Face Anti-Spoofing via Disentangled Representation Learning

Ke-Yue Zhang[1,2*], Taiping Yao[2*], Jian Zhang[2], Ying Tai[2†], Shouhong Ding[2], Jilin Li[2], Feiyue Huang[2], Haichuan Song[1], and Lizhuang Ma[1]

[1] East China Normal University, Shanghai, China
[2] Youtu Lab, Tencent, Shanghai, China
51184501178@stu.ecnu.edu.cn; hcsong@cs.ecnu.edu.cn; lzma@sei.ecnu.edu.cn;
{taipingyao,timmmyzhang,yingtai,ericshding,jerolinli,garyhuang}@tencent.com

Abstract. Face anti-spoofing is crucial to security of face recognition systems. Previous approaches focus on developing discriminative models based on the features extracted from images, which may be still entangled between spoof patterns and real persons. In this paper, motivated by the disentangled representation learning, we propose a novel perspective of face anti-spoofing that disentangles the liveness features and content features from images, and the liveness features is further used for classification. We also put forward a Convolutional Neural Network (CNN) architecture with the process of disentanglement and combination of low-level and high-level supervision to improve the generalization capabilities. We evaluate our method on public benchmark datasets and extensive experimental results demonstrate the effectiveness of our method against the state-of-the-art competitors. Finally, we further visualize some results to help understand the effect and advantage of disentanglement.

Keywords: Face anti-spoofing, generative model, disentangled representation

1 Introduction

With superior performance than human, face recognition techniques are widely used in smart devices, access control and security scenarios. However, the associated safety issues raise concern of public since the accessment of human face is low-cost and a well-designed makeup can easily fool this biometric mechanism. These face spoofs, also called Presentation Attacks (PA), vary from simpler printed facial images, video replays to more complicated 3D mask and facial cosmetic makeup. Theoretically, face recognition systems are vulnerable to all spoofs without specific defense, which incurs malicious attacks of hackers, but also encourages the boosting of robust face anti-spoofing algorithm.

* equal contribution.
† corresponding author.

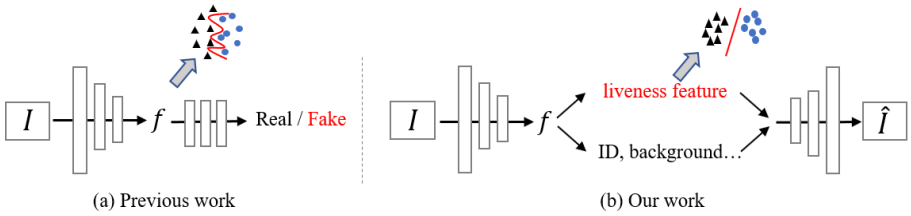


Fig. 1. Comparison between previous entangled framework and our disentangled framework. Previous works learn entangled features which are easily overfitting to the training dataset. In contrast, our disentangled framework distills the liveness features with proper constraints and supervision.

Since the primary facial spoof images or videos contain artifacts, researchers put forward several methods based on texture analysis. Some handcrafted features are combined with anti-spoofing algorithms, such as Local Binary Pattern(LBP) [6, 14, 15, 27], Histogram of Oriented Gridients(HOG) [21, 41], Scale Invariant Feature Transform(SIFT) [30], *etc.* These cue-based methods use handcrafted features to detect the motion cues such as lip movement or eye blinking for authentication. However, these methods couldn't deal with the replay attacks with high-fidelity. Recently, Convolutional Neural Network(CNN)-based methods have achieved great progress in face anti-spoofing [40, 29, 23]. Basically, these methods treat the security issue as a binary classification problem with softmax loss. However, they are lack of generalization capability for overfitting on the training dataset. Despite many methods use auxiliary information (*i.e.*, facial depth map, rppg signals, *etc.*) to further guide the network in telling the difference between real and spoof [18, 19, 24], these pre-defined characteristics are still insufficient for depicting the authentic abstract spoof patterns since exhausting all possible constraints is impossible.

Thus the crucial step of face anti-spoofing does not lie in how to precisely pre-define the spoof patterns, but how to achieve the spoof patterns from high-dimensional extracted representations. One possible solution is disentangling representations into separate parts. In disentangle learning [38, 17], it's a consensus that high-dimensional data can be explained by substantially lower dimensional and semantically meaningful latent representation variables. While in face anti-spoofing, the spoof patterns can be viewed as one kind of attributes of face, not just a certain irrelevant noise type or the combination. Hence, the problem is transformed into how we can directly target to the liveness information from all the variations of facial images.

As shown in Fig. 1, we propose a novel disentangled face anti-spoofing technique via separating the latent representation. Motivated by [17], we assume the latent space of facial images can be decomposed into two sub-spaces: liveness space and content space. Liveness features corresponds to the liveness-related information, while content features integrate remaining liveness-irrelated information in the input images, such as ID and lighting. However, in disentangled learning procedure, there exist two challenges on missing 1) corresponding gen-

uine images for spoof images in translation process and vice versa, 2) clear research about properties of liveness features in face anti-spoofing literature.

To tackle above challenges, we introduce low-level texture and high-level depth characteristics to further facilitate disentanglement. For the first challenge, we adopt a Generative Adversarial Network(GAN)-like discriminator to guarantee the plausibility of the translated images. An auxiliary depth estimator is then introduced to ensure that the liveness information has also been exchanged between genuine and spoof images. For the second challenge, checking the properties of liveness features is equivalent to making liveness and content features independent in disentangled framework. In order to spilt liveness and content space, we encode the translated images to get reconstructed liveness features again. With bidirectional reconstruction loss on images and latent codes, liveness features of diverse spoof patterns are thoroughly extracted in a self-supervised way. To further regularize liveness space, we introduce a novel LBP map supervision. Finally, the spoof classification could be solved in a smaller and more discriminative liveness feature space. Hence, our architecture is more likely to achieve good generalization capability.

To sum up, the contributions of this work are three-fold:

• We address face anti-spoofing via disentangled representation learning, which separates latent representation into liveness features and content features.

• We combine low-level texture and high-level depth characteristics to regularize liveness space, which facilitates disentangled representation learning.

• Abundant experiments and visualizations are presented to reveal the properties of liveness features, which demonstrates the effectiveness of our method against the state-of-the-art competitors.

2 Related Work

Our method introduces disentangled representation learning to solve face anti-spoofing. Previous related work lies in two perspectives: face anti-spoofing and attributes disentanglement.

Face Anti-spoofing. Early researches focused on hand-crafted feature descriptors, such as LBP [6, 14, 15, 27], HOG [21, 41], SIFT [30] and SURF [7], to project the faces into a low-dimension feature space, where traditional classifiers such as SVM are utilized for judgement. There are also some methods using information from different domains, such as HSV and YCrCb color space [6, 8], temporal domain [34, 2, 12, 39], and Fourier spectrum [22]. However, these hand-crafted feature-based methods cannot achieve high accuracy due to limited representation capacity.

With the rise of deep learning, researchers attempted to tackle the face anti-spoofing with CNN-based features. Initially, [40, 29, 23] treated the task as a binary classification problem with softmax loss. Compared to hand-crafted features, such models gained higher accuracy in intra-testing settings. However, due to the overfitting on training data, their generalization ability are relatively poor.

In order to improve the generalization ability, many methods attempted to utilize auxiliary supervision to guide networks. [24] attempted to guide networks with auxiliary supervision of facial depth information and remote-photoplethysmography (r-ppg) signal. [18] utilized the spoof images to estimate the spoofing-relevant noise pattern. [33] adopted the strategy of domain generalization to achieve improvements in cross-testing. These auxiliary supervision indeed improve generalization. However these methods all handle this problem in the whole feature space, which is disturbed by irrelevant factors.

Disentangled Representation. The key intuition about disentangling is that disentangled representation could factorize the data into distinct informative factors of variations [25]. [16, 10] aimed to learn disentangled representations without supervision. [38] divided latent features of an facial image into different parts, where each part encodes a single attribute. [17] assumed that latent space of images can be decomposed into a content space and a style space.

These works inspire us decompose the features of an facial image into content features and liveness features. In face anti-spoofing, content features correspond to the liveness-irrelated information in the images, such as ID, background, Scene lighting, *etc.* On the contrary, liveness features are the key to distinguishing between real persons and attacks. Obviously, we could tackle the face anti-spoofing in the liveness feature space. However, there are many challenges in disentangled learning procedure, such as without the ground truth of the recombined images, diverse styles of spoof, *etc.* In this paper, we combine low-level texture and high-level depth characteristics to facilitate disentangled representation learning.

3 Disentanglement Framework

Our framework mainly consists of two parts: the disentanglement process and the auxiliary supervision. As the core component of our framework explained in Sec.3.1, the disentanglement process separates the representation into two independent factors, which are liveness features and content features, respectively. As illustrated in Sec.3.2, depth, texture, and discriminative constraints are utilized as auxiliary supervision. By introducing these three auxiliary nets, we consolidate liveness features and further facilitate the disentanglement process. Fig.2 illustrates the overview of our method and the entire learning process.

3.1 Disentanglement Process

Disentanglement process is designed to separate liveness features and content features by exchanging and recombining these two features. Inputs of the disentanglement part are two unpaired images A and B, where A is randomly chosen from live face images and B is chosen from spoof images. In the encoder part, we first use a convolution block to extract latent code Z from inputs. And then two independent convolutional sub-networks encode latent code Z into liveness features L and content features C respectively. This specific structure separates

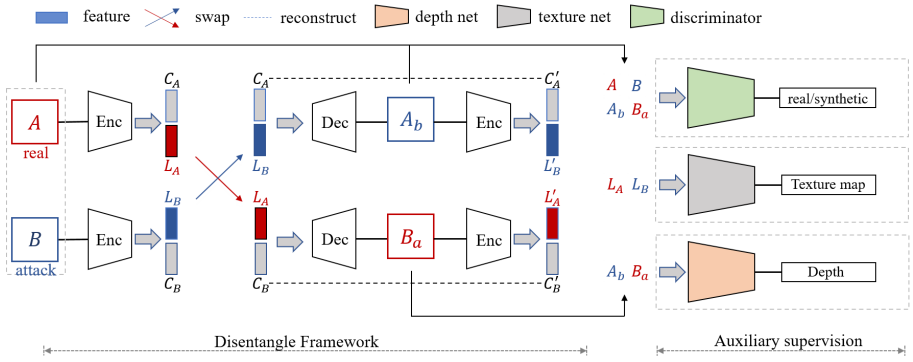


Fig. 2. Overview of our disentanglement framework. The features of an image are divided into two parts, content features and liveness features. By exchanging the liveness features of the real person and the attack, we can get different reconstructed images with the same content but their liveness attributes are changed. Texture net, depth net and discriminator are proposed to facilitate disentangled representation learning.

two features from convolving with each other. According to the above process, we can get L_A, C_A and L_B, C_B respectively by encoding images A, B. Then, we exchange the liveness part L_A and L_B to obtain images A_b and B_a.

$$A_b = Dec(C_A, L_B), B_a = Dec(C_B, L_A). \tag{1}$$

Because the liveness features determine the liveness attributes of the image, we suppose that A_b is a spoof version of image A, and B_a is a genuine version of image B. To better decode the latent code back into images, the architecture we used for the decoder is symmetrical with the encoder. Besides, following the U-Net [32] structure, the shortcuts are added from the middle layers in encoder to the corresponding layers in decoder to bring the original information as an auxiliary context for improving visual quality. To further guarantee that liveness information and content information can be split completely, we encode images A_b, B_a again to get C'_A, L'_B and C'_B, L'_A, and introduce a bidirectional reconstruction loss [17] to encourage reconstruction in two sequential processes (*i.e.*, from images to images and from latent features to latent features).

Image Reconstruction. The combination of the encoder and decoder should be capable of reconstructing any image x_i from the datasets:

$$\mathcal{L}_{x_i}^{rec} = \mathbb{E}_{x_i \sim p(x_i)} \left\| D(E(x_i)) - x_i \right\|_1, \tag{2}$$

where $p(x_i)$ is the distribution of original images in the datasets, E is the encoder and D is the decoder.

Latent Reconstruction. Given a pair of liveness features and content features at translation time, we should be able to reconstruct it after decoding and encoding.

$$\mathcal{L}_{z_i}^{rec} = \mathbb{E}_{z_i \sim q(z_i)} \left\| E(D(z_i)) - z_i \right\|_1 \tag{3}$$

Table 1. The details of Auxiliary nets of our method.

LBP Net			Depth Net			Discriminator		
Layer	chan./Stri.	Out.Size	Layer	chan./Stri.	Out.Size	Layer	chan./Stri.	Out.Size
Input:liveness features			**Input**:image			**Input**:image		
			conv2-0	64/1	256			
conv1-0	384/1	32	conv2-1	128/1	256	conv3-1	64/1	256
			conv2-2	196/1	256	pool3-1	-/2	128
			conv2-3	128/1	256			
			pool2-1	-/2	128			
conv1-1	128/1	32	conv2-4	128/1	128	conv3-2	128/1	128
			conv2-5	196/1	128	pool3-2	-/2	64
			conv2-6	128/1	128			
			pool2-2	-/2	64			
conv1-2	64/1	32	conv2-7	128/1	64	conv3-3	256/1	64
			conv2-8	196/1	64	pool3-3	-/2	32
			conv2-9	128/1	64			
			pool2-3	-/2	32			
conv1-2			pool2-1+pool2-2+pool2-3			vectorize		
			conv2-10	128/1	32			
			conv2-11	64/1	32			
conv1-3	1/1	32	conv2-12	1/1	32	fc3-1	1/1	2

where z_i is the combination of liveness features L_i and content features C_i, and $q(z_i)$ is the distribution of latent code.

3.2 Auxiliary Supervision

In this section, we introduce three auxiliary supervision: LBP map, depth map and discrimination supervision, which promote the disentanglement process collaboratively. Discrimination supervision ensures the visual quality of generated image. Depth and LBP supervision are plugged into different parts to guarantee the generated image being in correct category when their liveness features are exchanged. The LBP map and depth map together regularize the liveness feature space, making it the key factor to distinguish between real persons and spoof patterns. The detailed structure of three auxiliary nets are illustrated in Tab. 1. Each convolutional layer is followed by a batch normalization layer and a Rectified Linear Unit (ReLU) activation function with 3×3 kernel size.

Texture Auxiliary Supervision. Liveness features are the essential characteristic of a face image, which determine the liveness categories of the image. Thus when swapping liveness features between a real person and an attack, categories of images and estimated depth maps should be changed simultaneously. And the estimated depth map is usually considered to be related to factors such as facial lighting and shadows, which are contained in the texture information of the face. What's more, previous works have proven that texture is an important clue in face anti-spoofing. Therefore, LBP map is adopted to regularize the liveness features in disentanglement framework. Although LBP features contain some additional information, proposed disentanglement framework utilize Latent Reconstruction Loss to constrain liveness features to learn only essential information. To make the features distinctive, for the genuine faces, we use the

LBP map extracted by the algorithm in [1] as texture supervision. While for the spoof face, a zero map serves as the ground truth.

$$\begin{aligned}\mathcal{L}_{lbp} = &\mathbb{E}_{l_i \sim P(l_i), x_i \sim P(x_i)} \left\| LBP(l_i) - lbp_{x_i} \right\|_1 \\ &+ \mathbb{E}_{l_i \sim N(l_i), x_i \sim N(x_i)} \left\| LBP(l_i) - \mathbf{0} \right\|_1\end{aligned} \tag{4}$$

where LBP is the LBP Estimator Net, $P(x_i)$ is the distribution of live face images in the datasets, $P(l_i)$ is the distribution of liveness space of live face images, $N(x_i)$ is the distribution of spoof images in the datasets, $N(l_i)$ is the distribution of liveness space of spoof images, lbp_{x_i} means the lbp map of live face images x_i and $\mathbf{0}$ means the zero maps for spoof images.

Depth Supervision. Depth map is commonly used as an auxiliary supervision in face anti-spoofing tasks. In our disentanglement framework, we combine LBP map and depth map supervision to regularize the liveness feature space. Similarly as LBP branch, we use pseudo-depth as ground truth for live face images and zero map for spoof images. The pseudo-depth is estimated by the 3D face alignment algorithm in [13]. During training stage, depth net only provides the supervision and does not update parameters. Since the reconstructed images A' and generated B_a are live images, and the reconstructed images B' and generated A_b are spoof images, the corresponding depth map of above images should be the depth of the face in images A, B and two zero maps. Then the loss of depth is formulated as:

$$\mathcal{L}_{dep} = \mathbb{E}_{x_i \sim N(x_i)} \left\| Dep(x_i) - \mathbf{0} \right\|_1 + \mathbb{E}_{x_i \sim P(x_i)} \left\| Dep(x_i) - dep_{x_i} \right\|_1 \tag{5}$$

where Dep is the parameters fixed depth net, $P(x_i)$ is the distribution of live face images, $N(x_i)$ is the distribution of spoof images, dep_{x_i} is the depth map of live face images x_i and $\mathbf{0}$ means the zero maps for spoof images correspondingly.

Discriminative Supervision. For ensuring the visual plausibility of generated images, we apply discriminative supervision on the generated images. Discriminative supervision is used for distinguishing between the generated images (A', B', A_b, B_a) and the original images (A, B). At the same time, disentanglement framework aims to produce plausible images which would be classified as non-synthetic images under discriminative supervision. Nevertheless, the receptive field of a single discriminator is limited for large images. We use multi-scale discriminators [36] to address this problem. Specifically, we deploy two identical discriminators with varied input resolution. The discriminator with a larger input scale is denoted as D_1, which guides the disentanglement net to generate finer details. And the other discriminator with a smaller input scale is denoted as D_2, which guides the disentanglement net to preserve more global information. In the training process, there are two consecutive steps in each iteration. In the first step, we fix disentanglement net and update the discriminator,

$$\begin{aligned}\mathcal{L}_D^{Dis} = &- \mathbb{E}_{I\in R} log(D_1(I)) - \mathbb{E}_{I\in G} log(1 - D_1(I)) \\ &- \mathbb{E}_{I\in R} log(D_2(I)) - \mathbb{E}_{I\in G} log(1 - D_2(I))\end{aligned} \tag{6}$$

where R and G are the sets of real and generated images respectively. In the second step, we fix the discriminator and update the disentanglement net,

$$\mathcal{L}_D^{Gen} = -\mathbb{E}_{I\in G} log(D_1(I)) - \mathbb{E}_{I\in G} log(D_2(I)) \tag{7}$$

Loss Function. The final loss function of training process is the weighted summation of the loss functions above,

$$\mathcal{L} = \mathcal{L}_D^{Gen} + \lambda_1 \mathcal{L}_{x_i}^{rec} + \lambda_2 \mathcal{L}_{z_i}^{rec} + \lambda_3 \mathcal{L}_{dep} + \lambda_4 \mathcal{L}_{lbp} \tag{8}$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are the weights. Following common adversarial training pipeline, we alternately optimize discriminator and disentanglement net. The weights are empirically selected to balance each loss term.

4 Experimental Results

4.1 Experimental Setting

Databases. We test our method on four face anti-spoofing databases: Oulu-NPU [9], SiW [24], CASIA-MFSD [43] and Replay-Attack [11]. We evaluate our intra-testing performance on Oulu-NPU and SiW datasets, and conduct cross-testing by training on Replay-Attack or CASIA-MFSD and testing on the other.
Metrics. To compare with previous works, we report the performance via the following metrics: Attack Presentation Classification Error Rate (*APCER*) [4], Bona Fide Presentation Classification Error Rate (*BPCER*) [4], Average Classification Error Rate (*ACER*) = (*APCER*+*BPCER*)/2 [4] and Half Total Error Rate (*HTER*) = (False Acceptance Rate + False Rejection Rate)/2 [4].
Implementation Details. All datasets above are stored in video format. We use a face detector or face location files in datasets to crop the face and resize it to 256×256. For each frame, we combine scores of estimated LBP map and Depth map to detect attack for fully utilizing the low-level texture information and high-level global information, as the methods in [18], *i.e.*, $score = (\|map_{lbp}\| + \|map_{depth}\|)/2$. We implement method in Pytorch [28]. Models are trained with batch size of 4. In each epoch, we select negative images and positive images with the ratio $1:1$. To train network, we use learning rate of 1e-5 with Adam optimizer [20] and set λ_1 to λ_4 in Eqn. 8 as 10, 1, 1 and 2. Depth net is pre-trained and remains fixed during the training of other three nets, and all networks are trained with the same data in each protocol. In inference stage, reconstruction and translation procedure are both detached, thus the speed of our method is acceptable, which achieves 77.97±0.18 FPS on GeForce GTX 1080.

Table 2. The intra-testing results of four protocols of Oulu-NPU dataset.

Protocol	Method	APCER(%)	BPCER(%)	ACER(%)
1	STASN[42]	1.2	2.5	1.9
	Auxiliary[24]	1.6	1.6	1.6
	FaceDe-S[18]	**1.2**	1.7	1.5
	FAS-TD[37]	2.5	**0.0**	**1.3**
	Ours	1.7	0.8	**1.3**
2	Auxiliary[24]	2.7	2.7	2.7
	GRADIANT[5]	3.1	1.9	2.5
	STASN[42]	4.2	**0.3**	2.2
	FAS-TD[37]	1.7	2.0	**1.9**
	Ours	**1.1**	3.6	2.4
3	FaceDe-S[18]	4.0±1.8	3.8±1.2	3.6±1.6
	Auxiliary[24]	2.7±1.3	3.1±1.7	2.9±1.5
	STASN[42]	4.7±3.9	**0.9±1.2**	2.8±1.6
	BASN[19]	**1.8±1.1**	3.6±3.5	2.7±1.6
	Ours	2.8±2.2	1.7±2.6	**2.2±2.2**
4	FAS-TD[37]	14.2±8.7	4.2±3.8	9.2±6.0
	STASN[42]	6.7±10.6	8.3±8.4	7.5±4.7
	FaceDe-S[18]	**5.1±6.3**	6.1±5.1	5.6±5.7
	BASN[19]	6.4±8.6	**3.2±5.3**	4.8±6.4
	Ours	5.4±2.9	3.3±6.0	**4.4±3.0**

4.2 Experimental Comparison

In this section, we show the superiority of disentanglement and further illustrate translation results. To verify the performance of our method, we conduct experiments on Oulu-NPU and SiW for intra-testing results, CASIA and Replay-Attack for cross-testing results. Then we demonstrate some examples to show details of translation, which verifies the validity of the liveness features.

Intra-Testing. Intra-testing is evaluated on Oulu-NPU and SiW datasets. We utilize the protocols defined in each dataset. Tab. 2 shows the comparison of our method with the best four methods on Oulu dataset. Our method achieves better results in protocols 1, 3 and 4, while gets slightly worse ACER in protocol 2. For protocol 4 evaluating all variations in Oulu, our method gets the best results, which verifies that our method has better generalization performance. Following [19], we report the ACER on three protocols of SiW. Tab. 3 shows that our method achieves better results among the frame based methods.

Cross-Testing. We evaluate the generalization capability by conducting cross-dataset evaluations. Following the related work, CASIA-MFSD and Replay-Attack are used for the experiments and the results are measured in HTER. The results are shown in Tab. 4. For fair comparison, we compare with methods using only single frame information. Our method achieves 1.2 pp lower HTER

Table 3. The intra-testing results of three protocols of SiW dataset.

Protocol	Method	APCER(%)	BPCER(%)	ACER(%)
1	Auxiliary[24]	3.58	3.58	3.58
	STASN[42]	-	-	1.00
	FAS-TD[37]	0.96	0.50	0.73
	BASN[19]	-	-	0.37
	Ours	0.07	0.50	**0.28**
2	Auxiliary[24]	0.57±0.69	0.57±0.69	0.57±0.69
	STASN[42]	-	-	0.28±0.05
	FAS-TD[37]	0.08±0.17	0.21±0.16	0.15±0.14
	BASN[19]	-	-	0.12±0.03
	Ours	0.08±0.17	0.13±0.09	**0.10±0.04**
3	STASN[42]	-	-	12.10±1.50
	Auxiliary[24]	8.31±3.81	8.31±3.80	8.31±3.81
	BASN[19]	-	-	6.45±1.80
	FAS-TD[37]	3.10±0.79	3.09±0.83	**3.10±0.81**
	Ours	9.35±6.14	1.84±2.60	5.59±4.37

Table 4. The cross-testing results on CASIA-MFSD and Replay-Attack.

Method	Train CASIA MFSD	Test Replay Attack	Train Replay Attack	Test CASIA MFSD
Motion-Mag[3]	50.1%		47.0%	
Spectral cubes[31]	34.4%		50.0%	
LowPower[35]	30.1%		35.6%	
CNN[40]	48.5%		45.5%	
STASN[42]	31.5%		30.9%	
FaceDe-S[18]	28.5%		41.1%	
Auxiliary[24]	27.6%		**28.4%**	
BASN[19]	23.6%		29.9%	
Ours	**22.4%**		30.3%	

than the state-of-the-art from CASIA-MFSD to Replay-Attack and gets comparable HTER from Replay-Attack to CASIA-MFSD. This results also prove that our method with disentanglement has better generalization capability.

Translation Result. We demonstrate some examples of translation from Oulu protocol 1 in three groups: live-spoof, live-live, spoof-spoof, as shown in Fig. 3. In the live-spoof group, depth map changes with the exchange of the liveness features. While in live-live group and spoof-spoof group, the liveness features changing doesn't result in the change of depth map, which implies that liveness features indeed determine whether the image is live. The difference between each

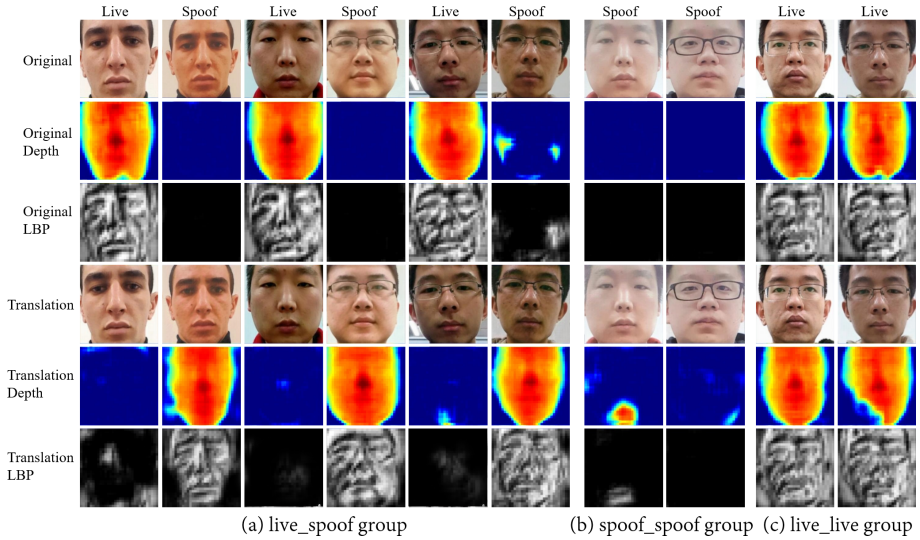


Fig. 3. Illustrations of translation results with corresponding depth map and LBP map. We swap liveness features between every two columns. The exchanging of depth and LBP map verifies that liveness features are the key part of live face images.

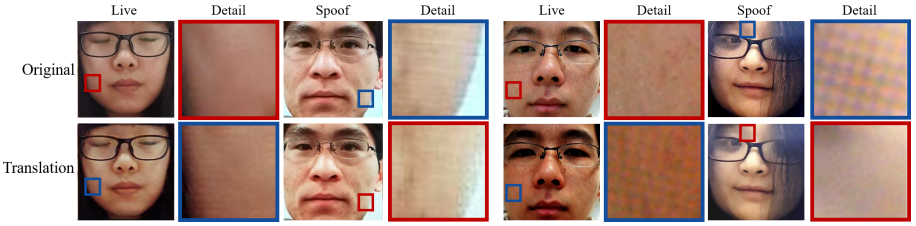


Fig. 4. Illustrations of exchanging live and spoof details. The first row is the original image, and the second row is the translation results. Red rectangular is referred as the details of live images while blue refer to the details of the spoof images.

two columns of live face and spoof images is **light, ID, background** respectively. As the translation shows, there are no changes about these factors with the category changing, which means that liveness features do not contain these factors. Fig. 4 shows two sets of live and attack images and their local area details. As shown in the figure, there is a big difference between the local details of the real person and the attack, and the attack images often have some repetitive streaks. And after combining the liveness features from the attack images, the local details of the translation results are similar to the corresponding attacks, which shows that the liveness features have not only learned the difference between real people and attacks, but also learned different attack details.

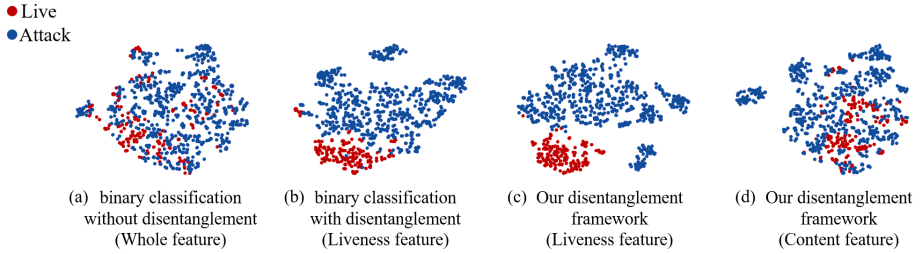


Fig. 5. Visualization of feature distributions from different methods. We use different constraint on livness feature or whole feature and draw the corresponding feature in brackets by t-SNE [26].

Table 5. The comparison of different supervision and combination.

Method / ACER	BC-Depth	0/1 Map-Depth	LBP-LBP	Depth-Depth	Depth-LBP	Ours
liveness features	3.64	3.02	1.87	1.69	1.65	**1.56**
fusion	2.78	2.50	2.40	1.80	1.50	**1.25**

4.3 Ablation Study

To study the effect of disentanglement, different supervision and score fusion methods, we conduct ablation experiments on Oulu-NPU protocol 1 respectively.

Liveness Feature Distribution. We use t-SNE [26] to visualize the features from different methods, which includes 500 live face images and 2,000 spoof images, as illustrated in Fig. 5. Comparing (a) with (b), we conclude that disentanglement indeed finds a sub-space where the features of live and spoof can be distinguished more easily. For comparison between (b) and our method (c), low level LBP supervision on the liveness features improves discrimination between live and attack. The difference between (c) and (d) proves that liveness features indeed can distinguish between real and attack while content features can't.

Different Supervision. a In our method, we propose the supervision combining low-level LBP texture and high-level depth information. We compare this combination of supervision with other five ablation methods, which are all based on the proposed disentanglement framework: (1) Binary classification (BC-Depth) method which uses binary classification on the liveness space. (2) 0/1 Map-Depth method means restricting liveness space by regressing the features to 0/1 Map, where 0 map is for attack and 1 map is for live. (3) LBP-LBP method supervises feature space and translated images with LBP map. (4) Depth-Depth method refers to two depth supervision on feature space and image space. (5) Depth-LBP method uses depth supervision on feature space and LBP supervision on translated images, which is a reverse version of our method.

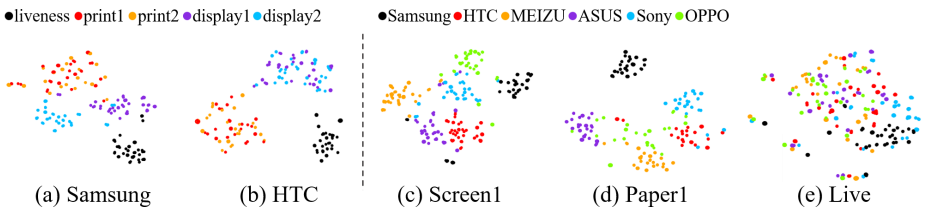


Fig. 6. Distribution of liveness features under two different settings: (a) and (b) display liveness features of different attack and live with the same devices; (c), (d) and (e) are about features of different devices with the same attack or live.

Table 6. The results of score fusion.

Method	LBP Map	Depth Map	Fusion	
			Maximum	Average
APCER	1.25	2.50	2.92	1.67
BPCER	1.67	0.83	0.83	0.83
ACER	1.56	1.67	1.88	1.25

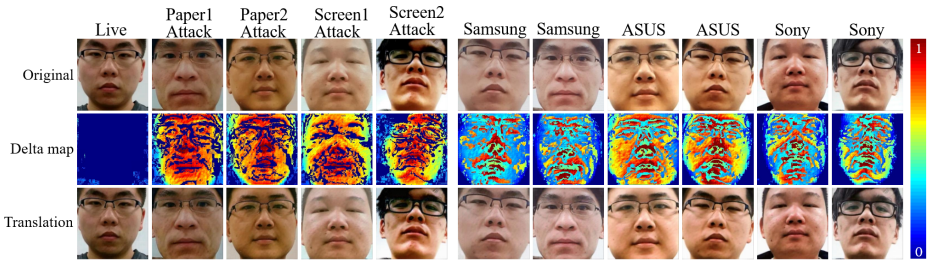
Tab. 5 shows the performance of each method on liveness features and the fusion results with depth network. Compared with different supervison on liveness features, LBP as a low-level texture supervision regularizes the feature space efficiently and performs better. The results of four combinations about LBP and Depth supervision show that the same supervision on feature space and images performs worse than different supervision. And the order of the two supervisions has little effect on the results, but the result of our method is slightly better.

Score Fusion. Using Oulu-NPU protocol 1, we perform studies on the effect of score fusion. Tab. 6 shows the results of each output and the fusion with maximum and average. It shows using LBP map or depth map, the performance is similiar. And the fusion of LBP map and depth map achieves the best performance. Hence, for all experiments, we evaluate the performance by utilizing the fusion score of the LBP map and the depth map, $score = (\|map_{lbp}\| + \|map_{depth}\|)/2$.

5 Further Exploration

We have ruled out the effects of some factors on liveness features in Sec. 4.2. For better understanding the essence of the liveness features, we do some qualitative experiments to explore what factors are related to it.

Spoof Type. We randomly pick up 200 images, which are collected by one certain device. Then we extract the liveness features of images and visualize them by t-SNE [26]. We demonstrate results under Samsumng and HTC mobiles in Fig. 6(a) and (b). Although no additional constraints on attacks are used, there



(a) Attack and Live under HTC (b) Screen1 Attack under three devices

Fig. 7. The delta maps for different attacks with same device and different devices with same attack.

are at least three distinct clusters: live images, paper attack and screen attack in all equipment, which implies liveness features may be related to the spoof type.

Collection Equipment. We randomly pick up 200 images for each type of attack and live with six different devices. Then we visualize the liveness features in Fig. 6(c), (d) and (e). The liveness features from different devices are clustered for attacks, but scattered for live person. It shows that the liveness features of real person may not related to collection equipment. However, the liveness features of attack may include information on collection equipment.

We further display the pixel-wise delta map between generated images and original images of each type, as shown in Fig. 7. The original images, which are shown in the first row, exchange the liveness features with the same one live image to generate the results in the third row. Then we subtract translation images from original images to get delta maps, which are mapped into color space for a better visualization in the second row. From Fig. 7, we may get the following conclusions: (1) When exchanging liveness features between real faces, the delta maps are almost zero. However the delta maps become bigger when between live faces and spoof images. (2) Delta maps of the same type of attack (paper or screen) are similar but are distinguishing between two kinds of attacks. (3) For the same type of attack, delta maps are different under different collection equipment.

6 Conclusions

This paper introduces a new perspective for face anti-spoofing that disentangles the liveness and content features from images. A novel architecture combining the process of disentanglement is proposed with multiple appropriate supervisions. We combine low-level texture and high-level depth characteristics to regularize the liveness space. We visualize the translation process and analyze the content of the liveness features which provides a deeper understanding of face anti-spoofing task. Our method is evaluated on widely-used face anti-spoofing databases and achieves outstanding results.

References

1. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: Application to face recognition. IEEE transactions on pattern analysis and machine intelligence **28**(12), 2037–2041 (2006)
2. Bao, W., Li, H., Li, N., Jiang, W.: A liveness detection method for face recognition based on optical flow field. In: 2009 international conference on image analysis and signal processing. pp. 233–236. IEEE (2009)
3. Bharadwaj, S., Dhamecha, T.I., Vatsa, M., Singh, R.: Computationally efficient face spoofing detection with motion magnification. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops (2013)
4. Biometcs., I.J.S..: information technology biometric presentation attack detection part 1: Framework. (2016)
5. Boulkenafet, Z., Komulainen, J., Akhtar, Z., Benlamoudi, A., Samai, D., Bekhouche, S.E., Ouafi, A., Dornaika, F., Taleb-Ahmed, A., Qin, L., et al.: A competition on generalized software-based face presentation attack detection in mobile scenarios. In: 2017 IEEE international joint conference on biometrics (IJCB). pp. 688–696. IEEE (2017)
6. Boulkenafet, Z., Komulainen, J., Hadid, A.: Face anti-spoofing based on color texture analysis. In: 2015 IEEE international conference on image processing (ICIP). pp. 2636–2640. IEEE (2015)
7. Boulkenafet, Z., Komulainen, J., Hadid, A.: Face antispoofing using speeded-up robust features and fisher vector encoding. IEEE signal processing letters **24**(2), 141–145 (2016)
8. Boulkenafet, Z., Komulainen, J., Hadid, A.: Face spoofing detection using colour texture analysis. IEEE transactions on information forensics and security **11**(8), 1818–1830 (2016)
9. Boulkenafet, Z., Komulainen, J., Li, L., Feng, X., Hadid, A.: Oulu-npu: A mobile face presentation attack database with real-world variations. In: 2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017). pp. 612–618. IEEE (2017)
10. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: interpretable representation learning by information maximizing generative adversarial nets. In: advances in neural information processing systems (2016)
11. Chingovska, I., Anjos, A., Marcel, S.: 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). pp. 1–7. IEEE (2012)
12. 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 **38** (2016)
13. Feng, Y., Wu, F., Shao, X., Wang, Y., Zhou, X.: Joint 3d face reconstruction and dense alignment with position map regression network. In: proceedings of the european conference on computer vision (ECCV). pp. 534–551 (2018)
14. de Freitas Pereira, T., Anjos, A., De Martino, J.M., Marcel, S.: Lbp- top based countermeasure against face spoofing attacks. In: asian conference on computer vision. pp. 121–132. Springer (2012)
15. de Freitas Pereira, T., Anjos, A., De Martino, J.M., Marcel, S.: Can face anti-spoofing countermeasures work in a real world scenario? In: 2013 international conference on biometrics (ICB). pp. 1–8. IEEE (2013)

16. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework. Iclr **2**(5), 6 (2017)
17. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: proceedings of the european conference on computer vision (ECCV). pp. 172–189 (2018)
18. Jourabloo, A., Liu, Y., Liu, X.: Face de-spoofing: Anti-spoofing via noise modeling. In: proceedings of the european conference on computer vision (ECCV) (2018)
19. Kim, T., Kim, Y., Kim, I., Kim, D.: Basn: Enriching feature representation using bipartite auxiliary supervisions for face anti-spoofing. In: proceedings of the IEEE international conference on computer vision Workshops (2019)
20. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
21. Komulainen, J., Hadid, A., Pietikäinen, M.: Context based face anti-spoofing. In: 2013 IEEE sixth international conference on biometrics: theory, applications and systems (BTAS). pp. 1–8. IEEE (2013)
22. Li, J., Wang, Y., Tan, T., Jain, A.K.: Live face detection based on the analysis of fourier spectra. In: biometric technology for human identification. vol. 5404, pp. 296–303. international society for optics and photonics (2004)
23. Li, L., Feng, X., Boulkenafet, Z., Xia, Z., Li, M., Hadid, A.: An original face anti-spoofing approach using partial convolutional neural network. In: 2016 sixth international conference on image processing theory, tools and applications (IPTA). pp. 1–6. IEEE (2016)
24. Liu, Y., Jourabloo, A., Liu, X.: Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 389–398 (2018)
25. Locatello, F., Bauer, S., Lucic, M., Rätsch, G., Gelly, S., Schölkopf, B., Bachem, O.: Challenging common assumptions in the unsupervised learning of disentangled representations. arXiv preprint arXiv:1811.12359 (2018)
26. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. journal of machine learning research **9**(Nov), 2579–2605 (2008)
27. Määttä, J., Hadid, A., Pietikäinen, M.: Face spoofing detection from single images using micro-texture analysis. In: 2011 international joint conference on biometrics (IJCB). pp. 1–7. IEEE (2011)
28. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
29. Patel, K., Han, H., Jain, A.K.: Cross-database face antispoofing with robust feature representation. In: chinese conference on biometric recognition. pp. 611–619. Springer (2016)
30. Patel, K., Han, H., Jain, A.K.: Secure face unlock: Spoof detection on smartphones. IEEE transactions on information forensics and security **11**(10), 2268–2283 (2016)
31. Pinto, A., Pedrini, H., Schwartz, W.R., Rocha, A.: Face spoofing detection through visual codebooks of spectral temporal cubes. IEEE transactions on image processing **24**(12), 4726–4740 (2015)
32. 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. pp. 234–241. Springer (2015)
33. Shao, R., Lan, X., Li, J., Yuen, P.C.: Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In: proceedings of the IEEE conference on computer vision and pattern recognition. pp. 10023–10031 (2019)

34. Siddiqui, T.A., Bharadwaj, S., Dhamecha, T.I., Agarwal, A., Vatsa, M., Singh, R., Ratha, N.: Face anti-spoofing with multifeature videolet aggregation. In: 2016 23rd international conference on pattern recognition (ICPR). IEEE (2016)
35. Vareto, R.H., Diniz, M.A., Schwartz, W.R.: Face spoofing detection on low-power devices using embeddings with spatial and frequency-based descriptors. In: Iberoamerican Congress on Pattern Recognition. pp. 187–197. Springer (2019)
36. 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: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8798–8807 (2018)
37. Wang, Z., Zhao, C., Qin, Y., Zhou, Q., Qi, G., Wan, J., Lei, Z.: Exploiting temporal and depth information for multi-frame face anti-spoofing. arXiv preprint arXiv:1811.05118 (2018)
38. Xiao, T., Hong, J., Ma, J.: Elegant: Exchanging latent encodings with gan for transferring multiple face attributes. In: proceedings of the european conference on computer vision (ECCV). pp. 168–184 (2018)
39. Xu, Z., Li, S., Deng, W.: Learning temporal features using lstm-cnn architecture for face anti-spoofing. In: 2015 3rd IAPR asian conference on pattern recognition (ACPR). pp. 141–145. IEEE (2015)
40. Yang, J., Lei, Z., Li, S.Z.: Learn convolutional neural network for face anti-spoofing. arXiv preprint arXiv:1408.5601 (2014)
41. Yang, J., Lei, Z., Liao, S., Li, S.Z.: Face liveness detection with component dependent descriptor. In: 2013 international conference on biometrics (ICB). pp. 1–6. IEEE (2013)
42. 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: proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3507–3516 (2019)
43. Zhang, Z., Yan, J., Liu, S., Lei, Z., Yi, D., Li, S.Z.: A face antispoofing database with diverse attacks. In: 2012 5th IAPR international conference on biometrics (ICB). pp. 26–31. IEEE (2012)