IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

*Digital Object Identifier 10.1109/ACCESS.xxx.DOI*

# DeepFake Detection for Human Face Images and Videos: A Survey

**ASAD MALIK[1] (Member, IEEE), MINORU KURIBAYASHI[2] (Senior Member, IEEE), SANI M. ABDULLAHI[3] (Member, IEEE) and AHMAD NEYAZ KHAN[4] (Member, IEEE)**

[1]Department of Computer Science, Aligarh Muslim University, Aligarh, 202002, India
[2]Department of Electrical and Communication Engineering, Okayama University, Okayama 7008530, Japan
[3]College of Computer and Information Technology, China Three Gorges University, Yichang, 443002, China
[4]Department of Computer Application, Integral University, Lucknow, 611731, India

Corresponding author: A. Malik (amalik@myamu.ac.in)

This research was supported by JSPS KAKENHI Grant Number 19K22846, JST SICORP Grant Number JPMJSC20C3, and JST CREST Grant Number JPMJCR20D3, Japan

**ABSTRACT** Techniques for creating and manipulating multimedia information have progressed to the point where they can now ensure a high degree of realism. DeepFake is a generative deep learning algorithm that creates or modifies face features in a superrealistic form, in which it is difficult to distinguish between real and fake features. This technology has greatly advanced and promotes a wide range of applications in TV channels, video game industries, and cinema, such as improving visual effects in movies, as well as a variety of criminal activities, such as misinformation generation by mimicking famous people. To identify and classify DeepFakes, research in DeepFake detection using deep neural networks (DNNs) has attracted increased interest. Basically, DeepFake is the regenerated media that is obtained by injecting or replacing some information within the DNN model. In this survey, we will summarize the DeepFake detection methods in face images and videos on the basis of their results, performance, methodology used and detection type. We will review the existing types of DeepFake creation techniques and sort them into five major categories. Generally, DeepFake models are trained on DeepFake datasets and tested with experiments. Moreover, we will summarize the available DeepFake dataset trends, focusing on their improvements. Additionally, the issue of how DeepFake detection aims to generate a generalized DeepFake detection model will be analyzed. Finally, the challenges related to DeepFake creation and detection will be discussed. We hope that the knowledge encompassed in this survey will accelerate the use of deep learning in face image and video DeepFake detection methods.

**INDEX TERMS** Deep learning, DeepFake, CNNs, GANs.

## I. INTRODUCTION

FAKE document detection is not a new issue. Rather, this issue has existed for quite some time. In the past, the process of legitimizing documents was confined to proofing, verification, and inquiry, and digital data had no significant role in this process. The recent growth of digital data throughout the Internet, as well as its relevance in everyday life, such as digital marketing, legal forensics imagery, medical imagery, sensitive satellite image processing, and many other applications, cannot be overlooked. Moreover, digital data in different applications are evolving in such a way that they are also fueling an uptick in cybercrime. In this context, the trend indicates serious vulnerabilities and a decrease in the trustworthiness of digital data. Furthermore, discerning whether the acquired digital data are authentic or altered and legitimizing digital documents are currently major problems.

Multimedia forensics research [1] has been active for at least 15 years and comes from not only research communities but also major IT businesses and government organizations. The U.S. Department of Defense's Defense Advanced Research Projects Agency (DARPA) established the large-scale Media Forensic project (MediFor) in 2016 to encourage research on media integrity, with significant results in terms of methodologies and benchmark datasets. Digital media confirmation may check for physical, digital, and semantic integrity, according to the MediFor taxonomy. Deep learning models' efficacy can no longer be overlooked; in fact, they are gradually replacing most technology and are being

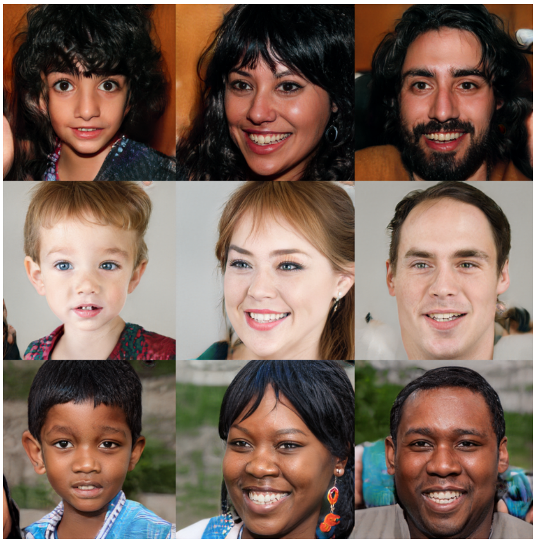

FIGURE 1: An example of Style-GAN [4] images.

rapidly embraced by many research communities and large IT firms.

The combination of deep learning and computer vision techniques, e.g., GANs [2] and autoencoders [3], has opened the door to producing superrealistic fake images and videos, which are known as DeepFakes. DeepFakes (a combination of the terms "deep learning" and "fakes") allow attackers or even nontechnical machine learning users to modify a picture or video by swapping out the content and generating a new image or video that cannot be differentiated by humans or computers. The creation of DeepFakes reduces people's trust in digital media content since they can no longer believe the images they are seeing. In the absence of deep learning, research on identifying or detecting fake manipulated media is considered traditional research.

At present, generative deep models are very powerful for creating DeepFakes, which are difficult to distinguish by traditional methods. This gap creates the need for DeepFake detection research to maintain people's trust in digital multimedia. For example, FaceSwap[1] is a technology that creates DeepFake videos of genuine individuals performing fictional activities, with even humans having difficulties differentiating what is fake from what is authentic. These technologies can cause distress for and negatively affect those who are targeted, promote disinformation and hate speech, and even heighten political tensions, spark controversy, terrorism, or violence. An example of different fake images generated by Style-GAN [4] is shown in Figure 1, which looks very realistic. The AI-based generation of DeepFakes has a wide range of applications in the computer vision and graphics industries, including human face synthesis and stunning scenery production. This breakthrough, however, is vulnerable to misuse. Many people with sinister intentions have utilized these technologies to make fake videos of female celebrities and members of the general public in ways that have created significant societal issues. According to recent research[2], 96 percent of DeepFakes come from porn films. Due to the lack of supporting data, the recognition of these DeepFakes or fabricated images/videos[3] is difficult. Many malicious applications have made use of DeepFakes, such as DeepNude[4], as they can take a fully dressed woman's photograph and generate an image with her clothes removed.

Because of the use of deep learning to construct DeepFakes and web-based tools to quickly create DeepFakes, forgery detection is extremely difficult for forensics professionals. Thus, researchers are developing a DNN model to detect DeepFakes.

In essence, the model is trained on DeepFake datasets and then tested in trials to see how well it performs. We will discuss picture and video DeepFake detection techniques in depth in this article. We will also review the DeepFake production methods and datasets that are employed to detect DeepFakes. Recently, studies based on DeepFake generation and detection in pictures, audio, and videos [5]–[9] have been published.

The main goals of this article are highlighted below:

- to introduce DeepFake tools that are used to manipulate the different aspects of images and videos;
- to introduce DeepFake datasets and some traditional datasets for forensic evaluation; and
- to review some recent existing DeepFake detection techniques used in images and videos.

The review starts with providing a technical background in Section II. Then, DeepFake tools and applications are discussed in Section III, and Section IV proceeds to understand the types of manipulation methods. Section V discusses the available image and video datasets and their fungibility. A brief survey of image and video detection methods is presented in Section VI. Then, additional major challenges for DeepFake creation and detection are discussed in Section VI, and conclusions are drawn in Section VII.

## II. TECHNICAL BACKGROUND

### A. CNN background

The CNN or ConvNet is a special kind of deep-learning architecture that has gained much attention in computer vision and robotics. The initial idea of CNN, called neocognitron, was presented in 1979 by Kunihiko Fukushima [10], which later became known as the predecessor of CNN. Furthermore, the CNN architecture has been explained by Le-Cun *et al.* [11]; later, an improved version was explained in [12]. A developed CNN network called LeNet-5 was found to be able to classify handwritten digits. Popular architectures from 2012 to 2015 are examined in [13], along with their basic components, and their applications are discussed in [14].

[1] https://faceswap.dev/

[2] https://rb.gy/9ffkom

[3] https://rb.gy/bv5530

[4] https://rb.gy/lgho24

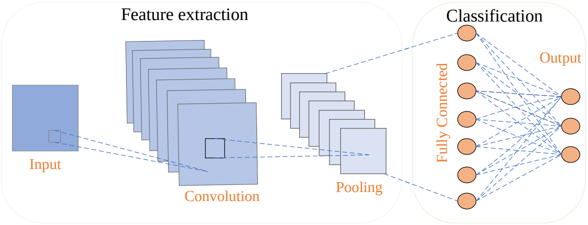

FIGURE 2: The basic architecture of CNN.

The basic structure of the CNN model comprises three types of layers: convolutional, pooling, and fully connected. Figure 2 presents the basic structure of the CNN model. The purpose of the convolution layer is to perform feature extraction. In the convolutional operation, an array of numbers (kernel) is applied across inputs (tensor) to construct the feature map. The procedure of constructing a feature map is an elementwise product between each element of the kernel and the input tensor, and the outputs are summed to obtain the element of the kernel. The kernel convolves across all the elements on the input tensor to construct the elements of the feature map for that kernel. An arbitrary number of feature maps can be obtained by implementing the convolution operation with different kernels. While training, the convolution operation is called forward propagation; during backpropagation, the gradient descent optimization technique updates the learnable parameters (kernels and weights) according to the loss value. The feature value $(Z^l_{i,j,k})$ at location $(i,j)$ in the $k^{th}$ feature map of the $l^{th}$ layer in [13] is as follows:

$$Z^l_{i,j,k} = (W^l_k)^T x^l_{i,j} + b^l_k \tag{1}$$

where $W^l_k$ and $b^l_k$ are the weight vector and bias term of the $k^{th}$ filter of the $l^{th}$ layer, respectively. $x^l_{i,j}$ is the input patch centered at location $(i,j)$ of the $l^{th}$ layer. Then, a nonlinear activation function is applied to detect nonlinear features such as sigmoid, tanh and ReLU. A nonlinear activation function $A(\cdot)$ can be expressed as:

$$a^l_{i,j,k} = A(Z^l_{i,j}), \tag{2}$$

where $a^l_{i,j,k}$ is the output value after applying the nonlinear activation function.

A pooling layer provides a typical downsampling operation to reduce the dimensionality of the feature maps to introduce translation invariance to small shifts and distortions and thereby decrease the number of subsequent learnable parameters. The pooling function is $pool(\cdot)$; for each feature map $a^l_{:,:,k}$, we have:

$$y^l_{i,j,k} = pool(a^l_{m,n,k}), \quad \forall (m,n) \in R_{i,j}, \tag{3}$$

where $R_{i,j}$ is a local neighborhood around location $(i,j)$. The fully connected layers are the final outputs of the CNN, such as the probabilities for each class in classification tasks. The number of output nodes in the final fully connected layer is usually equal to the number of classes. A nonlinear function, such as ReLU, follows each fully connected layer. Finally, a loss function is calculated to assess the compatibility of the CNN's forward propagation output predictions with the provided ground truth labels. The loss of CNN can be calculated as follows:

$$\mathfrak{L} = \frac{1}{N}\sum_{n=1}^{N} \ell(\theta; y^{(n)}, o^{(n)}), \tag{4}$$

where $N$ denotes the number of input-output relations $(x^{(n)}, y^{(n)})$, $x^{(n)}$ is the $n^{th}$ input data, $y^{(n)}$ is its target label, and $o^{(n)}$ is the output of the CNN [13]. Training a CNN determines the global minima, which identify the best-fitting set of parameters by minimizing the loss function. Currently, many CNN models exist, such as AlexNet [15], ZFNet [16], VGGNet [17], GoogLeNet/Inception [18] and ResNet [19].

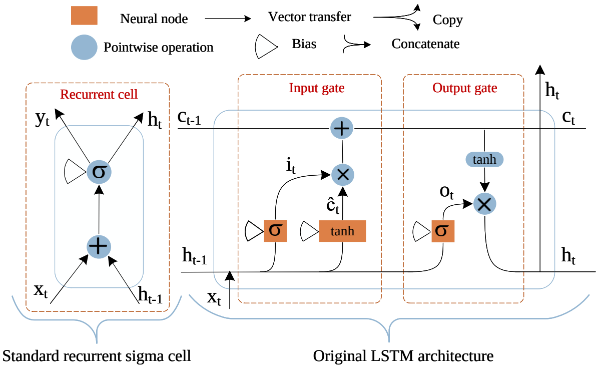

FIGURE 3: The basic architecture of RNN.

### *B. RNN background*

An RNN is a neural network in which the output from the previous step is used as input in the next phase. All inputs and outputs in typical neural networks are independent of one another; however, in some situations, such as when predicting the next word of a phrase, the prior words are necessary, and therefore, the previous words must be remembered. Consequently, RNNs were created, which use a hidden layer to overcome the problem. The hidden state, which remembers certain information about a sequence, is the most significant aspect of RNNs. RNNs have a “memory" that stores all information about the calculations. This memory utilizes the same settings for each input since it produces the same outcome by performing the same job on all inputs or hidden layers. Unlike in other neural networks, this method minimizes the complexity of the parameters. When the gap between the relevant input data is large, Hochreiter and Schmidhuber [20] proposed long short-term memory (LSTM) in 1997, which handles long-term dependencies. LSTM has been the focus of deep learning since it accomplishes nearly all the exciting outcomes based on RNNs. The recurrent layers, also known as hidden layers in RNNs, are made up of recurrent cells whose states are influenced by both previous states and current input via feedback connections. The classic recurrent sigma cell and LSTM with only input and output gates are depicted in Figure 3. The LSTM mathematical expressions are as follows:

$$\begin{aligned} i_t &= \sigma(W_i h_{t-1} + W_i x_t + b_i) \\ \hat{c}_t &= tanh(W_{\hat{c}} h_{t-1} + W_{\hat{c}} x_t + b_{\hat{c}}) \\ c_t &= c_{t-1} + i_t \cdot \hat{c}_t \\ o_t &= \sigma(W_o h_{t-1} + W_o x_t + b_o) \\ h_t &= o_t \cdot tanh(c_t), \end{aligned} \tag{5}$$

where $x_t$, $c_t$, $o_t$ and $h_t$ denote the input, the recurrent information, and the output of the cell at time $t$, respectively; $W_i$, $W_{\hat{c}}$, and $W_o$ are the weights; and $b$ is the bias. $c_t$ denotes the cell state of LSTM, and the operator '$\cdot$' denotes the pointwise multiplication of two vectors.

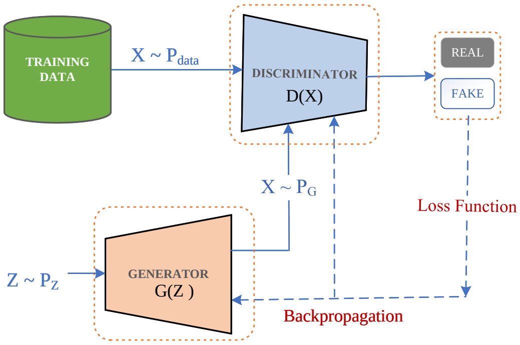

FIGURE 4: The basic architecture of a GAN.

### C. GANs background

GANs are a revolutionary tool used for teaching generative models to generate realistic examples from a data distribution [2]. Basically, GANs are a combination of two neural networks: the generator, $(G)$, and the discriminator, $(D)$. These two neural networks compete in a dynamic minimax game. The intuition behind this idea is that $G$ attempts to create fake samples, while $D$ attempts to determine which samples are fake and which are real. If the two models are allowed to compete for a long time, they will ultimately improve. In other words, the generator $G$ aims to capture the data distribution, whereas a $D$ aims to estimate the probability that a sample comes from the training data rather than from $G$. The basic structure of the GAN model can be visualized in Figure 4. The mathematical minmax optimization $(G^*)$ of neural networks $G$ and $D$ is as follows:

$$\begin{aligned} G^* &\in \arg\min\max V(G, D) \\ &= \arg\min\max \mathbb{E}_{X\sim P_{data(X)}}[\log(D(X))] \\ &\quad + \mathbb{E}_{Z\sim P_{Z(Z)}}[1 - \log(D(G(Z))], \end{aligned} \tag{6}$$

where $Z$ is the input for generator $G(Z)$ with probability distribution $P_Z$ and return $X$ with certain probability distribution $P_g$. The discriminator $D(X)$ estimates the probability that $X$ is from the distribution of training data $P_{data}$. Recently, various kinds of GANs, such as DCGAN [21], WGAN [22], PGGAN [23], BigGAN [24], and Style-GAN [4], [25], [26], were created to improve designs, losses, and training techniques.

## III. TOOLS USED TO CREATE A DEEPFAKE

In recent years, deep learning has achieved remarkable progress in computer vision and robotics. Moreover, the areas of digital face images and video manipulation are of leading interest because they use the power of GANs, which are capable of producing very realistic results. However, GANs still have challenges in establishing disentangled and controllable syntheses, particularly in the high-resolution domain. Disentangling distinct elements allows us to regulate changes across all factors independently. Nevertheless, without further adjustments such as regularization to encourage greater disentanglement, this technique is difficult to apply in GANs. Table 1 shows the tools used to create deep-fake images and videos. Mobile-based applications such as the Chinese apps ZAO, Auto FaceSwap and FaceApp allow ordinary internet users to easily create fake images and videos, which greatly helps the spread of DeepFakes. Several spoof videos created using GAN-based face-swapping techniques have been uploaded to YouTube and other video sites. Face swapping is very popular for moving a face from a source image to a target image to obtain realistic, unedited results. The main idea behind realistic face swapping is GANs [2]. Increasing numbers of face-swapping-, face synthesis-, face reenactment- and attribute manipulation-based applications are becoming popular; for example, images produced using Style-GAN [4], Style-GAN2 [25] and StyleGAN2-Ada [26] are becoming increasingly realistic and completely indistinguishable from human vision systems. By manipulating skin color or eye size without influencing other facial parameters, StyleGAN [4] cannot be utilized to generate high-fidelity human faces, and BigGAN [24] is unable to alter the color or length of a dog's hair without altering other aspects of the image.

Basically, face manipulation methods can be divided into five types [7]: entire face synthesis, identity swap, attribute manipulation, expression swap and miscellaneous. Table 2 shows the underlying idea of face manipulation methods. Detailed information on the face manipulation categories is summarized below.

### A. Entire face synthesis

This type of method generates nonexisting face images, usually using a powerful GAN, such as Style-GAN [4], Style-GAN2 [25] and StyleGAN2-Ada [26]. These approaches produce incredible outcomes, such as high-resolution facial images with a great degree of realism. Moreover, realistic face syntheses are becoming increasingly advanced. Entire face synthesis is based on datasets such as Generated-Images [4](100k-StyleGAN), Faces [27](100k-StyleGAN), DFFD [28](100k-StyleGAN, 200k-ProGAN), and iFake-FaceDB [29](250k-StyleGAN, 80k-ProGAN). This kind of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3151186, IEEE Access

TABLE 1: Tools used to create a DeepFake.

| Tool | Information | URL | Environment |
|---|---|---|---|
| ZAO | It allows users to use a single image to superimpose their faces over television and movie footage. | https://apps.apple.com/cn/app/id1465199127 | Application |
| AutoFaceSwap | Drag and drop the face or use the webcam for live face-swapping. | https://www.microsoft.com/en-us/p/auto-face-swap/9nblggh3m5nq | Application |
| FaceApp | Face image and video editing to make the face smile, look younger, look older, or change the gender. | https://apps.apple.com/gb/app/faceapp-ai-face-editor/id1180884341 | Application |
| FaceSwap | A deep learning-based method used to swap faces in images and videos. | https://github.com/deepfakes/faceswap | TensorFlow |
| FSGAN | A novel RNN face-swapping and face reenactment method for a single image or a video sequence. | https://github.com/YuvalNirkin/fsgan | PyTorch |
| FaceSwap-GAN | A GAN-based face-swapping model used for images and videos by adding adversarial loss and perceptual loss. | https://github.com/shaoanlu/faceswap-GAN | TensorFlow |
| FewShotFace translation | A GAN-based face image swap model that is also capable of translating to Asian faces. | https://github.com/shaoanlu/fewshot-face-translation-GAN | TensorFlow |
| StyleGAN | A Style-Based Generator Architecture for GANs. | https://github.com/NVlabs/stylegan | TensorFlow |
| StyleGAN2 | Improves image quality by proposing weight demodulation, regularizing path length, redesigning the generator, and removing progressive growth. | https://github.com/NVlabs/stylegan2 | TensorFlow |
| StyleGAN2-ADA | An adaptive discriminator augmentation mechanism that significantly stabilizes training in limited data regimes. | https://github.com/NVlabs/stylegan2-ada | TensorFlow |
| DFaker | Inputs are $64 \times 64$ images, outputs are a pair of $128 \times 128$ images, with one RGB with a reconstructed face. Structural dissimilarity (DSSIM) loss function is used to reconstruct faces. | https://github.com/dfaker/df | TensorFlow |
| DeepFake_tf | DeepFake_tf employs a similar idea as that used in DFaker. | https://github.com/StromWine/DeepFake_tf | TensorFlow |
| Deepfakes web | A face-swapping, video creation model that uses deep learning algorithms. | https://deepfakesweb.com/ | Web based |
| StarGAN | Unified GANs for Multi-Domain Image-to-Image Translation. | https://github.com/yunjey/stargan | PyTorch |
| StarGAN-V2 | Diverse image synthesis for multiple domains. | https://github.com/clovaai/stargan-v2 | PyTorch |
| DeepFaceLab | Generates better face-swapping videos. | https://github.com/iperov/DeepFaceLab | TensorFlow |
| DiscoFaceGAN | Disentangled and controllable face image generation via 3D imitative-contrastive learning. | https://github.com/microsoft/DiscoFaceGAN | TensorFlow |

manipulation might help a variety of businesses, including video games and 3D modelling, but it could also be used for negative purposes, such as the development of very realistic false accounts on social media to spread disinformation. Figure 5 depicts the nonexisting face images created by StyleGAN2 [25].

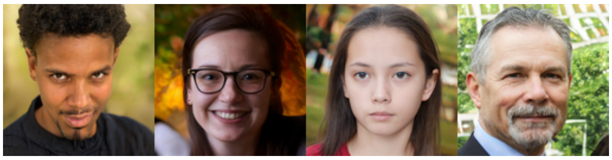

FIGURE 5: Example of entire face synthesis in [25].

TABLE 2: Facial manipulation techniques used to create DeepFakes.

| Facial manipulation | Key idea |
|---|---|
| Entire Face Synthesis | Creates entire non-existent face images are generated through a powerful GAN model, e.g., StyleGAN, StyleGAN2-Ada. |
| Identity Swap | Replacing the face of one person in image or video with the face of another person, e.g., FaceSwap, DeepFake. |
| Attribute Manipulation | Modifying some attributes of the face such as the color, hair, skin, gender, age, adding glasses, etc., e.g., StarGAN. |
| Expression Swap | Altering the facial expression of one person in a video with the facial expression of another person, e.g., Face2Face, Neural-Textures. |
| Miscellaneous | Face Morphing: create artificial biometric face samples that resemble the given biometric information.<br>Face De-Identification: remove the identity information present on a face image or video.<br>Audio-to-Video & Text-to-Video: facial expression swap is the synthesis of video from audio or text, also known as lip-sinc deep fakes. |

### *B. Identity swap*

The identity swap technique, also called the face-swap method, is very popular for replacing the face of one person in an image or video with that of another person. An example of an identity swap can be seen in Figure 6, where the source image shows the identity, the target image provides the attributes and a swapped face image is generated. Such swaps can be divided into two major types: i) graphics-based approaches such as FaceSwap and ii) deep learning technique-based approaches such as DeepFakes. The existing face-swap datasets are UADFV (49-FakeApp), D-TIMIT (620-faceswap-GAN), FF++ (1k-FaceSwap,1k-DeepFake), DFD(3k-DeepFake), Celeb-DF (5k-DeepFake) and DFDC Preview (4k-Unknown). This kind of manipulation might be useful in a variety of industries, including the entertainment industry. However, it might also be used for malicious objectives, such as the production of celebrity pornographic videos, fraud, and financial fraud.

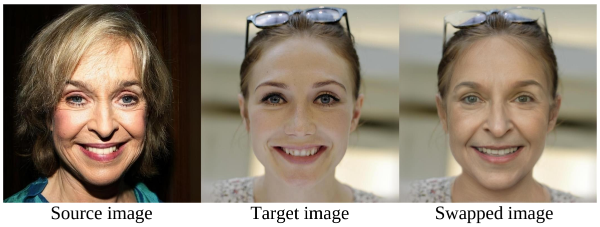

FIGURE 6: Example of identity swap in [30].

### C. Attribute manipulation

Attribute manipulation, also known as face editing or face retouching, entails changing aspects of the face, such as hair or skin color, gender, age, and the addition of spectacles [31]. An example of attribute manipulation can be seen in Figure 7, where Figure 7(a) shows the source image and the corresponding generated images: blond hair, gender, aged, and pale skin. Figure 7(b) shows the source image and the corresponding generated images: angry, happy, fearful. This manipulation process is usually carried out through a GAN, such as the StarGAN approach proposed in [31]. The popular AI face editor FaceApp, which is a mobile application, is an example of this type of manipulation. The existing attribute manipulation dataset is DFFD [28](80K-StarGAN, 12K-FaceAPP). Consumers may utilize this technology to test a wide range of items in a virtual environment, including cosmetics and makeup, spectacles, and hairstyles.

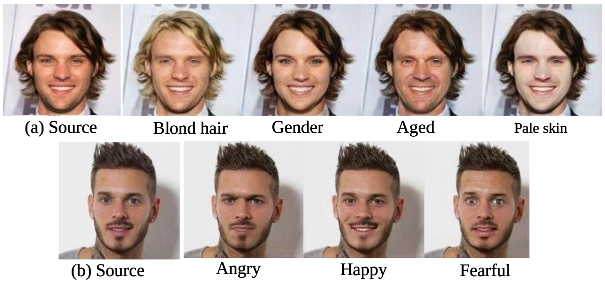

FIGURE 7: Example of attribute manipulation in [31].

### D. Expression swap

Expression swap, also known as face reenactment, modifies the facial expression of a person. An example of an expression swap can be seen in Figure 8, where the input expression is transferred to the targeted image, which then generates a reenactment result. The available techniques, such as image-level manipulation through popular GAN architectures [32], [33] and some popular video-based manipulation techniques, such as Face2Face [34] and neural textures [35], replace one person's facial expression in a video with another person's facial expression. The existing reenactment-based datasets are FF++(509k-Face2Face [34], 406k-Neural-Textures [36]). This form of fraud could have significant consequences, such as a video of someone saying something that he or she never said.

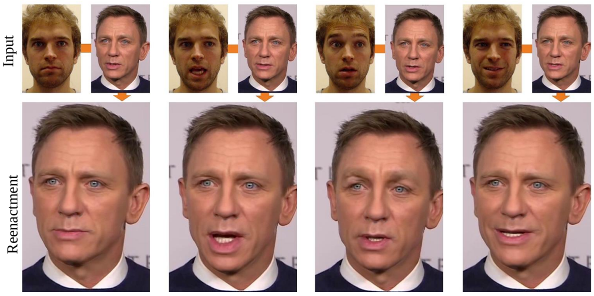

FIGURE 8: Example of expression Swap in [34].

### E. Miscellaneous

Regarding miscellaneous manipulation, we identified three types: face morphing, face deidentification, audio-to-video and text-to-video facial expression swaps.

Face morphing is a technique used for creating artificial biometric face samples that mimic the biometric data of multiple people. This type of manipulation leads to correctly verifying the created morphed face images against a manipulated reference in a facial recognition system database if a morphed face image is stored as a reference. Hence, morphed face images constitute a significant threat to face recognition systems, as they contradict the core principle of biometrics, which is the unique link between the sample and its matching person. [37] presented a comprehensive study of face morphing in 2019, covering both morphing strategies and morphing attack detectors.

Face deidentification is a type of manipulation used to remove artificial biometric fingerprints from images and videos. This technique can save artificial biometric fingerprint information for illegal verification. This action can be accomplished in a variety of ways. The most basic method is face blurring or pixelating. Other methods also exist, such as swapping an identity or synthesis identity swapping (applying some operations, i.e., pose, expression). An adversarial autoencoder-based video face deidentification method was demonstrated in [38].

Audio-to-video (A2V) and text-to-video (T2V) are also called lip-sync deep fakes [39]. Basically, the expression of the face in a video is synthesized using audio or text. An example of a fake video [40] describes a method used for synthesizing high-quality films of a person (in this case, Barack Obama) speaking with an accurate lip-sync track. Other important state-of-the-art methods are discussed in [41], [42]. In addition, [43] presents a procedure for blending counterfeit recordings from a text that takes information from a video of an individual talking and the necessary content to be spoken and makes another video wherein the individual's lips are synchronized with the new words.

TABLE 3: Publicly available forgeries detection datasets.

| Year | Dataset | Original | | Fake | | Methods | Details of dataset | URL |
|---|---|---|---|---|---|---|---|---|
| | | images | videos | images | videos | | | |
| 2011 | MICC-F220,<br>MICC-F2000,<br>MICC-F600 | 110,<br>1300,<br>440 | /<br>/<br>/ | 110,<br>700,<br>160 | /<br>/<br>/ | None | Used for image copy-move tampering detection [44]. | https://rb.gy/oecdyh |
| 2013 | IEEE IFS-TC | 1050 | / | 450 | / | None | The IEEE IFS-TC also contains 450 ground-truth mask for manipulation. | http://ifc.recod.ic.unicamp.br/fc.website/index.py |
| 2015 | WWD [45] | 13.5k | / | / | / | None | 82 cases of forgery, 92 forgery variants, and 101 unique masks for splice detection. | https://mklab.iti.gr/results/the-wild-web-tampered-image-dataset/ |
| 2015 | CelebA [46] | 202K | / | / | / | / | The images in this dataset cover large pose variations and background clutter. | https://liuziwei7.github.io/projects/FaceAttributes.html |
| 2017 | VISION [47] | 34.4k | 1914 | / | / | / | A video and image source identification application-based dataset (35 portable devices of 11 major brands). | https://lesc.dinfo.unifi.it/VISION/ |
| 2018 | UADFV [48] | 17.3k | 49 | 17.3k | 49 | 1 | The DeepFake videos are generated by using FakeAPP; it is straightforward and only has 2 classes: real and fake. | https://sites.google.com/view/grli-uavdt |
| 2018 | DF-TIMIT [49] | 34.0k | 320 | 68.0k | 640 | 2 | Two types of datasets, namely, low-quality DF-TIMIT-(LQ) and high-quality DF-TIMIT-(HQ), obtained using a face-swap GAN model. | https://www.idiap.ch/en/dataset/deepfaketimit |
| 2018 | FF [50] | 500.0k | 1004 | 521.4k | | 2 | Two ways to generate DeepFakes: Face2Face, and self-reenactment. | https://github.com/ondyari/FaceForensics/tree/original |
| 2019 | FF++ [51] | 509.9k | 1,000 | 509.0k | 4000 | 4 | Two graphics-based approaches (Face2Face [34] and FaceSwap) and two learning based approaches (DeepFakes and Neural Textures [36]). | https://github.com/ondyari/FaceForensics |
| 2019 | DFFD [28] | 58.7k | 1,000 | 240.3k | 3,000 | / | The DFFD dataset combines multiple forgery types in a single dataset. | http://cvlab.cse.msu.edu/dffd-diverse-fake-face-dataset.html |
| 2019 | DFD [52] | 315.4k | 363 | 2,242.7k | 3,068 | 5 | Google joined with the FF++ Dataset; additionally invited 28 paid actors in 16 different scenes, as well as over 3000 manipulated videos using DeepFakes. | https://github.com/ondyari/FaceForensics/tree/master/dataset |
| 2019 | DFDC-P [53] | 488.4k | 1,131 | 1,783.3k | 4,113 | 2 | The DFDC-P dataset works on deepfake detection technology to measure its performance. | https://ai.facebook.com/datasets/dfdc/ |
| 2020 | DFDC [54] | / | 23k | / | 104k | 8 | Eight facial modification algorithms have been used to extend the DFDC-P. | https://ai.facebook.com/datasets/dfdc/ |
| 2020 | Celeb-DF [55] | 225.4k | 590 | 2,116.8k | 5,639 | 1 | YouTube video clips of 59 celebrities of diverse genders, ages, and ethnic groups. The DeepFake videos are generated using an improved deepfake synthesis method. | https://github.com/yuezunli/celeb-deepfakeforensics |
| 2020 | DF-1.0 [56] | 12.6M | 50,000 | 5.0M | 10,000 | 1 | A large-scale dataset for real-world face forgery detection. | https://github.com/EndlessSora/DeeperForensics-1.0/tree/master/dataset |
| 2020 | WDF [57] | 11.8M | / | 7,314 | 707 | / | A small dataset that challenges the real-world dataset for DeepFake detection. | https://github.com/deepfakeinthewild/deepfake-in-the-wild |
| 2021 | OF [58] | 16K | / | 173K | / | / | Large-scale challenging dataset for multi-face forgery detection and segmentation in the wild. | https://sites.google.com/view/ltnghia/research/openforensics/ |

## IV. DATASETS

Forensics datasets can be classified into two broad types: traditional and DeepFake datasets. Traditional forensics datasets are created manually with extensive manual effort under carefully controlled conditions such as camera artifacts, splicing, inpainting, resampling and rotation detection. The Dresden Image Database (DID) [59] is based on camera fingerprinting and consists of 14,000 images from 73 cameras. The 73 different cameras were of 25 different models and camera fingerprinting types (indoor and outdoor scenes). While most traditional datasets incorporate image alteration forensics, only some of them cover video-based manipulation forensics. For example, MICC-F220, MICC F2000, and MICC-F600 are image datasets used to detect copy-move modifications. MICC-F220 is composed of 110 tampered and 110 original images, MICC-F2000 is composed of 700 tampered and 1300 original images, and MICC-F600 is composed of 160 tampered and 440 original images. The IEEE Information Forensics and Security Technical Committee (IFS-TC) conducted the First Image Forensics Challenge (2013), which is an international competition that collected thousands of photographs of varied scenes, both indoors and outdoors, using 25 digital cameras. The Wild Web Dataset (WWD) [45] contains 82 cases of 92 forgery variants and 101 unique mask splice detections. The WWD aims to address that gap in the evaluation of image tampering localization algorithms. The performance of [45] is evaluated in [60]. The CelebFaces Attributes Dataset (CelebA) is a large-scale face attribute dataset with more than 200K celebrity images, each with 40 attribute annotations. The images in this dataset cover large pose variations and background clutter. CelebA has large diversities, large quantities, and rich annotations, including 10,177 identities, 202,599 face images, 5 landmark locations, and 40 binary attribute annotations per image. In 2017, a VISION dataset was created that contained 11,732 original images and 648 original videos. The images were uploaded to social platforms such as Facebook and WhatsApp, and the videos were uploaded to YouTube and WhatsApp, resulting in a total of 34,427 images and 1,914 videos.

The second main type of forensics datasets are DeepFake datasets. These datasets are generally created by GAN-based models, which are very popular due to their realistic performance. The UADFV [48] consists of 49 real YouTube and 49 DeepFake videos. The DeepFake videos are generated using the DNN model with FakeAPP. The average length of these videos is approximately 11:14 seconds, with a typical resolution of 294 × 500. The DeepFake-TIMIT (DF-TIMIT) dataset [49] was created by using the VidTIMIT dataset [61] and FaceSwap-GAN; 16 similar-looking pairs of people from VidTIMIT [61] were selected, and for each of the 32 people, the database generated approximately 10 videos using low-quality of size 64×64, i.e., DF-TIMIT-(LQ), and high-quality of size 128 × 128, i.e., DF-TIMIT-(HQ) by using a face-swap GAN model. FaceFornesics (FF) [50] is a DeepFake dataset that aims to perform forensic tasks for facial identification and segmentation to forged images. It is composed of 1004 videos (face videos downloaded from YouTube) over 500,000 frames. The two types of manipulation are source-to-target, where facial expressions from a source video to a target video use Face2Face [34], and self-reenactment, where Face2Face reenacts the facial expressions of a source video. The FaceFornesics++ (FF++) [51] dataset has 1,000 real videos collected from YouTube, and 1,000 DeepFake videos were generated by applying each of the 4 face modification techniques: DeepFake, Face2Face [34], FaceSwap and Neural Texture [36] (4,000 face modification videos were created overall). These fake videos have produced 1.8 million manipulated face images. The Diverse Fake Face Dataset (DFFD) dataset combines multiple forgery types (FaceSwap, Deepfake, DeepFaceLab, FaceAPP, StarGAN and StyleGAN) in a single dataset. DeepFake Detection (DFD) [55] was developed by Google and JigSaw; 363 original videos were filmed with the assistance of 28 invited actors based on over 3,600 DeepFake videos using DeepFake techniques. In September 2019, Amazon Web Services, Facebook, Microsoft, and a number of academics collected a large-scale DeepFake dataset for the DeepFake Detection Challenge-Preview (DFDC-P) [53]. A full version of the DFDC-P was developed with eight manipulation methods and is known as the DeepFake Detection Challenge (DFDC). The Celeb-DF dataset [55] contains 590 actual videos and 5,639 DeepFake videos. Recently, the DeeperForensics-1.0 dataset (DF-1.0) [56] was found to consist of 60,000 videos with a total of 17.6 million frames for real-world face forgery detection. In addition, 100 paid actors were invited from 26 countries to collect high-resolution images of size 1920 × 1080. The new end-to-end face-swapping method (i.e., DF-VAE) was introduced and systematically applied to seven types of perturbations of fake videos at five intensity levels. More recently, a small WildDeepfake dataset (WDF) [57] was found to consist of 7,314 face sequences extracted from 707 DeepFake videos collected completely from the internet. WildDeepfake is a small dataset that can be used in addition to extending the existing datasets. Moreover, WDF is used to develop and test the effectiveness of DeepFake detectors against real-world DeepFakes. On the other hand, research on DeepFakes is also expanding to examine more than one face in a single image to detect DeepFake forgery, such as the OpenForensics dataset (OF) [58]. The OF dataset consists of 115K unrestricted images with 334K human faces. Table 3 summarizes these existing datasets.

## V. DEEPFAKE DETECTION

DeepFake face images and video detection dominate research on monitoring multimedia information and have the positive intention to improve the confidentiality and integrity of multimedia content. In addition, it is not an easy task to detect such altered multimedia content. This task has become more challenging after the emergence of generative models. Basically, forgery detection in multimedia content entails analyzing the multimedia content to determine whether the generated multimedia has been tampered with or is original. In the past, forgery detection techniques were considered traditional research; however, in recent years, DNN (AI-based)-based generated multimedia detection has become more popular. In this section, we will discuss both traditional and DeepFakes forensics-based techniques.

### A. Traditional forensic-based techniques

To modify image content, various traditional image processing technologies are employed, such as copy-move (splicing), resampling (resize, rotate, stretch), and the addition and/or removal of any part of the image. Traditional forensics-based techniques are commonly divided into two types: active and passive.

Active techniques require prior knowledge of multimedia for the authentication process. Basically, at the time of multimedia generation, some information is encoded, such as watermarks and digital signatures. For instance, a watermark is information that is added to a source image without degrading the visible artifact. Watermark extraction procedure is used to recover the watermark on the target image to discern whether the image has been manipulated. The manipulated portions in the target image can be detected using the extracted watermark. Over the past few years, mimicking aspects of genuine users or generating hyperrealistic masks at the presentation side for face images and videos have highlighted one kind of biometric vulnerability (biometric attack). To monitor or identify such biometric attacks, a variety of anti-spoofing techniques are used to counter these attacks, including eye blink detection in live stream scenarios, challenge-response techniques, 3D cameras, Active Flash and deep learning.

Facial recognition [110] is essential for face image and video detection before applying a traditional or a deep fake method. In this context, many researchers are interested in recognizing face images to identify authentic expressions, such as gestures made by the human face, which commu-

TABLE 4: DeepFake detection methods.

| Year | Study | Methods | Detection type | Techniques | Dataset Used |
|---|---|---|---|---|---|
| *Traditional techniques for DeepFake detection* | | | | | |
| 2018 | Koopman *et al.* [62] | PRNU Analysis | Facewap detection | PRNU | Self dataset by using GUI OpenFaceSwap application |
| 2018 | Afchar *et al.* [63] | Meso-4, MesoInception-4 | DeepFake images | CNN | DeepFake online videos, FF. |
| 2019 | Nataraj *et al.* [64] | Co-occurrence matrices | DeepFake images | DNNs | CycleGAN and StarGAN datasets |
| 2020 | Li *et al.* [65] | Color components | DeepFake images | DNNs with linear discriminative | LFW, LSUN, FFHQ, CelebA, FFHQ |
| 2021 | Haliassos *et al.* [66] | Semantic irregularities | DeepFake videos | ResNet-18 | FF++, DF-1.0, Celeb-DF, DFDC |
| 2021 | Lugstein *et al.* [67] | PRNU-based Analysis | DeepFake videos | PRNU, SVM | FF++, DFD, DF-TIMIT |
| *DNN-based techniques for DeepFake detection* | | | | | |
| 2018 | Güera and Delp [68] | ntra-frame and temporal inconsistencies | FaceSwap detection | CNN, LSTM | A collection of 600 videos obtained from multiple websites. |
| 2019 | Nguyen *et al.* [69] | Capsule-forensics | Replay attack, Face-swapping, Facial reenactment, Computer-generated images | VGG-19, Capsule networks | DeepFake online videos, FF, REPLAY-ATTACK database [70]. |
| 2019 | Xuan *et al.* [71] | Preprocessing combined with deep network | Generalization ability on unseen types of fake image | DCGAN, WGAN-GP, PG-GAN | CelebA-HQ |
| 2019 | Sabir *et al.* [72] | Temporal discrepancies | DeepFake, Face2Face and FaceSwap detection | CNN and RNN | FF++ |
| 2020 | Jeon *et al.* [73] | Fine-Tune and transformer | DeepFake images | SqueezeNet, ShallowNet, ResNet, Xception | CelebA, PGGAN, DF, FF |
| 2020 | Jeon *et al.* [74] | Self-training | DeepFake images | EfficientNet and ResNext | TPGGAN and StyleGAN-dataset |
| 2020 | Hsu *et al.* [75] | Pairwise learning | DeepFake images | CNN concatenated to CFFN | The CelebA and corresponding DeepFakes are created by the GAN method. |
| 2020 | Gandhi and Jain [76] | Adversarial perturbations | Adversarial perturbations to enhance DeepFakes and fool DeepFake detectors | VGG, ResNet | The CelebA and corresponding DeepFakes are created by the GAN method. |
| 2020 | Wu *et al.* [77] | Steganalysis features | DeepFake images | XceptionNet, LSTM | FF++ dataset |
| 2020 | Liu *et al.* [78] | Analyzing global image texture | DeepFake images | ResNet model | CelebA-HQ and FFHQ images. |
| 2020 | Khalid and Woo [79] | One-class Variational Autoencoder (VAE) | DeepFake image | OC-FakeDect model | FF++ |
| 2021 | Fung *et al.* [80] | Unsupervised Contrastive Learning | DeepFake detection | Xception network, SVM, and Bayes classifier | FF++, UADFV and Celeb-DF. |
| 2021 | Tariq *et al.* [81] | Spatial and temporal information | DeepFake videos | Convolutional LSTM-based Residual Network (CLRNet) | FF++, DFD, DeepFake-in-the-Wild videos (self). |
| *Artifact analysis for DeepFake detection* | | | | | |
| 2017 | Zhang *et al.* [82] | Bag of words and shallow classifiers | Facewap detection | SVM, RF, MLP | LFW face database [83] |
| 2018 | Li *et al.* [84] | Eye blinking | DeepFake videos | CNN, LRCN | CEW Dataset [85] for CNN and EBV dataset [86] for LRCN |
| 2018 | Li and Lyu [87] | Face-warping artifacts | DeepFake images | VGG-16, ResNet | UADFV and DF-TIMIT |
| 2019 | Agarwal *et al.* [88] | Facial expressions and movements | DeepFake videos | SVM | Person of interest (POI) videos. |
| 2019 | McCloskey and Albright [89] | Camera imagery | DeepFake images | SVM | CelebA HQ |
| 2019 | Marra *et al.* [90] | PRNU based | DeepFake images | GAN models | A raw images dataset for digital image forensics [91]. |
| 2019 | Yu *et al.* [92] | Image fingerprint | DeepFake images | | CelebA, LSUN bedroom scene dataset [93] |
| 2019 | Yang et al. [48] | Head poses | AI-generated fake face images and videos | SVM | UADFV and DARPA MediF |
| 2019 | Matern *et al.* [94] | Missing reflections, eye color, teeth, and eye tears | Generated faces, DeepFakes, Face2Face detection | Logistic regression, MLP | CelebA, ProGAN, Glow [95]. |
| 2019 | Fernandes et al. [96] | Heart rate variations | DeepFake videos | Neural-ODE | COHFACE (https://deepfakesweb.com/), VidTIMIT database |
| 2020 | Agarwal *et al.* [97] | Using appearance and behavior | DeepFake videos | ResNet-101, VGG | The world leaders dataset [88], FF++, DFD, DFDC and Celeb-DF. |
| 2020 | Chai *et al.* [98] | Patch-based classification | DeepFake images | Resnet-18, Xception, MesoInception4, CNN | CelebA-HQ |
| 2020 | Mittal *et al.* [99] | Using emotion audio-visual affective | DeepFake videos | Siamese network architecture | DF-TIMIT and DFDC. |
| 2020 | Agarwal *et al.* [39] | Phoneme-viseme mismatches | DeepFake videos | CNN | Instagram and YouTube [100], [101], A2V [40], T2V [43] |
| 2020 | Chugh *et al.* [102] | Modality Dissonance Score (MDS) | DeepFake videos | MDS network | DFDC, DF-TIMIT |
| 2020 | Guarnera *et al.* [103] | Analyzing convolutional traces | Forensics trace detection in DeepFake images | K-NN, SVM, and linear discriminant | CelebA and LFW [83] datasets. |
| 2020 | Fernandes *et al.* [104] | Attribution-based confidence (ABC) metric | DeepFake videos | Pre-trained ResNet50 on VGGFace2 [105] | VidTIMIT, COHFACE (idiap.ch/dataset/cohface), DF-TIMIT |
| 2020 | Qi *et al.* [106] | Heartbeat rhythms | DeepFake videos | DeepRhythm | FF++, DFDC-P |
| 2021 | Hu *et al.* [107] | lack of physical/physiological constraints | DeepFake images | Canny edge detector, Hough transform | FFHQ [4] dataset |
| 2021 | Demir and Ciftci [108] | Consistency of eyes and gazes | DeepFake videos | 3 dense layers network architecture | FF++, CelebDF |
| 2021 | Nirkin*et al.* [109] | Discrepancies between the two regions | DeepFake images | Xception networks | FF++, Celeb-DF, DFDC |

nicate information such as fear, disgust, happiness, sadness, surprise, anger, and neutrality. Umer et al. [111], [112] proposed a method to identify human facial expressions using data augmentation and fine-tuning the CNN model. A brief survey of biometric anti-spoofing methods for face recognition is available in [113]. To check the validity of the face images, Umer et al. [114] proposed a method that combines preprocessing, feature extraction and classification techniques. Initially, the landmark is extracted from the face images to identify the face region of the person; next, the detected face region is used to extract features. Finally, features are extracted from the detected facial region, and the scores are fused to calculate the final result based on the performance of the classifier according to these features.

In contrast to active techniques, passive techniques do not require prior knowledge of multimedia for the authentication process. In fact, statistical information about the source image (multimedia) that is highly consistent between distinct images is used. Consequently, the inherent statistical information of images is utilized to detect any fake areas of the image. Moreover, in the absence of digital watermarks, signatures, or specialized hardware, passive forensic techniques are used [115]. In Table 5, passive forensic techniques used in specific types of applications are summarized.

TABLE 5: Traditional forensics methods.

| Type | Forensics method |
|---|---|
| Format-Based Forensics | Fourier, JPEG, Double JPEG, JPEG Header, JPEG Ghost |
| Pixel-Based Forensics | Resampling, Cloning, Thumbnails |
| Statistical-Based Forensics | PCA, Linear Discriminant Analysis, Computer Generated |
| Printer Forensics | Clustering, Banding, Profiling |
| Geometric-Based Forensics | Camera Model, Calibration, Rectification, Lens Distortion, Composite, Reflection Shadow, Reflection Perception, Shadow Perception |
| Video Forensics | Motion, Re-Projected, Projectile, Enhancement |
| Camera-Based Forensics | Least-Squares, Expectation Maximization, Color Filter Array, Chromatic Aberration |
| Physics-Based Forensics | 2-D Lighting, 2-D Light Environment, Lee Harvey Oswald, 3-D Lighting |

### B. DeepFakes forensics-based techniques

Currently, DeepFake forensics-based techniques are a very active research area. Due to the popularity of DeepFake tools on the internet, it is very easy to create fake content that looks highly realistic and is difficult to distinguish with traditional techniques. To mitigate this challenging task or classify the content as either fake or pristine, researchers are developing DeepFake detection models. In contrast, many researchers are focusing on generating generalized realistic models to create DeepFakes. Creating DeepFakes is fun for users because many web-based tools are available online to perform such manipulations, which can still identify people and cause them to be misused for unwanted activities. However, it is also a technique that cyber attackers employ to penetrate identification or authentication systems to gain illegitimate access, thus violating privacy and compromising social security and democracy.

To combat the destructive impacts of DeepFakes, researchers have also turned dedicated attention to multimedia forensic techniques to identify DeepFakes. Existing methods have focused on either spatial and temporal artifacts left from the generation process or data-driven classification. Recently, researchers have used features such as those in Figure 9 to generate DeepFake detection models. This section reviews these features to create detection methods, and a summary of typical approaches is provided in Table 4. Inconsistencies, irregularities in the background, and GAN fingerprints are examples of spatial artifacts. Detecting fluctuations in a person's behavior, physiological signals, coherence, and video frame synchronization are all examples of temporal artifacts.

In this part, we will review recent DeepFake detection-based techniques grouped into three types: (1) traditional-based techniques for DeepFakes, (2) DNN-based techniques for DeepFakes, and (3) artifact analysis for DeepFakes.

#### 1) Traditional-based techniques for deepfake

In this method, pixel-level differences in the image and videos are examined to identify DeepFakes. Focusing on pixels and exploiting the correlations are easy to understand and provides hints in the detection process to clarify the variations between real and counterfeit (fake). When images or videos are modified by basic transformations, however, these efforts suffer from robustness concerns.

A novel photoresponse nonuniformity (PRNU) analysis method has been tested for its effectiveness at detecting DeepFake video manipulation [62]. This PRNU analysis reveals a statistically significant difference in mean normalized cross-correlation scores between real and DeepFake videos. However, the model has been tested on a very small dataset. The DeepFake GUI OpenFaceSwap application was used to create 10 authentic and 16 DeepFake images. The results shows that the cut-off value of 0.05 has a 3.8% false positive rate and a 0% false negative rate. In [64], a steganalysis method was adopted to identify DeepFake images. In fact, the co-occurrence matrices were constructed from RGB images, and the resulting values were trained with a deep convolutional neural network to identify the fakes. The experimental result shows 99% classification accuracy for cycleGAN- and StarGAN-based fake images. Li *et al.* [65] evaluated the statistical properties of deep network-generated images, such as the correlation between adjacent pixels in HSV and YCbCr color spaces, to distinguish DeepFake images. In Lips Don't Lie, Haliassos *et al.* [66] suggested a generalizable and robust approach to detect face forgery in videos also known as LipForensics. The fundamental theme is monitoring lip movements with high-level semantic inconsistencies that are present in many synthesized videos. Lugstein *et al.* [67] designed a novel pipeline to detect DeepFakes using photoresponse nonuniformity (PRNU). Basically, the PRNU technique is famous for detecting facial retouching and face morphing attacks. In Lugstein *et al.* [67], the PRNU feature

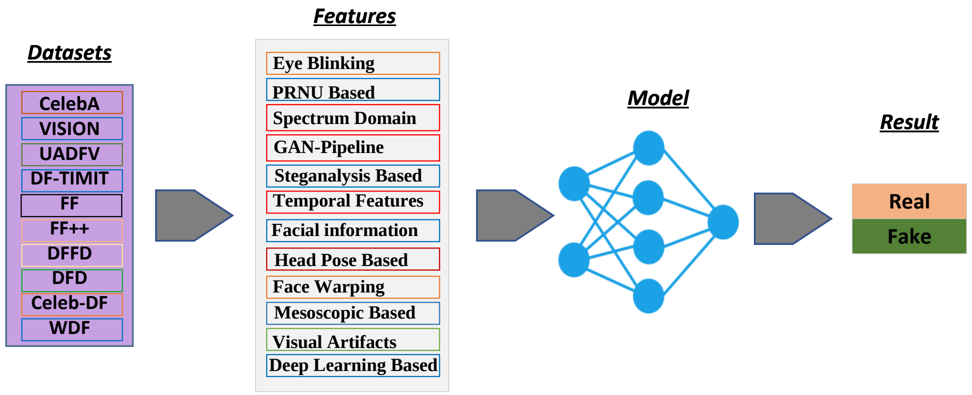

FIGURE 9: Some important features used for detection.

detection is similar to that in [116], [117] and adds a face image extraction stage, as well as an SVM classification stage. Two types of mesoscopic (a compact facial video forgery detection network) models (Meso-4 and MesoInception-4) have been proposed by Afchar *et al.* [63] to classify hyperrealistic forged videos based on DeepFake and Face2Face. It is obvious that uncompressed videos are severely degraded by image noise, wherein microscopic investigation-based image noise is not applicable. Moreover, the models are efficient in detecting hyperrealistic forged videos at a low computational cost. The average detection efficiency rate was found to be 98% for DeepFake videos and 95% for Face2Face videos under real conditions of diffusion on the internet.

### 2) DNN-based techniques for deepfakes

In this method, existing DNN models are used to analyze spatial characteristics, boost detection efficacy and improve the generalization capacity to detect DeepFakes. These methods are entirely data-driven. However, all of these DNN-based detection approaches are vulnerable to adversarial attacks, and very few studies have been able to assess their performance in combating adversarial attacks. Existing studies that use DNN to detect DeepFakes can be divided into three types. A fine-tuning approach is employed to improve the detection capacity of existing DNN models, explore artifact clues and train DNN models on different types of datasets to improve the generalization capacity. Güera and Delp [68] proposed a face-swapping-based detection method combining CNN and LSTM. InceptionV3 (CNN) is used to extract frame-level features, and the output of CNN is fed to LSTM to construct a sequence descriptor that is used for classification. The highest accuracy of the model is greater than 97% when classifying a video as pristine or DeepFake.

A capsule network is used to detect forged images and videos in a variety of forging scenarios, including replay attack detection and (both full and partial) computer-generated image/video detection in [69], where a capsule network was developed to resolve computer vision challenges and digital forensics issues. The ability of a capsule network based on a dynamic routing algorithm [118] to represent hierarchical pose relationships between object pieces has recently been demonstrated. To distinguish between fake and real images, a dynamic routing algorithm is used to route the outputs of the three capsules to the output capsules over a series of iterations. Four datasets are used to test the approach, which cover a wide spectrum of fabricated image and video attacks. In these four datasets, the suggested strategy outperforms existing methods. This outcome demonstrates the capsule network's utility in developing a generic detection system that can effectively detect a variety of counterfeit image and video attacks.

A generalized fake face image detection method was proposed by Xuan *et. al.* [71] in 2019. The key aim is to explicitly add a preprocessing step in the training stage to remove low-level unstable artifacts of GAN images and force the forensics classifier to focus on higher intrinsic forensic indications to detect such GAN-based images. In the preprocessing step, Xuan et al. used Gaussian blur and Gaussian noise methods. Adding Gaussian blur and Gaussian noise to low-level pixel data can depress low-level unstable artifacts. DCGAN [21], WGAN-GP [22] and PGGAN [23] are used to generate the GAN images, where pristine images are taken from CelebA-HQ. The generated image is used for PGGAN [23] to train the CNN and other DCGANs [21], and WGAN-GP [22] is used for testing purposes. However, the model shows little improvement in generalization ability on unseen types of fake image datasets.

Investigating the artifact clues in the image and videos is also a prominent scheme to detect DeepFakes. In [72], a combination of a recurrent convolutional model and face alignment approach was introduced to detect the three types of manipulations: DeepFake, Face2Face and FaceSwap. Initially, preprocessing operations are applied on video to detect, crop and align faces in a sequence of frames. Next, a combination of appropriate CNN models ResNet [19] or DenseNet [119] with alignment and a bidirectional recurrent network is used to test the accuracy. The model [72] is able to utilize micro-, meso- and macroscopic features for manipulation detection. Finally, according to the experimental results, landmark-based face alignment with bidirectional recurrent

DenseNet performs the best for detecting face manipulation in videos.

Jeon *et al.* [73] introduced an FDFtNet method to improve the capability of existing CNN models, such as SqueezeNet, ShallowNetV3, ResNetV2, and Xception. In this method, the fine-tuning method is used to extract the features using MBblockV3, and the method can be called fine-tuning transformation. This method shows a higher performance than that of the existing classical models. Moreover, the preference for unseen types of GAN-based image permutation attacks has not been calculated. Jeon *et al.* [74] proposed a transferable GAN-image detection framework (T-GD) technique, which efficiently detects DeepFake images. The model works on teacher and student relations, which mutually improve the detection performance.

Hsu *et. al.* [75] proposed a pairwise learning model to detect GAN-based generated fake images. The model was designed by combining the architecture of the improved version of the DenseNet backbone network and the Siamese network and is also called a common fake feature network (CFFN). To learn the discriminative common fake feature, pairwise information (labeled training dataset) is provided to the CFFN. The trained CFFN is capable of performing the classification task indicating whether the image is real or fake.

Gandhi and Jain [76] proposed a method to enhance the performance of existing DeepFake models by adding adversarial perturbations in DeepFake images. The fast gradient sign method and the Carlini and Wagner L2 norms are used to create adversarial perturbations in both black box and white box settings, and Lipschitz regularization and deep image prior (DIP) are introduced to increase the robustness of CNN (ResNet and VGG)-based deep-fake detectors. Lipschitz regularization increases the detection of perturbed DeepFakes, with a 10 percent improvement in the black box scenario, and DIP defense obtains a 95 percent accuracy with an original 98 percent accuracy. Moreover, there are two models with some limitations. The performance of Lipschitz regularization in the white box scenario only improves by 2.2 percent, and the DIP method shows higher performance than that of Lipschitz regularization; however, the detection process is highly time-consuming even after a high-performance configuration. Wu *et al.* [77] introduced an SSTNet method that combines spatial, steganalysis and feature extracted procedures to detect DeepFakes. Basically, XceptionNet is used to monitor the spatial features and statistical information of the image. Moreover, steganalysis operations are applied, and RNN is also used to mine the temporal features. Finally, all the extracted information is combined for binary classification to detect DeepFakes.

Liu *et al.* [78], using global texture data, increased the robustness and generalization capabilities of existing CNNs in identifying synthetic fake faces. Gram-Net shows significant resistance to perturbation attacks such as downsampling, JPEG compression, blur, and noise, according to experimental data. Gram-Net, which has demonstrated encouraging results in the wild, also has a proven generalization capacity in working with various GANs.

The current DeepFake detection methods use small datasets for specific types of manipulation. These types of generated deep fakes are highly realistic. The detection techniques for such DeepFakes suffer from performance. To solve this issue, Khalid and Woo [79] proposed the OC-FakeDect method, which uses a one-class variational autoencoder (VAE) to train only on real face images and detects nonreal images such as DeepFakes by treating them as anomalies.

Fung *et. al.* [80] introduced a unique unsupervised learning method for detecting facial modification. Two modified copies of a face image are generated using two distinct transformations and fed into two sequential subnetworks (Xception and projection head network). Furthermore, the outputs of the projection head networks maximize the agreement. The model architecture was inspired by the method proposed by Chen *et al.* [120], which shows high accuracy of visual representations over previous state-of-the-art methods.

By improving the generalization ability, conventional DNNs have been frequently used to detect fake faces; however, they can overfit specific manipulation types and suffer from transferability concerns when unknown manipulation methods are not available. Tariq *et. al.* [81] proposed a generalized method to detect multiple types of DeepFakes. Additionally, the model was tested on unseen types of DeepFakes, such as the DeepFake-in-the-Wild video dataset (Shahroztariq/CLRNet/blob/main/dataset_samples). The main idea is to trace the spatial and temporal information in DeepFakes by a convolutional LSTM-based residual network (CLRNet), which has a unique type of training strategy. The best performance of the CLRNet model on the DeepFake-in-the-Wild video dataset is 93.86%.

### 3) Artifact analysis for DeepFakes

DeepFakes frequently produce artifacts that are difficult to identify by humans but are quickly recognized by machine and forensic analysis. Inconsistencies, irregularities in the background, and GAN fingerprints are examples of spatial artifacts. Detecting fluctuation in a person's behavior, physiological signals, coherence, and video frame synchronization are all examples of temporal artifacts. Agarwal *et al.* [88], [97] proposed a combination of static biometrics on facial identity with temporal behavioral biometrics on facial expressions and head movements for DeepFake detection. According to Chai et al. [98], redundant artifacts can be evaluated from local patches to identify the fake face. This idea has been tested using different existing models, such as Resnet-18 [19], Xception [121], MesoInception4 [63], and CNN [122], with p values of 0.1 and 0.5 on the CelebA-HQ and FFHQ datasets, respectively[5]. This idea shows generalized characteristics with different network architectures and different datasets. Zhang *et. al.* [82] raised the concern about

[5] https://github.com/NVlabs/ffhq-dataset

the applications used for face swapping in less than a minute. This issue can be a serious problem for face authentication on the internet. To solve this issue, automated face swapping and its detection method were proposed with a combination of basic machine learning techniques. Initially, the key points from the face image are detected and presented as descriptors (capturing local information about the key point). Because each key point is independent, a further clustering operation is applied to generate the codebook for each image. This codebook is taken as input for linear or nonlinear-based machine learning to estimate its legitimacy. However, the features are extracted using speeded-up robust features (SURF) [123], and bag of words (bow) [124] methods are used to generate the codebook. The codebook information is then fed into support vector machines (SVMs), random forests (RFs) and multilayer perceptrons (MLPs) for binary classification. In the experiments, the best solution for detection accuracy is greater than 92%. Nirkin *et al.* [109] used the discrepancy between faces and their context to identify fake faces. In other words, two networks are trained; the first network is trained to identify the person's face, and the second context recognition network takes the face's context into account, such as the person's hair, ears, and neck. To identify fake faces, discrepancies are calculated by comparing these two networks. This method exhibits a high generalization ability.

Rather than looking at the visual artifacts in fake faces, other researchers are looking at the imperfect designs of the current GANs, which offer signals for distinguishing between genuine and DeepFake faces. McCloskey and Albright [89] explored the architecture of a GAN generator, which intended to enhance methods for detecting visual artifacts in DeepFake images. In fact, the generator's normalization processes are taken into account, which will reduce the frequency of saturated and underexposed pixels. Finally, the generated features are classified by SVM. Marra *et al.* [90] proposed GAN fingerprints (unique artifacts of Pro-GAN and Cycle-GAN fingerprints), which aim to detect DeepFake images.

Yu et al. [92] studied GAN fingerprints for image attribution and used them to classify images as real or produced GANs. This study also identified the source of GAN-generated images. If the model is trained by very little change in the dataset, then the model fingerprint will be distinct, which lends greater granularity to model authentication. Additionally, finetuning is an effective technique used to immunize the DNN model against adversarial perturbations in fingerprint images.

Analyzing artifacts in biological signals is also gaining prominent attention from researchers who aim to identify DeepFakes. In the synthesized fake faces, biological signal artifacts provide evident signals for fake detection. These biological signals are divided into the following groups: visual-audio inconsistency, visual inconsistency and biological signal-in-video. The visual-audio irregularity in DeepFake videos is a very important clue to detect the synthesized video. The techniques [39], [99], [102] can clearly demonstrate why the video is a fake. Mittal *et al.* [99] distinguish "real" and "fake" videos using a correlation between modalities and affective signals. For modelling the visual and audio in videos, a Siamese network is used, along with a mixture of the two triplet loss functions to determine similarity. One loss function aims to calculate the similarity between visual and auditory stimuli, while the other is designed to calculate effect cues such as perceived emotion. The experimental results show that the idea of estimating the audio-visual correlation is efficient in estimating DeepFake videos. Agarwal *et al.* [39] introduced a fake video detection method that takes advantage of abnormalities in the dynamics of the mouth shape (visemes) and the pronounced phoneme. Mama, baba, and papa are examples of phonemes that require the lips to be totally closed to be properly spoken. The authors' recommended strategy worked well, especially as the video became longer. The Modality Dissonance Score (MDS) was proposed by Chugh *et al.* [102] to detect DeepFake videos. Basically, dissimilarity scores are calculated between audio-visual segments over 1-second video segments, and the MDS is estimated after applying aggregation to all the segments. The resultant value can efficiently estimate the DeepFake video. This method can also be utilized for temporal forgery localization, which identifies the video segment that has been tampered with.

The idea of monitoring the lack of visual consistency in [48], [84], [87], [94], which is used to estimate DeepFake videos, particularly the shape, facial features, and landmarks of faces, is not based in nature. Li *et al.* [84] proposed an eye blinking-based fake face video detection method using a CNN and an RNN, which is an LRCN model. Basically, the LRCN model consists of three steps: feature extraction from the eye sequence by using VGG16, sequence learning by using LSTM, a special kind of RNN, and finally, state prediction, which generates the likelihood of eye open and closure states based on the output of LSTM. The best performance of the model under the ROC curve was 0.99. Li and Lyu [87] described a new deep learning-based model that can distinguish DeepFake videos from real videos. The model takes leverage of the warping step during DeepFake creation. This step leaves a resolution discrepancy between the warped face area and the surrounding context, and noticeable artifacts appear. Then, CNN models are used to detect such artifacts. CNN is specifically trained to recognize faces first and then extract landmarks to compute transform matrices to align the faces to a standard configuration. Gaussian blurring is applied to the aligned face, and then the inverse of the predicted transformation matrix is used to affine and warp it back to the original image. Faces are aligned into several scales to boost data diversity and to simulate more varied resolution scenarios of affine warped faces. The performance was calculated on four CNN models, namely, VGG16, ResNet50, ResNet101 and ResNet152, and on DeepFake datasets (UADFV and DF-TIMIT with two qualities, LQ and HQ). The ResNet50-based DeepFake detection model outperforms the DeepFake datasets.

Yang *et al.* [48] suggested a method for detecting changes between 3D head pose movement, which includes head orientation and position. To detect such orientation and positioning, 68 facial landmarks of the central face region are used. The 3D head postures are investigated since the DeepFake face generator pipeline has a flaw. After obtaining the detection results, the retrieved features are passed into an SVM classifier. Experiments on two datasets (UADFV, DARPA MediFor) reveal that the detection method outperforms the other methods. Guarnera *et. al.* [103] proposed a model for DeepFake detection by monitoring the hidden forensics traces in images. Basically, the expectation maximization (EM) algorithm [125] is used to extract a set of local features to model the underlying convolutional generative process. The model was evaluated with five different types of DeepFake creation techniques, namely, GDWCT, StarGAN, ATTGAN, StyleGAN and StyleGAN2, and on the CELEBA dataset using naïve classifiers to discriminate between originals and fakes.

Matern *et al.* [94] investigated a way to exploit DeepFake and face manipulation artifacts based on visual attributes such as eyes, teeth, and facial features. The visual artifacts are caused by a lack of global consistency, an incorrect or inadequate estimate of incident illumination, or an inaccurate estimate of the actual geometry. To detect DeepFakes, geometrical inconsistencies in reflections, eye and tooth areas are monitored, and textural characteristics collected from the face region based on facial landmarks and other factors are taken into account. Consequently, eye, teeth, and full-face crop features are employed. Following feature extraction, two classifiers, namely, logistic regression and a shallow neural network, are used to distinguish DeepFakes from original videos. The model works well on YouTube videos, with a best result of 0.851 in terms of the area under the receiver operating characteristics curve. The drawback of this method is that it requires pictures that satisfy specific criteria, such as open eyes or visible teeth. Fernandes *et. al.* [104] proposed an attribution-based confidence (ABC) metric [126] for detecting DeepFake videos. Initially, DeepFake videos were created using a commercial website (https://deepfakesweb.com/). Then, the generated DeepFake was tested on a pretrained ResNet50 model, where the model was trained with the VGGFace2 dataset [105]. According to the obtained attribution score, a threshold value of 0.94 was considered for the ABC metric that can differentiate a pristine from a DeepFake video. Hu *et al.* [107] analyzed the inconsistency between two eyes for detecting DeepFake face images. The detection model takes advantage of physical/physiological restrictions in GAN-based images and then sufficiently estimates the discrepancy between two eyes to identify fakes. These restrictions provide solid assurances for explaining the choice to differentiate a real from a fake; however, when improved GANs are suggested, they will be invalid. In addition, the model's resistance against perturbation attacks is unknown. Demir and Ciftci [108] proposed a model to detect DeepFakes by analyzing the gaze in videos.

The biological signs in such videos are difficult to duplicate. Heart rate has been demonstrated in studies to be useful in detecting DeepFake videos. Extracting the heart rate from videos is another challenging task. Taking advantage of the neural ordinary differential equation (Neural-ODE [127]) to identify DeepFake videos was presented by Fernandes *et al.* [96]. Qi et al. [106] proposed a DeepRhythm model that also exposes DeepFake videos using heartbeat rhythms. The authors created motion-magnified spatial-temporal representation (MMSTR) for the video to highlight heart rhythm signals. Finally, based on the output of MMSTR, a dual-spatial-temporal attentional network was built to identify fraudulent videos.

## VI. CHALLENGES FOR DEEPFAKE CREATION AND DETECTION

In recent years, many DeepFake tools have become available that have highly realistic performance levels, and many more are in development. In contrast, the development of the DeepFake generation model is creating large challenges for forensics experts in terms of combatting them. DeepFakes are AI-generated hyperrealistic images or videos that have been digitally edited using techniques such as face swapping, changing the attributes and representing individuals speaking and doing things that never happened.

GANs, which are popular artificial intelligence (AI) techniques, consist of two discriminative and generative models that compete against each other to improve their performance to generate believable fakes. These impersonations of real persons are frequently highly viral and spread swiftly across social media platforms, thereby making them an effective tool for propaganda. In digital forensics, as in other security-related disciplines, it is necessary to account for the presence of an adversary who is actively attempting to fool investigators. In reality, a knowledgeable attacker who understands the concepts on which the forensic tools are based may take a variety of counterforensic steps to avoid detection [128]. Forensics tools should be able to detect such situational threats, as well as any real-world situations that tend to degrade test accuracy. Therefore, the numerous counter forensics approaches intended to confuse current detectors are a valuable aid in the development of multimedia forensics, as they expose the flaws in current solutions and encourage research to find a more robust resolution.

To date, many models are available to create or detect fakes, but they still have weaknesses. In the following subsection, we will discuss the main challenges, point by point, in creating or detecting DeepFakes.

### A. Challenges for DeepFake creation

Despite the fact that significant efforts have been made to increase the visual quality of created DeepFakes, there are still a number of hurdles to overcome. Some challenges related to creating DeepFakes include generalization, temporal coherence, illumination stipulations, lack of realism in eyes and lips, hand movement behavior and identity leakage.

- **Generalization:** The characteristics of generative models depend on the type of dataset provided during training. Therefore, after finishing training on a particular dataset, the output produced by the model reflects the learned characteristics (fingerprint). In addition, the output quality depends on the size of the dataset provided during training. Thus, to generate high-quality output, the model should be fed a dataset large enough to achieve a particular type of characteristic. Moreover, creating a convincing model requires hours of training. It is usually simpler to obtain a dataset that contains relevant content; however, finding enough data for a single victim might be difficult. Retraining the model for each unique target identification is also time-consuming. Figure 10 shows the fingerprints left by different DeepFake generator models, which can be easily detected by a DeepFake detector.

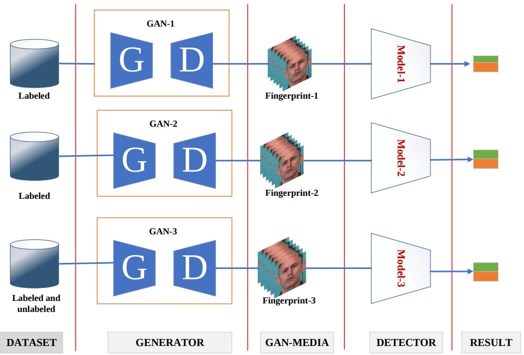

FIGURE 10: An example of a GAN fingerprint present in DeepFake-generated media using different environments can be discovered easily by a DeepFake detector.

- **Temporal coherence:** Other flaws include visible abnormalities such as flickering and jittering between frames. These flaws occur because the DeepFake generation frameworks work on each frame without considering temporal consistency. To overcome these flaws, some researchers offer this context to the generator or discriminator, consider temporal coherence losses, use RNNs, or use a combination of these approaches. Visible abnormalities can be seen in Figure 11.

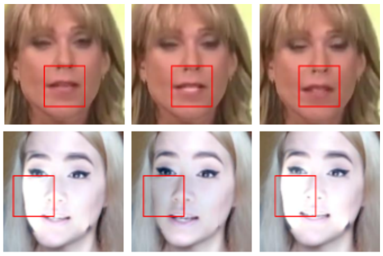

FIGURE 11: Abnormalities of temporal coherence.

- **Illumination stipulations:** Most available DeepFake datasets are produced in a controlled environment, such as using the same type of lighting and background. However, a sudden shift in lighting circumstances in indoor/outdoor scenarios causes color discrepancies and odd abnormalities in the resultant output.
- **Lack of realism in eyes and lips:** The lack of natural emotions, interruptions, and the rate at which the target talks are the primary difficulties of eye and lip synchronization-based DeepFake creation. Eye blinking abnormalities in DeepFake-generated video can be seen in Figure 12.

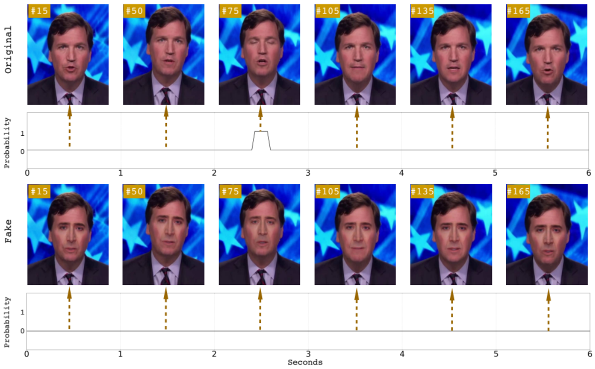

FIGURE 12: Abnormalities of eye blinking in [84].

- **Hand movement behavior:** Another issue is that when the target expresses emotion through hand movement, it is difficult for the DeepFake model to reflect such expressions. Moreover, this kind of expression dataset is limited; therefore, producing this type of DeepFake is challenging.
- **Identity leakage:** Target identity preservation becomes a challenge when there is considerable discrepancy between the target identity and the driving identity, such as in face reenactment tasks where target expressions are driven by some source identity. The driving 'identity facial data are partially transmitted to the manufactured face. This event occurs when training is performed on a single identity or many identities, yet data pairing is performed on the same identity.

Many DeepFake tools are available, but they are not perfect. In fact, the available tools are uniquely designed and focus only on certain types of characteristics. Given the abovementioned challenges, generating DeepFake tools requires more research to improve performance. To summarize, developing a DeepFake generation tool is a challenging task.

### B. Challenges for DeepFake detection

Although significant progress has been achieved in the performance of DeepFake detectors, several issues related to the current detection algorithms need to be addressed. Some of the difficulties faced by DeepFake detection techniques include a lack of datasets, unknown types of attacks on media, temporal aggregation and unlabeled data.

- **Lack of DeepFake datasets:** The performance of a DeepFake detection model depends on the variety of large datasets used during training. If the model is tested on downloaded media, which have an unknown type of manipulation, then designing the model to identify the unknown type of manipulation is challenging. Due to the popularity of web-based applications, postprocessing operations are applied to DeepFake multimedia with the intention of fooling the DeepFake detector; such manipulation could consist of removing temporal artifices, blurring, smoothing, cropping, etc.

- **Unknown type of attack:** Another challenging task is to design a robust DeepFake detection model against unknown types of attacks such as the fast gradient sign method (FGSM) [129] and the Carlini and Wagner L2 norm attack (CW-L2) [130]. These attacks are used to fool classifiers in their actual output. An example of a DeepFake creation using source and target faces, with adversarial perturbations, can be seen in Figure 13. DeepFakes are accurately classified as fake by a DeepFake detector, but adversarially perturbed DeepFakes are classified as real.

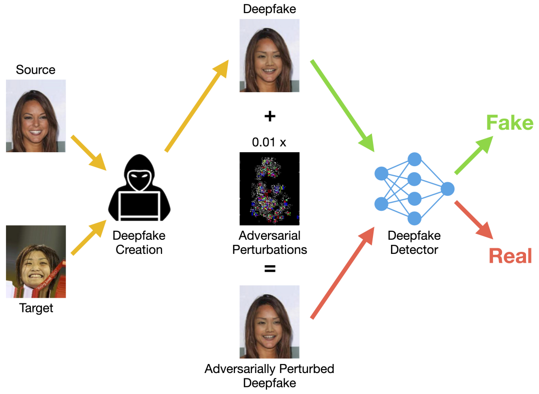

FIGURE 13: An example of an adversarial attack on a DeepFake detector in [76].

- **Temporal Aggregation:** Existing DeepFake detection algorithms use binary frame-level classification, which involves determining whether each video frame is real or fake. However, as these methods do not take interframe temporal consistency into consideration, they may encounter issues, such as exhibiting temporal abnormalities and real/artificial frames occurring in consecutive intervals. Furthermore, these methods necessitate an extra step to compute the video integrity score, which must be integrated for each frame to obtain the final result.
- **Unlabeled data:** Usually, DeepFake detection models are trained with large datasets. However, in some cases, such as journalism or law enforcement-based DeepFake detection, only a small dataset may be available. Moreover, this kind of dataset needs an additional effort to label the score corresponding to the type of forgery used. Consequently, further study is required to understand journalism or law enforcement-based forgery cases. Most DeepFake detection models, particularly those based on deep learning approaches, lack such an explanation because of their black-box nature. Therefore, designing a DeepFake detection model using a small and unlabeled dataset is challenging.

## VII. CONCLUSION

This article offers a comprehensive survey of a new and prominent technology, namely, DeepFake. It communicates the basics, benefits and threats associated with DeepFake, GAN-based DeepFake applications. In addition, DeepFake detection models are also discussed. The inability to transfer and generalize is common in most existing deep learning-based detection methods, which implies that multimedia forensics has not yet reached its zenith. Much interest has been shown by different important organizations and experts that are contributing to the improvement of applied techniques. However, much effort is still needed to ensure data integrity, hence the need for other protection methods. Furthermore, experts are anticipating a new wave of DeepFake propaganda in AI against AI encounters where none of the sides has an edge over the other.

## REFERENCES

[1] H. Farid, “Image forgery detection,” IEEE Signal processing magazine, vol. 26, no. 2, pp. 16–25, 2009.
[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” Advances in neural information processing systems, vol. 27, 2014.
[3] P. Baldi, “Autoencoders, unsupervised learning, and deep architectures,” in Proceedings of ICML workshop on unsupervised and transfer learning. JMLR Workshop and Conference Proceedings, 2012, pp. 37–49.
[4] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4401–4410.
[5] Y. Mirsky and W. Lee, “The creation and detection of deepfakes: A survey,” ACM Computing Surveys (CSUR), vol. 54, no. 1, pp. 1–41, 2021.
[6] M. Masood, M. Nawaz, K. M. Malik, A. Javed, and A. Irtaza, “Deepfakes generation and detection: State-of-the-art, open challenges, countermeasures, and way forward,” arXiv preprint arXiv:2103.00484, 2021.
[7] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, “Deepfakes and beyond: A survey of face manipulation and fake detection,” Information Fusion, vol. 64, pp. 131–148, 2020.
[8] T. T. Nguyen, C. M. Nguyen, D. T. Nguyen, D. T. Nguyen, and S. Nahavandi, “Deep learning for deepfakes creation and detection: A survey,” arXiv preprint arXiv:1909.11573, 2019.
[9] L. Verdoliva, “Media forensics and deepfakes: an overview,” IEEE Journal of Selected Topics in Signal Processing, vol. 14, no. 5, pp. 910–932, 2020.
[10] K. Fukushima, “A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” Biol. Cybern., vol. 36, pp. 193–202, 1980.
[11] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, “Handwritten digit recognition with a back-propagation network,” Advances in neural information processing systems, vol. 2, 1989.
[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.
[13] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai et al., “Recent advances in convolutional neural networks,” Pattern Recognition, vol. 77, pp. 354–377, 2018.
[14] S. Dong, P. Wang, and K. Abbas, “A survey on deep learning and its applications,” Computer Science Review, vol. 40, p. 100379, 2021.

[15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein et al., "Imagenet large scale visual recognition challenge," International journal of computer vision, vol. 115, no. 3, pp. 211–252, 2015.

[16] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in European conference on computer vision. Springer, 2014, pp. 818–833.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," arXiv preprint arXiv:1511.06434, 2015.

[22] A. Creswell and A. A. Bharath, "Inverting the generator of a generative adversarial network," IEEE transactions on neural networks and learning systems, vol. 30, no. 7, pp. 1967–1974, 2018.

[23] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," arXiv preprint arXiv:1710.10196, 2017.

[24] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," arXiv preprint arXiv:1809.11096, 2018.

[25] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8110–8119.

[26] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," arXiv preprint arXiv:2006.06676, 2020.

[27] "100,000 generated photos by stylegan." [Online]. Available: https://generated.photos/face-generator

[28] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. K. Jain, "On the detection of digital face manipulation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition, 2020, pp. 5781–5790.

[29] J. C. Neves, R. Tolosana, R. Vera-Rodriguez, V. Lopes, H. Proença, and J. Fierrez, "Ganprintr: Improved fakes and evaluation of the state of the art in face manipulation detection," IEEE Journal of Selected Topics in Signal Processing, vol. 14, no. 5, pp. 1038–1048, 2020.

[30] Y. Zhu, Q. Li, J. Wang, C.-Z. Xu, and Z. Sun, "One shot face swapping on megapixels," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4834–4844.

[31] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8789–8797.

[32] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen, "Arbitrary facial attribute editing: Only change what you want," arXiv preprint arXiv:1711.10678, vol. 1, no. 3, 2017.

[33] M. Liu, Y. Ding, M. Xia, X. Liu, E. Ding, W. Zuo, and S. Wen, "Stgan: A unified selective transfer network for arbitrary image attribute editing," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3673–3682.

[34] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of rgb videos," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2387–2395.

[35] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred neural rendering: Image synthesis using neural textures," ACM Transactions on Graphics (TOG), vol. 38, no. 4, pp. 1–12, 2019.

[36] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1125–1134.

[37] U. Scherhag, C. Rathgeb, J. Merkle, R. Breithaupt, and C. Busch, "Face recognition systems under morphing attacks: A survey," IEEE Access, vol. 7, pp. 23 012–23 026, 2019.

[38] O. Gafni, L. Wolf, and Y. Taigman, "Live face de-identification in video," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9378–9387.

[39] S. Agarwal, H. Farid, O. Fried, and M. Agrawala, "Detecting deepfake videos from phoneme-viseme mismatches," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 660–661.

[40] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing obama: learning lip sync from audio," ACM Transactions on Graphics (ToG), vol. 36, no. 4, pp. 1–13, 2017.

[41] Y. Song, J. Zhu, D. Li, X. Wang, and H. Qi, "Talking face generation by conditional recurrent adversarial network," arXiv preprint arXiv:1804.04786, 2018.

[42] L. Song, W. Wu, C. Qian, R. He, and C. C. Loy, "Everybody's talkin': Let me talk as you want," arXiv preprint arXiv:2001.05201, 2020.

[43] O. Fried, A. Tewari, M. Zollhöfer, A. Finkelstein, E. Shechtman, D. B. Goldman, K. Genova, Z. Jin, C. Theobalt, and M. Agrawala, "Text-based editing of talking-head video," ACM Transactions on Graphics (TOG), vol. 38, no. 4, pp. 1–14, 2019.

[44] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra, "A sift-based forensic method for copy–move attack detection and transformation recovery," IEEE transactions on information forensics and security, vol. 6, no. 3, pp. 1099–1110, 2011.

[45] "Wild web tampered image dataset," 2015. [Online]. Available: https://mklab.iti.gr/results/the-wild-web-tampered-image-dataset/

[46] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in Proceedings of the IEEE international conference on computer vision, 2015, pp. 3730–3738.

[47] D. Shullani, M. Fontani, M. Iuliani, O. Al Shaya, and A. Piva, "Vision: a video and image dataset for source identification," EURASIP Journal on Information Security, vol. 2017, no. 1, pp. 1–16, 2017.

[48] X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses," in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 8261–8265.

[49] P. Korshunov and S. Marcel, "Deepfakes: a new threat to face recognition? assessment and detection," arXiv preprint arXiv:1812.08685, 2018.

[50] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics: A large-scale video dataset for forgery detection in human faces," arXiv preprint arXiv:1803.09179, 2018.

[51] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1–11.

[52] D. Nicholas, G. Andrew, K. Per, V. V. Alexey, L. Thomas, C. Jeremiah, and B. Christoph, "Deepfakes detection dataset by google & jigsaw," https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html.

[53] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C. Ferrer, "The deepfake detection challenge (DFDC) preview dataset," arXiv preprint arXiv:1910.08854, 2019.

[54] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer, "The deepfake detection challenge (DFDC) dataset," arXiv preprint arXiv:2006.07397, 2020.

[55] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-df: A large-scale challenging dataset for deepfake forensics," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 3207–3216.

[56] L. Jiang, R. Li, W. Wu, C. Qian, and C. C. Loy, "Deeperforensics-1.0: A large-scale dataset for real-world face forgery detection," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2889–2898.

[57] B. Zi, M. Chang, J. Chen, X. Ma, and Y.-G. Jiang, "Wilddeepfake: A challenging real-world dataset for deepfake detection," in Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 2382–2390.

[58] T.-N. Le, H. H. Nguyen, J. Yamagishi, and I. Echizen, "Openforensics: Large-scale challenging dataset for multi-face forgery detection and segmentation in-the-wild," in International Conference on Computer Vision, 2021.

[59] T. Gloe and R. Böhme, "The'dresden image database'for benchmarking digital image forensics," in Proceedings of the 2010 ACM Symposium on Applied Computing, 2010, pp. 1584–1590.

[60] M. Zampoglou, S. Papadopoulos, and Y. Kompatsiaris, “Detecting image splicing in the wild (web),” in 2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW). IEEE, 2015, pp. 1–6.

[61] C. Sanderson, “The vidtimit database,” IDIAP, Tech. Rep., 2002.

[62] M. Koopman, A. M. Rodriguez, and Z. Geradts, “Detection of deepfake video manipulation,” in The 20th Irish machine vision and image processing conference (IMVIP), 2018, pp. 133–136.

[63] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, “Mesonet: a compact facial video forgery detection network,” in 2018 IEEE International Workshop on Information Forensics and Security (WIFS). IEEE, 2018, pp. 1–7.

[64] L. Nataraj, T. M. Mohammed, B. Manjunath, S. Chandrasekaran, A. Flenner, J. H. Bappy, and A. K. Roy-Chowdhury, “Detecting gan generated fake images using co-occurrence matrices,” Electronic Imaging, vol. 2019, no. 5, pp. 532–1, 2019.

[65] H. Li, B. Li, S. Tan, and J. Huang, “Identification of deep network generated images using disparities in color components,” Signal Processing, vol. 174, p. 107616, 2020.

[66] A. Haliassos, K. Vougioukas, S. Petridis, and M. Pantic, “Lips don’t lie: A generalisable and robust approach to face forgery detection,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 5039–5049.

[67] F. Lugstein, S. Baier, G. Bachinger, and A. Uhl, “Prnu-based deepfake detection,” in Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security, 2021, pp. 7–12.

[68] D. Güera and E. J. Delp, “Deepfake video detection using recurrent neural networks,” in 2018 15th IEEE international conference on advanced video and signal based surveillance (AVSS). IEEE, 2018, pp. 1–6.

[69] H. H. Nguyen, J. Yamagishi, and I. Echizen, “Capsule-forensics: Using capsule networks to detect forged images and videos,” in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 2307–2311.

[70] I. Chingovska, A. Anjos, and S. Marcel, “On the effectiveness of local binary patterns in face anti-spoofing,” in 2012 BIOSIG-proceedings of the international conference of biometrics special interest group (BIOSIG). IEEE, 2012, pp. 1–7.

[71] X. Xuan, B. Peng, W. Wang, and J. Dong, “On the generalization of gan image forensics,” in Chinese conference on biometric recognition. Springer, 2019, pp. 134–141.

[72] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan, “Recurrent convolutional strategies for face manipulation detection in videos,” Interfaces (GUI), vol. 3, no. 1, pp. 80–87, 2019.

[73] H. Jeon, Y. Bang, and S. S. Woo, “Fdftnet: Facing off fake images using fake detection fine-tuning network,” in IFIP International Conference on ICT Systems Security and Privacy Protection. Springer, 2020, pp. 416–430.

[74] H. Jeon, Y. Bang, J. Kim, and S. S. Woo, “T-gd: Transferable gan-generated images detection framework,” arXiv preprint arXiv:2008.04115, 2020.

[75] C.-C. Hsu, Y.-X. Zhuang, and C.-Y. Lee, “Deep fake image detection based on pairwise learning,” Applied Sciences, vol. 10, no. 1, p. 370, 2020.

[76] A. Gandhi and S. Jain, “Adversarial perturbations fool deepfake detectors,” in 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020, pp. 1–8.

[77] X. Wu, Z. Xie, Y. Gao, and Y. Xiao, “Sstnet: Detecting manipulated faces through spatial, steganalysis and temporal features,” in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020, pp. 2952–2956.

[78] Z. Liu, X. Qi, and P. H. Torr, “Global texture enhancement for fake face detection in the wild,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8060–8069.

[79] H. Khalid and S. S. Woo, “Oc-fakedect: Classifying deepfakes using one-class variational autoencoder,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 656–657.

[80] S. Fung, X. Lu, C. Zhang, and C.-T. Li, “Deepfakeucl: Deepfake detection via unsupervised contrastive learning,” arXiv preprint arXiv:2104.11507, 2021.

[81] S. Tariq, S. Lee, and S. Woo, “One detector to rule them all: Towards a general deepfake attack detection framework,” in Proceedings of the Web Conference 2021. New York, NY, USA: Association for Computing Machinery, 2021, pp. 3625–3637. [Online]. Available: https://doi.org/10.1145/3442381.3449809

[82] Y. Zhang, L. Zheng, and V. L. Thing, “Automated face swapping and its detection,” in 2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP). IEEE, 2017, pp. 15–19.

[83] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database forstudying face recognition in unconstrained environments,” in Workshop on faces in’Real-Life’Images: detection, alignment, and recognition, 2008.

[84] Y. Li, M.-C. Chang, and S. Lyu, “In ictu oculi: Exposing ai created fake videos by detecting eye blinking,” in 2018 IEEE International Workshop on Information Forensics and Security (WIFS). IEEE, 2018, pp. 1–7.

[85] “Cew dataset.” [Online]. Available: http://parnec.nuaa.edu.cn/xtan/data/ClosedEyeDatabases.html

[86] “Ebv dataset.” [Online]. Available: http://www.cs.albany.edu/lsw/downloads.html

[87] Y. Li and S. Lyu, “Exposing deepfake videos by detecting face warping artifacts,” arXiv preprint arXiv:1811.00656, 2018.

[88] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li, “Protecting world leaders against deep fakes.” in CVPR workshops, vol. 1, 2019.

[89] S. McCloskey and M. Albright, “Detecting gan-generated imagery using saturation cues,” in 2019 IEEE International Conference on Image Processing (ICIP). IEEE, 2019, pp. 4584–4588.

[90] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi, “Do gans leave artificial fingerprints?” in 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR). IEEE, 2019, pp. 506–511.

[91] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, “Raise: A raw images dataset for digital image forensics,” in Proceedings of the 6th ACM multimedia systems conference, 2015, pp. 219–224.

[92] N. Yu, L. S. Davis, and M. Fritz, “Attributing fake images to gans: Learning and analyzing gan fingerprints,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 7556–7566.

[93] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, “Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop,” arXiv preprint arXiv:1506.03365, 2015.

[94] F. Matern, C. Riess, and M. Stamminger, “Exploiting visual artifacts to expose deepfakes and face manipulations,” in 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW). IEEE, 2019, pp. 83–92.

[95] D. P. Kingma and P. Dhariwal, “Glow: Generative flow with invertible 1x1 convolutions,” arXiv preprint arXiv:1807.03039, 2018.

[96] S. Fernandes, S. Raj, E. Ortiz, I. Vintila, M. Salter, G. Urosevic, and S. Jha, “Predicting heart rate variations of deepfake videos using neural ode,” in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, Oct 2019.

[97] S. Agarwal, H. Farid, T. El-Gaaly, and S.-N. Lim, “Detecting deepfake videos from appearance and behavior,” in 2020 IEEE International Workshop on Information Forensics and Security (WIFS). IEEE, 2020, pp. 1–6.

[98] L. Chai, D. Bau, S.-N. Lim, and P. Isola, “What makes fake images detectable? understanding properties that generalize,” in European Conference on Computer Vision. Springer, 2020, pp. 103–120.

[99] T. Mittal, U. Bhattacharya, R. Chandra, A. Bera, and D. Manocha, “Emotions don’t lie: A deepfake detection method using audio-visual affective cues,” arXiv preprint arXiv:2003.06711, vol. 3, 2020.

[100] “Four in-the-wild lip-sync deep fakes, instagram.” [Online]. Available: https://www.instagram.com/bill_posters_uk

[101] “Four in-the-wild lip-sync deep fakes, youtube.” [Online]. Available: https://www.youtube.com/watch?v=VWMEDacz3L4

[102] K. Chugh, P. Gupta, A. Dhall, and R. Subramanian, “Not made for each other-audio-visual dissonance-based deepfake detection and localization,” in Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 439–447.

[103] L. Guarnera, O. Giudice, and S. Battiato, “Deepfake detection by analyzing convolutional traces,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 666–667.

[104] S. Fernandes, S. Raj, R. Ewetz, J. S. Pannu, S. K. Jha, E. Ortiz, I. Vintila, and M. Salter, “Detecting deepfake videos using attribution-based confidence metric,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 308–309.

[105] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, “Vggface2: A dataset for recognising faces across pose and age,” in 2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018). IEEE, 2018, pp. 67–74.

[106] H. Qi, Q. Guo, F. Juefei-Xu, X. Xie, L. Ma, W. Feng, Y. Liu, and J. Zhao, "Deeprhythm: Exposing deepfakes with attentional visual heartbeat rhythms," in Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 4318–4327.

[107] S. Hu, Y. Li, and S. Lyu, "Exposing gan-generated faces using inconsistent corneal specular highlights," in ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021, pp. 2500–2504.

[108] I. Demir and U. A. Ciftci, "Where do deep fakes look? synthetic face detection via gaze tracking," in ACM Symposium on Eye Tracking Research and Applications, 2021, pp. 1–11.

[109] Y. Nirkin, L. Wolf, Y. Keller, and T. Hassner, "Deepfake detection based on discrepancies between faces and their context," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.

[110] M. Wang and W. Deng, "Deep face recognition: A survey," Neurocomputing, vol. 429, pp. 215–244, 2021.

[111] S. Umer, R. K. Rout, C. Pero, and M. Nappi, "Facial expression recognition with trade-offs between data augmentation and deep learning features," Journal of Ambient Intelligence and Humanized Computing, pp. 1–15, 2021.

[112] S. Hossain, S. Umer, V. Asari, and R. K. Rout, "A unified framework of deep learning-based facial expression recognition system for diversified applications," Applied Sciences, vol. 11, no. 19, p. 9174, 2021.

[113] J. Galbally, S. Marcel, and J. Fierrez, "Biometric antispoofing methods: A survey in face recognition," IEEE Access, vol. 2, pp. 1530–1552, 2014.

[114] S. Umer, B. C. Dhara, and B. Chanda, "Face recognition using fusion of feature learning techniques," Measurement, vol. 146, pp. 43–54, 2019.

[115] H. Farid, "Digital image forensics," Scientific American, vol. 298, no. 6, pp. 66–71, 2008.

[116] C. Rathgeb, A. Botaljov, F. Stockhardt, S. Isadskiy, L. Debiasi, A. Uhl, and C. Busch, "Prnu-based detection of facial retouching," IET Biometrics, vol. 9, no. 4, pp. 154–164, 2020.

[117] U. Scherhag, L. Debiasi, C. Rathgeb, C. Busch, and A. Uhl, "Detection of face morphing attacks based on prnu analysis," IEEE Transactions on Biometrics, Behavior, and Identity Science, vol. 1, no. 4, pp. 302–317, 2019.

[118] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," arXiv preprint arXiv:1710.09829, 2017.

[119] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.

[120] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in International conference on machine learning. PMLR, 2020, pp. 1597–1607.

[121] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251–1258.

[122] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "Cnn-generated images are surprisingly easy to spot... for now," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8695–8704.

[123] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in European conference on computer vision. Springer, 2006, pp. 404–417.

[124] G. Qiu, "Indexing chromatic and achromatic patterns for content-based colour image retrieval," Pattern Recognition, vol. 35, no. 8, pp. 1675–1686, 2002.

[125] T. K. Moon, "The expectation-maximization algorithm," IEEE Signal processing magazine, vol. 13, no. 6, pp. 47–60, 1996.

[126] S. Jha, S. Raj, S. Fernandes, S. K. Jha, S. Jha, B. Jalaian, G. Verma, and A. Swami, "Attribution-based confidence metric for deep neural networks," 2019.

[127] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," arXiv preprint arXiv:1806.07366, 2018.

[128] T. Gloe, M. Kirchner, A. Winkler, and R. Böhme, "Can we trust digital image forensics?" in Proceedings of the 15th ACM international conference on Multimedia, 2007, pp. 78–86.

[129] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv preprint arXiv:1412.6572, 2014.

[130] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in 2017 ieee symposium on security and privacy (sp). IEEE, 2017, pp. 39–57.