{(i-K):(i)}}$

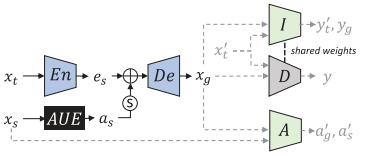
[83] Deep Video Portrait:

m_{eye}: mask of eye region (gaze), x_{UV}: UV correspondence map, x_{cgi}: 3D rendered image of x



$\mathcal{L}_1$: $x_t^{(i)}, x_g^{(i)}$ $\quad$ $\mathcal{L}_{adv}$: $y, \{x_t^{(i-N)}, \dots x_t^{(i)}\}, {x_t^{(i)}}/{x_g^{(i)}}$

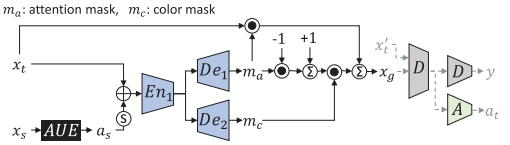
[126] GATH:



$\mathcal{L}_{adv}$: $y, {x_g}/{x'_t}$ $\quad$ $\mathcal{L}_{AU(A)}$: $\mathcal{L}_2(a'_g, a'_s)$ $\quad$ $\mathcal{L}_1$: x_t, x_g $\quad$ $\mathcal{L}_{CE(I)}$: y'_t, y_g $\quad$ $\mathcal{L}_{tv}$: x_g

[127] GANimation:

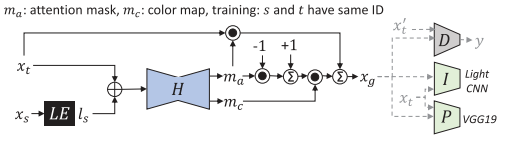
m_a: attention mask, m_c: color mask



$\mathcal{L}_1$: $G(G(x_t, a_s), a_t), x_t$ $\quad$ $\mathcal{L}_{tv}$: m_a $\quad$ $\mathcal{L}_{attn}$: m_a -see paper
$\mathcal{L}_{expression}$: $a_t, {x_g}/{x'_t}$ -see paper $\quad$ $\mathcal{L}_{adv}$: $y, {x_g}/{x'_t}$

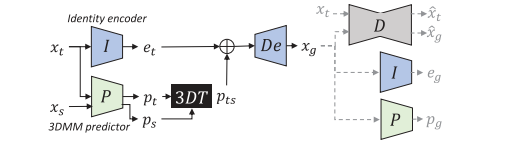
[134] GANotation:

m_a: attention mask, m_c: color map, training: s and t have same ID



$\mathcal{L}_{tv}$: x_s, x_g $\quad$ $\mathcal{L}_2$: x_s, x_g (same ID) $\quad$ $\mathcal{L}_{consistency}$: $H(x_g, l_t), x_t$ $\quad$ $\mathcal{L}_{adv}$: y, x_g, x'_t
$\mathcal{L}_{3\times consistency}$: $H(x_g, l'_s), H(x_t, l'_s)$ $\quad$ $\mathcal{L}_{perc}$: $l_{\ell_k,\ell_{k-1}}(x_g), l_{\ell_k,\ell_{k-1}}(x_t)$
$\mathcal{L}_{perc}$: $P_{\ell_k,\ell_{k-1}}(x_g), P_{\ell_k,\ell_{k-1}}(x_t)$

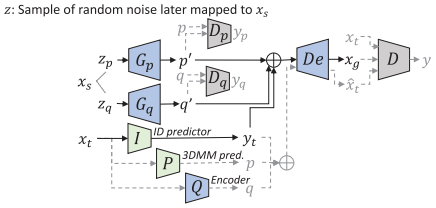
[140] FaceID-GAN:



$\mathcal{L}_{adv}$: $\mathcal{L}_1(x_t, \hat{x}_t) - \alpha\mathcal{L}_1(x_g, \hat{x}_g)$ $\quad$ $\mathcal{L}_{CE}(I)$: e_g, y_{id} $\quad$ $\mathcal{L}_{CE(I)}$: e_t, y_{id}
$\mathcal{L}_1(De)$: $x_g, \hat{x}_g$ $\quad$ $\mathcal{L}_2(De)$: p_{ts}, p_g $\quad$ $\mathcal{L}_{cos}(De)$: e_g, e_t
$\mathcal{L}_{pose}(P)$: $p_t, 3DMM(x_t)$

[141] FaceFeat-GAN:

z: Sample of random noise later mapped to x_s

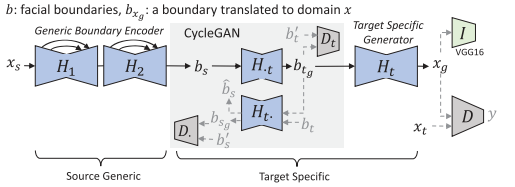


$\mathcal{L}_{adv}(D)$: $y, {x_t}/{x_g, \hat{x}_t}$ $\quad$ $\mathcal{L}_{adv}(D_p)$: $y_p, {p}/{p'}$ $\quad$ $\mathcal{L}_{adv}(D_q)$: $y_q, {q}/{q'}$ $\quad$ $\mathcal{L}_{CE}(I)$: y_t, y'_t
$\mathcal{L}_{pose}(P)$: $p, 3DMM(x_t)$ $\quad$ $\mathcal{L}_1(Q)$: $x_t, G(y_t, p, Q(x_t))$ $\quad$ $\mathcal{L}_1(G)$: $x_t, \hat{x}_t$
$\mathcal{L}_2(G)$: $I(\hat{x}_t), y_t$ & $I(x_g), y_t$

Fig. 6. Architectural schematics of *reenactment networks*. Black lines indicate prediction flows used during deployment; dashed gray lines indicate dataflows performed during training. Zoom in for more detail.

[173] Reenact GAN:

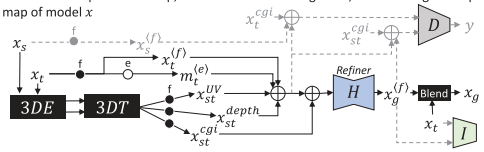
b: facial boundaries, b_{x_g}: a boundary translated to domain x



$\mathcal{L}_{adv}$: $y, {x_g}/{x_t}$ $\quad$ $\mathcal{L}_1$: x_t, x_g $\quad$ $\mathcal{L}_{content}$: $l_{\ell_2}(x_g), l_{\ell_3}(x_g)$ $\quad$ $\mathcal{L}_{cycle}$: $\hat{b}_t, b_t$
$\mathcal{L}_{cycle}$: $\hat{b}_s, b_s$ $\quad$ $\mathcal{L}_{adv(\cdot)}$: $y, {b'}/{b_{s_g}}$ $\quad$ $\mathcal{L}_{adv(t)}$: $y_t, {b'_t}/{b_{t_g}}$

[112] paGAN:

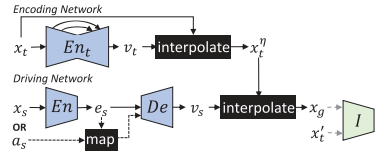
x^{UV}: UV correspondence map, x^{cgi}: 3D rendered image of x, x^{depth}: image of depth map of model x



$\mathcal{L}_{adv}: x_t^{\langle f \rangle}, m_t^{\langle e \rangle}, x_{st}^{UV}, x_{st}^{depth}, {}^{x_s^{\langle f \rangle}, x_t^{cgi}}/_{x_g^{\langle f \rangle}, x_{st}^{cgi}}$ $\mathcal{L}_{perc}(I): x_t^{\langle f \rangle}, x_g^{\langle f \rangle}$ $\mathcal{L}_1: x_{st}^{cgi}, x_g^{\langle f \rangle}$

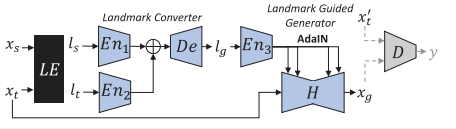
[171] X2Face:

v: vector map of pixel deltas (changes), x^{η}: a face with a neutral expression/pose, a: some other modality (e.g., audio)



$\mathcal{L}_1: x_s, x_g$ $\mathcal{L}_{perc}(I): x_t, x_g$

[184] FaceSwapNet:



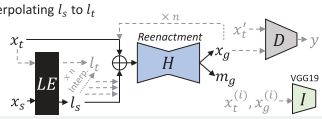
$\mathcal{L}_1: x_g, x_t'$ (with same expr.) $\mathcal{L}_2: l_g, l_t'$ (with same expr.) $\mathcal{L}_{adv}: y, x_g, x_t'$

$\mathcal{L}_{triplet}: \mathcal{L}_{perc}(x_{t1}, x_{t2}), \mathcal{L}_{perc}(x_{t1}, x_s)$

Same ID, different expression — Different IDs

[120] FSGAN:

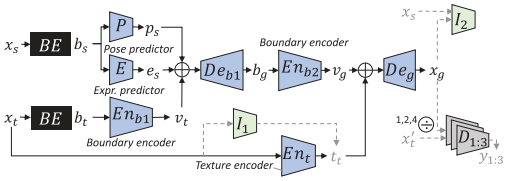
m: segmentation mask (face, hair, other), l: 3D facial landmarks, H^n: n passes through H while interpolating l_s to l_t



$\mathcal{L}_1: x_g, x_t$ $\mathcal{L}_1: H^n(x_g, l_s), x_t$ $\mathcal{L}_{perc}(H): l_{\ell_x}(x_t), l_{\ell_x}(x_t)$

$\mathcal{L}_{perc}(H): l_{\ell_x}(H^n(x_g, l_s)), l_{\ell_x}(x_t)$ $\mathcal{L}_{adv}: y, x_g, x_t'$

[56] Fu et al. 2019:

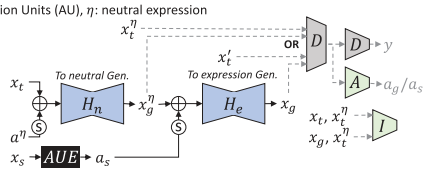


$\mathcal{L}_{reg}(De_b): De_b(p_s', e_s', v_t), P(p_s'), E(e_s')$ $\mathcal{L}_1(De_b): b_t, b_g$ $\mathcal{L}_1(De_g): {}^{1,2,4}\oplus x_g, x_s$

$\mathcal{L}_{thr}: l_1(x_t), En_t(x_t)$ $\mathcal{L}_{adv\ 1,2,3}: {}^{1,2,4}\oplus y, {}^{x_t'}/_{x_g}$ $\mathcal{L}_{FM}(De_g): l_{2,\ell}(x_s), l_{2,\ell}(x_g)$

[159] ICFace:

a: Action Units (AU), η: neutral expression

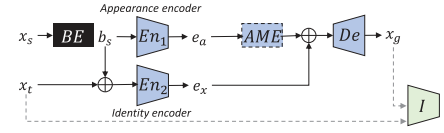


$\mathcal{L}_1: x_t, x_g$ $\mathcal{L}_1: x_g^{\eta}, H_e(x_t', a^{\eta})$ $\mathcal{L}_{adv(\eta)}: y, {}^{x_g^{\eta}}/_{x_t^{\eta}}$ $\mathcal{L}_{adv(e)}: y, {}^{x_g}/_{x_t'}$

$\mathcal{L}_2: A(x_t), a_s$ $\mathcal{L}_2: A(x_g), a_s$ $\mathcal{L}_2: A(x_g^{\eta}), a^{\eta}$ $\mathcal{L}_{perc}(I): x_t, x_t^{\eta}$ $\mathcal{L}_{perc}(I): x_g, x_t^{\eta}$

[48] AF-VAE:

AME: Additive Memory Encoder – *models e_a as a Gaussian mixture of clustered facial boundaries.*

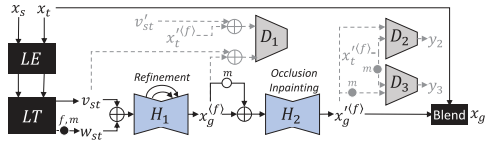


$\mathcal{L}_{KL}: x_g, e_a, e_x, AME$ $\mathcal{L}_{FM}(I): x_g, x_t$

[60] wg-GAN:

v_{st}: vector map of the warp from x_t to x_s, $w_{st}: x_t$ warped according to v_{st}

Training: for each $x_t^{(i)}$, $x_s = x_t^{(i-10)}$ taken from the same video clip

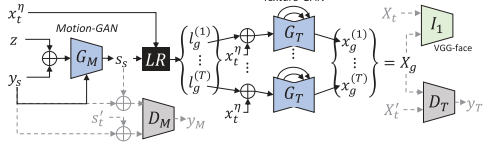


$\mathcal{L}_{adv1}: {}^{v_{st}', x_t'^{(f)}}/_{v_{st}, x_g^{(f)}}$ $\mathcal{L}_{adv2}: y_2, {}^{x_t'^{(f)}}/_{x_g'^{(f)}}$ $\mathcal{L}_{adv3}: y_3, {}^{x_t'^{(m)}}/_{x_g'^{(m)}}$

$\mathcal{L}_1: x_g'^{(f)}, x_t^{(f)}$ $\mathcal{L}_2: x_g'^{(m)}, x_t'^{(m)}$

[124] Motion&Texture-GAN:

x^{η}: cropped neutral expression face, y_s: face expression label of source, s: an SRVF point on a spherical manifold, LR: landmark reconstruction from s, l: facial landmarks

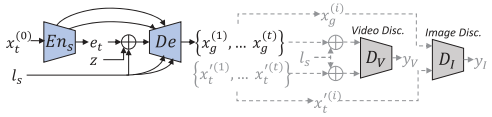


$\sum_{i=1}^{T} \mathcal{L}_1: x_t'^{(i)}, x_g^{(i)}$ $\sum_{i=1}^{T} \mathcal{L}_{adv(T)}: y_T, {}^{x_t'^{(i)}}/_{x_g^{(i)}}$ $\sum_{i=1}^{T} \mathcal{L}_{perc}(I): x_t'^{(i)}, x_g^{(i)}$

$\mathcal{L}_1: s_s, s_t'$ $\mathcal{L}_{adv(M)}: y_S, y_m, {}^{s_s}/_{s_t'}$

[170] ImaGINator:

l: One-hot label encoding of expression, z: Random value $z \sim N(0,1)$



$\mathcal{L}_1: \{x_t'^{(1)}, \ldots x_t'^{(t)}\}, \{x_g^{(1)}, \ldots x_g^{(t)}\}$ $\mathcal{L}_{adv(V)}: y_V, l_s, \{x_t'^{(1)}, \ldots\}, \{x_g^{(1)}, \ldots\}$

$\mathcal{L}_{adv(I)}: y_I, x_t'^{(i)}, x_g^{(i)}$

Fig. 7. Architectural schematics of *reenactment networks*. Black lines indicate prediction flows used during deployment; dashed gray lines indicate dataflows performed during training. Zoom in for more detail.

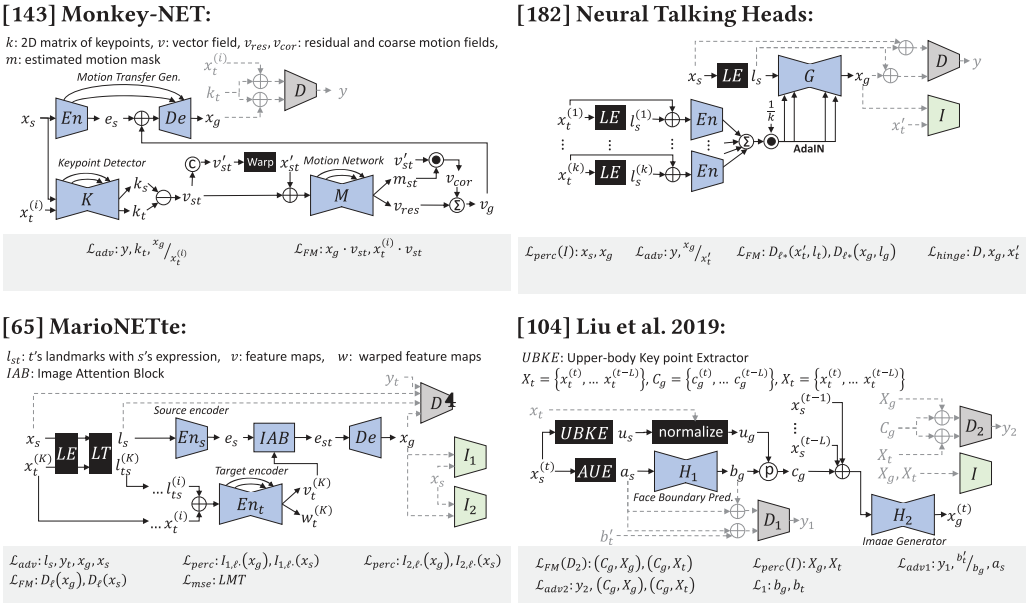


Fig. 8. Architectural schematics of the **reenactment networks**. Black lines indicate prediction flows used during deployment; dashed gray lines indicate dataflows performed during training. Zoom in for more detail.

The authors of Ref. [83] took temporal deepfakes one step further achieving complete facial reenactment (gaze, blinking, pose, mouth, etc.) with only 1 minute of training video. Their approach was to extract the source and target's 3D facial models from 2D images using monocular reconstruction. Then, for each frame, (1) transfer the facial pose and expression of the source's 3D model to the target's; and (2) produce x_g with a modified pix2pix framework, using the last 11 frames of rendered heads, UV maps, and gaze masks as the input.

4.1.3 Many-to-Many (Multiple IDs to Multiple IDs). **Label Driven Reenactment.** The first attempts at identity agnostic models were made in 2017, where the authors of Ref. [123] used a conditional GAN (CGAN) for the task. Their approach was to (1) extract the inner-face regions as (x_t, x_s), and then (2) pass them to an ED to produce x_g subjected to $\mathcal{L}_1$ and $\mathcal{L}_{adv}$ losses. The challenge of using a CGAN was that the training data had to be paired (images of different identities with the same expression).

Going one step further, in Ref. [189] the authors reenacted full portraits at low resolutions. Their approach to decoupling the identities was to use a conditional adversarial autoencoder to disentangle the identity from the expression in the latent space. However, their approach is limited to driving x_t with discreet AU expression labels (fixed expressions) that capture x_s. A similar label-based reenactment was presented in the evaluation of StarGAN [29]; an architecture similar to CycleGAN but for N domains (poses, expressions, etc).

Later, in 2018, the authors of Ref. [126] proposed GATH which can drive x_t using continuous AU as an input, extracted from x_s. Using continuous AUs enables smoother reenactments over previous approaches [29, 123, 189]. Their generator is ED network trained on the loss signals from using three other networks: (1) a discriminator, (2) an identity classifier, and (3) a pretrained AU estimator. The classifier shares the same hidden weights as the discriminator to disentangle the identity from the expressions.

Self-Attention Modeling. Similar to Ref. [126], another work called GANimation [127] reenacts faces through AU value inputs estimated from x_s. Their architecture uses an AU-based generator that uses a self attention model to handle occlusions, and mitigate other artifacts. Furthermore, another network penalizes G with an expression prediction loss, and shares its weights with the discriminator to encourage realistic expressions. Similar to CycleGAN, GANimation uses a cycle consistency loss, which eliminates the need for image pairing.

Instead of relying on AU estimations, the authors of Ref. [134] propose GANnotation, which uses facial landmark images. Doing so enables the network to learn facial structure directly from the input but is more susceptible to identity leakage compared to AUs that are normalized. GANotation generates x_g based on (x_t, l_s), where l_s is the facial landmarks of x_s. The model uses the same self attention model as GANimation, but proposes a novel "triple consistency loss" to minimize artifacts in x_g. The loss teaches the network how to deal with intermediate poses/expressions not found in the training set. Given l_s, l_t and l_z sampled randomly from the same video, the loss is computed as

$$\mathcal{L}_{trip} = \|G(x_t, l_s) - G(G(x_t, l_z), l_s)\|^2 \tag{4}$$

3D Parametric Approaches. Concurrent to the work of Ref. [83], other works also leveraged 3D parametric facial models to prevent identity leakage in the generation process. In Ref. [140], the authors propose FaceID-GAN, which can reenact t at oblique poses and high resolution. Their ED generator is trained in tandem with a 3DMM face model predictor, where the model parameters of x_t are used to transform x_s before being joined with the encoder's embedding. Furthermore, to prevent identity leakage from x_s to x_g, FaceID-GAN incorporates an identification classifier within the adversarial game. The classifier has $2N$ outputs where the first N outputs (corresponding to training set identities) are activated if the input is real and the rest are activated if it's fake.

Later, the authors of Ref. [140] proposed FaceFeat-GAN, which improves the diversty of the faces while preserving the identity [141]. The approach is to use a set of GANs to learn facial feature distributions as encodings, and then use these generators to create new content with a decoder. Concretely, three encoder/predictor neural networks P, Q, and I, are trained on real images to extract feature vectors from portraits. P predicts 3DMM parameters p, Q encodes the image as q capturing general facial features using feedback from I, and I is an identity classifier trained to predict label y_i. The next two GANs, seeded with noise vectors, produce p' and q', while a third GAN is trained to reconstruct x_t from (p, q, y_i) and x_g from (p', q', y_i). To reenact x_t, (1) y_t is predicted using I (even if the identity was previously unseen), (2) z_p and z_q are selected empirically to fit x_s, and (3) the third GAN's generator uses (p', q', y_t) to create x_g. Although FaceFeat-GAN improves image diversty, it is less practical than FaceID-GAN since the GAN's input seed z can be selected empirically to fit x_s.

In Ref. [112], the authors present paGAN, a method for complete facial reenactment of a 3D avatar, using a single image of the target as input. An expression neutral image of x_t is used to generate a 3D model, which is then driven by x_s. The driven model is used to create inputs for a U-Net generator: the rendered head, its UV map, its depth map, a masked image of x_t for texture, and a 2D mask indicating the gaze of x_s. Although paGAN is very efficient, the final deepfake is 3D rendered, which detracts from the realism.

Using Multi-Modal Sources. In Ref. [171], the authors propose X2Face, which can reenact x_t with x_s or some other modality such as audio or a pose vector. X2Face uses two ED networks: an embedding network and a driving network. First, the embedding network encodes 1–3 examples of the target's face to v_t: the optical flow field required to transform x_t to a neutral pose and expression. Next, x_t is interpolated according to m_t producing $x_t^{'}$. Finally, the driving network maps x_s to the vector map v_s, crafted to interpolate $x_t^{'}$ to x_g, having the pose and expression of x_s.

During training, first $\mathcal{L}_1$ loss is used between x_t and x_g, and then an identity loss is used between x_s and x_g using a pre-trained identity model trained on the VGG-Face Dataset. All interpolation is performed with a tensorflow interpolation layer to enable back propagation using $x_t^{'}$ and x_g. The authors also show how the embedding of driving network can be mapped to other modalities such as audio and pose.

In 2019, nearly all works pursued identity agnostic models:

Facial Landmark and Boundary Conversion. In Ref. [184], the authors propose FaceSwapNet, which tries to mitigate the issue of identity leakage from facial landmarks. First, two encoders and a decoder are used to transfer the expression in landmark l_s to the face structure of l_t, denoted l_g. Then, a generator network is used to convert x_t to x_g where l_g is injected into the network with AdaIn layers like a Style-GAN. The authors found that it is crucial to use triplet perceptual loss with an external VGG network.

In Ref. [56], the authors propose a method for high-resolution reenactment and at oblique angles. A set of networks encode the source's pose, expression, and the target's facial boundary for a decoder that generates the reenacted boundary b_g. Finally, an ED network generates x_g using an encoding of x_t's texture in its embedding. A multi-scale loss is used to improve quality, and the authors utilize a small labeled dataset by training their model in a semi-supervised way.

In Ref. [120], the authors present FSGAN: a face swapping and facial reenactment model that can handle occlusions. For reenactment, a pix2pixHD generator receives x_t and the source's 3D facial landmarks l_s, represented as a 256x256x70 image (one channel for each of the 70 landmarks). The output is x_g and its segmentation map m_g with three channels (background, face, and hair). The generator is trained recurrently where each output is passed back as input for several iterations, while l_s is interpolated incrementally from l_s to l_t. To improve results further, Delaunay Triangulation and barycentric coordinate interpolation are used to generate content similar to the target's pose. In contrast to other facial conversion methods [56, 184], FSGAN uses fewer neural networks enabling real-time reenactment at 30fps.

Latent Space Manipulation. In Ref. [159], the authors present a model called ICFace where the expression, pose, mouth, eye, and eyebrows of x_t can be driven independently. Their architecture is similar to a CycleGAN in that one generator translates x_t into a neutral expression domain as x_t^{η} and another generator translates x_t^{η} into an expression domain as x_g. Both generators ar conditioned on the target AU.

In Ref. [48], the authors propose an Additive Focal Variational Auto-encoder (AF-VAE) for high-quality reenactment. This is accomplished by separating a CVAE's latent code into an appearance encoding e_a and identity-agnostic expression coding e_x. To capture a wide variety of factors in e_a (e.g., age, illumination, complexion), the authors use an additive memory module during training, which conditions the latent variables on a Gaussian mixture model, fitted to a clustered set of facial boundaries. Subpixel convolutions were used in the decoder to mitigate artifacts and improve fidelity.

Warp-based Approaches. In the past, facial reenactment was done by warping the image x_t to the landmarks l_s [13]. In Ref. [60], the authors propose wgGAN, which uses the same approach but creates high-fidelity facial expressions by refining the image though a series of GANs: one for refining the warped face and another for in-painting the occlusions (eyes and mouth). A challenge with wgGAN is that the warping process is sensitive to head motion (change in pose).

In Ref. [185], the authors propose a system that can also control the gaze: a decoder generates x_g with an encoding of x_t as the input and a segmentation map of x_s as reenactment guidance via SPADE residual blocks. The authors blend x_g with a warped version, guided by the segmentation, to mitigate artifacts in the background.

To overcome the issue of occlusions in the eyes and mouth, the authors of Ref. [62] use multiple images of t as a reference, in contrast to Refs [60] and [185], which only use one. In their approach (FLNet), the model is provided with N samples of t (X_t) having various mouth expressions, along with the landmark deltas between X_t and x_s (L_t). Their model is an ED (configured like GANimation [127]), which produces (1) N encodings for a warped x_g, (2) an appearance encoding, and (3) a selection (weight) encoding. The encodings are then coverted into images using separate CNN layers and merged together through masked multiplication. The entire model is trained end-to-end in a self-supervised manner using frames of t taken from different videos.

Motion-Content Disentanglement. In Ref. [124], the authors propose a GAN to reenact neutral expression faces with smooth animations. The authors describe the animations as temporal curves in 2D space, summarized as points on a spherical manifold by calculating their square-root velocity function (SRVF). A WGAN is used to complete this distribution given target expression labels, and a pix2pix GAN is used to convert the sequences of reconstructed landmarks into video frames of the target.

In contrast to MoCoGAN [161], the authors of Ref. [170] propose ImaGINator: a conditional GAN that fuses both motion and content and uses them with transposed 3D convolutions to capture the distinct spatio-temporal relationships. The GAN also uses a temporal discriminator, and to increase diversity, the authors train the temporal discriminator with some videos using the wrong label.

A challenge with works such as Refs [124] and [170] is that they are label driven and produce videos with a set number of frames. This makes the deepfake creation process manual and less practical. In contrast, the authors of Ref. [143] propose Monkey-Net: a self supervised network for driving an image with an arbitrary video sequence. Similar to MoCoGAN [161], the authors decouple the source's content and motion. First a series of networks produce a motion heat map (optical flow) using the source and target's key-points, and then an ED generator produces x_g using x_s and the optical flow (in its embedding).

Later, in Ref. [144], the authors extend Monkey-Net by improving the object appearance when large pose transformations occur. They accomplish this by (1) modeling motion around the key-points using affine transformations, (2) updating the key-point loss function accordingly, and (3) having the motion generator predict an occlusion mask on the preceding frame for in-painting inference. Their work has been implemented as a free real-time reenactment tool for video chats, called Avitarify.[6]

4.1.4 Few-Shot Learning. Toward the end of 2019 and into the beginning of 2020, researchers began looking into minimizing the amount of training data further via one-shot and few-shot learning.

In Ref. [182], the authors propose a few-shot model that works well at oblique angles. To accomplish this, the authors perform meta-transfer learning, where the network is first trained on many different identities and then fine-tuned on the target's identity. Then, an identity encoding of x_t is obtained by averaging the encodings of k sets of (x_t, l_t). Then, a pix2pix GAN is used to generate x_g using l_s as an input, and the identity encoding via AdaIN layers. Unfortunately, the authors note that their method is sensitive to identity leakage.

In Ref. [167], the authors of Vid2Vid (Section 4.1.2) extend their work with few-shot learning. They use a network weight generation module, which utilizes an attention mechanism. The module learns to extract appearance patterns from a few samples of x_t, which are injected into the video synthesis layers. In contrast to FLNet, Refs [62], [182], and [167] merge the multiple

[6] https://github.com/alievk/avatarify.

representations of t *before* passing it through the generator. This approach is more efficient because it involves fewer passes through the model's networks.

In Ref. [65], the authors propose MarioNETte, which alleviates identity leakage when the pose of x_s is different than x_t. In contrast to other works that encode the identity separately or use of AdaIN layers, the authors use an image attention block and target feature alignment. This enables the model to better handle the differences between face structures. Finally, the identity is also preserved using a novel landmark transformer inspired by Ref. [21].

4.2 Mouth Reenactment (Dubbing)

In contrast to expression reenactment, mouth reenactment (a.k.a. video or image dubbing) is concerned with driving a target's mouth with a segment of audio. Figure 9 presents the relevant schematics for this section.

4.2.1 *Many-to-One (Multiple Identities to a Single Identity).*

Obama Puppetry. In 2017, the authors of Ref. [151] created a realistic reenactment of former president Obama. This was accomplished by (1) using a time delayed RNN over Mel-frequency cepstral coefficient (MFCC) audio segments to generate a sequence of mouth landmarks (shapes), (2) generating the mouth textures (nose and mouth) by applying a weighted median to images with similar mouth shapes via PCA-space similarity, (3) refining the teeth by transferring the high-frequency details other frames in the target video, and (4) by using dynamic programming to re-time the target video to match the source audio and blend in the texture.

Later that year, the authors of Ref. [89] presented ObamaNet: a network that reenacts an individual's mouth and voice using text as input instead of audio, like in Ref. [151]. The process is to (1) convert the source text to audio using Char2Wav [147], (2) generate a sequence of mouth-keypoints using a time-delayed LSTM on the audio, and (3) use a U-Net CNN to perform in-painting on a composite of the target video frame with a masked mouth and overlayed keypoints.

Later in 2018, Jalalifar et al. [73] proposed a network that synthesizes the entire head portrait of Obama, and therefore does not require pose re-timing and can be trained end-to-end, unlike Refs [151] and [89]. First, a bidirectional LSTM coverts MFCC audio segments into sequence of mouth landmarks, and then a pix2pix-like network generates frames using the landmarks and a noise signal. After training, the pix2pix network is fine-tuned using a single video of the target to ensure consistent textures.

3D Parametric Approaches. Later on in 2019, the authors of Ref. [55] proposed a method for editing a transcript of a talking heads which, in turn, modifies the target's mouth and speech accordingly. The approach is to (1) align phenomes to a_s, (2) fit a 3D parametric head model to each frame of X_t like Ref. [83], (3) blend matching phenomes to create any new audio content, (4) animate the head model with the respective frames used during the blending process, and (5) generate X_g with a CGAN RNN using composites as inputs (rendered mouths placed over the original frame).

The authors of Ref. [153] had a different approach: (1) animate the reconstructed 3D head with the predicted blend shape parameters from a_s using a DeepSpeech model for feature extraction, (2) use Deferred Neural Rendering [154] to generate the mouth region, and then (3) use a network to blend the mouth into the original frame. Compared to previous works, the authors found that their approach only requires 2–3 minutes of video while producing very realistic results. This is because neural rendering can summarize textures with a high fidelity and operate on UV maps, mitigating artifacts on how the textures are mapped to the face.

4.2.2 *Many-to-Many (Multiple IDs to Multiple IDs).*

One of the first works to perform identity agnostic video dubbing was Ref. [142]. There, the authors used an LSTM to map MFCC audio

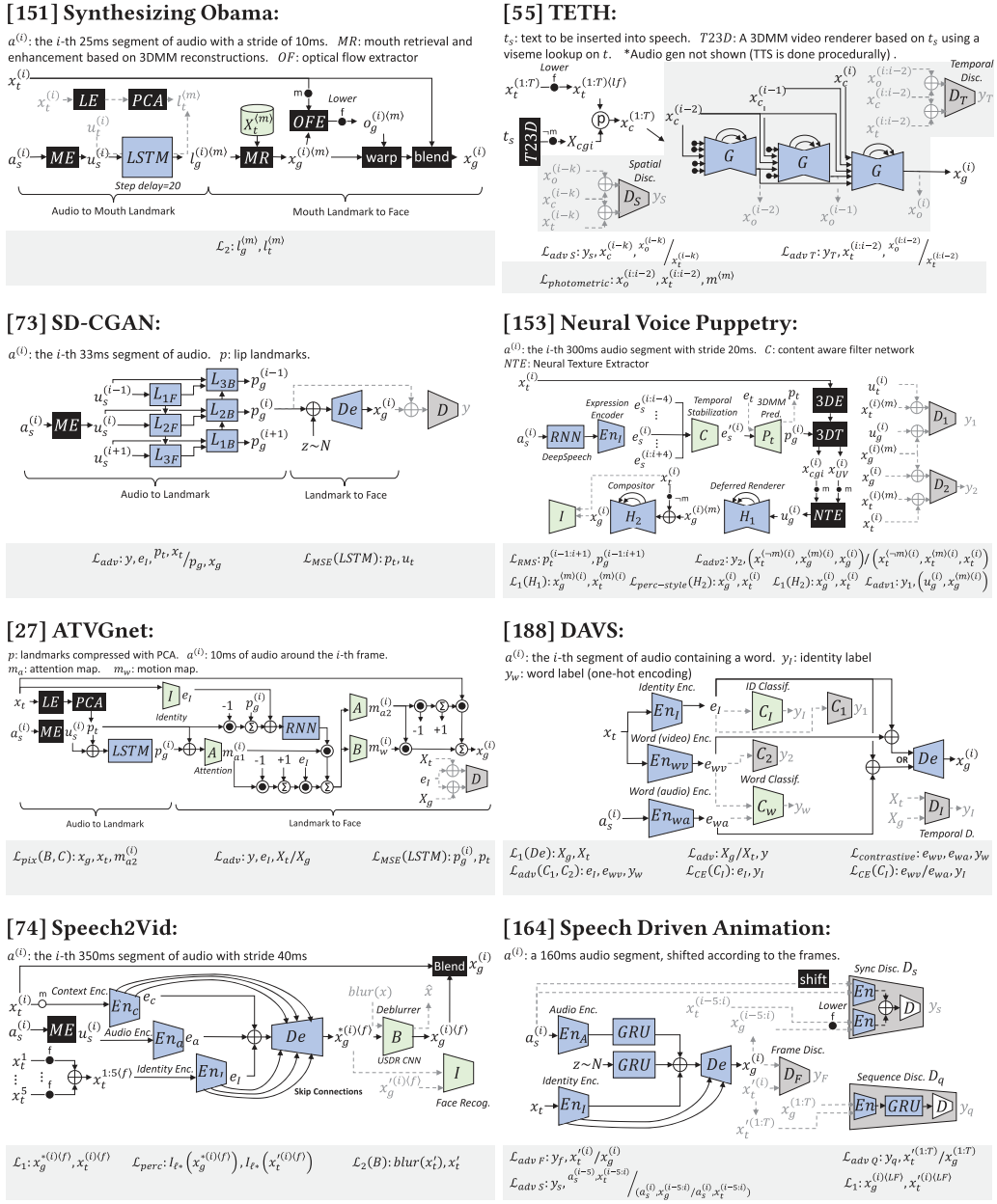


Fig. 9. Architectural schematics for some *mouth reenactment networks*. Black lines indicate prediction flows used during deployment; dashed gray lines indicate dataflows performed during training.

segments to the face shape. The face shapes were represented as the coefficients of an active appearance model (AAM), which were then used to retrieve the correct face shape of the target.

Improvements in Lip-Sync. Noting a human's sensitivity to temporal coherence, the authors of Ref. [146] use a GAN with three discriminators: on the frames, video, and lip-sync. Frames

are generated by (1) encoding each MFCC audio segment $a_s^{(i)}$ and x_t with separate encoders, (2) passing the encodings through an RNN, and (3) decoding the outputs as $x_g^{(i)}$ using a decoder.

In Ref. [178], the authors try to improve the lipsyncing with a textual context. A time-delayed LSTM is used to predict mouth landmarks given MFCC segments and the spoken text using a text-to-speech model. The target frames are then converted into sketches using an edge filter, and the predicted mouth shapes are composited into them. Finally, a pix2pix-like GAN with self-attention is used to generate the frames with both video and image conditional discriminators.

Compared to direct models such as the direct models of Refs [146] and [178], the authors of Ref. [27] improve the lip-syncing by preventing the model from learning irrelevant correlations between the audiovisual signal and the speech content. This was accomplished with LSTM audio-to-landmark network and a landmark-to-identity CNN-RNN used in sequence. There, the facial landmarks are compressed with PCA, and the attention mechanism from Ref. [127] is used to help focus the model on the relevant patterns. To improve synchronization further, the authors proposed a regression-based discriminator, which considers both sequence and content information.

EDs for Preventing Identity Leakage. The authors in Ref. [188] mitigate identity leakage by disentangling the speech and identity latent spaces using adversarial classifiers. Since their speech encoder is trained to project audio and video into the same latent space, the authors show how x_g can be driven using x_s or a_s.

In Ref. [74], the authors propose Speech2Vid, which also uses separate encoders for audio and identity. However, to capture the identity better, the identity encoder En_I uses a concatenation of five images of the target, and there are skip connections from the En_I to the decoder. To blend the mouth in better, a third "context" encoder is used to encourage in-painting. Finally, a VDSR CNN is applied to x_g to sharpen the image.

A disadvantage with Refs [188] and [74] is that they cannot control facial expressions and blinking. To resolve this, the authors in Ref. [163] generate frames with a stride transposed CNN decoder on GRU-generated noise, in addition to the audio and identity encodings. Their video discriminator uses two RNNs for both the audio and video. When applying the L1 loss, the authors focus on the lower half of the face to encourage better lip-sync quality over facial expressions.

Later, in Ref. [164], the same authors improve the temporal coherence by splitting the video discriminator into two: (1) for temporal realism in mouth to audio synchronization and (2) for temporal realism in overall facial expressions. Then, in Ref. [79], the authors tune their approach further by fusing the encodings (audio, identity, and noise) with a polynomial fusion layer as opposed to simply concatenating the encodings together. Doing so makes the network less sensitive to large facial motions compared to Refs [164] and [74].

4.3 Pose Reenactment

Most deep learning works in this domain focus on the problem of face frontalization. However, there are some works that focus on facial pose reenactment.

In Ref. [70], the authors use a U-Net to convert (x_t, l_t, l_s) into x_g using a GAN with two discriminators: one conditioned with the neutral pose image, and the other conditioned with the landmarks. In Ref. [158], the authors propose DR-GAN for pose-invariant face recognition. To adjust the pose of x_t, the authors use an ED GAN, which encodes x_t as e_t and then decodes (e_t, p_s, z) as x_g, where p_s is the source's pose vector, and z is a noise vector. Compared to Ref. [70], Ref. [158] has the flexibility of manipulating the encodings for different tasks and the authors improve the quality of x_g by averaging multiple examples of the identity encoding before passing it through the decoder (similar to Refs [62], [167], and [182]). In Ref. [23], the authors suggest using two GANs: The first frontalizes the face and produces a UV map, and second rotates the face, given the target

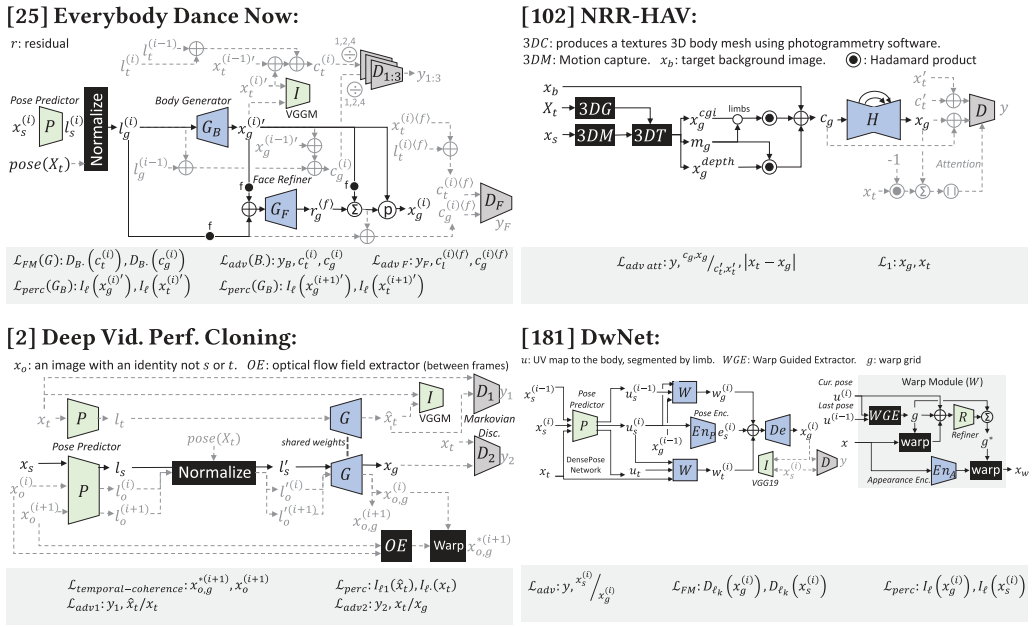


Fig. 10. Architectural schematics for some *body reenactment networks*. Black lines indicate prediction flows used during deployment; dashed gray lines indicate dataflows performed during training.

angle as an injected embedding. The result is that each model performs a less complex operation, and the models can, therefore, collectively produce a higher quality image.

4.4 Gaze Reenactment

There are only a few deep learning works that have focused on gaze reenactment. In Ref. [57], the authors convert a cropped eye x_t, its landmarks, and the source angle, to a flow (vector) field using a two-scale CNN. x_g is then generated by applying a flow field to x_t to warping it to the source angle. The authors then correct the illumination of x_g with a second CNN. A challenge with Ref. [57] is that the head must be frontal to avoid inconsistencies due to pose and perspective. To mitigate this issue, the authors of Ref. [180] proposed the Gaze Redirection Network (GRN). In GRN, the target's cropped eye, head pose, and source angle are encoded separately and then passed though an ED network to generate an optical flow field. The field is used to warp x_t into x_g. To overcome the lack of training data and the challenge of data pairing, the authors (1) pre-train their network on 3D synthesized examples, (2) further tune their network on real images, and then (3) fine tune their network on 3–10 examples of the target.

4.5 Body Reenactment

Several facial reenactment papers from Section 4.1 discuss body reenactment, too, for example, Vid2Vid [167, 168], MocoGAN [161], and others [143, 144]. In this section, we focus on methods that specifically target body reenactment. Schematics for some of these architectures can be found in Figure 10.

4.5.1 One-to-One (Identity-to-Identity). In the work of Ref. [104], the authors perform facial reenactment with the upper-body as well (arms and hands). The approach is to (1) use a pix2pixHD GAN to convert the source's facial boundaries to the targets, (2) then paste them onto a captured pose skeleton of the source and (3) use a pix2pixHD GAN to generate x_g from the composite.

4.5.2 *Many-to-One (Multiple Identities to a Single Identity).*

Dance Reenactment. In Ref. [25], the authors make people dance using a target specific pix2pixHD GAN with a custom loss function. The generator receives an image of the captured pose skeleton and the discriminator receives the current and last image conditioned on their poses. The quality of face is then improved with a residual predicted by an additional pix2pixHD GAN, given the face region of the pose. A many-to-one relationship is achieved by normalizing the input pose to that of the target's.

The authors of Ref. [102] then tried to overcome artifacts that occur in Ref. [25] such as stretched limbs due to incorrectly detected pose skeletons. They used photogrammetry software on hundreds of images of the target and then reenacted the 3D rendering of the target's body. The rendering, partitioned depth map, and background are then passed to a pix2pix model for image generation using an attention loss.

Another artifact in Ref. [25] was that the model could not generalize well to unseen poses. To improve the generalization, the authors of Ref. [2] trained their network on many identities other than s and t. First, they trained the GAN on paired data (the same identity doing different poses) and then later added another discriminator to evaluate the temporal coherence given (1) $x_g^{(i)}$ driven by another video and (2) the optical flow predicted version.

A challenge with the previous works was that they required lots of training data. This was reduced from about an hour of video footage to only 3 minutes in Ref. [190] by segmenting and orienting the limbs of x_t according to x_s before the generation step. Then, a pix2pixHD GAN used this composition and the last k frames' poses to generate the body. Finally, another pix2pixHD GAN is used to blend the body into the background.

4.5.3 *Many-to-Many (Multiple IDs to Multiple IDs).*

Pose Alignment. In Ref. [145] the authors try to resolve the issue of misalignment when using pix2pix like architectures. They propose "deformable skip connections," which help orient the shuttled feature maps according to the source pose. The authors also propose a novel nearest neighbor loss instead of using L1 or L2 losses. To modify unseen identities at test time, an encoding of x_t is passed to the decoder's inner layers.

Although the work of Ref. [145] helps align the general images, artifacts can still occur when x_s and x_t have very different poses. To resolve this, the authors of Ref. [192] use novel Pose-Attentional Transfer blocks (PATB) inside their GAN-based generator. The architecture passes x_t and the poses p_s concatenated with p_t through separate encoders that are passed though a series of PATBs before being decoded. The PATBs progressively transfer regional information of the poses to regions of the image to ultimately create a body that has better shape and appearance consistency.

Pose Warping. In Ref. [116], the authors use a pre-trained DensePose network [9] to refine a predicted pose with a warped and in-painted DensePose UV spatial map of the target. Since the spatial map covers all surfaces of the body, the generated image has improved texture consistency. In contrast to Refs [145] and [192], which use feature mappings to alleviate misalignment, the authors of Ref. [181] use warping, which reduces the complexity of the network's task. Their model, called DwNet, uses a "warp module" in an ED network to encode $x_t^{(i-1)}$ warped to $p_s^{(i)}$, where p is a UV body map of a pose obtained as a DensePose network.

A challenge with the alignment techniques of the previous works is that the body's 3D shape and limb scales are not considered by the network, resulting in identity leakage from x_s. In Ref. [103], the authors counter this issue with their Liquid Warping GAN. This is accomplished by predicting target and source's 3D bodies with the model in Ref. [77] and then by translating the

Table 2. Summary of Deep Learning Replacement Models

				Replacement	Retraining for new...		Model					Repr.		Model Training					Model Execution		Model Outp.
				Transfer Swap	Source (s)	Target (t)	Identity Agnostic	Encoders	Decoders	Discriminators	Other Netw. 3DMM/Rendering	Segmentation	Landmark / Keypoint	Labeling of: ID	Labeling of: Other	No Pairing	Paring within Same Video	Paring ID to Diff. Actions	Requires Video	Source (x_s...)	Target (x_t...)	Resolution
One-to-One	[1]	2017	Deepfakes for All	•	2k-5k portraits			1	2	0-1	0			•		•			•	-	portrait	256x256
	[138]	2018	FaceSwap-GAN	•	2k-5k portraits			1	2	2	1			•		•			•	-	portrait	256x256
	[71]	2018	DeepFaceLab	•	2k-5k portraits			1	2	0-1	0			•		•			•	-	portrait	256x256
One-to-Many	[88]	2017	Fast Face Swap	•	60 portraits	None		0	0	0	2	•	•	•		•				portrait	portrait	256x256
Many-to-Many	[114]	2018	RSGAN	•	None	None		4	3	2	1	•				•				portrait	portrait	128x128
	[113]	2018	FSNet	•	None	None		3	4	5	0		•	•	•	•				portrait	portrait	128x128
	[17]	2018	OSIP-FS	•	None	None		2	1	2	0			•		•				portrait	portrait	128x128
	[111]	2018	DepthNets	•	None	None	•	3	2	2	1		•	•				•		portrait	portrait	80x80
	[120]	2018	FSGAN		None	None	•	4	4	3	1	•	•	•				•	•	portrait	portrait	256x256
	[150]	2018	IO-FR	•	None	None	•	1	1	1	1	•		•		•				portrait	portrait	256x256
	[139]	2019	FS Face Trans.	•	None	None	•	1	1	2	2	•		•		•				portraits	portrait	128x128
	[174]	2019	IHPT	•	None	None	•	2	1	2	0				•		•		•	cropped	cropped	128x128
	[93]	2019	FaceShifter	•	None	None	•	3	3	3	0			•		•				portrait	portrait	256x256

two through a novel liquid warping block (LWB) in their generator. Specifically, the estimated UV maps of x_s and x_t, along with their calculated transformation flow, are passed through a three-stream generator, which produces (1) the background via in-painting, (2) a reconstruction of the x_s and its mask for feature mapping, and (3) the reenacted foreground and its mask. The latter two streams use a shared LWB to help the networks address multiple sources (appearance, pose, and identity). The final image is obtained through masked multiplication, and the system is trained end-to-end.

Background Foreground Compositing. In Ref. [14], the authors break the process down into three stages, trained end-to-end: (1) use a U-Net to segment x_t's body parts and then orient them according to the source pose p_s, (2) use a second U-Net to generate the body x_g from the composite, and (3) use a third U-Net to perform in-painting on the background and paste x_g into it. The authors of Ref. [46] then streamlined this process by using a single ED GAN network to disentangle the foreground appearance (body), background appearance, and pose. Furthermore, by using an ED network, the user gains control over each of these aspects. This is accomplished by segmenting each of these aspects before passing them through encoders. To improve the control over the compositing, the authors of Ref. [35] used a CVAE-GAN. This enabled the authors to change the pose and appearance of bodies individually. The approach was to condition the network on heatmaps of the predicted pose and skeleton.

4.5.4 Few-Shot Learning. In Ref. [91], the authors demonstrate the few-shot learning technique of Ref. [53] on a pix2pixHD network and the network of Ref. [14]. Using just a few sample images, they were able to transfer the resemblance of a target to new videos in the wild.

5 REPLACEMENT

The network schematics and summary of works for replacement deepfakes can be found in Figure 12 and Table 2, respectively.

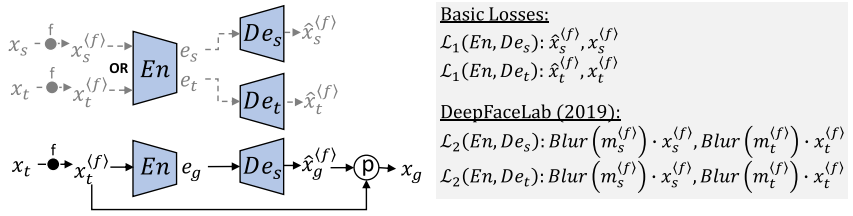


Fig. 11. The basic schematic for the Reddit "deepfakes" model and its variants [1, 71, 138].

5.1 Swap

At first, face swapping was a manual process accomplished using tools such as Photoshop. More automated systems first appeared between 2004–08 in Refs [20] and [18]. Later, fully automated methods were proposed in Refs [34], [80], [121], and [162] using methods such as warping and reconstructed 3D morphable face models.

5.1.1 One-to-One (Identity-to-Identity).

Online Communities. After the Reddit user "deepfakes" was exposed in the media, researchers and online communities began finding improved ways to perform face swapping with deep neural networks. The original deepfake network, published by the Reddit user, is an ED network (visualized in Figure 11). The architecture consists of one encoder En and two decoders De_s and De_t. The components are trained concurrently as two autoencoders: $De_s(En(x_s)) = \hat{x}_s$ and $De_t(En(x_t)) = \hat{x}_t$, where x is a cropped face image. As a result, En learns to map s and t to a shared latent space, such that

$$De_s(En(x_t)) = x_g \tag{5}$$

Currently, there are a number of open source face swapping tools on GitHub based on the original network. One of the most popular is DeepFaceLab [71]. Their current version offers a wide variety of model configurations, including adversarial training, residual blocks, a style transfer loss, and masked loss to improve the quality of the face and eyes. To help the network map the target's identity into arbitrary face shapes, the training set is augmented with random face warps.

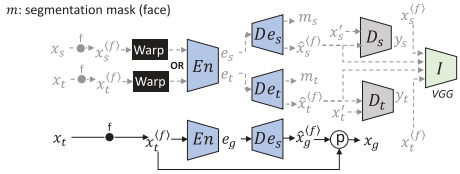
Another tool called FaceSwap-GAN [138] follows a similar architecture but uses a denoising autoencoder with self-attention mechanisms, and offers cycle-consistency loss, which can reduce the identity leakage and increase the image fidelity. The decoders in FaceSwap-GAN also generate segmentation masks, which helps the model handle occlusions and is used to blend x_g back into the target frame. Finally, Ref. [1] is another open source tool that provides a graphical user interface (GUI). Their software comes with 10 popular implementations, including that of Ref. [71], and multiple variations of the original Redit user's code.

5.1.2 One-to-Many (Single Identity to Multiple Identities). In Ref. [88], the authors use a modified style transfer with CNN, where the content is x_t and the style is the identity of x_s. The process is (1) align x_t to a reference x_s, (2) transfer the identity of s to the image using a multi scale CNN, trained with style loss on images of s, and (3) align the output to x_t and blend the face back in with a segmentation mask.

5.1.3 Many-to-Many (Multiple IDs to Multiple IDs). One of the first identity agnostic methods was Ref. [123], mentioned in Section 4.1.3. However, to train this CGAN, one needs a dataset of paired faces with different identities having the same expression.

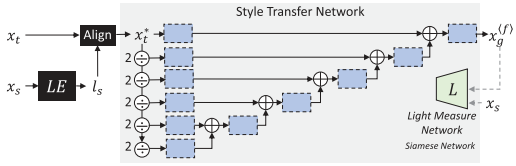
Disentanglement with EDs. However, to provide more control, the authors in Ref. [17] use an ED to disentangle the identity from the attributes (pose, hair, background, and lighting) during

[138] Face Swap GAN:



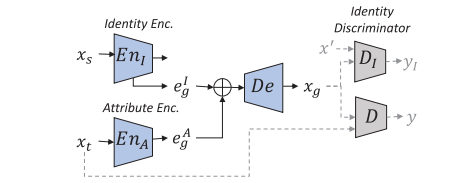
$\mathcal{L}_{tv}(s): x_s^{\langle f \rangle}$ $\mathcal{L}_{tv}(t): x_t^{\langle f \rangle}$ $\mathcal{L}_{MAE}: x_s^{\langle e \rangle}, x_t^{\langle e \rangle}$ $\mathcal{L}_{adv}(s): y_s, x_s'/\hat{x}_s^{\langle f \rangle}$
$\mathcal{L}_{adv}(t): y_t, x_t'/\hat{x}_t^{\langle f \rangle}$ $\mathcal{L}_1(s): \hat{x}_s^{\langle f \rangle}, x_s^{\langle f \rangle}$ $\mathcal{L}_1(t): \hat{x}_t^{\langle f \rangle}, x_t^{\langle f \rangle}$ $\mathcal{L}_{perc}(s): l_\ell.\left(\hat{x}_s^{\langle f \rangle}\right), l_\ell.\left(x_s^{\langle f \rangle}\right)$
$\mathcal{L}_{perc}(t): l_\ell.\left(\hat{x}_t^{\langle f \rangle}\right), l_\ell.\left(x_t^{\langle f \rangle}\right)$ $\mathcal{L}_{cyc}(s): G_{ts}(G_{st}(x_s)), x_s$ $\mathcal{L}_{cyc}(t): G_{st}(G_{ts}(x_t)), x_t$

[88] Fast Face Swap:



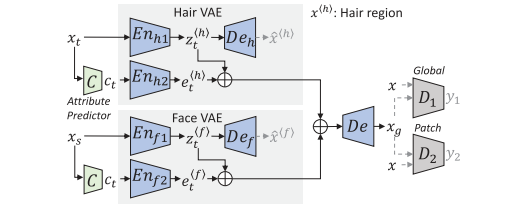
$\mathcal{L}_{perc}: L\left(x_g^{\langle f \rangle}\right), L(x_s)$ $\mathcal{L}_{style}: x_g^{\langle f \rangle}, X_t$
$\mathcal{L}_{FM}: L_\ell.\left(x_g^{\langle f \rangle}\right), L_\ell.\left(x_s^{\langle f \rangle}\right)$ $\mathcal{L}_{tv}: x_g^{\langle f \rangle}$

[17] OSIP-FS:



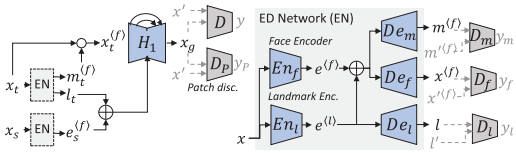
$\mathcal{L}_{CE}(En_I): y_I, x$ $\mathcal{L}_2(En_A): x_g, x_t$ -weighted $\mathcal{L}_{KL}: e^A$
$\mathcal{L}_{adv}(D): y, x_t/x_g$ $\mathcal{L}_{FM}(De): D_{I_\ell}(x_g), D_{I_\ell}(x_t)$ $\mathcal{L}_{CE}(D_I): y_I, x'/x_g$
$\mathcal{L}_{FM}(De): D_{I_\ell}(x_g), D_{I_\ell}(x')$

[114] RSGAN:



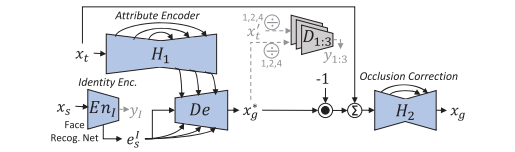
$\mathcal{L}_1: x^{\langle f \rangle}, \hat{x}^{\langle f \rangle}$ $\mathcal{L}_1: x^{\langle h \rangle}, \hat{x}^{\langle h \rangle}$ $\mathcal{L}_1: x, \hat{x}$ $\mathcal{L}_{KL}: x, c$
$\mathcal{L}_{adv\,1}: y, x/x_g$ $\mathcal{L}_{adv\,2}: y, x/x_g$ $\mathcal{L}_{CE}(C): c, y_c$

[113] FSNet:



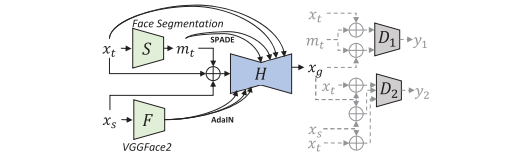
VAE Objectives: $\mathcal{L}_{CE}(ED): m^{\langle f \rangle}, \hat{m}^{\langle f \rangle}$ $\mathcal{L}_{CE}(ED): l, \hat{l}$ $\mathcal{L}_1(ED): x^{\langle f \rangle}, \hat{x}^{\langle f \rangle}$ $\mathcal{L}_1(ED): x^{\langle f \rangle} \cdot m^{\langle f \rangle}, \hat{x}^{\langle f \rangle} \cdot m^{\langle f \rangle}$
$\mathcal{L}_{adv}(ED): y_m, m^{\langle f \rangle}/m'^{\langle f \rangle}$ $\mathcal{L}_{adv}(ED): y_f, x^{\langle f \rangle}/x'^{\langle f \rangle}$ $\mathcal{L}_{adv}(ED): y_l, l/l'^{\langle f \rangle}$ $\mathcal{L}_{adv}: y, x'/x_g$
$\mathcal{L}_{adv,P}: y_P, x'/x_g$ $\mathcal{L}_{trip}(ED): (x_{t1}, x_{t2}, x_s), (x_{g1}, x_{t1}, x_s), (x_{g2}, x_{t2}, x_s)$

[93] FaceShifter:



$\mathcal{L}_{adv1}: y_1, x/x_g^*$ $\mathcal{L}_{adv2}: y_2, \quad x/x_g^*$ $\mathcal{L}_{adv3}: y_3, \quad x/x_g^*$
$\mathcal{L}_{cosine}(De, H_1): e_s^I, e_g^I$ $\mathcal{L}_{perc}(De, H_1): H_{1\,\ell}.(x_t), H_{1\,\ell}.(x_g^*)$ $\mathcal{L}_2(De, H_1): x_g, x_t$ -if s=t
$\mathcal{L}_{cosine}(H_2): e_s^I, e_g^I$ $\mathcal{L}_1(H_2): x_g, x_t - x_g^*$ $\mathcal{L}_2(H_2): x_g, x_t$ -if s=t

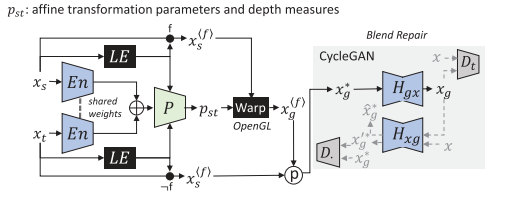
[139] Few-Shot Face Translation:



Losses unknown: Training code was not released as of writing

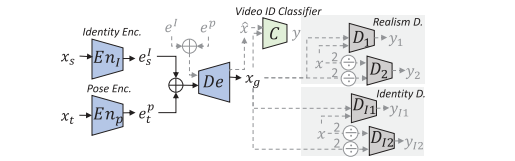
[111] Depth Nets:

p_{st}: affine transformation parameters and depth measures



$\mathcal{L}_{custom}(En, P): p_{st}$ $\mathcal{L}_{cyc}(H_{gx}, H_{xg}): x_g^*, x$ $\mathcal{L}_{adv}(D_t): y_t, x/x_g$ $\mathcal{L}_{adv}(D_\cdot): y_t, x_g^{**}/x_g^*$

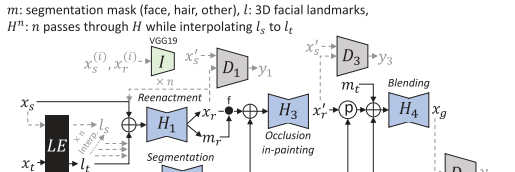
[174] IHPT:



$\mathcal{L}_1: x_g, x_t'$ $\mathcal{L}_1: e^I, \hat{e}^I$ $\mathcal{L}_1: e^P, \hat{e}^P$ $\mathcal{L}_{pixel-verif}: x_g, x_t', x_s$ $\mathcal{L}_{CE}(De, C): y, \hat{x}$
$\mathcal{L}_{adv\,1}: y_1, x/x_g$ $\mathcal{L}_{adv\,2}: y_2, 2\ominus x/x_g$ $\mathcal{L}_{adv\,I1}: y_{I1}, x/x_g$ $\mathcal{L}_{adv\,I2}: y_{I2}, 2\ominus x/x_g$

[120] FSGAN:

m: segmentation mask (face, hair, other), l: 3D facial landmarks, H^n: n passes through H while interpolating l_s to l_t



$\mathcal{L}_1(H_1): x_t^{\langle f \rangle}, x_r^{\langle f \rangle}$ $\mathcal{L}_{perc}(H_1): l_\ell.\left(x_t^{\langle f \rangle}\right), l_\ell.\left(x_r^{\langle f \rangle}\right)$ $\mathcal{L}_{adv1}: y_1, x_t^{\langle f \rangle}/x_r$ $\mathcal{L}_1(H_1): m_r, m_t$
$\mathcal{L}_1(H_2): m_r, H_{1,m}(x_t)$ $\mathcal{L}_1(H_3): x_r', x_t^{\langle f \rangle}$ $\mathcal{L}_{perc}(H_3): l_\ell.(x_r'), l_\ell.\left(x_t^{\langle f \rangle}\right)$
$\mathcal{L}_{adv3}: y_3, x_t'^{\langle f \rangle}$ $\mathcal{L}_{adv4}: y_4, x_g/x_t^{\langle f \rangle}$ $\mathcal{L}_{poisson-perc}(H_4): x_t, m_t, paste(x_r', x_t)$

Fig. 12. Architectural schematics of the *replacement networks* with their generation and training dataflows.

the training process. The identity encodings are the last pooling layer of a face classifier, and the attribute encoder is trained using a weighted L2 loss and a KL divergence loss to mitigate identity leakage. The authors also show that they can adjust attributes, expression, and pose via interpolation of the encodings. Instead of swapping identities, the authors of Ref. [150] wanted to *variably* obfuscate the target's identity. To accomplish this, the authors used an ED to predict the 3D head parameters, which were either modified or replaced with the source's. Finally, a GAN was used to in-paint the face of x_t given the modified head model parameters.

Disentanglement with VAEs. In Ref. [114], the authors propose RSGAN: a VAE-GAN consisting of two VAEs and a decoder. One VAE encodes the hair region and the other encodes the face region, where both are conditioned on a predicted attribute vector c describing x. Since VAEs are used, the facial attributes can be edited through c.

In contrast to Ref. [114], the authors of Ref. [113] use a VAE to prepare the content for the generator, and use a network to perform the blending via in-painting. A single VAE-ED network is run on x_s and then x_t, producing encodings for the face of x_s and the landmarks of x_t. To perform a face swap, a generator receives the masked portrait of x_t and performs in-painting on the masked face. The generator uses the landmark encodings in its embedding layer. During training, randomly generated faces are used with triplet loss on the encodings to preserve identities.

Face Occlusions. FSGAN [120], mentioned in Section 4.1.3, is also capable of face swapping and can handle occlusions. After the face reenactment generator produces x_r, a second network predicts the target's segmentation mask m_t. Then, $(x_r^{\langle f \rangle}, m_t)$ is passed to a third network that performs in-painting for occlusion correction. Finally, a fourth network blends the corrected face into x_t while considering ethnicity and lighting. Instead of using interpolation like Ref. [120], the authors of Ref. [93] propose FaceShifter, which uses novel Adaptive Attentional Denormalization (AAD) layers to transfer localized feature maps between the faces. In contrast to Ref. [120], FaceShifter reduces the number of operations by handling the occlusions through a refinement network trained to consider the delta between the original x_t and a reconstructed $\hat{x}_t$.

5.1.4 Few-Shot Learning. The same author of FaceSwap-GAN [138] also hosts few-shot approach online dubbed "One Model to Swap Them All" [139]. In this version, the generator receives $(x_s^{\langle f \rangle}, x_t^{\langle f \rangle}, m_t)$ where its encoder is conditioned on VGGFace2 features of x_t using FC-AdaIN layers, and its decoder is conditioned on x_t and the face structure m_t via layer concatenations and SPADE-ResBlocks, respectively. Two discriminators are used: one on image quality given the face segmentation and the other on the identities.

5.2 Transfer

Although face transfers precede face swaps, today there are very few works that use deep learning for this task. However, we note that a face transfer is equivalent to performing *self-reenactment* on a face swapped portrait. Therefore, high-quality face transfers can be achieved by combining a method from Section 4.1 and Section 5.1.

In 2018, the authors of Ref. [111] proposed DepthNets: an unsupervised network for capturing facial landmarks and translating the pose from one identity to another. The authors use a Siamese network to predict a transformation matrix that maps the x_s's 3D facial landmarks to the corresponding 2D landmarks of x_t. A 3D renderer (OpenGL) is then used to warp $x_s^{\langle f \rangle}$ to the source pose l_t, and the composition is refined using a CycleGAN. Since warping is involved, the approach is sensitive to occlusions.

Later in 2019, the authors of Ref. [174] proposed a self-supervised network that can change the identity of an object within an image. Their ED disentangles the identity from an objects pose using

Table 3. Summary of Deepfake Detection Models

			Type			Modality			Content			Method			Eval. Dataset								Performance*		
			Reenactment	Replacement	Image	Video	Audio	Feature	Body Part	Face Image	Model	Indicates Affected Area	Input Resolution	DeepfakeTIMIT [86]	DFFD [149]	FaceForensics [130]	FaceForensics++ [131]	FFW [82]	Celeb-DF [101]	Other Deepfake DB	Custom	ACC	EER	AUC
Classic ML	[186]	2017		●	●	●				●	SVM-RBF		250x250								●	92.9		
	[4]	2017		●	●	●				●	SVM		*							●			18.2	
	[177]	2018	●	●	●	●				●	SVM		*							●				0.97
	[86]	2018	●	●		●				●	SVM		128x128	●									3.33	
	[42]	2019		●	●					●	SVM, Kmeans...		1024x1024								●	100		
	[8]	2019	●	●	●	●				●	SVM		*	●									13.33	
	[6]	2019	●	●		●				●	SVM		*								●			0.98
Deep Learning	[110]	2018	●	●	●	●				●	CNN		256x256								●	99.4		
	[96]	2018	●	●				●	●		LSTM-CNN		224x224								●			0.99
	[118]	2018	●	●	●	●				●	Capsule-CNN		128x128			●						99.3		
	[17]	2018		●	●	●				●	ED-GAN		128x128								●	92		
	[39]	2018	●	●	●					●	CNN		1024x1024							●				0.81
	[63]	2018	●	●		●				●	CNN-LSTM		299x299								●	97.1		
	[105]	2018	●	●	●					●	CNN		256×256								●	94.4		
	[33]	2018	●	●	●	●				●	CNN AE		256x256			●					●	90.5		
	[3]	2018	●	●		●				●	CNN		256x256								●			0.99
	[131]	2019	●	●		●				●	CNN-LSTM		224x224				●					96.9		
	[117]	2019	●	●	●	●				●	CNN-DE	●	256x256			●	●					92.8	8.18	
	[38]	2019		●	●	●				●	CNN		-								●	98.5		
	[41]	2019	●	●	●	●				●	CNN AE GAN	●	256x256				●				●	99.2		
	[148]	2019	●	●	●	●				●	CNN+Attention	●	299x299		●								3.11	0.99
	[97]	2019	●	●	●	●				●	CNN		128x128							●	●			0.99
	[100]	2019		●		●				●	CNN		*						●					0.64
	[52]	2019	●	●	●	●				●	CNN+HMN		224x224			●	●	●				99.4		
	[92]	2019	●	●	●	●				●	FCN		256x256				●					98.1		
	[176]	2019	●	●	●	●				●	CNN		128x128								●	94.7		
	[160]	2019	●	●	●	●				●	CNN		224x224							●		86.4		
	[152]	2019	●	●	●					●	CNN		1024x1024								●			94
	[30]	2019	●	●		●				●	CNN		128x128			●					●	96		
	[98]	2019	●	●	●	●				●	CNN	●	224x224	●									93.2	
	[11]	2019	●	●		●				●	CNN		224x224				●					81.6		
	[?]	2019	●			●	●	●	●		LSTM		*								●		22	
	[47]	2019	●			●	●	●	●		LSTM-DNN		*								●		16.4	
	[25]	2019	●			●				●	CNN		256x256								●	97		
	[179]	2019	●	●	●				●	●	CNN		128x128			●		●			●	99.6	0.53	
	[165]	2019	●	●	●	●				●	SVM+VGGnet		224x224				●					85		
	[94]	2020	●	●	●	●			●		CNN	●	64x64		●		●		●	●				99.2
	[94]	2020	○	●	●					●	HRNet-FCN	●	64x64		●		●			●			20.86	0.86
	[95]	2020	●	●	●					●	PP-CNN		-	●		●				●				0.92
	[122]	2020	●	●	●					●	ED-CNN		299x299				●		●	●				0.99
	[107]	2020	●	●		●				●	ED-LSTM		224x224				●		●					
	[166]	2020	●	●	●					●	CNN ResNet		224x224				●				●	Avrg.	Prec.=	0.93
	[64]	2020	●	●	●					●	AREN-CNN		128x128			●					●	98.52		
	[109]	2020	●	●		●	●	●		●	ED-CNN		*	●	●					●				0.92
	[5]	2020	●	●		●	●		●		CNN		128x128							●		89.6		
	[10]	2020	●	●		●				●	LSTM		256x256				●					94.29		
	[69]	2020	●	●	●					●	Siamese CNN		64x64								●	TPR=0.91		
	[128]	2020	●	●	●					●	Ensemble		224x224				●					99.65		1.00
	[36]	2020	●	●		●				●	*		112x112						●			98.26		99.73
	[81]	2020	●	●	●					●	OC-VAE		100x100				●					TPR=0.89		
	[51]	2020	●	●	●					●	ABC-ResNet		224x224								●		?	
Statistics & Steganalysis	[85]	2018		●		●				●	PRNU		1280x720								●	*TPR=1*	*FPR=*	*0.03*
	[149]	2019		●		●				●	Statistics	●	-								●			
	[106]	2019	●	●	●					●	PRNU		*								●	90.3		

*Only the best reported performance, averaged over the test datasets, is displayed to capture the 'best-case' scenario.

a novel disentanglement loss. Furthermore, to handle misaligned poses, an L1 loss is computed using a pixel mapped version of x_g to x_s (using the weights of the identity encoder). Similarly, the authors of Ref. [99] proposed a method disentangled identity transfer. However, neither Ref. [174] nor Ref. [99] were explicitly performed on faces.

6 COUNTERMEASURES

In general, countermeasures to malicious deepfakes can be categorized as either detection or prevention. We will now briefly discuss each accordingly. A summary and systematization of the deepfake detection methods can be found in Table 3.

6.1 Detection

The subject of image forgery detection is a well-researched subject [187]. In our review of detection methods, we will focus on works that specifically deal with detecting deepfakes of humans.

6.1.1 Artifact-Specific. Deepfakes often generate artifacts that may be subtle to humans but can be easily detected using machine learning and forensic analysis. Some works identify deepfakes by searching for specific artifacts. We identify seven types of artifacts: Spatial artifacts in blending, environments, and forensics; temporal artifacts in behavior, physiology, synchronization, and coherence.

Blending (*spatial*). Some artifacts appear where the generated content is blended back into the frame. To help emphasize these artifacts to a learner, researchers have proposed edge detectors, quality measures, and frequency analysis [4, 8, 42, 110, 186]. In Ref. [94], the authors follow a more explicit approach to detecting the boundary. They trained a CNN network to predict an image's blending boundary and a label (real or fake). Instead of using a deepfake dataset, the authors trained their network on a dataset of face swaps generated by splicing similar faces found through facial landmark similarity. By doing so, the model has the advantage that is focused on the blending boundary and not other artifacts caused by the generative model.

Environment (*spatial*). The content of a fake face can be anomalous in context to the rest of the frame. For example, residuals from face warping processes [97, 98, 100], lighting [149], and varying fidelity [86] can indicate the presence of generated content. In Ref. [95], the authors follow a different approach by contrasting the generated foreground to the (untampered) background using a patch and pair CNN. The authors of Ref. [122] also contrast the fore/background but enable a network to identify the distinguishing features automatically. They accomplish this by (1) encoding the face and context (hair and background) with an ED and (2) passing the difference between the encodings with the complete image (encoded) to a classifier.

Forensics (*spatial*). Several works detect deepfakes by analyzing subtle features and patterns left by the model. In Refs [179] and [106], the authors found that GANs leave unique fingerprints and show how it is possible to classify the generator given the content, even in the presence of compression and noise. In Ref. [85] the authors analyze a camera's unique sensor noise, called photo response non-uniformity (PRNU), to detect pasted content. To focus on the residuals, the authors of Ref. [107] use a two-stream ED to encode the color image and a frequency enhanced version using "Laplacian of Gaussian layers" (LoG). The two encodings are then fed through an LSTM, which then classifies the video based on a sequence of frames.

Instead of searching for residuals, the authors of Ref. [177] search for imperfections and found that deepfakes tend to have inconsistent head poses. Therefore, they detect deepfakes by predicting and monitoring facial landmarks. The authors of Ref. [166] had a different approach by training classifiers to focus on the imperfections instead of the residuals. This was accomplished by using a dataset generated using a ProGAN instead of other GANs since the ProGAN's images contain the least amount of frequency artifacts. In contrast to Ref. [166], the authors in Ref. [64] use a network to *emphasize* the residuals and suppress the imperfections in a preprocessing step for a classifier. Their network uses adaptive convolutional layers that predict residuals to maximize the artifacts' influence. Although this approach may help the network identify artifacts better, it may not generalize as well to new types of artifacts.

Behavior (*temporal*). With large amounts of data on the target, mannerisms and other behaviors can be monitored for anomalies. For example, in Ref. [6] the authors protect world leaders from a wide variety of deepfake attacks by modeling their recorded stock footage. Recently, the authors of Ref. [109] showed how behaviors can be used with no reference footage of the target. The approach is to detect discrepancies in the perceived emotion extracted from the clip's audio and video content. The authors use a custom Siamese network to consider the audio and video emotions when contrasted to real and fake videos.

Physiology (*temporal*). In 2014, researchers hypothesized that generated content will lack physiological signals and identified computer generated faces by monitoring their heart rate [32]. Regarding deepfakes, Ref. [30] monitored blood volume patterns (pulse) under the skin, and Ref. [96] took a more robust approach by monitoring irregular eye blinking patterns. Instead of detecting deepfakes, the authors of Ref. [31] use the pulse signal to help determine the model used to create the deepfake.

Synchronization (*temporal*). Inconsistencies are also a revealing factor. In Refs [87] and [47], the authors noticed that video dubbing attacks can be detected by correlating the speech to landmarks around the mouth. Later, in Ref. [5], the authors refined the approach by detecting when visemes (mouth shapes) are inconsistent with the spoken phonemes (utternaces). In particular, they focus on phonemes where the mouth is fully closed (B, P, M) since deepfakes in the wild tend to fail in generating these visemes.

Coherence (*temporal*). As noted in Section 4.1, realistic temporal coherence is challenging to generate, and some authors capitalize on the resulting artifacts to detect the fake content. For example, Ref. [63] uses an RNN to detect artifacts such as flickers and jitter, and Ref. [131] uses an LSTM on the face region only. In Ref. [25], a classifier is trained pairs of sequential frames, and in Ref. [11], the authors refine the network's focus by monitoring the frames' optical flow. Later, the same authors use an LSTM to predict the next frame and expose deepfakes when the reconstruction error is high [10].

6.1.2 Undirected Approaches. Instead of focusing on a specific artifact, some authors train deep neural networks as generic classifiers and let the network decide which features to analyze. In general, researchers have taken one of two approaches: classification or anomaly detection.

Classification. In Refs [105], [118], and [130], it was shown that deep neural networks tend to perform better than traditional image forensic tools on compressed imagery. Various authors then demonstrated how standard CNN architectures can effectively detect deepfake videos [3, 38, 39, 152]. In Ref. [69], the authors train the CNN as a Siamese network using contrasting examples of real and fake images. In Ref. [52], the authors were concerned that a CNN can only detect the attacks on which they trained. To close this gap, the authors propose using Hierarchical Memory Network (HMN) architecture, which considers the contents of the face and previously seen faces. The network encodes the face region, which is then processed using a bidirectional GRU while applying an attention mechanism. The final encoding is then passed to a memory module, which compares it to recently seen encodings and makes a prediction. Later, in Ref. [128], the authors use an ensemble approach and leverage the predictions of seven deepfake CNNs by passing their predictions to a meta classifer. Doing so produces results that are more robust (fewer false positives) than using any single model. In Ref. [36], the authors tried a variety of different classic spatio-temproal networks and feature extractors as a baseline for temporal deepfake detection. They found that a 3D CNN, which looks at multiple frames at once, out performs both recurrent networks and the state-of-the-art ID3 architecture.

To localize the tampered areas, some works train networks to predict masks learned from a ground truth dataset, or by mapping the neural activations back to the raw image [41, 92, 117, 148].

In general, we note that the use of classifiers to detect deepfakes is problematic since an attacker can evade detection via adversarial machine learning. We will discuss this issue further in Section 7.2.

Anomaly Detection. In contrast to classification, anomaly detection models are trained on the normal data and then detect outliers during deployment. By doing so, these methods do not make

assumptions on how the attacks look and thus generalize better to unknown creation methods. The authors of Ref. [165] follow this approach by measuring the neural activation (coverage) of a face recognition network. By doing so, the model is able to overcome noise and other distortions by obtaining a stronger signal than from just using the raw pixels. Similarly, in Ref. [81], a one-class VAE is trained to be used to reconstruct real images. Then, for new images, an anomaly score is computed by taking the MSE between mean component of the encoded image and the mean component of the reconstructed image. Alternatively, the authors of Ref. [17] measure an input's embedding distance to real samples using an ED's latent space. The difference between these works is that Refs [165] and [81] rely on a model's inability to process unknown patterns while Ref. [17] contrasts the model's representations.

Instead of using a neural network directly, the authors of Ref. [51] use a state-of-the-art attribution-based confidence (ABC) metric. To detect a fake image, the ABC is used to determine if the image fits the training distribution of a pretrained face recognition network (e.g., VGG).

6.2 Prevention and Mitigation

Data Provenance. To prevent deepfakes, some have suggested that data provenance of multimedia should be tracked through distributed ledgers and blockchain networks [54]. In Ref. [44], the authors suggest that the content should be ranked by participants and AI. In contrast, Ref. [68] proposes that the content should be authenticated and managed as a global file system over Etherium smart contracts.

Counter Attacks. To combat deepfakes, the authors of Ref. [101] show how adversarial machine learning can be used to disrupt and corrupt deepfake networks. The authors perform adversarial machine learning to add crafted noise perturbations to x, which prevents deepfake technologies from locating a proper face in x. In a different approach, the authors of Ref. [137] use adversarial noise to change the identity of the face so that web crawlers will not be able find the image of t to train their model.

7 DISCUSSION

7.1 The Creation of Deepfakes

7.1.1 Tradeoffs between the Methodologies. In general, there is a different cost and payoff for each deepfake creation method. However, the most effective and threatening deepfakes are those that are (1) the most practical to implement (Training *Data*, Execution *Speed*, and *Accessibility*) and (2) are the most believable to the victim (*Quality*):

Data vs. Quality. Models trained on numerous samples of the target often yield better results (e.g., see Refs [25], [55], [71], [73], [89], [104], [151], and [173]). For example, in 2017, Ref. [151] produced an extremely believable reenactment of Obama, which exceeds the quality of recent works. However, these models require many hours footage for training, and are therefore are only suitable for exposed targets such as actors, CEOs, and political leaders. An attacker who wants to commit defamation, impersonation, or a scam on an arbitrary individual will need to use a many-to-many or few-shot approach. On the other hand, most of these methods rely on a single reference of t and are therefore prone to generating artifacts. This is because the model must "imagine" missing information (e.g., different poses and occlusions). Therefore, approaches that provide the model with a limited number of reference samples [62, 65, 158, 167, 171, 180, 182] strike the best balance between data and quality.

Speed vs. Quality. The tradeoff between these aspects depends on whether the attack is online (interactive) or offline (stored media). Social engineering attacks involving deepfakes

are likely to be online and thus require real-time speeds. However, high-resolution models have many parameters and sometimes use several networks (e.g., Ref. [56]) and some process multiple frames to provide temporal coherence (e.g., Refs [15], [83], and [168]). Other methods may be slowed down due to their pre/post-processing steps, such as warping [60, 62, 185], UV mapping or segmentation prediction [23, 112, 123, 180], and the use of refinement networks [25, 60, 83, 93, 111, 153]. To the best of our knowledge, [74], [88], [112], [120], and [144] are the only papers that claim to generate real-time deepfakes, yet they subjectively tend to be blurry or distort the face. Regardless, a victim is likely fall for an imperfect deepfake in a social engineering attack when placed under pressure in a false pretext [172]. Moreover, it is likely that an attacker will implement a complex method at a lower resolution to speed up the frame rate. In which case, methods that have texture artifacts would be preferred over those that produce shape or identity flaws (e.g., Refs [144] vs. [182]). For attacks that are not real-time (e.g, fake news), resolution and fidelity is critical. In these cases, works that produce high-quality images and videos with temporal coherence are the best candidates (e.g., Refs [65] and [168]).

Availability vs. Quality. We also note that availability and reproducibility are key factors in the proliferation of new technologies. Works that publish their code and datasets online (e.g., Refs [79], [134], [144], [161], [171], and [173]) are more likely to be used by researchers and criminals compared to those that are unavailable [2, 55, 65, 83, 120, 126, 153, 170, 184] or require highly specific or private datasets [57, 112, 180]. This is because the payoff in implementing a paper is minor compared to using a functional and effective method available online. Of course, this does not include state-actors who have plenty of time and funding.

We have also observed that approaches that augment a network's inputs with synthetic ones produce better results in terms of quality and stability, for example, by rotating limbs [104, 190], refining rendered heads [14, 55, 112, 153, 169, 178], providing warped imagery [60, 111, 116, 181], and UV maps [23, 62, 83, 124, 181]. This is because the provided contextual information reduces the problem's complexity for the neural network.

Given these considerations, in our opinion, the most significant and available deepfake technologies today are [144] for facial reenactment because of its efficiency and practicality; [27] for mouth reenactment because of its quality; and [71] for face replacement because its high fidelity and widespread use. However, this is a subjective opinion based on the samples provided online and in the respective papers. A comparative research study, where the methods are trained on the same dataset and evaluated by a number of people, is necessary to determine the best quality deepfake in each category.

7.1.2 Research Trends. Over the last few years, there has been a shift toward identity agnostic models and high-resolution deepfakes. Some notable advancements include (1) unpaired self-supervised training techniques to reduce the amount of initial training data; (2) one/few-shot learning, which enables identity theft with a single profile picture; (3) improvements of face quality and identity through AdaIN layers, disentanglement, and pix2pixHD network components; (4) fluid and realistic videos through temporal discriminators and optical flow prediction; and (5) the mitigation of boundary artifacts by using secondary networks to blend composites into seamless imagery (e.g., Refs [55], [153], and [169]).

Another large advancement in this domain was the use of perceptual loss on a pre-trained VGG face recognition network. The approach boosts the facial quality significantly, and, as a result, has been adopted in popular online deepfake tools [1, 138]. Another advancement being adopted is the use of a network pipeline. Instead of enforcing a set of global losses on a single network, a pipeline

of networks is used where each network is tasked with a different responsibility (conversion, generation, occlusions, blending, etc.) This give more control over the final output and has been able to mitigate most of the challenges mentioned in Section 3.7.

7.1.3 Current Limitations. Aside from quality, there are a few limitations with the current deepfake technologies. First, for reenactment, content is always driven and generated with a frontal pose. This limits the reenactment to a very static performance. Today, this is avoided by face swapping the identity onto a lookalike's body, but a good match is not always possible and this approach has limited flexibility. Second, reenactments and replacements depend on the driver's performance to deliver the identity's personality. We believe that next generation deepfakes will utilize videos of the target to stylize the generated content with the expected expressions and mannerisms. This will enable a much more automatic process of creating believable deepfakes. Finally, a new trend is real-time deepfakes. Works such as Refs [74] and [120] have achieved real-time deepfakes at 30fps. Although real-time deepfakes are an enabler for phishing attacks, the realism is not quite there yet. Other limitations include the coherent rendering of hair, teeth, tongues, shadows, and the ability to render the target's hands (especially when touching the face). Regardless, deepfakes are already very convincing [130] and are improving at a rapid rate. Therefore, it is important that we focus on effective countermeasures.

7.2 The Deepfake Arms Race

Like any battle in cyber security, there is an arms race between the attacker and defender. In our survey, we observed that the majority of deepfake detection algorithms assume a static game with the adversary: They are either focused on identifying a specific artifact, or do not generalize well to new distributions and unseen attacks [33]. Moreover, based on the recent benchmark of Ref. [100], we observe that the performance of state-of-the-art detectors are decreasing rapidly as the quality of the deepfakes improve. Concretely, the three most recent benchmark datasets (DFD by Google [119], DFDC by Facebook [40], and Celeb-DF by Ref. [100]) were released within one month of each other at the end of 2019. However, the deepfake detectors only achieved an area under the curve (AUC) of 0.86, 0.76, and 0.66 on each, respectively. Even a false alarm rate of 0.001 is far too low considering the millions of images published online daily.

Evading Artifact-based Detectors. To evade an artifact-based detector, the adversary only needs to mitigate a single flaw to evade detection. For example, G can generate the biological signals monitored by Refs [30] and [96] by adding a discriminator that monitors these signals. To avoid anomalies in extensive neuron activation [165], the adversary can add a loss that minimizes neuron coverage. Methods that detect abnormal poses and mannerisms [6] can be evaded by reenacting the entire head and by learning the mannerisms from the same databases. Models that identify blurred content [110] are affected by noise and sharpening GANs [73, 84]; and models that search for the boundary where the face was blended in [4, 8, 42, 94, 110, 186] do not work on deepfakes passed through refiner networks, which use in-painting, or those which output full frames (e.g., Refs [83], [93], [102], [112], [113], [120], [181], and [190]). Finally, solutions that search for forensic evidence [85, 106, 179] can be evaded (or at least raise the false alarm rate) by passing x_g through filters, or by performing physical replication or compression.

Evading Deep Learning Classifiers. There are a number of detection methods that apply deep learning directly to the task of deepfake detection (e.g., Refs [3], [38], [39], [52], and [152]). However, an adversary can use adversarial machine learning to evade detection by adding small perturbations to x_g. Advances in adversarial machine learning has shown that these attacks transfer across multiple models regardless of the training data used [125]. Recent works have shown

how these attacks not only work on deepfakes classifiers [115] but also work with no knowledge of the classifier or its training set [24].

Moving Forward. Nevertheless, deepfakes are still imperfect, and these methods offer a modest defense for the time being. Furthermore, these works play an important role in understanding the current limitations of deepfakes, and raise the difficulty threshold for malicious users. At some point, it may become too time-consuming and resource-intensive for a common attacker to create a good-enough fake to evade detection. However, we argue that solely relying on the development of content-based countermeasures is not sustainable and may lead to a reactive arms-race. Therefore, we advocate for more out-of-band approaches for detecting and preventing deepfakes, for example, the establishment of content provenance and authenticity frameworks for online videos [44, 54, 68], and proactive defenses such as the use of adversarial machine learning to protect content from tampering [101].

7.3 Deepfakes in Other Domains

In this survey, we put a focus on human reenactment and replacement attacks—the type of deepfakes that has made the largest impact so far [12, 66]. However, deepfakes extend beyond human visuals, and have spread to many other domains. In healthcare, the authors of Ref. [108] showed how deepfakes can be used to inject or remove medical evidence in CT and MRI scans for insurance fraud, disruption, and physical harm. In Ref. [75], it was shown how one's voice can be cloned with only five seconds of audio, and in September 2019, a CEO was scammed out of $250K via a voice clone deepfake [37]. The authors of Ref. [22] have shown how deep learning can generate realistic human fingerprints that can unlock multiple users' devices. In Ref. [135], it was shown how deepfakes can be applied to financial records to evade the detection of auditors. Finally, it has been shown how deepfakes of news articles can be generated [183] and that deepfake tweets exist as well [50].

These examples demonstrate that deepfakes are not just attack tools for misinformation, defamation, and propaganda, but also sabotage, fraud, scams, obstruction of justice, and potentially many more.

7.4 What's on the Horizon

We believe that in the coming years, we will see more deepfakes being weaponized for monetization. The technology has proven itself in humiliation, misinformation, and defamation attacks. Moreover, the tools are becoming more practical [1] and efficient [75]. Therefore, is seems natural that malicious users will find ways to use the technology for a profit. As a result, we expect to see an increase in deepfake phishing attacks and scams targeting both companies and individuals.

As the technology matures, real-time deepfakes will become increasingly realistic. Therefore, we can expect that the technology will be used by hacking groups to perform reconnaissance as part of an advanced persistent threat (APT), and by state actors to perform espionage and sabotage by reenacting officials or family members.

To keep ahead of the game, we must be proactive and consider the adversary's next step, not just the weaknesses of the current attacks. We suggest that more work be done on evaluating the theoretical limits of these attacks. For example, by finding a bound on a model's delay can help detect real-time attacks such as Ref. [75], and determining the limits of GANs like Ref. [7] can help us devise the appropriate strategies. As mentioned earlier, we recommend further research on solutions that do not require analyzing the content itself. Moreover, we believe it would be beneficial for future works to explore the weaknesses and limitations of current deepfakes detectors. By identifying and understanding these vulnerabilities, researchers will be able to develop stronger countermeasures.

8 CONCLUSION

Not all deepfakes are malicious. However, because the technology makes it so easy to create believable media, malicious users are exploiting it to perform attacks. These attacks are targeting individuals and causing psychological, political, monetary, and physical harm. As time goes on, we expect to see these malicious deepfakes spread to many other modalities and industries.

In this survey, we focused on reenactment and replacement deepfakes of humans. We provided a deep review of how these technologies work, the differences between their architectures, and what is being done to detect them. We hope this information will be helpful to the community in understanding and preventing malicious deepfakes.

REFERENCES

[1] 2017. deepfakes/faceswap: Deepfakes Software For All. Retrieved January 27, 2020 from https://github.com/deepfakes/faceswap.

[2] Kfir Aberman, Mingyi Shi, Jing Liao, D. Liscbinski, Baoquan Chen, and Daniel Cohen-Or. 2019. Deep video-based performance cloning. In *Computer Graphics Forum*. Wiley Online Library.

[3] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. 2018. Mesonet: A compact facial video forgery detection network. In *Proceedings of the 2018 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 1–7.

[4] Akshay Agarwal, Richa Singh, Mayank Vatsa, and Afzel Noore. 2017. SWAPPED! Digital face presentation attack detection via weighted local magnitude pattern. In *Proceedings of the 2017 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 659–665.

[5] Shruti Agarwal, Hany Farid, Ohad Fried, and Maneesh Agrawala. 2020. Detecting deep-fake videos from phoneme-viseme mismatches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 660–661.

[6] Shruti Agarwal, Hany Farid, Yuming Gu, Mingming He, Koki Nagano, and Hao Li. 2019. Protecting world leaders against deep fakes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 38–45.

[7] Sakshi Agarwal and Lav R. Varshney. 2019. Limits of deepfake detection: A robust estimation viewpoint. *arXiv:1905.03493* (2019).

[8] Zahid Akhtar and Dipankar Dasgupta. [n.d.]. A comparative evaluation of local feature descriptors for DeepFakes detection. ([n.d.]).

[9] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. 2018. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[10] Irene Amerini and Roberto Caldelli. 2020. Exploiting prediction error inconsistencies through LSTM-based classifiers to detect deepfake videos. In *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security*. 97–102.

[11] Irene Amerini, Leonardo Galteri, Roberto Caldelli, and Alberto Del Bimbo. 2019. Deepfake video detection through optical flow based CNN. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 0–0.

[12] Arije Antinori. 2019. Terrorism and DeepFake: From hybrid warfare to post-truth warfare in a hybrid world. In *ECIAIR 2019 European Conference on the Impact of Artificial Intelligence and Robotics*. Academic Conferences and Publishing Limited, 23.

[13] Hadar Averbuch-Elor, Daniel Cohen-Or, Johannes Kopf, and Michael F. Cohen. 2017. Bringing portraits to life. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2017)* 36, 6 (2017), 196.

[14] Guha Balakrishnan, Amy Zhao, Adrian V. Dalca, Fredo Durand, and John Guttag. 2018. Synthesizing images of humans in unseen poses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8340–8348.

[15] Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. 2018. Recycle-GAN: Unsupervised video retargeting. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[16] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. 2017. CVAE-GAN: Fine-grained image generation through asymmetric training. In *Proceedings of the IEEE International Conference on Computer Vision*. 2745–2754.

[17] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. 2018. Towards open-set identity preserving face synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[18] Dmitri Bitouk, Neeraj Kumar, Samreen Dhillon, Peter Belhumeur, and Shree K. Nayar. 2008. Face swapping: Automatically replacing faces in photographs. *ACM Transactions on Graphics (TOG)* 27, 3 (2008), 1–8.
[19] Volker Blanz, Curzio Basso, Tomaso Poggio, and Thomas Vetter. 2003. Reanimating faces in images and video. In *Computer Graphics Forum*, Vol. 22. Wiley Online Library, 641–650.
[20] Volker Blanz, Kristina Scherbaum, Thomas Vetter, and Hans-Peter Seidel. 2004. Exchanging faces in images. In *Computer Graphics Forum*, Vol. 23. Wiley Online Library, 669–676.
[21] Volker Blanz and Thomas Vetter. 1999. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. 187–194.
[22] Philip Bontrager, Aditi Roy, Julian Togelius, Nasir Memon, and Arun Ross. 2018. DeepMasterPrints: Generating masterprints for dictionary attacks via latent variable evolution. In *Proceedings of the 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE.
[23] Jie Cao, Yibo Hu, Bing Yu, Ran He, and Zhenan Sun. 2019. 3D aided duet GANs for multi-view face image synthesis. *IEEE Transactions on Information Forensics and Security* 14, 8 (2019), 2028–2042.
[24] Nicholas Carlini and Hany Farid. 2020. Evading deepfake-image detectors with white-and black-box attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 658–659.
[25] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A. Efros. 2019. Everybody dance now. In *Proceedings of the IEEE International Conference on Computer Vision*. 5933–5942.
[26] Yao-Jen Chang and Tony Ezzat. 2005. Transferable videorealistic speech animation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 143–151.
[27] Lele Chen, Ross K. Maddox, Zhiyao Duan, and Chenliang Xu. 2019. Hierarchical cross-modal talking face generation with dynamic pixel-wise loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7832–7841.
[28] Robert Chesney and Danielle Keats Citron. 2018. Deep fakes: A looming challenge for privacy, democracy, and national security. (2018).
[29] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. 2018. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
[30] Umur Aybars Ciftci and Ilke Demir. 2019. FakeCatcher: Detection of synthetic portrait videos using biological signals. *arXiv preprint arXiv:1901.02212* (2019).
[31] Umur Aybars Ciftci, Ilke Demir, and Lijun Yin. 2020. How do the hearts of deep fakes beat? Deep fake source detection via interpreting residuals with biological signals. *arXiv preprint arXiv:2008.11363* (2020).
[32] Valentina Conotter, Ecaterina Bodnari, Giulia Boato, and Hany Farid. 2014. Physiologically-based detection of computer generated faces in video. In *Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, 248–252.
[33] Davide Cozzolino, Justus Thies, Andreas Rossler, Christian Riess, Matthias Niessner, and Luisa Verdoliva. 2018. Forensictransfer: Weakly-supervised domain adaptation for forgery detection. *arXiv preprint arXiv:1812.02510* (2018).
[34] Kevin Dale, Kalyan Sunkavalli, Micah K. Johnson, Daniel Vlasic, Wojciech Matusik, and Hanspeter Pfister. 2011. Video face replacement. In *Proceedings of the 2011 SIGGRAPH Asia Conference*. 1–10.
[35] Rodrigo De Bem, Arnab Ghosh, Adnane Boukhayma, Thalaiyasingam Ajanthan, N Siddharth, and Philip Torr. 2019. A conditional deep generative model of people in natural images. In *Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE.
[36] Oscar de Lima, Sean Franklin, Shreshtha Basu, Blake Karwoski, and Annet George. 2020. Deepfake detection using spatiotemporal convolutional networks. *arXiv preprint arXiv:2006.14749* (2020).
[37] Jesse Demiani. 2019. A Voice Deepfake Was Used to Scam a CEO Out of $243,000. *Forbes*. https://bit.ly/38sXb1I.
[38] Xinyi Ding, Zohreh Raziei, Eric C. Larson, Eli V. Olinick, Paul Krueger, and Michael Hahsler. 2019. Swapped face detection using deep learning and subjective assessment. *arXiv preprint arXiv:1909.04217* (2019).
[39] Nhu-Tai Do, In-Seop Na, and Soo-Hyung Kim. 2018. Forensics Face Detection From GANs Using Convolutional Neural Network.
[40] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton Ferrer. 2019. The deepfake detection challenge (DFDC) preview dataset. *arXiv preprint arXiv:1910.08854* (2019).
[41] Mengnan Du, Shiva Pentyala, Yuening Li, and Xia Hu. 2019. Towards generalizable forgery detection with locality-aware AutoEncoder. *arXiv preprint arXiv:1909.05999* (2019).
[42] Ricard Durall, Margret Keuper, Franz-Josef Pfreundt, and Janis Keuper. 2019. Unmasking DeepFakes with simple features. *arXiv preprint arXiv:1911.00686* (2019).
[43] P. Ekman, W. Friesen, and J. Hager. 2002. Facial action coding system: Research Nexus. *Network Research Information, Salt Lake City, UT* 1 (2002).

[44] Chi-Ying Chen et al. 2019. A trusting news ecosystem against fake news from humanity and technology perspectives. In *Forbes 2019 19th International Conference on Computational Science and Its Applications (ICCSA)*. IEEE, 132–137.
[45] Daniil Kononenko et al. 2017. Photorealistic monocular gaze redirection using machine learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 11 (2017), 2696–2710.
[46] Liqian Ma et al. 2018. Disentangled person image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 99–108.
[47] Pavel Korshunov et al. 2019. Tampered speaker inconsistency detection with phonetically aware audio-visual features. In *International Conference on Machine Learning*.
[48] Shengju Qian et al. 2019. Make a face: Towards arbitrary high fidelity face manipulation. In *Proceedings of the IEEE International Conference on Computer Vision*.
[49] Facebook. 2018. Facing Facts. Retrieved March 2, 2020 from https://about.fb.com/news/2018/05/inside-feed-facing-facts/#watchnow.
[50] Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2020. TweepFake: About detecting deepfake tweets. *arXiv preprint arXiv:2008.00036* (2020).
[51] Steven Fernandes, Sunny Raj, Rickard Ewetz, Jodh Singh Pannu, Sumit Kumar Jha, Eddy Ortiz, Iustina Vintila, and Margaret Salter. 2020. Detecting deepfake videos using attribution-based confidence metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 308–309.
[52] Tharindu Fernando, Clinton Fookes, Simon Denman, and Sridha Sridharan. 2019. Exploiting human social cognition for the detection of fake and fraudulent faces via memory networks. *arXiv preprint arXiv:1911.07844* (2019).
[53] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1126–1135.
[54] Paula Fraga-Lamas and Tiago M. Fernandez-Carames. 2019. Leveraging distributed ledger technologies and blockchain to combat fake news. *arXiv preprint arXiv:1904.05386* (2019).
[55] Ohad Fried, Ayush Tewari, Michael Zollhofer, Adam Finkelstein, Eli Shechtman, Dan B. Goldman, Kyle Genova, Zeyu Jin, Christian Theobalt, and Maneesh Agrawala. 2019. Text-based editing of talking-head video. *arXiv preprint arXiv:1906.01524* (2019).
[56] Chaoyou Fu, Yibo Hu, Xiang Wu, Guoli Wang, Qian Zhang, and Ran He. 2019. High fidelity face manipulation with extreme pose and expression. *arXiv preprint arXiv:1903.12003* (2019).
[57] Yaroslav Ganin, Daniil Kononenko, Diana Sungatullina, and Victor Lempitsky. 2016. Deepwarp: Photorealistic image resynthesis for gaze manipulation. In *European Conference on Computer Vision*. Springer.
[58] Pablo Garrido, Levi Valgaerts, Ole Rehmsen, Thorsten Thormahlen, Patrick Perez, and Christian Theobalt. 2014. Automatic face reenactment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4217–4224.
[59] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2015. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576* (2015).
[60] Jiahao Geng, Tianjia Shao, Youyi Zheng, Yanlin Weng, and Kun Zhou. 2019. Warp-guided GANs for single-photo facial animation. *ACM Transactions on Graphics (TOG)* 37, 6 (2019), 231.
[61] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*. 2672–2680.
[62] Kuangxiao Gu, Yuqian Zhou, and Thomas S. Huang. 2020. FLNet: Landmark driven fetching and learning network for faithful talking facial animation synthesis. In *AAAI*. 10861–10868.
[63] David Guera and Edward J. Delp. 2018. Deepfake video detection using recurrent neural networks. In *IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 1–6.
[64] Zhiqing Guo, Gaobo Yang, Jiyou Chen, and Xingming Sun. 2020. Fake face detection via adaptive residuals extraction network. *arXiv preprint arXiv:2005.04945* (2020).
[65] Sungjoo Ha, Martin Kersner, Beomsu Kim, Seokjun Seo, and Dongyoung Kim. 2020. MarioNETte: Few-shot face reenactment preserving identity of unseen targets. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
[66] Holly Kathleen Hall. 2018. Deepfake videos: When seeing isn't believing. *Cath. UJL & Tech* 27 (2018), 51.
[67] Karen Hao. 2019. The biggest threat of deepfakes isn' t the deepfakes themselves. *MIT Tech Review*. Retrieved from https://www.technologyreview.com/s/614526/the-biggest-threat-of-deepfakes-isnt-the-deepfakes-themselves/.
[68] Haya R. Hasan and Khaled Salah. 2019. Combating deepfake videos using blockchain and smart contracts. *IEEE Access* 7 (2019), 41596–41606.
[69] Chih-Chung Hsu, Yi-Xiu Zhuang, and Chia-Yen Lee. 2020. Deep fake image detection based on pairwise learning. *Applied Sciences* 10, 1 (2020), 370.
[70] Yibo Hu, Xiang Wu, Bing Yu, Ran He, and Zhenan Sun. 2018. Pose-guided photorealistic face rotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8398–8406.

[71] iperov. 2019. DeepFaceLab: DeepFaceLab is a tool that utilizes machine learning to replace faces in videos. Retrieved December 31, 2019 from https://github.com/iperov/DeepFaceLab.
[72] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1125–1134.
[73] Seyed Ali Jalalifar, Hosein Hasani, and Hamid Aghajan. 2018. Speech-driven facial reenactment using conditional generative adversarial networks. *arXiv preprint arXiv:1803.07461* (2018).
[74] Amir Jamaludin, Joon Son Chung, and Andrew Zisserman. 2019. You said that?: Synthesising talking faces from audio. *International Journal of Computer Vision* (2019), 1–13.
[75] Ye Jia, Yu Zhang, Ron Weiss, Quan Wang, Jonathan Shen, Fei Ren, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, Yonghui Wu, et al. 2018. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In *Advances in Neural Information Processing Systems*. 4480–4490.
[76] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*. Springer, 694–711.
[77] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. 2018. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7122–7131.
[78] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2019. Analyzing and improving the image quality of styleGAN. *arXiv preprint arXiv:1912.04958* (2019).
[79] Triantafyllos Kefalas, Konstantinos Vougioukas, Yannis Panagakis, Stavros Petridis, Jean Kossaifi, and Maja Pantic. 2019. Speech-driven facial animation using polynomial fusion of features. *arXiv preprint arXiv:1912.05833* (2019).
[80] Ira Kemelmacher-Shlizerman. 2016. Transfiguring portraits. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 94.
[81] Hasam Khalid and Simon S. Woo. 2020. OC-FakeDect: Classifying deepfakes using one-class variational autoencoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 656–657.
[82] Ali Khodabakhsh, Raghavendra Ramachandra, Kiran Raja, Pankaj Wasnik, and Christoph Busch. 2018. Fake face detection methods: Can they be generalized? In *Proceedings of the 2018 International Conference of the Biometrics Special Interest Group (BIOSIG)*. IEEE, 1–6.
[83] Hyeongwoo Kim, Pablo Carrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Perez, Christian Richardt, Michael Zollhofer, and Christian Theobalt. 2018. Deep video portraits. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 163.
[84] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. 2016. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1646–1654.
[85] Marissa Koopman, Andrea Macarulla Rodriguez, and Zeno Geradts. 2018. Detection of deepfake video manipulation. In *Conference: IMVIP*.
[86] Pavel Korshunov and Sebastien Marcel. 2018. Deepfakes: A new threat to face recognition? Assessment and detection. *arXiv preprint arXiv:1812.08685* (2018).
[87] Pavel Korshunov and Sebastien Marcel. 2018. Speaker inconsistency detection in tampered video. In *Proceedings of the 2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2375–2379.
[88] Iryna Korshunova, Wenzhe Shi, Joni Dambre, and Lucas Theis. 2017. Fast face-swap using convolutional neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*.
[89] Rithesh Kumar, Jose Sotelo, Kundan Kumar, Alexandre de Brebisson, and Yoshua Bengio. 2017. Obamanet: Photo-realistic lip-sync from text. *arXiv preprint arXiv:1801.01442* (2017).
[90] Dami Lee. 2019. Deepfake Salvador Dalí takes selfies with museum visitors. *The Verge*. https://bit.ly/3cEim4m.
[91] Jessica Lee, Deva Ramanan, and Rohit Girdhar. 2019. MetaPix: Few-shot video retargeting. *arXiv preprint arXiv:1910.04742* (2019).
[92] Jia Li, Tong Shen, Wei Zhang, Hui Ren, Dan Zeng, and Tao Mei. 2019. Zooming into face forensics: A pixel-level analysis. *arXiv preprint arXiv:1912.05790* (2019).
[93] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. 2019. FaceShifter: Towards high fidelity and occlusion aware face swapping. *arXiv preprint arXiv:1912.13457* (2019).
[94] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. 2020. Face x-ray for more general face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5001–5010.
[95] Xurong Li, Kun Yu, Shouling Ji, Yan Wang, Chunming Wu, and Hui Xue. 2020. Fighting against deepfake: Patch&pair convolutional neural networks (PPCNN). In *Companion Proceedings of the Web Conference 2020*. 88–89.
[96] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. 2018. In ictu oculi: Exposing AI created fake videos by detecting eye blinking. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 1–7.
[97] Yuezun Li and Siwei Lyu. 2019. DSP-FWA: Dual Spatial Pyramid for Exposing Face Warp Artifacts in DeepFake Videos. Retrieved December 18, 2019 from https://github.com/danmohaha/DSP-FWA.
[98] Yuezun Li and Siwei Lyu. 2019. Exposing DeepFake videos by detecting face warping artifacts. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

[99] Yuheng Li, Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. 2019. MixNMatch: Multifactor disentanglement and encodingfor conditional image generation. *arXiv preprint arXiv:1911.11758* (2019).
[100] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. 2019. Celeb-DF: A new dataset for DeepFake forensics. *arXiv preprint:1909.12962* (2019).
[101] Yuezun Li, Xin Yang, Baoyuan Wu, and Siwei Lyu. 2019. Hiding faces in plain sight: Disrupting AI face synthesis with adversarial perturbations. *arXiv preprint arXiv:1906.09288* (2019).
[102] Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Hyeongwoo Kim, Florian Bernard, Marc Habermann, Wenping Wang, and Christian Theobalt. 2019. Neural rendering and reenactment of human actor videos. *ACM Transactions on Graphics (TOG)* 38, 5 (2019), 139.
[103] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. 2019. Liquid warping GAN: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*.
[104] Zhaoxiang Liu, Huan Hu, Zipeng Wang, Kai Wang, Jinqiang Bai, and Shiguo Lian. 2019. Video synthesis of human upper body with realistic face. *arXiv preprint arXiv:1908.06607* (2019).
[105] Francesco Marra, Diego Gragnaniello, Davide Cozzolino, and Luisa Verdoliva. 2018. Detection of GAN-generated fake images over social networks. In *Proceedings of the 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 384–389.
[106] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. 2019. Do GANs leave artificial fingerprints? In *Proceedings of the 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 506–511.
[107] Iacopo Masi, Aditya Killekar, Royston Marian Mascarenhas, Shenoy Pratik Gurudatt, and Wael AbdAlmageed. 2020. Two-branch recurrent network for isolating deepfakes in videos. *arXiv preprint arXiv:2008.03412* (2020).
[108] Yisroel Mirsky, Tom Mahler, Ilan Shelef, and Yuval Elovici. 2019. CT-GAN: Malicious tampering of 3D medical imagery using deep learning. In *USENIX Security Symposium 2019*.
[109] Trisha Mittal, Uttaran Bhattacharya, Rohan Chandra, Aniket Bera, and Dinesh Manocha. 2020. Emotions don't lie: A deepfake detection method using audio-visual affective cues. *arXiv preprint arXiv:2003.06711* (2020).
[110] Huaxiao Mo, Bolin Chen, and Weiqi Luo. 2018. Fake faces identification via convolutional neural network. In *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*. ACM.
[111] Joel Ruben Antony Moniz, Christopher Beckham, Simon Rajotte, Sina Honari, and Chris Pal. 2018. Unsupervised depth estimation, 3D face rotation, and replacement. In *Advances in Neural Information Processing Systems*. 9736–9746.
[112] Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, Hao Li, Richard Roberts, et al. 2018. paGAN: Real-time avatars using dynamic textures. *ACM Trans. Graph.* 37, 6 (2018), 1–12.
[113] Ryota Natsume, Tatsuya Yatagawa, and Shigeo Morishima. 2018. FSNet: An identity-aware generative model for image-based face swapping. In *Asian Conference on Computer Vision*. Springer, 117–132.
[114] Ryota Natsume, Tatsuya Yatagawa, and Shigeo Morishima. 2018. RSGAN: Face swapping and editing using face and hair representation in latent spaces. *arXiv preprint arXiv:1804.03447* (2018).
[115] Paarth Neekhara, Shehzeen Hussain, Malhar Jere, Farinaz Koushanfar, and Julian McAuley. 2020. Adversarial deepfakes: Evaluating vulnerability of deepfake detectors to adversarial examples. *arXiv preprint arXiv:2002.12749* (2020).
[116] Natalia Neverova, Riza Alp Guler, and Iasonas Kokkinos. 2018. Dense pose transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 123–138.
[117] Huy H. Nguyen, Fuming Fang, Junichi Yamagishi, and Isao Echizen. 2019. Multi-task learning for detecting and segmenting manipulated facial images and videos. *arXiv preprint arXiv:1906.06876* (2019).
[118] Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. 2019. Capsule-forensics: Using capsule networks to detect forged images and videos. In *Proceedings of the ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2307–2311.
[119] Andrew Gully and Nick Dufour. 2019. DFD. Retrieved from https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html.
[120] Yuval Nirkin, Yosi Keller, and Tal Hassner. 2019. FSGAN: Subject agnostic face swapping and reenactment. In *Proceedings of the IEEE International Conference on Computer Vision*. 7184–7193.
[121] Yuval Nirkin, Iacopo Masi, Anh Tran Tuan, Tal Hassner, and Gerard Medioni. 2018. On face segmentation, face swapping, and face perception. In *Proceedings of the 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 98–105.
[122] Yuval Nirkin, Lior Wolf, Yosi Keller, and Tal Hassner. 2020. DeepFake detection based on the discrepancy between the face and its context. *arXiv preprint arXiv:2008.12262* (2020).
[123] Kyle Olszewski, Zimo Li, Chao Yang, Yi Zhou, Ronald Yu, Zeng Huang, Sitao Xiang, Shunsuke Saito, Pushmeet Kohli, and Hao Li. 2017. Realistic dynamic facial textures from a single image using GANs. In *Proceedings of the IEEE International Conference on Computer Vision*. 5429–5438.

[124] Naima Otberdout, Mohamed Daoudi, Anis Kacem, Lahoucine Ballihi, and Stefano Berretti. 2019. Dynamic facial expression generation on Hilbert hypersphere with conditional Wasserstein generative adversarial nets. *arXiv preprint arXiv:1907.10087* (2019).

[125] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016. Transferability in machine learning: From phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277* (2016).

[126] Hai X. Pham, Yuting Wang, and Vladimir Pavlovic. 2018. Generative adversarial talking head: Bringing portraits to life with a weakly supervised neural network. *arXiv preprint arXiv:1803.07716* (2018).

[127] Albert Pumarola, Antonio Agudo, Aleix M. Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. 2019. GANimation: One-shot anatomically consistent facial animation. *International Journal of Computer Vision* (2019), 1–16.

[128] Md. Shohel Rana and Andrew H. Sung. 2020. DeepfakeStack: A deep ensemble-based learning technique for deepfake detection. In *Proceedings of the 2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*. IEEE, 70–75.

[129] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Niessner. 2018. Faceforensics: A large-scale video dataset for forgery detection in human faces. *arXiv preprint arXiv:1803.09179* (2018).

[130] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Niessner. 2019. Faceforensics++: Learning to detect manipulated facial images. *arXiv preprint:1901.08971* (2019).

[131] Ekraam Sabir, Jiaxin Cheng, Ayush Jaiswal, Wael AbdAlmageed, Iacopo Masi, and Prem Natarajan. 2019. Recurrent-convolution approach to DeepFake detection-state-of-art results on FaceForensics++. *arXiv preprint arXiv:1905.00582* (2019).

[132] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*. 2234–2242.

[133] Sigal Samuel. 2019. A guy made a deepfake app to turn photos of women into nudes. It didn't go well. https://www.vox.com/2019/6/27/18761639/ai-deepfake-deepnude-app-nude-women-porn.

[134] Enrique Sanchez and Michel Valstar. 2018. Triple consistency loss for pairing distributions in GAN-based face synthesis. *arXiv preprint arXiv:1811.03492* (2018).

[135] Marco Schreyer, Timur Sattarov, Bernd Reimer, and Damian Borth. 2019. Adversarial learning of deepfakes in accounting. *arXiv preprint arXiv:1910.03810* (2019).

[136] Oscar Schwartz. 2018. You thought fake news was bad? *The Guardian*. Retrieved March 2, 2020 from https://www.theguardian.com/technology/2018/nov/12/deep-fakes-fake-news-truth.

[137] Shawn Shan, Emily Wenger, Jiayun Zhang, Huiying Li, Haitao Zheng, and Ben Y. Zhao. 2020. Fawkes: Protecting privacy against unauthorized deep learning models. In *Proceedings of the 29th {USENIX} Security Symposium ({USENIX} Security 20)*. 1589–1604.

[138] Shaoanlu. 2018. faceswap-GAN: A denoising autoencoder + adversarial losses and attention mechanisms for face swapping. Retrieved December 17, 2020 from https://github.com/shaoanlu/faceswap-GAN.

[139] Shaoanlu. 2019. fewshot-face-translation-GAN: Generative adversarial networks integrating modules from FUNIT and SPADE for face-swapping. Retrieved from https://github.com/shaoanlu/fewshot-face-translation-GAN.

[140] Yujun Shen, Ping Luo, Junjie Yan, Xiaogang Wang, and Xiaoou Tang. 2018. FaceID-GAN: Learning a symmetry three-player GAN for identity-preserving face synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 821–830.

[141] Yujun Shen, Bolei Zhou, Ping Luo, and Xiaoou Tang. 2018. FaceFeat-GAN: A two-stage approach for identity-preserving face synthesis. *arXiv preprint arXiv:1812.01288* (2018).

[142] Taiki Shimba, Ryuhei Sakurai, Hirotake Yamazoe, and Joo-Ho Lee. 2015. Talking heads synthesis from audio with deep neural networks. In *Proceedings of the 2015 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 100–105.

[143] Aliaksandr Siarohin, Stephane Lathuiliere, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. 2019. Animating arbitrary objects via deep motion transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2377–2386.

[144] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. 2019. First order motion model for image animation. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 7135–7145. http://papers.nips.cc/paper/8935-first-order-motion-model-for-image-animation.pdf.

[145] Aliaksandr Siarohin, Enver Sangineto, Stephane Lathuiliere, and Nicu Sebe. 2018. Deformable GANs for pose-based human image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3408–3416.

[146] Yang Song, Jingwen Zhu, Xiaolong Wang, and Hairong Qi. 2018. Talking face generation by conditional recurrent adversarial network. *arXiv preprint arXiv:1804.04786* (2018).

[147] Jose Sotelo, Soroush Mehri, Kundan Kumar, Joao Felipe Santos, Kyle Kastner, Aaron Courville, and Yoshua Bengio. 2017. Char2wav: End-to-end speech synthesis. *Openreview.net* (2017).

[148] Joel Stehouwer, Hao Dang, Feng Liu, Xiaoming Liu, and Anil Jain. 2019. On the detection of digital face manipulation. *arXiv preprint arXiv:1910.01717* (2019).

[149] Jeremy Straub. 2019. Using subject face brightness assessment to detect "deep fakes" (Conference Presentation). In *Real-Time Image Processing and Deep Learning 2019*, Vol. 10996. International Society for Optics and Photonics, 109960H.

[150] Qianru Sun, Ayush Tewari, Weipeng Xu, Mario Fritz, Christian Theobalt, and Bernt Schiele. 2018. A hybrid model for identity obfuscation by face replacement. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 553–569.

[151] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. 2017. Synthesizing Obama: Learning lip sync from audio. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–13.

[152] Shahroz Tariq, Sangyup Lee, Hoyoung Kim, Youjin Shin, and Simon S. Woo. 2018. Detecting both machine and human created fake face images in the wild. In *Proceedings of the 2nd International Workshop on Multimedia Privacy and Security*. ACM, 81–87.

[153] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Niessner. 2019. Neural voice puppetry: Audio-driven facial reenactment. *arXiv preprint arXiv:1912.05566* (2019).

[154] Justus Thies, Michael Zollhofer, and Matthias Niessner. 2019. Deferred neural rendering: Image synthesis using neural textures. *arXiv preprint arXiv:1904.12356* (2019).

[155] Justus Thies, Michael Zollhofer, Matthias Niessner, Levi Valgaerts, Marc Stamminger, and Christian Theobalt. 2015. Real-time expression transfer for facial reenactment. *ACM Trans. Graph.* 34, 6 (2015), 183–1.

[156] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Niessner. 2016. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2387–2395.

[157] Justus Thies, Michael Zollhofer, Christian Theobalt, Marc Stamminger, and Matthias Niessner. 2018. Headon: Real-time reenactment of human portrait videos. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–13.

[158] Luan Tran, Xi Yin, and Xiaoming Liu. 2018. Representation learning by rotating your faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 12 (2018), 3007–3021.

[159] Soumya Tripathy, Juho Kannala, and Esa Rahtu. 2019. ICface: Interpretable and controllable face reenactment using GANs. *arXiv preprint arXiv:1904.01909* (2019).

[160] Xiaoguang Tu, Hengsheng Zhang, Mei Xie, Yao Luo, Yuefei Zhang, and Zheng Ma. 2019. Deep transfer across domains for face anti-spoofing. *arXiv preprint arXiv:1901.05633* (2019).

[161] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. 2018. MocoGAN: Decomposing motion and content for video generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1526–1535.

[162] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popovic. 2006. Face transfer with multilinear models. In *ACM SIGGRAPH 2006 Courses*. 24–es.

[163] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. 2019. End-to-end speech-driven realistic facial animation with temporal GANs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 37–40.

[164] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. 2019. Realistic speech-driven facial animation with GANs. *arXiv preprint arXiv:1906.06337* (2019).

[165] Run Wang, Lei Ma, Felix Juefei-Xu, Xiaofei Xie, Jian Wang, and Yang Liu. 2019. Fakespotter: A simple baseline for spotting AI-synthesized fake faces. *arXiv preprint arXiv:1909.06122* (2019).

[166] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A. Efros. 2020. CNN-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 7.

[167] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro. 2019. Few-shot video-to-video synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[168] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. Video-to-video synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[169] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. High-resolution image synthesis and semantic manipulation with conditional GANs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[170] Yaohui Wang, Piotr Bilinski, Francois Bremond, and Antitza Dantcheva. 2020. ImaGINator: Conditional spatio-temporal GAN for video generation.

[171] Olivia Wiles, A. Sophia Koepke, and Andrew Zisserman. 2018. X2face: A network for controlling face generation using images, audio, and pose codes. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 670–686.

[172] Michael Workman. 2008. Wisecrackers: A theory-grounded investigation of phishing and pretext social engineering threats to information security. *Journal of the American Society for Information Science and Technology* 59, 4 (2008), 662–674.

[173] Wayne Wu, Yunxuan Zhang, Cheng Li, Chen Qian, and Chen Change Loy. 2018. ReenactGAN: Learning to reenact faces via boundary transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 603–619.

[174] Fanyi Xiao, Haotian Liu, and Yong Jae Lee. 2019. Identity from here, pose from there: Self-supervised disentanglement and generation of objects using unlabeled videos. In *Proceedings of the IEEE International Conference on Computer Vision*. 7013–7022.

[175] Runze Xu, Zhiming Zhou, Weinan Zhang, and Yong Yu. 2017. Face transfer with generative adversarial network. *arXiv preprint:1710.06090* (2017).

[176] Xinsheng Xuan, Bo Peng, Wei Wang, and Jing Dong. 2019. On the generalization of GAN image forensics. In *Proceedings of the Chinese Conference on Biometric Recognition*. Springer, 134–141.

[177] Xin Yang, Yuezun Li, and Siwei Lyu. 2019. Exposing deep fakes using inconsistent head poses. In *Proceedings of the ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 8261–8265.

[178] Lingyun Yu, Jun Yu, and Qiang Ling. 2019. Mining audio, text and visual information for talking face generation. In *Proceedings of the 2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 787–795.

[179] Ning Yu, Larry S. Davis, and Mario Fritz. 2019. Attributing fake images to GANs: Learning and analyzing GAN fingerprints. In *Proceedings of the IEEE International Conference on Computer Vision*.

[180] Yu Yu, Gang Liu, and Jean-Marc Odobez. 2019. Improving few-shot user-specific gaze adaptation via gaze redirection synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 11937–11946.

[181] Polina Zablotskaia, Aliaksandr Siarohin, Bo Zhao, and Leonid Sigal. 2019. DwNet: Dense warp-based network for pose-guided human video generation. *arXiv preprint arXiv:1910.09139* (2019).

[182] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. 2019. Few-shot adversarial learning of realistic neural talking head models. *arXiv preprint arXiv:1905.08233* (2019).

[183] Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 9054–9065. Retrieved from http://papers.nips.cc/paper/9106-defending-against-neural-fake-news.pdf.

[184] Jiangning Zhang, Xianfang Zeng, Yusu Pan, Yong Liu, Yu Ding, and Changjie Fan. 2019. FaceSwapNet: Landmark guided many-to-many face reenactment. *arXiv preprint arXiv:1905.11805* (2019).

[185] Yunxuan Zhang, Siwei Zhang, Yue He, Cheng Li, Chen Change Loy, and Ziwei Liu. 2019. One-shot face reenactment. *arXiv preprint arXiv:1908.03251* (2019).

[186] Ying Zhang, Lilei Zheng, and Vrizlynn L. L. Thing. 2017. Automated face swapping and its detection. In *Proceedings of the 2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP)*. IEEE, 15–19.

[187] Lilei Zheng, Ying Zhang, and Vrizlynn L. L. Thing. 2019. A survey on image tampering and its detection in real-world photos. *Journal of Visual Communication and Image Representation* 58 (2019), 380–399.

[188] Hang Zhou, Yu Liu, Ziwei Liu, Ping Luo, and Xiaogang Wang. 2019. Talking face generation by adversarially disentangled audio-visual representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 9299–9306.

[189] Yuqian Zhou and Bertram Emil Shi. 2017. Photorealistic facial expression synthesis by the conditional difference adversarial autoencoder. In *Proceedings of the 2017 7th International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 370–376.

[190] Yipin Zhou, Zhaowen Wang, Chen Fang, Trung Bui, and Tamara L. Berg. 2019. Dance dance generation: Motion transfer for Internet videos. *arXiv preprint arXiv:1904.00129* (2019).

[191] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 2223–2232.

[192] Zhen Zhu, Tengteng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. 2019. Progressive pose attention transfer for person image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Received May 2020; revised September 2020; accepted September 2020