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 paper, we explore the creation and detection of deepfakes and provide an in-depth view of 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 which 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: Deepfakes, Deep fakes, 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. 1, 1 (May 2020), 36 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

A deepfake is content generated by artificial intelligence which seems 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 involves the generation and manipulation of human imagery. This technology has creative and productive applications. For example, realistic video dubbing of foreign films, education through the reanimation of historical figures [77], and virtually trying on clothes while shopping. There are also numerous online communities devoted to creating deepfake memes for entertainment, 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 capabilities. At the end of 2017, a Reddit user by the name of ‘deepfakes’ used deep learning to swap faces of celebrities into pornographic videos and posted them online.

1https://variety.com/2019/biz/news/ai-dubbing-david-beckham-multilingual-1203309213/
2www.forbes.com/sites/forbestechcouncil/2019/05/21/gans-and-deepfakes-could-revolutionize-the-fashion-industry/
3https://www.reddit.com/r/SFWdeepfakes/
4https://www.vice.com/en_us/article/gydydm/gal-gadot-fake-ai-porn

Authors’ addresses: Yisroel Mirsky, yisroel@gatech.edu, yisroel@post.bgu.ac.il, Georgia Institute of Technology, 756 W Peachtree St NW, Atlanta, Georgia, 30308, Ben-Gurion University, P.O.B. 653, Beer-Sheva, Israel, 8410501; Wenke Lee, wenke@cc.gatech.edu, Georgia Institute of Technology, 756 W Peachtree St NW, Atlanta, Georgia, 30308.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2020 Association for Computing Machinery.
XXXX-XXXX/2020/5-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

arXiv:2004.11138v2 [cs.CV] 12 May 2020
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 Barack Obama giving a talk on the subject of deepfakes. The video was made using the Reddit user’s software (FakeApp), and it raised concerns over identity theft, impersonation, and the spread of misinformation on social media. In Fig. 1, we present an information trust chart for deepfakes, inspired by [57], which organizes the influence of deepfakes on society.

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 papers published on the subject rose from 3 to over 150 (2018-19).

To understand where the threats are moving and how to mitigate them, we need a clear view of the technologies, challenges, limitations, capabilities, and trajectory. Unfortunately, to the best of our knowledge, there are no other works which present the techniques, advancements, and challenges in a technical and encompassing way. Therefore, the goals of this paper 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 which identifies basic building blocks of the technologies, limitations, and challenges (Section 3). We then provide a chronological and systematic review for each category of deepfake, and provide the network 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, 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 discuss 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 emphasize that deepfakes should not be confused with adversarial machine learning, which is the subject of fooling machine learning algorithms with maliciously crafted inputs (Fig. 2). The difference is that for deepfakes, the objective of the generated content is to fool a human and not a machine.

2 OVERVIEW & 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. Fig. 3 illustrates some examples of facial deepfakes in each of these categories and their sub-types. Throughout this paper we denote $s$ and $t$ as the source and the target identities respectively. 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 $x_t$:

**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 the 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 [41].

**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 [138].

**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 a 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-production, and in educational media where historical figures are reenacted.

**Body** reenactment, a.k.a. pose transfer or human pose synthesis, is similar to the facial reenactments listed above except it is the pose of $x_t$’s body being driven.

Fig. 3. Examples and illustrations of human face reenactment, replacement, editing, and synthesis deepfakes.
The Attack Model. Reenactment deepfakes 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 discreditability, spread misinformation, and tamper with evidence. For example, an attacker can impersonate \( t \) to exploit 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. Finally, the technology can be used to tamper surveillance footage or some other archival imagery in an attempt to plant false evidence in a trial.

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 shortcut to fully reenact \( t \) 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 [117].

2.3 Editing & 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 to edit other 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 [58], 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 [113].

Synthesis is where the deepfake \( x_g \) is created with no target as a basis. Human face and body synthesis techniques such as [66] (used in Fig. 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[11, 26, 56]. 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 which we will be using throughout the paper.
3.1 Neural Networks

Neural networks are non-linear models for predicting or generating content based on an input [52]. They are made up of layers of neurons, where each layer is connected sequentially via synapses. The synapses have associated weights which 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 $x$ is propagated through each layer and an activation function is used to summarize a neuron’s output (e.g., the Sigmoid or ReLU function).

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 $L$ is defined. The loss function is used 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 $L$ 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 to (1) pass information on $x' \in X'$ to the inner layers of $M$ during the feed-forward process, and (2) 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 [7]. There are various types of loss functions which can be applied in different ways depending on the learning objective. For example, when training $M$ as an $n$-class classifier, the 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 ($L_{CE}$) by comparing $y'$ to the ground truth label $y$, and then perform back-propagation to update the weights with the training signal. The loss $L_{CE}$ over the entire training set $X$ is calculated as

$$L_{CE} = -\sum_{i=1}^{[X]} \sum_{c=1}^{n} y_i[c] \log(y'_i[c])$$

where $y'_i[c]$ 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 $L_1 = |x - x_g|^1$ and $L_2 = |x - x_g|^2$. 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. Another approach to compare two unaligned images is to pass them through another network and measure the difference between the layer’s activations (feature maps). This loss is called perceptual loss ($L_{per}$) and, for deepfakes, is often performed using a face recognition network such as VGGFace. Similarly, there is a feature matching loss ($L_{FM}$) which uses the last output of a network, and content loss ($L_{CFM}$) which passes only $x_g$ and measures the difference between the activations of the network’s layers.

3.3 Generative Neural Networks (for deepfakes)

Deepfakes are often created using combinations or variations of six different networks, five of which are illustrated in Fig. 4.

**Encoder-Decoder Networks (ED).** An ED consists of at least two networks, an encoder $En$ and decoder $De$. The ED has narrower layers towards its center so that when it is 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 autoencoder (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 CNN learns pattern hierarchies in the data and is therefore much more efficient at handling imagery. A convolutional layer in a CNN learns filters, which 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 an ED CNN for imagery.

**Generative Adversarial Networks (GAN)** The GAN was first proposed in 2014 by Goodfellow et al. in [53]. A GAN consists of two neural networks which 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:

\[
L_{adv}(D) = \max \log D(x) + \log(1 - D(G(z)))
\]

\[
L_{adv}(G) = \min \log(1 - D(G(z)))
\]

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.

**Image-to-Image Translation (pix2pix).** Numerous variations and improvements on GANs have been proposed over the years. One popular version is the pix2pix framework which enables translations from one image domain to another [62]. 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 [147] for generating high resolution imagery with better fidelity.

**CycleGAN.** An improvement of pix2pix which enables image translation through unpaired training [168]. 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 (\( L_{cyc} \)).
Recurrent Neural Networks (RNN) An RNN is a type of neural network that can handle sequential and variable length data. The network remembers its internal state after processing \( x^i (i - 1) \) and can use it to process \( x^i (i) \) and so on. In deepfake creation, RNNs are often used to handle audio and sometimes video. More advanced versions of RNNs include long short-term memory (LSTM) and gate recurrent 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) [39]. 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 as the model’s input. 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 which can be efficiently tracked using the open source computer vision library (Open CV). 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 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 capture 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 Fig. 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.

![Fig. 5. The processing pipeline for making reenactment and face swap deepfakes. Usually only a subset of these steps are performed.](image-url)
(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. We identify three primary relationships in this
regard: **one-to-one** $x_g = M_t(E_s(x_s))$, **many-to-one** $x_g = M_t(E(x_s))$, and **many-to-many** $x_g = M(E_1(x_s), E_2(x_t))$, where $E$ is some model or process for representing or extracting features from $x$.

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, it is typically much easier to obtain access to
a large dataset of the driving content than a dataset of 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 trans-
ferred 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 occur when 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, such as cropped imagery or inconsistent facial
features, may appear. To mitigate this, works such as [103, 109, 124] perform segmentation
and in-painting on the obstructed areas.

**Temporal Coherence.** Deepfake videos often produce more obvious artifacts, such as flickering
and jitter [142]. This is because most deepfake networks process each frame individually
with no context of the preceding frames. To mitigate this, some researchers either provide
this context to $G$ and $D$, implement temporal coherence losses, use RNNs, or perform a
combination thereof.
| Model | Year | Technique | Target(s) | Model Training | Model Execution | Model Output |
|-------|------|------------|------------|----------------|-----------------|--------------|
| PT-GAN | 2017 | FusionNet, ResNet | portrait | portrait | - | 128x128 |
| Recycle-GAN | 2018 | 5-10 min. video | portrait | - | 512x512 |
| DeepFakeLab | 2019 | 1-3 hr. video | portrait video | - | 512x512 |
| Liu et al. | 2019 | 1-3 hr. video | upperbody video | - | 256x256 |
| Synth. Obama | 2017 | None | audio | portrait video | 2048x1024 |
| ObamaNet | 2017 | None | audio | 256x128 |
| Deep Video Posts | 2018 | None | video neural texture | - | 1024x1024 |
| RemaxGAN | 2018 | None | portrait video | - | 256x256 |
| Vid2vid | 2018 | None | video portrait | - | 2048x1024 |
| Mocogan | 2018 | None | expression label | identity label | 64x64 |
| SD-CGAN | 2018 | None | audio | 128x128 |
| GRN | 2019 | None | 3-10 eye images | - | 64x128 |
| TETH | 2019 | None | text | portrait video | 512x512 |
| N.V. Puppety | 2019 | None | video portrait | - | 512x512 |
| NIK-HAV | 2019 | None | body image | background | 512x512 |
| Deep Video P.C. | 2019 | None | body image | - | 256x256 |
| Everybody D. N. | 2019 | None | body image | - | 256x256 |
| D. D. Generation | 2019 | None | body video | - | 512x512 |
| N. Talking Heads | 2019 | None | portrait/landmarks | 1-3 portraits | 256x256 |
| Few-shot Vid2Vid | 2019 | None | portrait/body/video | 1-10 portraits | 2048x1024 |
| Shumma et al. | 2015 | None | audio face detection | - | - |
| DeepWarp | 2016 | None | eye image | >4x10 |
| CVAE-GAN | 2017 | None | latent variables | portrait | 128x128 |
| REFT | 2017 | None | portrait | - | 256x256 |
| FE-CDEA | 2016 | None | portrait | AU label | 32x32 |
| psGAN | 2018 | None | portrait | neutral | 512x512 |
| XFace | 2018 | None | portrait | 1-3 portraits | 256x256 |
| GaNStoration | 2018 | None | portrait/landmarks | portrait | 128x128 |
| GATH | 2018 | None | portrait/AUs | portrait | 1024x1024 |
| FaceID-GAN | 2018 | None | portrait | - | 128x128 |
| FaceFeat-GAN | 2018 | None | portrait | - | 128x128 |
| CAPG-GAN | 2018 | None | portrait | - | 128x128 |
| DR-GAN | 2018 | None | portrait | - | 1024x1024 |
| Deformable GAN | 2018 | None | body image | 1+ portraits | 96x96 |
| SHUP | 2018 | None | body image | body image | 256x256 |
| DPNG | 2018 | None | body image | body image | 128x64 |
| Dense Pose Tr. | 2018 | None | body image | body image | 256x256 |
| Song et al. | 2018 | None | audio | portrait | 128x128 |
| eg-GAN | 2018 | None | portrait | - | 256x256 |
| FS-GAN | 2018 | None | portrait/landmarks | portrait | 256x256 |
| GANimation | 2018 | None | portrait/AUs | portrait | 128x128 |
| ICFace | 2019 | None | portrait/landmarks | portrait/landmarks | 256x256 |
| FaschSwapNet | 2019 | None | portrait/body | body image | 256x256 |
| Monkey-Net | 2019 | None | pose | 1+ portraits | 64x64 |
| First-Order-Model | 2019 | None | portrait/landmarks | portrait/landmarks | 256x256 |
| M4T-GAN | 2019 | None | expression label | portrait | 64x64 |
| AF-VAE | 2019 | None | portrait/boudaries | portrait | 256x256 |
| Fu et al. 2019 | 2019 | None | portrait/label | - | 1024x1024 |
| FusionNet | 2019 | None | portrait/landmarks | portrait | 256x256 |
| AD-GAN | 2020 | None | pose | portrait | 128x128 |
| Speech D. Amm. 1 | 2019 | None | audio | portrait | 96x128 |
| Speech D. Amm. 2 | 2019 | None | audio | portrait | 96x128 |
| Speech D. Amm. 3 | 2019 | None | audio | portrait | 96x128 |
| SANS | 2019 | None | audio/po portrait | video | 256x256 |
| ATVGet | 2019 | None | audio | portrait video | 128x128 |
| Speech2Vid | 2019 | None | audio | portrait video | 1024x1024 |
| DeNet | 2019 | None | body video | body image | 256x256 |
| C-DGNet | 2019 | None | body image | body/pose image | 4x64 |
| FFAT-PIG | 2019 | None | body image | body image | 256x256 |
| ImagiGator | 2020 | None | expression label | portrait | 4x64 |
| MarioNET | 2020 | None | portrait | 1-8 portraits | 256x256 |

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

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 had been around long before deepfakes were popularized. In 2003, researchers morphed models of 3D scanned heads [18]. In 2005, it was shown how this can be done without a 3D model [24]. 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 ([135] and [136, 137]).

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 [152] used a CycleGAN for facial reenactment, without the need for data pairing. The two domains were video frames of \( x_s \) and \( x_t \). However, to avoid artifacts in \( x_d \), the authors note that both domains must share a similar distributions (e.g., poses and expressions).

In 2018, Bansal et al. proposed a generic translation network based on CycleGAN called RecycleGAN [14]. Their framework improved temporal coherence and mitigated artifacts by including next-frame predictor networks for each domain. For facial reenactment, the authors trained 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 [15] presented CVAE-GAN, a conditional VAE-GAN where the generator is conditioned on an attribute vector or class label. A weakness of reenactment with CVAE-GAN is that it 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 \( t \):\footnote{Although works such as [105] and [166] achieved fully agnostic models (many-to-many) in 2017, their works were on low resolution or partial faces.}

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 [150]. 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 \( x_d \) with a pix2pix-like generator.

Temporal GANs. To improve the temporal coherence of deepfake videos, the authors of [141] developed MoCoGAN: a temporal GAN which 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).

Instead of using an RNN, another way of handling temporal coherence is the network with some temporal context (e.g., the last few frames). The authors of [146] took this approach in 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).
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, E: reconstructed
m: Binary mask, s: Segmentation map, t: Landmark or Keypoint, z: Noise
C: Concatenate, S: Subtract, M: Multiply, A: Add
O: Paste content
δ: Crisp out region a from image where a ∈ {f: face, e: eye, m: mouth}
α: Image x cropped to the region of a ∈ {f: face, e: eye, m: mouth}
γ: Spatial replication of a vector (channel-wise or dim-wise)
X, Y, Z: Image, audio, video
LE, BE, AE, 3DE: Landmark, Boundary, Attention Unit (AU), and 3DMM facial model extractors (open source CV library)
LT, 3DT: Landmark and 3D model transformation, from s to t
ME: MFCC audio feature extractor
b, ℒ, H, 1: Target specific, Loss, Attention, Total Variance, KL Divergence

[x105] GANotation:
FaceFeat-GAN:
GANimation:
FaceID-GAN:
FaceFeat-GAN:

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.
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.
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.

Going one step further, in [166] the authors reenacted full portraits at low resolutions. Their approach was to extract the source and target’s 3D facial models from 2D images using monocular reconstruction, and 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)

The first attempts at identity agnostic models were made in 2017, where the authors of [105] used a conditional GAN (CGAN) for the task. Their approach was to (1) extract the inner-face regions as $(x_f, x_s)$, and then (2) pass them 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 [166] the authors reenacted full portraits at low resolutions. Their approach was to decouple the identities with a conditional adversarial autoencoder, disentangling 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 [27], an architecture similar to CycleGAN but for $N$ domains (poses, expressions, etc).

Later, in 2018, the authors of [108] suggested to drive $x_t$ with continuous action units (AU) as an input, extracted from $x_s$. Their generator, GATH, is an 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 [108], another work called GANimation [109] reenacted faces through AU value inputs estimated from \( x_t \). 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 [114] propose GANnotation which generates \( x_g \) based on \((x_t, l_s)\), where \( l_s \) denotes the facial landmarks of \( x_s \). GANnotation uses the same self-attention model as GANimation, but considers 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 \( l_e \) sampled randomly from the same video, the loss is computed as

\[
\mathcal{L}_{trip} = \|G(x_t, l_s) - G(G(x_t, l_e), l_s)\|^2
\]

(4)

3D Parametric Approaches. Concurrent to the work of [70], other works also leveraged 3D parametric facial models to prevent identity leakage in the generation process. In [120], the authors reenact \( t \) at oblique poses and high resolution. Their framework, FaceID-GAN, is an ED generator 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. 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 is fake.

Later, the authors of [120] advanced FaceID-GAN with FaceFeat-GAN which improves the diversity of the faces while preserving the identity [121]. Their approach was 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. Unfortunately, the feature GANs’ input seeds must be selected empirically to fit \( x_t \).

Instead of passing the 3DMM parameters to the network, one can use the network to refine the realism of a 3D head rendering. paGAN [96] follows this approach, and only needs 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 3D 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 \( x_t \).

Using Multi-Modal Sources. In the work of [149], the authors developed 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 \( u_t \): 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 \( u_s \), crafted to interpolate \( x_t' \) to \( x_g \), having the pose and expression of \( x_s \). During training, first \( L_1 \) 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 tensor-flow interpolation layer to enable back propagation using \( x_t' \) and \( x_g \). The authors also showed how the embedding of the driving network can be mapped to other modalities such as audio and pose.

In 2019, nearly all works pursued identity agnostic models:

Facial Landmark & Boundary Conversion. To mitigate the issue of identity leakage from facial landmarks, the authors of [161], convert the source’s intermediate facial representation to the target’s before passing it through the generator. Their framework, called FaceSwapNet, accomplished this by using two encoders and a decoder to transfer the expression in landmark \( l_t \) to the face structure of \( l_r \), 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.

Another work which transforms facial representations is FSGAN [103], a GAN pipeline which can perform facial reenactment and replacement while handling 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 their evaluations they achieve real time reenactment at 30fps.

Often, the quality of $x_g$ degrades when $x_s$ is at oblique angles. To counter this issue, [49] suggested using 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 utilized a small labeled dataset by training their model in a semi-supervised way.

**Latent Space Manipulation.** When using ED networks, the latent (or encoded) space can be manipulated to control the decoder’s output. The authors of [139] utilized this fact in a framework called ICFace, which enables the user to drive the target’s expression, pose, mouth, eyes, and eyebrows independently. Their architecture is similar to a CycleGAN in that one generator translates $x_t$ into a neutral expression domain as $x_\eta$ and another generator translates $x_\eta$ into an expression domain as $x_g$. Both generators are conditioned on the target AU.

Variational Auto-encoders are a natural choice to controlling the latent space of an ED. For the task of facial reenactment, the authors of [44] proposed the Additive Focal Variational Auto-encoder (AF-VAE). Their model separates a C-VAE’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 used an additive memory module during training which conditions the latent variables on a Gaussian mixture model, fitted to 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 $l_s$ [12]. In [51], the authors proposed wgGAN which uses the same approach but creates high-fidelity facial expressions by refining the warped 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).

To handle large pose changes, [162] suggested that the $x_g$ should instead be blended with a warped version. In their work the authors generated $x_g$ with a decoder using (1) an encoding of $x_t$ and (2) a segmentation map of $x_t$ as reenactment guidance via SPADE residual blocks. Then a separate network uses $x_g$ and the classical warped version to find a blending mask for joining them together.

**Motion-Content Disentanglement.** Several works noted that both motion (i.e., location tracking) and content (e.g., textures) can be decoupled to help the network process new content. Similar to MoCoGAN [141], the authors of [123] decoupled the source’s content and motion in a framework called Monkey-Net. Monkey-Net is a self-supervised network for driving an image with an arbitrary video sequence. 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 [124], the authors extended Monkey-Net by improving the object appearance when large pose transformations occur. They accomplished this by (1) modeling motion around the keypoints 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.

In contrast to MoCoGAN [141] the authors of [148] proposed ImaGINator: a conditional GAN which fuses both motion and content. Their GAN uses 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 disadvantage of [106] and [148] is that they are label driven and generating a fixed number of frames.

The authors of [106] had a different approach to motion-content disentanglement. There, the authors attempt to reenact neutral expression faces with smoother animations than previous works. The authors described 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 a video frames of the target.

4.1.4 Few-Shot Learning. Towards the end of 2019 and into the beginning of 2020, researchers began looking for ways to minimize the amount of required training data, using techniques such as one-shot and few-shot learning.

In [159], the authors developed a few-shot reenactment model which works well at oblique angles. To accomplish this, the authors performed 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.

To obtain temporal few-shot learning, the authors of Vid2Vid (Section 4.1.2) extended their work in [145]. They used a network weight generation module which utilizes an attention mechanism. The module learns to extract appearance patterns from a few sample $x_t$ which are injected into the video synthesis layers.

Often, the quality $x_g$ is poor in a few-shot setting when $x_s$ has a different pose then the reference image(s) in $X_t$. To alleviate identity leakage in these cases, the authors of [55] developed MarioNETte. In contrast to other works which encode the identity separately or use of AdaIN layers, MarioNETte uses 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 [20].

4.2 Mouth Reenactment (Dubbing)

In contrast to expression reenactment, mouth reenactment (a.k.a., video or image dubbing) drives a target’s mouth with a segment of audio. Fig. 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 [131] created a realistic reenactment of former President Obama. This was accomplished by (1) using a time delayed RNN over 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 from other frames into the target video, and (4) using dynamic programming to re-time the target video to match the source audio and blend the texture in.

Later that year, the authors of [76] presented ObamaNet: a network that reenacts an individual’s mouth and voice using text as input. The process is to (1) convert the source text to audio using
[131] Synthesizing Obama:
\[ a^{(i)}, \text{i-th } 1.25 \text{ms segment of audio with a stride of } 10 \text{ms. } \]
\[ \text{MR: mouth retrieval and enhancement based on 3DMM reconstructions. } \]
\[ OS: \text{optical flow extractor} \]

[63] SD-CGAN:
\[ a^{(i)}, \text{i-th } 3.3 \text{ms segment of audio. } \]
\[ p: \text{lips landmarks.} \]

[25] ATVNet:
\[ p: \text{landmarks compressed with PCA. } \]
\[ a^{(i)}: \text{10ms of audio around the } i \text{-th frame.} \]
\[ s_e: \text{attention map. } \]
\[ s_m: \text{motion map.} \]

[64] Speech22Vid:
\[ a^{(i)}, \text{i-th } 350 \text{ms segment of audio with stride } 40 \text{ms} \]

[133] Neural Voice Puppetry:
\[ a^{(i)}, \text{i-th } 300 \text{ms segment of audio with stride } 20 \text{ms. } \]
\[ p: \text{content aware filter network} \]
\[ NT: \text{Neural Texture Extractor} \]

[165] DAVS:
\[ a^{(i)}, \text{i-th } 300 \text{ms segment of audio containing a word. } \]
\[ y_i: \text{identity label.} \]
\[ x_i: \text{word label (one-hot encoding)} \]

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.

Char2Wav [127], (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. [63] created a network that synthesizes the entire head portrait of Obama, and therefore does not require pose re-timing like [76, 131]. 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 in 2019, the authors of [48] 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 [70], (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 [133] had a different approach: (1) animate a 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 [134] to generate the mouth region, and then (3) use a network to blend the mouth into the original frame. The authors found that their approach only requires 2-3 minutes of video of the target to produce very realistic results.

4.2.2 **Many-to-Many (Multiple IDs to Multiple IDs).** One of the first works to perform identity agnostic video dubbing was [122]. There, the authors used an LSTM to map MFCC audio 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 [126] 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.

Since the audio is processed in segments, lip-sync artifacts can occur. The authors of [155] tried to mitigate this issue by adding a textual context. There, a time-delayed LSTM was used to predict mouth landmarks given MFCC segments and the spoken text using a text-to-speech model. The target frames were then converted into sketches using an edge filter and the predicted mouth shapes were composited into them. Finally, a pix2pix like GAN with self-attention was used to generate the frames with both video and image conditional discriminators.

The authors of [25] stated that direct models are problematic since the model learns irrelevant correlations between the audiovisual signal and the speech content. To avoid this issue, an LSTM audio-to-landmark network and a landmark-to-identity CNN-RNN are used in sequence, where the facial landmarks are compressed with PCA and the attention mechanism from [109] is used. To improve synchronization, the authors recommended using a regression based discriminator which considers both sequence and content information.

**EDs for Preventing Identity Leakage.** The authors in [165] mitigated 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 showed how $x_g$ can be driven using $x_s$ or $a_s$.

In [64], the authors proposed 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.

In [142], the authors developed a dubbing network which can also generate facial expressions and blinking. Their approach was to 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.

Later in [143], the same authors improved 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 [67], the authors tuned 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 [143] and [64].

4.3 Pose Reenactment

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

In [60] the authors used 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 [138], the authors considered another approach for the task of pose-invariant face recognition. Their pipeline, called DR-GAN, uses 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 sources pose vector and \(z\) is a noise vector. The authors showed that the quality of \(x_g\) can be improved by averaging multiple examples of the identity encoding before passing it through the decoder. In [22], the authors suggested using two GANs: The first frontalizes the face and produces a UV map, and the second rotates the face, given the target angle as an injected embedding.

4.4 Gaze Reenactment

There are only a few deep learning works which have focused on gaze reenactment. In [50] the authors converted a cropped eye \(x_t\), its landmarks, and the source angle, to a flow (vector) field using a 2-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 corrected the illumination of \(x_g\) with a second CNN. In [157], the authors published the Gaze Redirection Network (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-trained their network on 3D synthesized examples, (2) further tuned their network on real images, and then (3) fine-tuned their network on 3-10 examples of the target.

4.5 Body Reenactment

Several facial reenactment papers from Section 4.1 discuss body reenactment as well, including: Vid2Vid [145, 146], MocoGAN [141], and others [123, 124]. In this section, we focus on methods which specifically target body reenactment. Schematics for some of these architectures can be found in Fig. 10.

4.5.1 One-to-One (Identity to Identity). In the work [89], the authors performed 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) paste them onto a captured pose skeleton of the source, and then (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 [23] the authors made 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.

In some cases, the method of [23] would produce artifacts such as stretched limbs due to incorrectly detected pose skeletons. To mitigate for this, the authors of [88] 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.

Further artifacts can occur when a body reenactment network attempts to generating poses which were unseen training. The authors of [2] alleviate this issue by having the network generalize to many other than s or t. This was accomplished by first training the GAN on paired data (the same identity doing different poses) and then later adding another discriminator that evaluates the temporal coherence given (1) \( x_{g}^{(i)} \) driven by another video, and (2) the optical flow predicted version.

To reduce the amount of training content for the target, the authors of [167] proposed a method that only requires three minutes of video: First, the target’s body are segmented by limb and the pieces are oriented to the source’s pose. Then a pix2pixHD GAN uses 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

Sometimes there are issues of misalignment when using translation networks, such as pix2pix, for reenactment. To resolve this issue, the authors of [125] suggested the use of novel ‘deformable skip connections’ which orients the shuttled feature maps according to the source pose. The authors also used a novel nearest neighbor loss instead of using L1 or L2 losses. To modify unseen identities at test time, the authors pass an encoding of \( x_{t} \) to the decoder’s inner layers.
To ensure that patterns, shapes, and textures are mapped realistically between frames, the authors of [169] developed a GAN which uses novel Pose-Attentional Transfer blocks (PATB) inside the generator. The architecture passes $x_t$ and the poses $p_s$ concatenated with $p_t$ through separate encoders which are passed through 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.** Similar to the image warping techniques used for facial reenactment, some authors also tried to reenact bodies. For example, in [99] the authors used 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. Another example is [158] where the authors proposed DwNet. There the authors tried to alleviate misalignment artifacts between $x_s$ and $x_t$ by using warping. To accomplish this, the authors used 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 a DensePose network.

**Background Foreground Compositing.** Some researchers noticed that the background of a video interferes with the generated foreground, and can contain temporal distortions. Therefore, some works used segmentation and compositions to focus the network on the relevant components. In [13], the authors broke the process down into three stages, trained end-to-end: (1) use a U-Net to segment $x_s$’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 [42] used an ED GAN network to disentangle the foreground appearance (body), background appearance, and pose to gain control over these aspects. This is accomplished by segmenting each of these aspects before passing them through encoders. In [32] the authors used a CVAE-GAN, conditioned on heatmaps of the detected pose and skeleton. As a result, the authors were able to change the pose and appearance of bodies individually.

### 4.5.4 Few-Shot Learning.

In [78], the authors demonstrated the few-shot learning technique of [46] on a pix2pixHD network and the network of [13]. 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 Fig. 12 and Table 2 respectively.

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 [19] and [17]. Later, fully automated methods were proposed in [31, 68] and [104] 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. 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 image.
face image. As a result, \( E_n \) learns to map \( s \) and \( t \) to a shared latent space, such that

\[
D_e(E_n(x_t)) = x_g
\]  

(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 [61]. 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 [118] follows a similar architecture, but uses a denoising autoencoder with a self-attention mechanisms, and offers cycle-consistency loss. The decoders in FaceSwap-GAN also generate segmentation masks which help the model handle occlusions and is used to blend \( x_g \) back into the target frame. Finally, [1] is another open-source tool that provides a GUI. Their software comes with 10 popular implementations, including that of [61], and multiple variations of the original Redit user’s code.

5.1.2 One-to-Many (Single Identity to Multiple Identities).

Some researchers tried to apply the technique of style transfer to perform a face swap. The concept is that an individual’s identity can be captured as a style which can then be applied to other fitting objects (heads). In [75], the authors applied this idea with a modified style transfer CNN, where the content was \( x_t \) and the style was the identity of \( x_s \). The process was to (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 [105], mentioned in Section 4.1.3. However, to train the CGAN, one needs a dataset of paired faces with different identities having the same expression.

Disentanglement with EDs. A few works created many-to-many face swapping networks by disentangling and then modifying an ED’s latent space. In [16] the approach was to disentangle the identity from the attributes (pose, hair, background, and lighting) during the training process. The identity encodings were the last pooling layer of a face classifier, and the attribute encoder was trained using a weighted L2 loss and a KL divergence loss to mitigate identity leakage. The authors also showed that they can adjust attributes, expression, and pose via interpolation of the encodings.

In [130] the authors proposed a system for identity obfuscation which swaps \( x_t \)’s face with a similar or different identity. An ED is used to predict the 3D head parameters with are then either modified or replaced with the source’s. Finally, a GAN is used to in-paint the face of \( x_t \) given the head model parameters.

Disentanglement with VAEs. Other authors took the disentanglement concept further by using VAEs. For example, the authors of [98] developed 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 \( c \).

Another example is [97] where the authors created a VAE ED network called FSNet. FSNET 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.

In many cases, \( x_t \) or \( x_s \) is partially covered with hair, a hand, or some other object. This results in inconsistencies and artifacts in \( x_q \) (e.g., varying eye color). To handle occlusions during face swapping, the authors of [80] developed FaceShifter: First, the last layer of a face recognition classifier is used to encode the identity of \( x_t \). Then the encoding is then passed to a generator: a series of novel Adaptive Attentional Denormalization layers (AAD) inside ResBlocks. The generator also receives localized information of \( x_t \) via layer concatenations and \( c \). Since VAEs are used, the facial attributes describing \( x_t \) are partially changed. To perform occlusions, a second network predicts the target’s segmentation mask \( m_f \). Then \((x_t^f, m_f)\) 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.

#### 5.1.4 Few-Shot Learning.

The author of FaceSwap-GAN [118] also hosted a few-shot approach online dubbed “One Model to Swap Them All” [119]. In this version, the generator receives \((x_s^f, x_t^f, m_t)\), 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.
Fig. 12. Architectural schematics of the replacement networks with their generation and training dataflows.
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 face transfer is equivalent to performing self-reenactment on a face swapped portrait. Therefore, high quality face transfers can be achieved by combining methods from Section 4.1 and Section 5.1.

In 2018, the authors of [95] proposed DepthNets: an unsupervised network for capturing facial landmarks and translating the pose from one identity to another. The authors used 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 \( x_s^{(f)} \) to the source pose \( I_l \), and the composition is refined using a CycleGAN. Since warping is involved, the approach is sensitive to occlusions.

Later in 2019, the authors of [151] created a self-supervised network which can change the identity of an object within an image. Their ED disentangles the identity from an object’s pose using 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 [85] proposed a method disentangled identity transfer. However, neither [151] or [85] 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 [164]. In our review of detection methods, we will focus on works which specifically deal with detecting deepfakes of humans.

6.1.1 Artifact-based Detection. Deepfakes often generate artifacts which may be subtle to humans, but can be easily detected using machine learning. In 2014, researchers suggested this hypothesis and monitored physiological signals, such as heart rate, to detect computer generated faces [29]. Regarding deepfakes, [82] monitored irregular eye blinking patterns and [28] monitored blood volume patterns (pulse) under the skin.

Inconsistencies are also a revealing factor. In [74] and [43], the authors noticed that video dubbing attacks can be detected by correlating the speech to landmarks around the mouth. In [154] it was shown that similar artifacts appear when predicting the facial landmarks. With large amounts of data on the target, mannerisms and other behaviors can be monitored for anomalies. For example, in [5] the authors protected world leaders from a wide variety of deepfake attacks by modeling their recorded stock footage.

Some artifacts appear where the generated content was blended back into the frame. The authors of [4, 8, 38, 94, 163] used edge detectors, quality measures, and frequency analysis to detect artifacts in the pasted content and borders. In [81] the authors detected deepfakes by decomposing them into to their sources while identifying the content’s boundary. Some works identify and visualize the tampered regions by either predicting masks learned from a ground truth, or by mapping the neural activations to the raw image [37, 79, 100, 128].

The content of a fake face can be anomalous in context to the rest of the frame. For example, residuals from face warping processes [83, 84, 86], lighting [129], and varying fidelity [73] indicate the presence of generated content. In [156] and [91], 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 [72] the authors analyzed a camera’s unique sensor noise (PRNU) to detect pasted content.

As noted in Section 4.1, realistic temporal coherence is challenging to generate, and some authors capitalized on the resulting artifacts to detect the fake content. For example, [54] uses an RNN to detect artifacts such as flickers and jitter, and [112] uses an LSTM on the face region only. In [10] the optical flow between frames is analyzed, and in [23] a classifier is trained on the two frames directly.

### 6.1.2 Undirected Approaches

Instead of focusing on a specific artifact, some authors trained deep neural networks as generic classifiers, and let the network decide which features to analyze [3, 34, 35, 45, 132]. In [90, 101, 111], it was shown that deep neural networks tend to perform better than traditional image forensic tools on compressed imagery. Alternatively, to overcome noise and other distortions, the authors of [144] measured the neural activation (coverage) of a face recognition network to obtain a stronger signal from than just using the raw pixels. In [16] the authors detected deepfakes by measuring an input’s embedding distance to real samples using an ED’s latent space. To improve performance of their models, some add noise to their training data [153] and others try to develop models which generalize better to unseen attacks/generators through disentanglement [30] and semi-supervised learning [140].

### 6.2 Prevention & Mitigation

To prevent deepfakes, some have suggested that data provenance of multimedia should be tracked through distributed ledgers and blockchain networks [47]. In [40] the authors suggested that the content should be ranked by participants and AI. In contrast, [59] proposed that the content should be authenticated and managed as a global file system over Etherium smart contracts. To combat deepfakes, the authors of [87] showed how adversarial machine learning can be used to disrupt and corrupt deepfake networks. The authors performed adversarial machine learning to add crafted noise perturbations to $x$, which prevents deepfake technologies from locating a proper face in $x$.

## 7 DISCUSSION

### 7.1 Creation

Over the last few years there has been a shift towards 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., [48, 133, 147]).

Significant progress in this domain was made when researchers began using perceptual loss with 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, 118]. 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 gives more control over the final output and has been able to mitigate most of the challenges mention in Section 3.7.

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 look-alike’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
Table 3. Summary of Deepfake Detection Models

| Type Modality | Content | Method | Eval. Dataset | Performance* |
|---------------|---------|--------|---------------|--------------|
| Reenactment   | Image   | SVM-RBF| 250x250       | 92.9         |
| Replacement   | Video   | SVM    | *             | 18.2         |
|               | Audio   | SVM    | *             | 0.97         |
|               | Feature | SVM    | 128x128       | 3.33         |
|               | Body Part| SVM   | 1024x1024     | 100          |
|               | Face    | SVM    | *             | 13.33        |
|               | Image   | SVM    | *             | 0.98         |
|               | Model   | FaceForensics [110] | 2017 | 92.9 |
|               |         | FaceForensics++ [111] | 2017 | 92.9 |
|               |         | FFW [69] | 2017 | 92.9 |
|               |         | Custom | ACC | 0.99 |
|               |         | DFFD [128] | 2017 | 0.99 |
|               |         | DeepfakeTIMIT [73] | 2017 | 0.99 |
|               |         | DeepfakeDB | 2017 | 0.99 |
|               |         | TPR=1 @FPR= 0.03 | 2017 | 0.99 |

*Only the best reported performance of each paper is displayed to capture the "best-case" scenario.

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 [64, 103] have achieved real-time deepfakes at 30fps. Although real-time deepfakes are an enabler for phishing attacks, the realism is not quite there yet. Regardless, deepfakes are already very convincing [111] 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 cybersecurity, there is an arms race between the attacker and defender. In our SoK, 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 [30]. Moreover, based on the recent benchmark of [86], 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 [102], DFDC by Facebook [36], and Celeb-DF by [86]) were released within one month of each other at the end of 2019. However, the deepfake detectors only achieved an AUC of 0.86, 0.76, and 0.66 on each of them 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 [28, 82] by adding a discriminator which monitors these signals. To avoid anomalies in extensive neuron activation [144], the adversary can add a loss which minimizes neuron coverage. Methods which detect abnormal poses and mannerisms [5] can be evaded by reenacting the entire head and by learning the mannerisms from the same databases. Models which identify blurred content [94] are affected by noise and sharpening GANs [63, 71], and models which search for the boundary where the face was blended in [4, 8, 38, 81, 94, 163] do not work on deepfakes passed through refiner networks or those which output full frames (e.g., [70, 96, 167] and [88, 158]). Finally, solutions which search for forensic evidence [72, 91, 156] can be evaded (or at least raise the false alarm rate) by passing \( x_d \) through filters, or by performing physical replication or compression.

Evading Deep Learning Classifiers. Deep convolutions neural networks are very good at face recognition, achieving state-of-the-art performance [92, 116]. This is because the hierarchical filters in a CNN can effectively and efficiently extract and match visual patterns (e.g., parts of the face). Therefore, it is understandable why many works have applied deep learning directly to the task of deepfake detection (e.g., [3, 34, 35, 45, 132]). However, by hiding or planting subtle patterns in an image, one can influence the output of the filters and alter the prediction. This attack, known as adversarial machine learning, can be applied by an adversary on \( x_d \) to evade detection. Research has also shown that these small adversarial perturbations transfer across multiple models regardless of the training data used [107]. Although more research is required to demonstrate that this attack works well on deepfakes, it is entirely feasible and should be addressed in future work.

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, including the establishment of content provenance and authenticity frameworks for online videos [40, 47, 59], and proactive defenses such as the use of adversarial machine learning to protect content from tampering [87].

7.3 Deepfakes in other Domains

In this SoK, we focus on human reenactment and replacement attacks; the type of deepfakes which has made the largest impact so far [11, 56]. However, deepfakes extend beyond human visuals, and have spread to many other domains. In healthcare, the authors of [93] showed how deepfakes can be used to inject or remove medical evidence in CT and MRI scan for insurance fraud, disruption, and physical harm. In [65] 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 [33]. The authors of [21] have shown how deep learning can generate realistic human fingerprints that can unlock multiple users’ devices. In [115] 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 which has generated some concern [160].

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 [65]. Therefore, it 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 of 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, finding a bound on a model’s delay can help detect real-time attacks such as [65], and determining the limits of GANs like [6] can help us devise the appropriate strategies. As mentioned earlier, we recommend further research on solutions which 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 SoK 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. https://github.com/deepfakes/faceswap. (Accessed on 01/27/2020).
[2] Kfir Aberman, Mingyi Shi, Jing Liao, D Lischinski, 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 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 2017 IEEE International Joint Conference on Biometrics (IJCB). IEEE, 659–665.
[5] 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.
[6] Sakshi Agarwal and Lav R Varshney. 2019. Limits of Deepfake Detection: A Robust Estimation Viewpoint. arXiv preprint arXiv:1905.03493 (2019).
The Creation and Detection of Deepfakes: A Survey

[34] 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).

[35] Nhu-Tai Do, In-Seop Na, and Soo-Hyung Kim. 2018. Forensics Face Detection From GANs Using Convolutional Neural Network.

[36] Brian Dollhansky, Russ Howes, Ben Pflaus, Nicole Baram, and Cristian Canton Ferrer. 2019. The Deepfake Detection Challenge (DFDC) Preview dataset. arXiv preprint arXiv:1910.08854 (2019).

[37] Mengnan Du, Shiva Pentyala, Yuening Li, and Xia Hu. 2019. Towards Generalizable Forgery Detection with Locality-aware AutoEncoder. arXiv preprint arXiv:1909.05999 (2019).

[38] Ricard Durall, Margret Keuper, Franz-Josef Pfreundt, and Janis Keuper. 2019. Unmasking DeepFakes with simple Features. arXiv preprint arXiv:1911.00686 (2019).

[39] P Ekman, W Friesen, and J Hager. 2002. Facial action coding system: Research Nexus. Network Research Information, Salt Lake City, UT 1 (2002).

[40] Chi-Ying Chen et al. 2019. A Trusting News Ecosystem Against Fake News from Humanity and Technology Perspectives. In 2019 19th International Conference on Computational Science and Its Applications (ICCSA). IEEE.

[41] 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.

[42] Liqian Ma et al. 2018. Disentangled person image generation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 99–108.

[43] Pavel Korshunov et al. 2019. Tampered Speaker Inconsistency Detection with Phonetically Aware Audio-visual Features. In International Conference on Machine Learning.

[44] Shengju Qian et al. 2019. Make a Face: Towards Arbitrary High Fidelity Face Manipulation. In Proceedings of the IEEE International Conference on Computer Vision.

[45] 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).

[46] 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.

[47] Paula Fraga-Lamas and Tiago M Fernandez-Carame. 2019. Leveraging Distributed Ledger Technologies and Blockchain to Combat Fake News. arXiv preprint arXiv:1904.05386 (2019).

[48] 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).

[49] 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).

[50] Yaroslav Ganin, Daniil Kononenko, Diana Sungatullina, and Victor Lempitsky. 2016. Deepwarp: Photorealistic image resynthesis for gaze manipulation. In European conference on computer vision. Springer.

[51] 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.

[52] I. Goodfellow, Y. Bengio, and A. Courville. 2016. Deep Learning. MIT Press. 163–174 pages.

[53] 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.

[54] 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.

[55] 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.

[56] Holly Kathleen Hall. 2018. Deepfake Videos: When Seeing Isn’t Believing. Cath. UJL & Tech 27 (2018), 51.

[57] Karen Hao. 2019. The biggest threat of deepfakes isnâĂŹt the deepfakes themselves – MIT Technology Review. https://www.technologyreview.com/2019/10/10/132667/the-biggest-threat-of-deepfakes- isnt-the-deepfakes-themselves/. (Accessed on 05/10/2020).

[58] Karen Hao. 2019. The biggest threat of deepfakes isnâĂŹt the deepfakes themselves - MIT Tech Review. https://www.technologyreview.com/s/614526/the-biggest-threat-of-deepfakes- isnt-the-deepfakes-themselves/.

[59] Haya R Hasan and Khaled Salah. 2019. Combating Deepfake Videos Using Blockchain and Smart Contracts. IEEE Access 7 (2019), 41596–41606.

[60] 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.

[61] Iperov. 2019. DeepFaceLab: DeepFaceLab is a tool that utilizes machine learning to replace faces in videos. https://github.com/iperov/DeepFaceLab. (Accessed on 12/31/2019).
[62] 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.

[63] Seyed Ali Jalalifar, Hosein Hasani, and Hamid Aghajan. 2018. Speech-driven facial reenactment using conditional generative adversarial networks. arXiv preprint arXiv:1803.07461 (2018).

[64] 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.

[65] 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.

[66] 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).

[67] 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).

[68] Ira Kemelmacher-Shlizerman. 2016. Transfiguring portraits. ACM Transactions on Graphics (TOG) 35, 4 (2016), 94.

[69] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. 2019. DSP-FWA: Dual Spatial Pyramid for Exposing Face Warp Artifacts in DeepFake Forensics. In Conference: IMVIP.

[70] 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.

[71] 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.

[72] Marissa Koopman, Andrea Macarulla Rodriguez, and Zeno Geradts. 2018. Detection of Deepfake Video Manipulation. In Conference: IMVIP.

[73] Pavel Korshunov and Sebastien Marcel. 2018. Deepfakes: a new threat to face recognition? assessment and detection. arXiv preprint arXiv:1812.08685 (2018).

[74] Pavel Korshunov and Sebastien Marcel. 2018. Speaker inconsistency detection in tampered video. In 2018 26th European Signal Processing Conference (EUSIPCO). IEEE, 2375–2379.

[75] 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.

[76] Alireza Khodabakhsh, Raghavendra Ramachandra, Kiran Raja, Pankaj Wasnik, and Christoph Busch. 2018. Face Fake Detection Methods: Can They Be Generalized?. In 2018 International Conference of the Biometrics Special Interest Group (BIOSIG). IEEE, 1–6.

[77] Hyeyoung 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.

[78] Dami Lee. 2019. Deepfake Salvador DalÃŋ takes selfies with museum visitors - The Verge. https://bit.ly/3cEim4m.

[79] Jessica Lee, Deva Ramanan, and Rohit Girdhar. 2019. MetaPix: Few-Shot Video Retargeting. arXiv preprint arXiv:1910.04742 (2019).

[80] Jia Li, Tong Shen, Wei Zhang, Hui Ren, Dan Zeng, and Tao Mei. 2019. Zooming into Face Forensics: A Pixel-level Analysis. arXiv:1912.05790 (2019).

[81] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, and Fang Wen. 2019. FaceX-ray for More General Face Forgery Detection. arXiv preprint arXiv:1912.13458 (2019).

[82] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. 2019. Face X-ray for More General Face Forgery Detection. arXiv preprint arXiv:1912.13458 (2019).

[83] Yuezun Li and Siwei Lyu. 2019. Exposing DeepFake Videos By Detecting Face Warping Artifacts. In Conference: IMVIP.

[84] Yuezun Li and Siwei Lyu. 2019. Exposing DeepFake Videos By Detecting Face Warping Artifacts. In Conference: IMVIP.

[85] Yuezun Li and Siwei Lyu. 2019. DSP-FWA: Dual Spatial Pyramid for Exposing Face Warp Artifacts in DeepFake Forensics. In Conference: IMVIP.

[86] Yuezun Li and Siwei Lyu. 2019. Exposing DeepFake Videos By Detecting Face Warping Artifacts. In Conference: IMVIP.

[87] 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).

[88] 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.
[114] Enrique Sanchez and Michel Valstar. 2018. Triple consistency loss for pairing distributions in GAN-based face synthesis. arXiv preprint arXiv:1811.03492 (2018).

[115] Marco Schreyer, Timur Sattarov, Bernd Reimer, and Damian Borth. 2019. Adversarial Learning of Deepfakes in Accounting. arXiv preprint arXiv:1910.03810 (2019).

[116] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition. 815–823.

[117] Oscar Schwartz. 2018. You thought fake news was bad? – The Guardian. https://www.theguardian.com/technology/2018/nov/12/deep-fakes-fake-news-truth. (Accessed on 03/02/2020).

[118] shoaanlu. 2018. faceswap-GAN: A denoising autoencoder + adversarial losses and attention mechanisms for face swapping. https://github.com/shoaanlu/faceswap-GAN. (Accessed on 12/17/2019).

[119] Shaoanlu. 2019. fewshot-face-translation-GAN: Generative adversarial networks integrating modules from FUNiT and SPADE for face-swapping. https://github.com/shaoanlu/fewshot-face-translation-GAN. (Accessed on 12/17/2019).

[120] Yujun Shen, Ping Luo, Junjie Yan, Xiaogang Wang, and Xiaou 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.

[121] Yujun Shen, Bolei Zhou, Ping Luo, and Xiaou Tang. 2018. FaceFeat-GAN: a Two-Stage Approach for Identity-Preserving Face Synthesis. arXiv preprint arXiv:1812.01288 (2018).

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

[123] Aliaksandr Sirarhine, 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.

[124] Aliaksandr Sirarhine, 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

[125] Aliaksandr Sirarhine, Enver Sanguineto, 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.

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

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

[128] Joel Stéhouwer, Hao Dang, Feng Liu, Xiaoming Liu, and Anil Jain. 2019. On the Detection of Digital Face Manipulation. arXiv preprint arXiv:1910.01717 (2019).

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

[130] 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).

[131] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. 2017. Synthesizing obama: learning lip patterns and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition. 815–823.

[132] 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.

[133] 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).

[134] Justus Thies, Michael Zollhofer, and Matthias Niessner. 2019. Deferred Neural Rendering: Image Synthesis using Neural Textures. arXiv preprint arXiv:1904.12356 (2019).

[135] 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.

[136] 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.

[137] 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), 164.

[138] 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.

[139] Soumya Tripathy, Juho Kannala, and Esa Rahtu. 2019. ICFace: Interpretable and Controllable Face Reenactment Using GANs. arXiv preprint arXiv:1904.01909 (2019).
[140] 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:1905.05633 (2019).

[141] 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.

[142] 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.

[143] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. 2019. Realistic Speech-Driven Facial Animation with GANs. arXiv preprint arXiv:1906.06337 (2019).

[144] 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).

[145] 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).

[146] 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).

[147] 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.

[148] Yaohui Wang, Piotr Bilinski, Francois Bremond, and Antitza Dantcheva. 2020. ImaGINator: Conditional Spatio-Temporal GAN for Video Generation.

[149] 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.

[150] 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.

[151] 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.

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

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

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

[155] Lingyun Yu, Jun Yu, and Qiang Ling. 2019. Mining Audio, Text and Visual Information for Talking Face Generation. In 2019 IEEE International Conference on Data Mining (ICDM). IEEE, 787–795.

[156] 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.

[157] 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.

[158] 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).

[159] 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).

[160] 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. http://papers.nips.cc/paper/9106-defending-against-neural-fake-news.pdf

[161] 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).

[162] Xinsheng Xuan, Bo Peng, Wei Wang, and Jing Dong. 2019. On the generalization of GAN image forensics. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 603–619.

[163] Ying Zhang, Lilei Zheng, and Vrizlynn LL Thing. 2017. Automated face swapping and its detection. In 2017 IEEE 2nd International Conference on Signal and Image Processing (ICISP). IEEE, 15–19.

[164] Lilei Zheng, Ying Zhang, and Vrizlynn LL 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.

[165] 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.
[166] Yuqian Zhou and Bertram Emil Shi. 2017. Photorealistic facial expression synthesis by the conditional difference adversarial autoencoder. In 2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII). IEEE, 370–376.

[167] 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).

[168] 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.

[169] 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.