On Disentangling Spoof Trace for Generic Face Anti-Spoofing

Yaojie Liu, Joel Stehouwer, and Xiaoming Liu

Michigan State University, East Lansing MI 48823, USA
{liuyaoj1,stehouw7,liuxm}@msu.edu

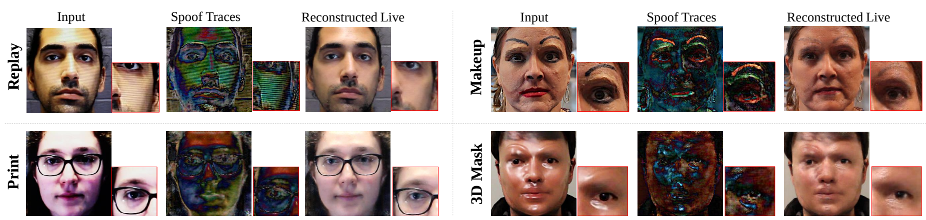

Fig. 1: Our approach can detect spoof faces, disentangle the spoof traces, and reconstruct the live counterparts. It can recognize diverse traces from various spoof types (*e.g.*, Moiré pattern in replay, artificial eyebrow and wax in makeup, color distortion in print, and specular highlights in 3D mask). Zoom in for details.

Abstract. Prior studies show that the key to face anti-spoofing lies in the subtle image pattern, termed "spoof trace", *e.g.*, color distortion, 3D mask edge, Moiré pattern, and many others. Designing a generic anti-spoofing model to estimate those spoof traces can improve both generalization and interpretability. Yet, this is a challenging task due to the diversity of spoof types and the lack of ground truth. This work designs a novel adversarial learning framework to disentangle the spoof traces from input faces as a hierarchical combination of patterns. With the disentangled spoof traces, we unveil the live counterpart from spoof face, and synthesize realistic new spoof faces after a proper geometric correction. Our method demonstrates superior spoof detection performance on both seen and unseen spoof scenarios while providing visually-convincing estimation of spoof traces. Code is available at `https://github.com/yaojieliu/ECCV20-STDN`.

1 Introduction

In recent years, the vulnerability of face biometric systems has been widely recognized and brought increasing attention to the vision community due to various physical and digital attacks. There are various physical and digital attacks, such as face morphing [13, 52, 55], face adversarial attacks [14, 20, 44], face manipulation attacks (*e.g.*, deepfake, face swap) [9, 45], and face spoofing (*i.e.*, presentation attacks) [5, 19, 40], that can be used to attack the biometric systems. Among all these attacks, face spoofing is the only physical attack to deceive the systems, where attackers present faces from spoof mediums, such as photograph, screen, mask and makeup, instead of a live human. These spoof mediums can be easily manufactured by ordinary people, therefore posing a huge threat to applications such as mobile face unlock, building access control, and transportation security. Therefore, face biometric systems need to be reinforced with face anti-spoofing techniques before performing face recognition tasks.

Face anti-spoofing[1] has been studied for over a decade, and one of the most common approaches is based on texture analysis [6, 7, 37]. Researchers notice that presenting faces from spoof mediums introduces special texture differences, such as color distortions, unnatural specular highlights, and Moiré patterns. Those texture differences are inherent within spoof mediums and thus hard to remove or camouflage. Early works build feature extractor with classifier, such as LBP+SVM and HOG+SVM [17, 26]. Recent works leverage deep learning techniques and show great progress [4, 29, 31, 41, 51].

However, there are two limitations in deep learning approaches. First, most works concern either print/replay or 3D mask alone, while a real-world scenario may encounter more spoof types at the same time. Second, most approaches formulate face anti-spoofing simply as a classification/regression problem. Although a few methods [29, 24, 51] attempt to offer insights via fixation, saliency, or noise analysis, there is little understanding on what the exact differences are between live and spoof, and what patterns the classifier's decision is based upon. We regard the face spoof detection for *all* spoof types as **generic face anti-spoofing**, and term the patterns differentiating spoof with live as **spoof trace**. Shown in Fig. 1, this work aims to equip generic face anti-spoofing models with the ability to explicitly extract the spoof traces from the input faces. We term this process as **spoof trace disentanglement**. This is a challenging objective due to spoof traces diversity and the lack of ground truth. However, we believe that solving this problem can bring several benefits:

1. Binary classification for face anti-spoofing would harvest any cue that helps classification, which might include spoof-irrelevant cues such as lighting, and thus hinder generalization. Spoof trace disentanglement explicitly tackles the most fundamental cue in spoofing, upon which the classification can be grounded and witnesses better generalization.
2. With the trend of pursuing explainable AI [1, 3], it is desirable to generate the patterns that support its decision. Spoof trace serves as a good visual explanation of the model's decision. Certain properties (*e.g.*, severity, methodology) of spoof attacks could potentially be revealed based on the traces.
3. Spoof traces are good sources for synthesizing realistic spoof samples. High-quality synthesis can address the issue of limited training data for the minority spoof types, such as special 3D masks and makeup.

[1] As most face recognition systems are based on a monocular camera, this work only concerns monocular face anti-spoofing methods, and terms as face anti-spoofing hereafter for simplicity.

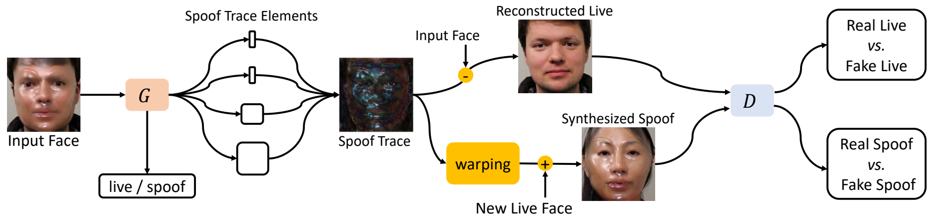

Fig. 2: Overview of the proposed Spoof Trace Disentanglement Network (STDN).

Shown in Fig. 2, we propose a Spoof Trace Disentanglement Network (STDN) to tackle this problem. Given only the binary labels of live *vs.* spoof, STDN adopts an overall GAN training strategy. The generator takes input faces, detect the spoof faces, and disentangles the spoof traces as a combination of multiple elements. With the spoof traces, we can reconstruct the live counterpart from the spoof and synthesize new spoof from the live. To correct possible geometric discrepancy during spoof synthesis, we propose a novel 3D warping layer to deform spoof traces toward the target face. We deploy multiscale discriminators to improve the fidelity of both the reconstructed live and synthesized spoof. Moreover, the synthesized spoof are further utilized to train the generator in a supervised fashion, thanks to disentangled spoof traces as ground truth for the synthesized sample. In summary, the main contributions are as follows:

- We for the first time study spoof trace for generic face anti-spoofing;
- We propose a novel model to disentangle spoof traces hierarchically;
- We utilize the spoof traces to synthesize new data and enhance the training;
- We achieve SOTA performance and provide convincing visualization.

2 Related Work

Face Anti-Spoofing has been studied for more than a decade and its development can be roughly divided into three stages. In early years, researchers leverage the spontaneous motion, such as eye blinking, to detect simple print photograph or static replay attacks [25, 35]. However, simple counter attacks would fail those methods, such as print with eye holes, and video replaying. Later, researchers focus on texture differences between live and spoof. Researchers extract hand-crafted features from faces, *e.g.*, LBP [6, 17, 18, 33], HoG [26, 50], SIFT [37] and SURF [7], and train a classifier such as SVM and LDA. Recently, deep learning demonstrates significant improvements over the conventional methods. [16, 27, 36, 49] train a deep neural network to do binary classification between live and spoof. [4, 29, 31, 41, 51] propose to learn additional information, such as face depth map and rPPG signal. With the latest approaches become saturating on several benchmarks, researchers start to explore more challenging cases, such as few-shot/zero-shot face anti-spoofing [31, 38, 54], domain adaptation in face anti-spoofing [41, 42], *etc.*

This work aims to disentangling the spoof traces from the input faces. [24, 43, 12] are the first few to estimate the different traces. However, they formulate the traces as low-intensity noises, which is limited to print/replay attacks with no convincing visual results. In contrast, we explore and visualize spoof traces for a wide range of spoof attacks, and evaluate the proposed method on challenging cases (*e.g.*, zero-shot face anti-spoofing).

Disentanglement Learning is often adopted to better represent complex data representation. DR-GAN [46, 47] disentangles face into identity and pose vectors for pose-invariant face recognition and synthesis. [53] disentangles the representations of appearance, canonical, and pose features for gait recognition. [28] disentangles the representation of a 3D face into identity, expressions, poses, albedo, and illuminations. To solve the problem of image synthesis, [15] disentangles an image into appearance and shape with U-Net and Variational Auto Encoder (VAE). In this work, we intend to disentangle features with different geometry and scales. We leverage outputs from different layers to represent features at different scales, and propose a novel warping layer to align the geometry.

3 Spoof Trace Disentanglement Network

3.1 Problem Formulation

Let the domain of live faces be denoted as $\mathcal{L} \subset \mathbb{R}^{N\times N\times 3}$ and spoof faces as $\mathcal{S} \subset \mathbb{R}^{N\times N\times 3}$, where N is the image size. We intend to obtain not only the correct prediction (live *vs.* spoof) of the input face, but also a convincing estimation of the spoof traces. Without the guidance of ground truth spoof traces, our key idea is to find a minimum change that transfers an input face to the live domain:

$$\arg\min_{\hat{\mathbf{I}}} \|\mathbf{I} - \hat{\mathbf{I}}\|_F \; s.t. \; \mathbf{I} \in (\mathcal{S} \cup \mathcal{L}) \text{ and } \hat{\mathbf{I}} \in \mathcal{L}, \tag{1}$$

where $\mathbf{I}$ is the input face from either domain, $\hat{\mathbf{I}}$ is the target face in the live domain, and $\mathbf{I} - \hat{\mathbf{I}}$ is defined as the spoof trace. For an input live face $\mathbf{I}_{\text{live}}$, the spoof traces should be 0 as it's already in $\mathcal{L}$. For an input spoof face $\mathbf{I}_{\text{spoof}}$, this L-2 regularization on spoof traces is also preferred, as there is no paired solution for the domain transfer and we hope the spoof traces to be bounded. Based on [24, 37], spoof traces can be partitioned into multiple elements based on scales: global traces, low-level traces, and high-level traces. Global traces, such as color balance bias and range bias, can be efficiently modeled by a single value. The color biases here only refer to those created by the interaction between spoof mediums and the capturing camera, and the model is expected to ignore those spoof-irrelevant color variations. Low-level traces consist of smooth content patterns, such as makeup strokes, and specular highlights. High-level traces include sharp patterns and high-frequency texture, such as mask edges and Moiré pattern. Denoted

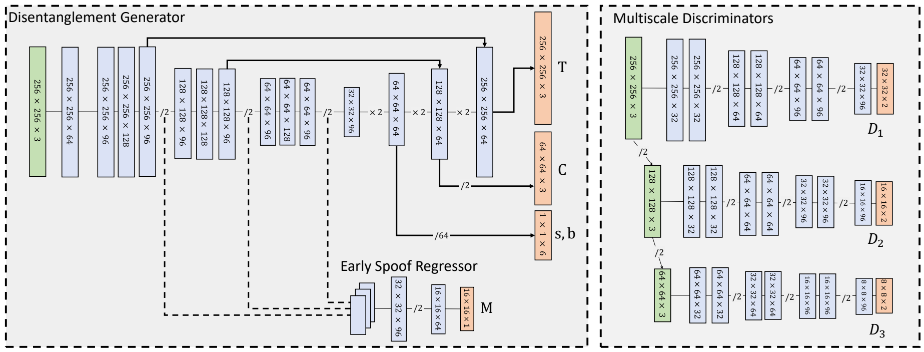

Fig. 3: The proposed STDN architecture. Except the last layer, each conv and transpose conv is concatenated with a Leaky ReLU layer and a batch normalization layer. **/2** denotes a downsampling by 2, and $\times\mathbf{2}$ denotes an upsampling by 2.

as $G(\cdot)$, the spoof trace disentanglement is formulated as a coarse-to-fine spoof effect build-up:

$$\begin{aligned} G(\mathbf{I}) &= \mathbf{I} - \hat{\mathbf{I}} \\ &= \mathbf{I} - ((1-\mathbf{s})\mathbf{I} - \mathbf{b} - \lfloor \mathbf{C} \rfloor_N - \mathbf{T}) \\ &= \mathbf{s}\mathbf{I} + \mathbf{b} + \lfloor \mathbf{C} \rfloor_N + \mathbf{T}, \end{aligned} \tag{2}$$

where $\mathbf{s}, \mathbf{b} \in \mathbb{R}^{1\times1\times3}$ represent color range bias and balance bias, $\mathbf{C} \in \mathbb{R}^{L\times L\times3}$ denotes the smooth content patterns ($L<N$ to enforce the smoothness), $\lfloor\cdot\rfloor$ is the resizing operation, and $\mathbf{T} \in \mathbb{R}^{N\times N\times3}$ is the high-level texture patterns. Compared to the single layer representation [24], this disentangled representation $\{\mathbf{s}, \mathbf{b}, \mathbf{C}, \mathbf{T}\}$ can largely improve disentanglement quality and suppress unwanted artifacts due to its coarse-to-fine process.

As shown in Fig. 3, Spoof Trace Disentanglement Network (STDN) consists of a generator and multiscale discriminators. They are jointly optimized to disentangle the spoof trace elements $\{\mathbf{s}, \mathbf{b}, \mathbf{C}, \mathbf{T}\}$ from the input faces. In the rest of this section, we discuss the details of the generator, face reconstruction and synthesis, the discriminators, and the training steps and losses used in STDN.

3.2 Disentanglement Generator

Spoof trace disentanglement is implemented via the generator. The disentanglement generator adopts an encoder-decoder as the backbone network. The encoder progressively downsamples the input face $\mathbf{I}\in\mathbb{R}^{256\times256\times3}$ to a latent feature tensor $\mathbf{F}\in\mathbb{R}^{32\times32\times96}$ via conv layers. The decoder upsamples the feature tensor $\mathbf{F}$ with transpose conv layers back to the input face size. To properly disentangle each spoof trace element, we leverage the natural upscaling property of the decoder structure: $\mathbf{s}, \mathbf{b}$ have the lowest spatial resolution and thus are disentangled in the very beginning of the decoder; $\mathbf{C}$ is extracted in the middle of the decoder with the size of 64; $\mathbf{T}$ is accordingly estimated in the last layer of the decoder. Similar to U-Net [39], we apply the short-cut connection between encoder and decoder to leak the high-frequency details for a high-quality estimation.

Unlike typical GAN scenarios where the generator only takes data from the source domain, our generator takes data from both source (spoof) and target (live) domains, and requires high accuracy in distinguishing two domains. Although the spoof traces should be significantly different between the two domains, they solely are not perfect hint for classification as the intensity of spoof traces varies from type to type. For this objective, we additionally introduce an Early Spoof Regressor (ESR) to enhance discriminativeness of the generator. ESR takes the bottleneck features $\mathbf{F}$ and outputs a $\mathbf{0/1}$ map $\mathbf{M} \in \mathbb{R}^{16\times16}$, where $\mathbf{0}$ means live and $\mathbf{1}$ means spoof. Moreover, we purposely make the encoder much heavier than the decoder, *i.e.*, more channels and deeper layers. This benefits the classification since ESR can better leverage the features learnt for spoof trace disentanglement.

In the testing phase, we use the average of the output from ESR and the intensity of spoof traces for classification:

$$\text{score} = \frac{1}{2K^2}\|\mathbf{M}\|_1 + \frac{\alpha_0}{2N^2}\|G(\mathbf{I})\|_1, \tag{3}$$

where α_0 is the weight for the spoof trace, $K=16$ is the size of $\mathbf{M}$, and $N=256$ is the image size.

3.3 Reconstruction and Synthesis

There are two ways to use the spoof traces:

- **Reconstruction**: obtaining the live face counterpart from the input as $\hat{\mathbf{I}} = \mathbf{I} - G(\mathbf{I})$;
- **Synthesis**: obtaining a new spoof face by applying the spoof traces $G(\mathbf{I}_i)$ disentangled from face image $\mathbf{I}_i$ to a live face $\mathbf{I}_j$.

To note that, spoof traces contain shape-dependent content associated with the original spoof face. Directly combining them with a new face with different shape or pose may result in poor alignment and strong visual implausibility. Hence, we propose an online 3D warping layer to correct the shape discrepancy. With ground truth traces, the synthesized spoof enable *supervised* training for the generator.

Online 3D Warping Layer The spoof traces for face i can be expressed as:

$$G_i = G(\mathbf{I}_i)[\mathbf{p}_0], \tag{4}$$

where $\mathbf{p}_0 = \{(0,0),(0,1),...,(255,255)\} \in \mathbb{R}^{256\times256\times2}$ enumerates pixel locations in $\mathbf{I}_i$. To warp the spoof trace, a dense offset $\Delta\mathbf{p}_{i\to j} \in \mathbb{R}^{256\times256\times2}$ is required to indicate the offset value from face i to face j. The warped traces can be denoted as:

$$G_{i\to j} = G(\mathbf{I}_i)[\mathbf{p}_0 + \Delta\mathbf{p}_{i\to j}], \tag{5}$$

$\Delta\mathbf{p}_{i\to j}$ can be fractional numbers, and the sampling of fractional pixel locations is implemented via bilinear interpolation. During data preparation, we use [30]

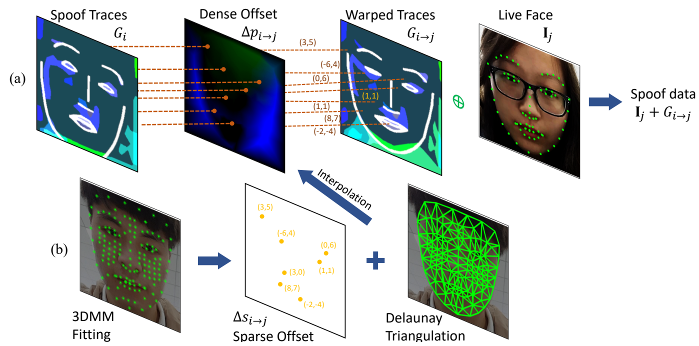

Fig. 4: 3D warping pipeline. (a) Given the corresponding dense offset, we warp the spoof trace and add them to the target live face to create a new spoof. E.g. pixel (x, y) with offset $(3, 5)$ is warped to pixel$(x+3, y+5)$ in the new image. (b) To obtain a dense offsets from the spare offsets of the selected face shape vertices, Delaunay triangulation interpolation is adopted.

to fit 3DMM model and extract the 2D locations of Q selected vertices for each face as:

$$\mathbf{s} = \{(x_0, y_0), (x_1, y_1), ..., (x_N, y_N)\} \in \mathbb{R}^{Q\times 2}, \tag{6}$$

A sparse offset on the corresponding vertices can then be computed between face i and j as $\varDelta\mathbf{s}_{i\to j} = \mathbf{s}_j - \mathbf{s}_i$. We select $Q = 140$ vertices to cover the face region so that they can represent non-rigid deformation, due to pose and expression. To convert the sparse offset $\varDelta\mathbf{s}_{i\to j}$ to the dense offset $\varDelta\mathbf{p}_{i\to j}$, we apply a triangulation interpolation:

$$\varDelta\mathbf{p}_{i\to j} = \mathrm{Tri}(\mathbf{p}_0, \mathbf{s}_i, \varDelta\mathbf{s}_{i\to j}), \tag{7}$$

where $\mathrm{Tri}(\cdot)$ is the bilinear interpolation operation based on Delaunay triangulation. Since the pixel values in the warped face are a linear combination of pixel values of the triangulation vertices, this whole process is differentiable. This process is illustrated in Fig. 4. Compared to previous methods [11, 29] that use offline face swapping or pre-computed dense offset, our warping layer only requires a sparse set of vertex locations, which is differentiable and computationally efficient.

Creating "harder" samples We can manipulate the spoof traces via tuning $\{\mathbf{s}, \mathbf{b}, \mathbf{C}, \mathbf{T}\}$, such as diminishing or amplifying certain element. Diminishing one or a few elements in $\{\mathbf{s}, \mathbf{b}, \mathbf{C}, \mathbf{T}\}$ would make the faces "less spoofed" as spoof traces are weakened. Those "less spoofed" data can be regarded as *harder* samples and may benefit the generalization. *E.g.*, removing the color distortion $\mathbf{s}$ may force the generator to explore high-level texture patterns. In this work, we randomly set one element from $\{\mathbf{s}, \mathbf{b}, \mathbf{C}, \mathbf{T}\}$ to be zero when synthesizing a new spoof face. Compared with other methods, such as brightness and contrast change [32], reflection and blurriness effect [51], or 3D distortion [21], our approach can introduce more realistic and effective data samples, as shown in Sec. 4.

3.4 Multi-scale Discriminators

Motivated by [48], we adopt 3 discriminators D_1, D_2, and D_3 at different resolutions (*i.e.*, 256, 128, and 64). The faces are resized to corresponding resolutions and sent to discriminators. D_1, working at the highest scale, processes the fine texture. D_2, working at the middle scale, focuses on the content pattern mostly in $\mathbf{C}$. D_3, working at the lowest scale, focuses on global elements as the higher-frequency details are erased. We adopt the structure of fully convolutional network, similar to PatchGAN [23], and each discriminator consists of 10 conv with 3 downsampling. It outputs a 2-channel map, where the first channel compares the reconstructed live with the real live, and the second channel compares the synthesized spoof with real spoof.

3.5 Training Steps and Loss Functions

We utilize multiple loss functions in three training steps.
ESR loss: $\mathbf{M}$ for live should be zero, and for spoof should be one:

$$L_{ESR} = \frac{1}{K^2}(\mathbb{E}_{i\sim\mathcal{L}}[\|\mathbf{M}_i\|_1] + \mathbb{E}_{i\sim\mathcal{S}\cup\hat{\mathcal{S}}}[\|\mathbf{M}_i - 1\|_1]), \tag{8}$$

where $\hat{\mathcal{S}}$ denotes synthesized spoof faces and $K=16$ is the size of $\mathbf{M}$.
Adversarial loss: We employ the LSGANs [34] on reconstructed live and synthesized spoof. For G:

$$L_G = \sum_{n=1,2,3} \{\mathbb{E}_{i\sim\mathcal{S}}[(D_n^1(\mathbf{I}_i - G_i) - \mathbf{1})^2] + \mathbb{E}_{i\sim\mathcal{L},j\sim\mathcal{S}}[(D_n^2(\mathbf{I}_i + G_{j\to i}) - \mathbf{1})^2]\}, \tag{9}$$

and for D:

$$\begin{aligned} L_D = \sum_{n=1,2,3} &\{\mathbb{E}_{i\sim\mathcal{L}}[(D_n^1(\mathbf{I}_i) - \mathbf{1})^2] + \mathbb{E}_{i\sim\mathcal{S}}[(D_n^2(\mathbf{I}_i) - \mathbf{1})^2] \\ &+ \mathbb{E}_{i\sim\mathcal{S}}[(D_n^1(\mathbf{I}_i - G_i(x)))^2] + \mathbb{E}_{i\sim\mathcal{L},j\sim\mathcal{S}}[D_n^2(\mathbf{I}_i + G_{j\to i}))^2]\}. \end{aligned} \tag{10}$$

where D_n^1 and D_n^2 denote the first and second channel of discriminator D_n.
Regularizer loss: In Eq. 1, the task regularizes the intensity of spoof traces while satisfying certain domain conditions. This regularizer loss is denoted as:

$$L_R = \beta\, \mathbb{E}_{\mathbf{x}\sim\mathcal{L}}[\|G(\mathbf{I}_i)\|_2^2] + \mathbb{E}_{\mathbf{i}\sim\mathcal{S}}[\|G(\mathbf{I}_i)\|_2^2], \tag{11}$$

where $\beta > 1$ is a weight to further compress the traces of live faces to be zero.
Pixel loss: Synthesized spoof data come with ground truth spoof traces. Therefore we can enable a supervised pixel loss for the generator to disentangle the exact spoof traces that were added to the live faces:

$$L_P = \mathbb{E}_{\mathbf{i}\sim\mathcal{L},\mathbf{j}\sim\mathcal{S}}[\|G(\lceil \mathbf{I}_i + G_{j\to i}\rceil) - \lceil G_{j\to i}\rceil\|_1], \tag{12}$$

where $\lceil\cdot\rceil$ is the `stop_gradient` operation. In this loss, we regard the traces $G_{j\to i}$ as ground truth, and the `stop_gradient` operation can prevent changing $G_{j\to i}$ to minimize the loss.

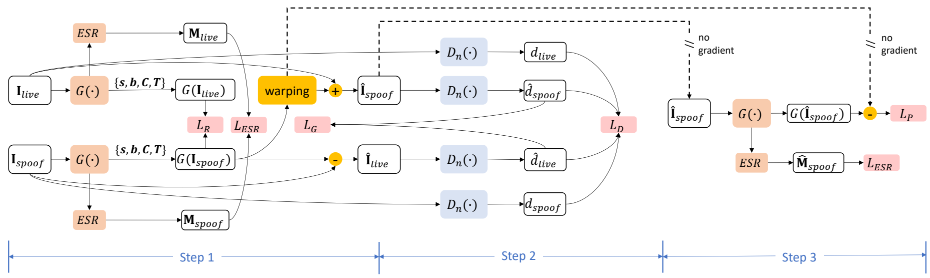

Fig. 5: The three training steps of STDN. Each mini-batch includes the same number of live and spoof samples.

Training steps and total loss: Fig. 5 shows the 3 training steps: generator step, discriminator step, and extra supervision step. In the generator step, live faces $\mathbf{I}_{live}$ and spoof faces $\mathbf{I}_{spoof}$ are fed to the generator to disentangle the spoof traces. The spoof traces are used to reconstruct the live counterpart $\hat{\mathbf{I}}_{live}$ and synthesize new spoof $\hat{\mathbf{I}}_{spoof}$. The generator is updated with respect to adversarial loss L_G, ESR loss L_{ESR}, and regularizer loss L_R:

$$L = \alpha_1 L_G + \alpha_2 L_{ESR} + \alpha_3 L_R. \tag{13}$$

For the discriminator step, $\mathbf{I}_{live}$, $\mathbf{I}_{spoof}$, $\hat{\mathbf{I}}_{live}$, and $\hat{\mathbf{I}}_{spoof}$ are fed into the discriminators $D_n(\cdot), n = \{1, 2, 3\}$. The discriminators are supervised with adversarial loss L_D to compete with the generator. For the extra supervision step, $\mathbf{I}_{live}$ and $\hat{\mathbf{I}}_{spoof}$ are fed into the generator with ground truth label and trace to enable pixel loss L_P and ESR loss L_{ESR}:

$$L = \alpha_4 L_{ESR} + \alpha_5 L_P, \tag{14}$$

α_1-α_5 are the weights to balance the multitask training. We execute 3 steps in every iteration, and reduce the learning rate for discriminator step by half.

4 Experiments

We first introduce the setup, and present the experiment results. Next, we quantitatively evaluate the spoof traces by performing a spoof medium classification, and conduct an ablation study on each design. Finally, we provide visualization on the spoof trace disentanglement and new spoof synthesis.

4.1 Experimental Setup

Databases We conduct experiments on three major databases: Oulu-NPU [8], SiW [29], and SiW-M [31]. Oulu-NPU and SiW include print/replay attacks, while SiW-M includes 13 spoof types. We follow all the testing protocols and compare with SOTA methods. Similar to most prior works, we only use the face region for training and testing.

Protocol	Method	APCER (%)	BPCER (%)	ACER (%)
1	STASN[51]	1.2	2.5	1.9
	Auxiliary [29]	1.6	1.6	1.6
	DeSpoof [24]	1.2	1.7	1.5
	Ours	**0.8**	**1.3**	**1.1**
2	Auxiliary [29]	2.7	2.7	2.7
	GRADIANT [8]	3.1	1.9	2.5
	STASN[51]	4.2	**0.3**	2.2
	Ours	**2.3**	1.6	**1.9**
3	DeSpoof [24]	4.0 ± 1.8	3.8 ± 1.2	3.6 ± 1.6
	Auxiliary [29]	2.7 ± 1.3	3.1 ± 1.7	2.9 ± 1.5
	STASN[51]	4.7 ± 3.9	$\mathbf{0.9 \pm 1.2}$	$\mathbf{2.8 \pm 1.6}$
	Ours	$\mathbf{1.6 \pm 1.6}$	4.0 ± 5.4	$\mathbf{2.8 \pm 3.3}$
4	Auxiliary [29]	9.3 ± 5.6	10.4 ± 6.0	9.5 ± 6.0
	STASN[51]	6.7 ± 10.6	8.3 ± 8.4	7.5 ± 4.7
	DeSpoof [24]	5.1 ± 6.3	6.1 ± 5.1	5.6 ± 5.7
	Ours	$\mathbf{2.3 \pm 3.6}$	$\mathbf{5.2 \pm 5.4}$	$\mathbf{3.8 \pm 4.2}$

(a)

Protocol	Method	APCER (%)	BPCER (%)	ACER (%)
1	Auxiliary[29]	3.6	3.6	3.6
	STASN[51]	–	–	1.0
	Meta-FAS-DR[54]	0.5	0.5	0.5
	Ours	**0.0**	**0.0**	**0.0**
2	Auxiliary[29]	0.6 ± 0.7	0.6 ± 0.7	0.6 ± 0.7
	Meta-FAS-DR[54]	0.3 ± 0.3	0.3 ± 0.3	0.3 ± 0.3
	STASN[51]	–	–	0.3 ± 0.1
	Ours	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$
3	STASN[51]	–	–	12.1 ± 1.5
	Auxiliary[29]	8.3 ± 3.8	8.3 ± 3.8	8.3 ± 3.8
	Meta-FAS-DR[54]	$\mathbf{8.0 \pm 5.0}$	$\mathbf{7.4 \pm 5.7}$	$\mathbf{7.7 \pm 5.3}$
	Ours	8.3 ± 3.3	7.5 ± 3.3	7.9 ± 3.3

(b)

Metrics(%)	Replay	Print	3D Mask					Makeup			Partial Attacks			Overall
			Half	Silic.	Trans.	Paper	Manne.	Ob.	Im.	Cos.	Funny.	Papergls.	Paper	
ACER(%)														
Auxiliary[29]	5.1	5.0	5.0	10.2	5.0	9.8	6.3	19.6	5.0	26.5	5.5	5.2	5.0	6.3
Ours	**3.2**	**3.1**	**3.0**	**9.0**	**3.0**	**3.4**	**4.7**	**3.0**	**3.0**	**24.5**	**4.1**	**3.7**	**3.0**	**4.1**
EER(%)														
Auxiliary[29]	4.7	0.0	1.6	10.5	4.6	10.0	6.4	12.7	0.0	19.6	7.2	7.5	0.0	6.6
Ours	**2.1**	**2.2**	**0.0**	**7.2**	**0.1**	**3.9**	**4.8**	**0.0**	**0.0**	**19.6**	**5.3**	**5.4**	**0.0**	**4.8**
TDR@FDR=0.5(%)														
Ours	90.1	76.1	80.7	71.5	62.3	74.4	85.0	100.0	100.0	33.8	49.6	30.6	97.7	70.4

(c)

Table 1: Known spoof detection on: (a) OULU-NPU (b) SiW (c) SiW-M Protocol I.

Evaluation metrics Two standard metrics are used in this work for comparison: EER and APCER/BPCER/ACER[22]. We also report True Detection Rate (TDR) at a given False Detection Rate (FDR). This metric describes the spoof detection rate at a strict tolerance to live errors, which is widely used to evaluate systems in real-world applications [2]. In this work, we report TDR at FDR$= 0.5\%$.
Parameter setting STDN is implemented in Tensorflow with an initial learning rate of $1e$-4. We train in total $150,000$ iterations with a batch size of 8, and decrease the learning rate by a ratio of 10 every $45,000$ iterations. We initialize the weights with $[0, 0.02]$ normal distribution. $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \beta\}$ are set to be $\{1, 100, 1e\text{-}3, 50, 1, 1e4\}$. α_0 is empirically determined from the training or validation set. We use open source face alignment [10] and 3DMM fitting [30] to crop the face and provide 140 landmarks.

4.2 Anti-Spoofing for Known Spoof Types

Oulu-NPU [8] is a common benchmark due to its high quality and challenging testing. Shown in Tab. 1(a), our approach achieves the best performance in all four protocols. Specifically, we demonstrate significant improvement in protocol 1 and 4, reducing the ACER by 30% and 32% relative to prior works. We notice, in protocol 3 and 4, the performances of camera 6 are much lower than those of cameras 1-5: the ACER for camera 6 are 9.5% and 8.6%, while the average ACER for the other cameras are 1.7% and 3.1%. Compared to other cameras, camera 6 has stronger sensor noises and STDN recognizes them as unknown

Methods	Replay	Print	3D Mask					Makeup			Partial Attacks			Average
			Half	Silic.	Trans.	Paper	Manne.	Ob.	Im.	Cos.	Fun.	Papergls.	Paper	
APCER(%)														
LBP+SVM [8]	19.1	15.4	40.8	20.3	70.3	**0.0**	4.6	96.9	35.3	**11.3**	53.3	58.5	0.6	32.8 ± 29.8
Auxiliary[29]	23.7	7.3	27.7	18.2	97.8	8.3	16.2	100.0	18.0	16.3	91.8	72.2	0.4	38.3 ± 37.4
DTL [31]	**1.0**	**0.0**	0.7	24.5	58.6	0.5	3.8	**73.2**	13.2	12.4	17.0	17.0	0.2	17.1 ± 23.3
Ours	1.6	**0.0**	**0.5**	**7.2**	**9.7**	0.5	**0.0**	96.1	**0.0**	21.8	**14.4**	6.5	**0.0**	**12.2 ± 26.1**
BPCER(%)														
LBP+SVM [8]	22.1	21.5	21.9	21.4	20.7	23.1	22.9	21.7	12.5	22.2	18.4	20.0	22.9	21.0 ± 2.9
Auxiliary[29]	**10.1**	**6.5**	**10.9**	**11.6**	**6.2**	**7.8**	**9.3**	11.6	9.3	**7.1**	**6.2**	**8.8**	**10.3**	**8.9 ± 2.0**
DTL [31]	18.6	11.9	29.3	12.8	13.4	8.5	23.0	11.5	9.6	16.0	21.5	22.6	16.8	16.6 ± 6.2
Ours	14.0	14.6	13.6	18.6	18.1	8.1	13.4	**10.3**	**9.2**	17.2	27.0	35.5	11.2	16.2 ± 7.6
ACER(%)														
LBP+SVM [8]	20.6	18.4	31.3	21.4	45.5	11.6	13.8	59.3	23.9	16.7	35.9	39.2	11.7	26.9 ± 14.5
Auxiliary[29]	16.8	6.9	19.3	14.9	52.1	8.0	12.8	55.8	13.7	**11.7**	49.0	40.5	**5.3**	23.6 ± 18.5
DTL [31]	9.8	**6.0**	15.0	18.7	36.0	4.5	13.4	**48.1**	11.4	14.2	**19.3**	**19.8**	8.5	16.8 ± 11.1
Ours	**7.8**	7.3	7.1	**12.9**	13.9	**4.3**	**6.7**	53.2	**4.6**	19.5	20.7	21.0	5.6	**14.2 ± 13.2**
EER(%)														
LBP+SVM [8]	20.8	18.6	36.3	21.4	37.2	7.5	14.1	51.2	19.8	16.1	34.4	33.0	7.9	24.5 ± 12.9
Auxiliary[29]	14.0	4.3	11.6	**12.4**	24.6	7.8	10.0	72.3	10.1	**9.4**	21.4	**18.6**	4.0	17.0 ± 17.7
DTL [31]	10.0	**2.1**	14.4	18.6	26.5	5.7	9.6	50.2	10.1	13.2	**19.8**	20.5	8.8	16.1 ± 12.2
Ours	**7.6**	3.8	**8.4**	13.8	14.5	**5.3**	4.4	**35.4**	**0.0**	19.3	21.0	20.8	**1.6**	**12.0 ± 10.0**
TDR@FDR=0.5(%)														
Ours	45.0	40.5	45.7	36.7	11.7	40.9	74.0	0.0	67.5	16.0	13.4	9.4	62.8	35.7 ± 23.9

Table 2: The evaluation on SiW-M Protocol II: unknown spoof detection. **Bold** indicates the best score in each protocol. **Red** indicates protocols that our method improves over 50% than SOTA.

spoof traces, which leads to an increasing BPCER. Separating sensor noises from spoof traces can be an important future research.

SiW [29] Compared to Oulu, SiW includes fewer cameras but more spoof mediums and environment variations, such as pose, illumination, and expression. The comparisons are shown in Tab. 1(b). We outperform the previous works on the first two protocols and have a competitive performance on protocol 3. Protocol 3 requires the model to be trained on one spoof attack (print or replay) and tested on the other. Shown in Fig. 8, the traces of print and replay are significantly different, which may prevent the model from generalizing well.

SiW-M [31] contains a large amount of spoof types, including print, replay, 3D mask, makeup, and partial attacks. To use SiW-M, we randomly split the data into train/test set with a ratio of 60% and 40%, and the results are shown in Tab. 1(c). Compared to one of the best anti-spoofing models [29], our method outperforms on all spoof types as well as the overall performance, which demonstrates the superiority of our anti-spoofing on known spoof attacks.

4.3 Anti-Spoofing for Unknown Spoof Types

Another important aspect of anti-spoofing model is to generalize to the unknown/unseen. SiW-M comes with the testing protocol to evaluate the performance of unknown attack detection. Shown in Tab. 2, STDN achieves significant improvement over the previous best model by relatively 24.8% on the overall EER and 15.5% on the overall ACER. This is especially noteworthy because DTL was specifically designed for detecting unknown spoof types, while our proposed approach shines in *both known and unknown spoof detection*. Specifically, we reduce the EERs of transparent mask, mannequin head, impersonation makeup

Label \ Predict	Live	Print	Replay
Live	60(+1)	0(−1)	0
Print	3(+3)	108(+20)	9(−23)
Replay	1(−12)	11(+3)	108(+9)

Label \ Predict	Live	Print1	Print2	Replay1	Replay2
Live	56(−4)	1(+1)	1(+1)	1(+1)	1(+1)
Print1	0	43(+2)	11(+9)	3(−8)	3(−3)
Print2	0	9(−25)	48(+37)	1(−8)	2(−4)
Replay1	1(−9)	2(−1)	3(+3)	51(+38)	3(−28)
Replay2	1(−7)	2(−5)	2(+2)	3(−3)	52(+13)

Table 3: Confusion matrices of spoof mediums classification based on spoof traces. The left table is 3-class classification, and the right is 5-class classification. The results are compared with the previous method [24]. Green represents improvement over [24]. Red represents performance drop.

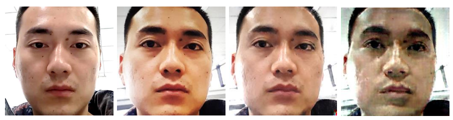

Fig. 6: Live reconstruction comparison: (a) live, (b) spoof, (c) ESR+D-GAN, (d) ESR+GAN.

Method	APCER (%)	BPCER (%)	ACER (%)
ESR	0.8	4.3	2.6
ESR+GAN	1.5	2.7	2.1
ESR+D-GAN	0.8	2.4	1.6
ESR+GAN+L_P	0.8	8.2	4.5
ESR+D-GAN+L_P	**0.8**	**1.3**	**1.1**

Table 4: Quantitative ablation study of components in our approach.

and partial paper attack relatively by 45.3%, 54.2%, 100.0%, 81.8%, respectively. Among all, obfuscation makeup is the most challenging one, where we predict almost all the spoof samples as live. This is due to the fact that such makeup looks very similar to the live faces, while being dissimilar to any other spoof types. Once we obtain a few samples, our model can quickly recognize the spoof traces on the eyebrow and cheek, and successfully detect the attack (0% in Tab. 1(c)). However, with the TDR= 35.7% at FDR= 0.5%, the proposed method is still far from applicable in practices when dealing with unknown spoof types, which warrant future research.

4.4 Spoof Traces Classification

To quantitatively evaluate the spoof trace, we perform a spoof medium classification on the disentangled spoof traces. After convergence, we fix STDN and apply a simple CNN to classify the spoof mediums given the estimated spoof traces. We follow the same setting in [24] on Oulu-NPU Protocol 1. Shown in Tab. 3, our 3-class model and 5-class model can achieve classification accuracy of 92.0% and 83.3% respectively. Compared to [24], we improve 10% on the 3-class model and 29% on the 5-class model. In addition, we train the same CNN on the original images for spoof medium classification, and the classification accuracy is 86.3% (3-class) and 80.6% (5-class). This demonstrates that STDN distills significant information to distinguish different spoof mediums.

4.5 Ablation Study

In this section, we show the importance of each design on Oulu Protocol 1. Our baseline is the encoder with ESR (denoted as ESR), which is a conventional

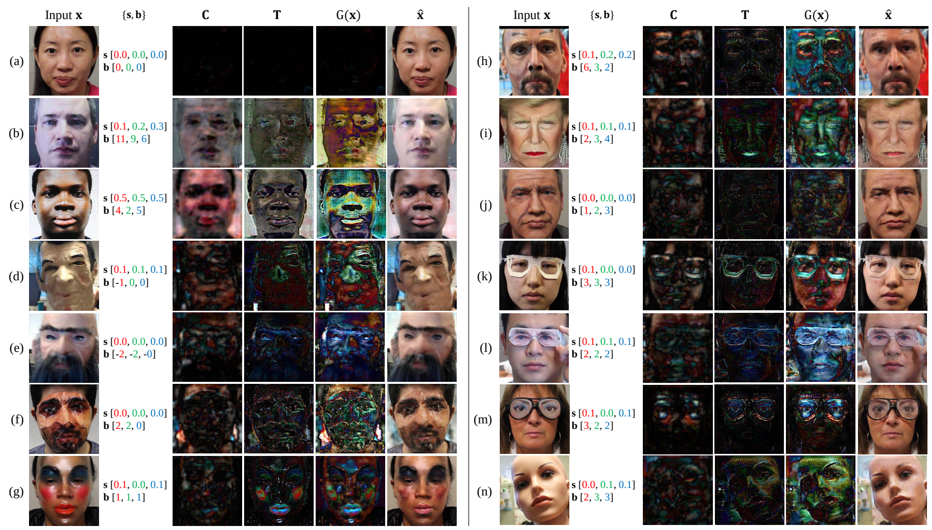

Fig. 7: Examples of spoof trace disentanglement on SiW-M. The (a)-(n) items are live, print, replay, half mask, silicone mask, paper mask, transparent mask, obfuscation makeup, impersonation makeup, cosmetic makeup, paper glasses, partial paper, funny eye glasses, and mannequin head. The first column is the input face, the 2nd-4th columns are the spoof trace elements $\{\mathbf{s}, \mathbf{b}, \mathbf{C}, \mathbf{T}\}$, the 5th column is the overall spoof traces, and the last column is the reconstructed live.

regression model. To validate the effectiveness of GAN, we report the results of ESR+GAN. In this case, the generator outputs a single-layer spoof trace with the input size, instead of the proposed four elements. To demonstrate the effectiveness of disentangled 4-element spoof trace, we change the single layer to the proposed $\{\mathbf{s}, \mathbf{b}, \mathbf{C}, \mathbf{T}\}$, denoted as ESR+D-GAN. In addition, we evaluate the effect of synthesized data via enabling training step 3, denoted as ESR+GAN+L_P and ESR+D-GAN+L_P (*i.e.*, our final approach).

Shown in Tab. 4, the baseline achieves a decent performance of ACER 2.6%. Using a generative model can improve the ACER from 2.6% to 2.1%, while a proper disentanglement can improve to 1.6%. Shown in Fig. 6, ESR+D-GAN produces higher-quality spoof traces than ESR+GAN. If feeding bad-quality spoof samples in the training step 3, it would increase the error rate from 2.1% to 4.5%. But if feeding the good-quality spoof samples, it can achieve a significant improvement from 1.6% to 1.1%.

4.6 Visualization

As shown in Fig. 7, we successfully disentangle various spoof traces. *E.g.*, strong color distortion shows up in print/replay attacks (Fig. 7b-c). Moiré patterns in the replay attack are well detected (Fig. 7c). For makeup attacks (Fig. 7h-j), the fake eyebrows, lipstick, artificial wax, and cheek shade are clearly detected. The

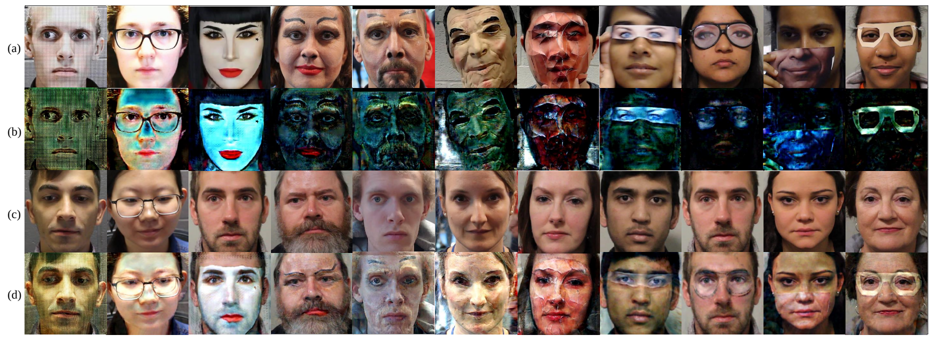

Fig. 8: Examples of the spoof data synthesis. (a) The source spoof samples $\mathbf{I}_i$. (b) The disentangled spoof traces $G(\mathbf{I}_i)$. (c) The target live faces $\mathbf{I}_j$. (d) The synthesized spoof $\mathbf{I}_j + G_{i \rightarrow j}$.

folds and edges in paper-crafted mask (Fig. 7f) are well detected. Although our method cannot provide a convincing estimation for a few spoof types (*e.g.*, funny eye glasses in Fig. 7m), the model effectively focuses on the correct region and disentangles parts of the traces.

Additionally, we show some examples of spoof synthesis using the disentangled spoof traces in Fig. 8. The spoof traces can be precisely transferred to a new face without changing the identity of the target face. Thanks to the proposed 3D warping layer, the geometric discrepancy between the source spoof trace and the target face is corrected during the synthesis. These two figures demonstrate that our approach disentangles visually convincing spoof traces that help face anti-spoofing.

5 Conclusions

This work proposes a network (STDN) to tackle a challenging problem of disentangling spoof traces from faces. With the spoof traces, we reconstruct the live faces as well as synthesize new spoofs. To correct the geometric discrepancy in synthesis, we propose a 3D warping layer to deform the traces. The disentanglement not only improves the SOTA of both known and unknown anti-spoofing, but also provides visual evidence to support the model's decision.

Acknowledgment This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. 2017-17020200004. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

References

1. Explainable Artificial Intelligence (XAI). `https://www.darpa.mil/program/explainable-artificial-intelligence`
2. IARPA research program Odin). `https://www.iarpa.gov/index.php/research-programs/odin`
3. Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al.: Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion (2020)
4. Atoum, Y., Liu, Y., Jourabloo, A., Liu, X.: Face anti-spoofing using patch and depth-based CNNs. In: IJCB. IEEE (2017)
5. Bigun, J., Fronthaler, H., Kollreider, K.: Assuring liveness in biometric identity authentication by real-time face tracking. In: International Conference on Computational Intelligence for Homeland Security and Personal Safety (CIHSPS). IEEE (2004)
6. Boulkenafet, Z., Komulainen, J., Hadid, A.: Face anti-spoofing based on color texture analysis. In: ICIP. IEEE (2015)
7. Boulkenafet, Z., Komulainen, J., Hadid, A.: Face antispoofing using speeded-up robust features and fisher vector encoding. Signal Processing Letters (2016)
8. Boulkenafet, Z., Komulainen, J., Li, L., Feng, X., Hadid, A.: OULU-NPU: A mobile face presentation attack database with real-world variations. In: FG. IEEE (2017)
9. Boylan, J.F.: Will deep-fake technology destroy democracy? In: The New York Times (2018)
10. Bulat, A., Tzimiropoulos, G.: How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3d facial landmarks). In: ICCV. IEEE (2017)
11. Chang, H., Lu, J., Yu, F., Finkelstein, A.: Paired cycleGAN: Asymmetric style transfer for applying and removing makeup. In: CVPR. IEEE (2018)
12. Chen, C., Xiong, Z., Liu, X., Wu, F.: Camera trace erasing. In: CVPR (2020)
13. Dale, K., Sunkavalli, K., Johnson, M.K., Vlasic, D., Matusik, W., Pfister, H.: Video face replacement. In: TOG. ACM (2011)
14. Deb, D., Zhang, J., Jain, A.K.: Advfaces: Adversarial face synthesis. arXiv preprint arXiv:1908.05008 (2019)
15. Esser, P., Sutter, E., Ommer, B.: A variational U-Net for conditional appearance and shape generation. In: CVPR. IEEE (2018)
16. Feng, L., Po, L.M., Li, Y., Xu, X., Yuan, F., Cheung, T.C.H., Cheung, K.W.: Integration of image quality and motion cues for face anti-spoofing: A neural network approach. Journal of Visual Communication and Image Representation (2016)
17. de Freitas Pereira, T., Anjos, A., De Martino, J.M., Marcel, S.: LBP-TOP based countermeasure against face spoofing attacks. In: ACCV. Springer (2012)
18. de Freitas Pereira, T., Anjos, A., De Martino, J.M., Marcel, S.: Can face anti-spoofing countermeasures work in a real world scenario? In: ICB. IEEE (2013)
19. Frischholz, R.W., Werner, A.: Avoiding replay-attacks in a face recognition system using head-pose estimation. In: International SOI Conference. IEEE (2003)
20. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
21. Guo, J., Zhu, X., Xiao, J., Lei, Z., Wan, G., Li, S.Z.: Improving face anti-spoofing by 3D virtual synthesis. arXiv preprint arXiv:1901.00488 (2019)

22. ISO/IEC JTC 1/SC 37 Biometrics: Information technology biometric presentation attack detection part 1: Framework. international organization for standardization. https://www.iso.org/obp/ui/iso (2016)
23. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR. IEEE (2017)
24. Jourabloo, A., Liu, Y., Liu, X.: Face de-spoofing: Anti-spoofing via noise modeling. In: ECCV. Springer (2018)
25. Kollreider, K., Fronthaler, H., Faraj, M.I., Bigun, J.: Real-time face detection and motion analysis with application in "liveness" assessment. TIFS (2007)
26. Komulainen, J., Hadid, A., Pietikäinen, M.: Context based face anti-spoofing. In: BTAS. IEEE (2013)
27. Li, L., Feng, X., Boulkenafet, Z., Xia, Z., Li, M., Hadid, A.: An original face anti-spoofing approach using partial convolutional neural network. In: Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA). IEEE (2016)
28. Liu, F., Zeng, D., Zhao, Q., Liu, X.: Disentangling features in 3D face shapes for joint face reconstruction and recognition. In: CVPR. IEEE (2018)
29. Liu, Y., Jourabloo, A., Liu, X.: Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In: CVPR. IEEE (2018)
30. Liu, Y., Jourabloo, A., Ren, W., Liu, X.: Dense face alignment. In: ICCV Workshops. IEEE (2017)
31. Liu, Y., Stehouwer, J., Jourabloo, A., Liu, X.: Deep tree learning for zero-shot face anti-spoofing. In: CVPR. IEEE (2019)
32. Liu, Y., Stehouwer, J., Jourabloo, A., Liu, X.: Presentation attack detection for face in mobile phones. Selfie Biometrics (2019)
33. Määttä, J., Hadid, A., Pietikäinen, M.: Face spoofing detection from single images using micro-texture analysis. In: IJCB. IEEE (2011)
34. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: ICCV. IEEE (2017)
35. Pan, G., Sun, L., Wu, Z., Lao, S.: Eyeblink-based anti-spoofing in face recognition from a generic webcamera. In: ICCV. IEEE (2007)
36. Patel, K., Han, H., Jain, A.K.: Cross-database face antispoofing with robust feature representation. In: CCBR. Springer (2016)
37. Patel, K., Han, H., Jain, A.K.: Secure face unlock: Spoof detection on smartphones. TIFS (2016)
38. Qin, Y., Zhao, C., Zhu, X., Wang, Z., Yu, Z., Fu, T., Zhou, F., Shi, J., Lei, Z.: Learning meta model for zero-and few-shot face anti-spoofing. arXiv preprint arXiv:1904.12490 (2019)
39. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. Springer (2015)
40. Schuckers, S.A.: Spoofing and anti-spoofing measures. Information Security technical report (2002)
41. Shao, R., Lan, X., Li, J., Yuen, P.C.: Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In: CVPR. IEEE (2019)
42. Shao, R., Lan, X., Yuen, P.C.: Regularized fine-grained meta face anti-spoofing. arXiv preprint arXiv:1911.10771 (2019)
43. Stehouwer, J., Jourabloo, A., Liu, Y., Liu, X.: Noise modeling, synthesis and classification for generic object anti-spoofing. In: CVPR. IEEE (2020)

44. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
45. Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., Nießner, M.: Face2face: Real-time face capture and reenactment of RGB videos. In: CVPR. IEEE (2016)
46. Tran, L., Yin, X., Liu, X.: Disentangled representation learning GAN for pose-invariant face recognition. In: CVPR. IEEE (2017)
47. Tran, L., Yin, X., Liu, X.: Representation learning by rotating your faces. IEEE Trans. on Pattern Analysis and Machine Intelligence **41**(12), 3007–3021 (2019)
48. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional GANs. In: CVPR. IEEE (2018)
49. Yang, J., Lei, Z., Li, S.Z.: Learn convolutional neural network for face anti-spoofing. arXiv preprint arXiv:1408.5601 (2014)
50. Yang, J., Lei, Z., Liao, S., Li, S.Z.: Face liveness detection with component dependent descriptor. In: ICB. IEEE (2013)
51. Yang, X., Luo, W., Bao, L., Gao, Y., Gong, D., Zheng, S., Li, Z., Liu, W.: Face anti-spoofing: Model matters, so does data. In: CVPR. IEEE (2019)
52. Zakharov, E., Shysheya, A., Burkov, E., Lempitsky, V.: Few-shot adversarial learning of realistic neural talking head models. arXiv preprint arXiv:1905.08233 (2019)
53. Zhang, Z., Tran, L., Yin, X., Atoum, Y., Wan, J., Wang, N., Liu, X.: Gait recognition via disentangled representation learning. In: CVPR. IEEE (2019)
54. Zhao, C., Qin, Y., Wang, Z., Fu, T., Shi, H.: Meta anti-spoofing: Learning to learn in face anti-spoofing. arXiv preprint arXiv:1904.12490 (2019)
55. Zollhöfer, M., Thies, J., Garrido, P., Bradley, D., Beeler, T., Pérez, P., Stamminger, M., Nießner, M., Theobalt, C.: State of the art on monocular 3D face reconstruction, tracking, and applications. In: Computer Graphics Forum. Wiley Online Library (2018)