Domain Generalization via Shuffled Style Assembly for Face Anti-Spoofing

Zhuo Wang[1] Zezheng Wang[2*] Zitong Yu[3] Weihong Deng[1*]
Jiahong Li[2] Tingting Gao[2] Zhongyuan Wang[2]

[1]Beijing University of Posts and Telecommunications [2]Kuaishou Technology [3]CMVS, University of Oulu

{wz2019, whdeng}@bupt.edu.cn zitong.yu@oulu.fi

{wangzezheng, lijiahong, wangzhongyuan}@kuaishou.com tinagao2019@gmail.com

Abstract

*With diverse presentation attacks emerging continually, generalizable face anti-spoofing (FAS) has drawn growing attention. Most existing methods implement domain generalization (DG) on the complete representations. However, different image statistics may have unique properties for the FAS tasks. In this work, we separate the complete representation into content and style ones. A novel **S**huffled **S**tyle **A**ssembly **N**etwork (SSAN) is proposed to extract and reassemble different content and style features for a stylized feature space. Then, to obtain a generalized representation, a contrastive learning strategy is developed to emphasize liveness-related style information while suppress the domain-specific one. Finally, the representations of the correct assemblies are used to distinguish between living and spoofing during the inferring. On the other hand, despite the decent performance, there still exists a gap between academia and industry, due to the difference in data quantity and distribution. Thus, a new large-scale benchmark for FAS is built up to further evaluate the performance of algorithms in reality. Both qualitative and quantitative results on existing and proposed benchmarks demonstrate the effectiveness of our methods. The codes will be available at https://github.com/wangzhuo2019/SSAN.*

1. Introduction

As the most successful computer vision technology, face recognition (FR) [11, 51] has been widely employed in different application scenarios, such as mobile access control and electronic payments. Despite great success, FR systems may still suffer from presentation attacks (PAs), including print attacks, video replay, and 3D masks. To tackle these issues, a series of face anti-spoofing (FAS) methods have been proposed, from hand-craft descriptors based methods [9, 38] to deep representation based ones [52, 55, 57, 59, 61].

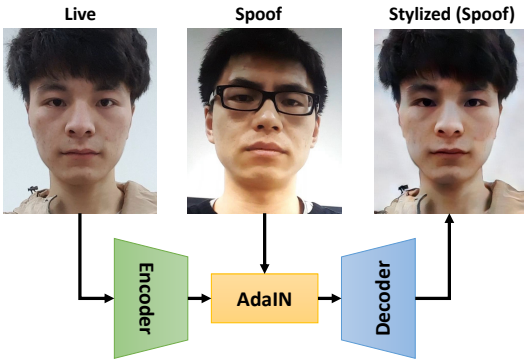

Figure 1. The illustration of style transfer using the method of [18] when live face as content input and spoof face as style input.

The previous FAS methods have achieved promising performance in intra-domain scenarios, but may encounter dramatic degradation under the cross-domain settings. The major reason behind this lies in the conflict between the limitations of training data and the capability of networks [17, 32, 58], which makes the models trapped in dataset bias [43] and leads to poor generalization toward new domains. To address this problem, domain adaptation (DA) techniques [23, 49] are used to alleviate the discrepancy between source and target domains by using unlabeled target data. However, in most real-world FAS scenarios, it is inefficient to collect sufficient unlabeled target data for training.

Thus, domain generalization (DG) methods are proposed to generalize well on the unseen target domain, which can be coarsely classified into three categories: learning a common feature space [21, 41], learning for disentangled representations [48], and learning to learn [40, 42]. These methods almost implement DG on the complete representations from common modules ($i.e.$, CNN-BN-ReLU), but ignore fully taking advantage of subtle properties of global and local image statistics in FAS. Specifically, different normalization approaches lay stress on different statistics information in FAS. For example, Batch Normalization (BN) [19] based structures are usually used to summarize global image statistics, such as semantic features and physical at-

[*] denotes the corresponding author.

tributes. Instance Normalization (IN) [45] based structures focus on the specific sample for distinctive characteristics, such as liveness-related texture and domain-specific external factors. Thus, to mine different statistics in FAS, [30] adopts an adaptive approach to adjust the ratio of IN and BN in feature extraction. Differently, we adopt BN and IN based structures to separate the complete representation into global and local image statistics, denoted as content and style features respectively, then implement specific measures on them for generalizable FAS.

Besides, style transfer [18] can be used to reassemble the pairs of content features as global statistics and style features as local statistics to form stylized features for specific supervision. As shown in Fig. 1, spoofing cues as style input can be applied to live faces to generate the corresponding spoof manipulations. Thus, [35, 54] directly utilize this approach for data augmentation before the training in FAS. However, these two-stage methods are inefficient in large-scale training. Thus, an end-to-end approach is adopted based on style transfer at the feature level in this work.

Combined with the abovementioned viewpoints, we propose a novel framework, called shuffled style assembly network (SSAN), based on style transfer at the feature level. Specifically, a two-stream structure is utilized to extract content and style features, respectively. For content information, they mainly record some global semantic features and physical attributes, thus a shared feature distribution is easily acquired by using adversarial learning. For style information, they preserve some discriminative information that is beneficial to enhance the distinction between living and spoofing. Different from the image-to-image style transfer proposed in [18], we stack up successive shuffled style assembly layers to reassemble various content and style features for a stylized feature space. Then, a contrastive learning strategy is adopted to enhance liveness-related style information and suppress domain-specific one. Lastly, our end-to-end architecture and training approach are more suitable for large-scale training in reality.

Due to the data distribution difference between academic and industrial scenarios, previous evaluation protocols are limited to reflect the genuine performance of algorithms in reality. Thus, to simulate the data quantity and distribution in reality, we combine twelve datasets to build a large-scale evaluation benchmark and further verify the effectiveness of algorithms. Specifically, the TPR@FPR at specific values as the metrics are utilized to evaluate the performance of different models on each dataset, where all live samples as negative cases and partial spoof samples as positive cases.

The main contributions of this work are four-fold:

• To utilize the global and local statistics separately for their unique properties, we propose a novel architecture called shuffled style assembly network (SSAN) for generalizable face anti-spoofing.

• To enhance liveness-relative style information and suppress domain-specific one, we adopt a contrastive learning approach to control the stylized features close or far from the anchor feature. The corresponding loss function is utilized to supervise our network.

• Based on the real-world data distribution, we combine twelve public datasets into a large-scale benchmark for face anti-spoofing in reality. The metric of single-side TPR@FPR is proposed for a comprehensive assessment.

• Our proposed methods achieve the state-of-the-art performance on existing and proposed benchmarks.

2. Related Work

Face Anti-Spoofing. Traditional methods usually extract hand-crafted features such as LBP [9] and SIFT [38] to split living and spoofing. In the era of deep learning, [55] trains CNNs to learn a binary classifier. Auxiliary information such as depth map [2], reflection map [56], and rPPG [25] is utilized to explore additional details for FAS.

To make the algorithm generalize well to unseen scenarios, domain adaptation (DA) and domain generation (DG) techniques are developed. [23] minimizes MMD [15] to pull close between different distributions. [47] leverages adversarial domain adaptation to learn a shared embedding space. [41] utilizes multiple domain discriminators to learn a generalized feature space. [21] forms single-side adversarial learning to further improve the performance. [48, 63] utilize disentangled representation learning to isolate the liveness-related features for classification. To obtain general learning, meta-learning based methods [6, 39, 40, 42, 50] are introduced and developed for regular optimization.

Different from previous DG methods, we split the complete representation into content and style ones with various supervision. Then, a generalized feature space is obtained by resembling features under a contrastive learning strategy.

Normalization and Style Transfer. Normalization layers are essential in deep networks to eliminate covariate shifts and accelerate training. Batch Normalization (BN) [19] utilizes the statistics of the mini-batch to induce universal characteristics. Differently, Instance Normalization (IN) [45] is proposed to exploit stylized characteristics for specific samples. Thus, the former lays stress on the global statistics and the latter emphasizes specific ones. [18] proposes Adaptive Instance Normalization (AdaIN) for style transfer by utilizing target samples to control the scaling and shifting of source image normalized features. This style manipulation is widely used in generative tasks for texture synthesis [37] and style transfer [22]. Observing its effect on texture patterns, our method adopts this module to FAS.

Different from previous methods [30, 35, 54] operating on normalization and image-level transformation, our method adopts AdaIN based layers to assemble different content and style features for a generalized feature space.

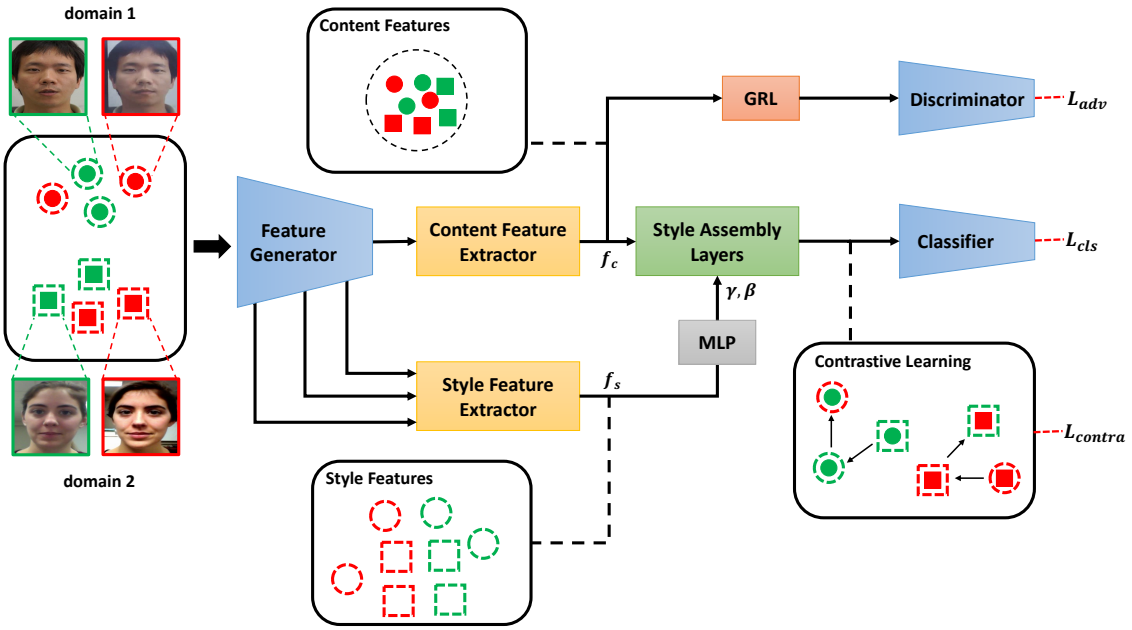

Figure 2. The overall architecture of our shuffled style assembly network (SSAN). Firstly, RGB images from different domains are fed into the feature generator to obtain feature embeddings. Then, the feature extractor with GRL is trained to make the content feature indistinguishable for different domains by using adversarial learning. Meanwhile, another feature extractor collects multi-scale generated features to capture coarse-to-fine style information. Furthermore, to refine the style information related to FAS, a cascade of style assembly layers (SAL) are utilized to reassemble different content and style features when the corresponding contrastive learning strategy is designed.

Protocols for Face Anti-Spoofing. To evaluate the effectiveness of FAS methods, various protocols have been established, including intra-dataset intra-type protocol [4, 32], cross-dataset intra-type protocol [23], intra-dataset cross-type protocol [14, 33], and cross-dataset cross-type protocol [1, 60]. Especially, most protocols are merely constituted of single or double datasets, which may limit their evaluation capabilities for multiply data distributions. Thus, protocol OCIM [41, 42] is used to evaluate their domain-generalization performance across multiple domains.

Moreover, due to the limited amount of data, [8] proposes an open-source framework to aggregate heterogeneous datasets for specific evaluation. Differently, we focus on the real-world data distribution, and more complex domain fields with different data distributions are obtained by fusing twelve different datasets including image and video formats. Thus, the merged dataset contains more sophisticated attack types, such as print, replay, mask, makeup, waxworks, *etc*. Besides, the evaluations under intra- and cross- domain scenarios among multiple datasets have been investigated by using the metric of single-side TPR@FPR, which is more suitable for realistic spectacles.

3. Proposed Approach

In this section, we introduce our shuffled style assembly network (SSAN) shown in Fig. 2. Firstly, we present the two-stream part in our network for content and style information aggregation. Secondly, a shuffled style assembly approach is proposed to recombine various content and style features for a stylized feature space. Then, to suppress domain-specific style information and enhance liveness-related ones, contrastive learning is used in the stylized feature space. Lastly, the overall loss is integrated to optimize the network for stable and reliable training.

3.1. Content and Style Information Aggregation

Content information is usually represented by common factors in FAS, mainly including semantic features and physical attributes. Differently, style information describes some discriminative cues that can be divided into two parts in FAS tasks: domain-specific and liveness-related style information. Thus, content and style features are captured in the two-stream paths separately in our network. Specifically, the feature generator as a shallow embedding network captures multi-scale low-level information. Then, content and style feature extractors collect different image statistics by using specific normalization layers (*i.e.*, BN and IN).

For content information aggregation, we conjecture that small distribution discrepancies exist in different domains, based on the following facts: 1) Considering samples from various domains, they both contain facial areas, thus share a common semantic feature space; 2) Whether bona fide or attack presentation, their physical attributes such as shape and size are often similar. Therefore, we adopt adversarial learning to make generated content features indistinguish-

able for different domains. Specifically, the parameters of the content feature generator are optimized by maximizing the adversarial loss function while the parameters of the domain discriminator are optimized in the opposite direction. Thus, this process can be formulated as follows:

$$\begin{aligned} &\min_{D} \max_{G} L_{adv}(G, D) = \\ &-\mathbb{E}_{(x,y)\sim(X,Y_D)} \sum_{i=1}^{M} \mathbb{1}[i = y] \, logD(G(x)), \end{aligned} \tag{1}$$

where Y_D is the set of domain labels and M is the number of different data domains. G and D represent the content feature generator and domain discriminator, respectively. To optimize G and the D simultaneously, the gradient reversal layer (GRL) [13] is used to reverse the gradient by multiplying it by a negative scalar during the backward propagation.

For style information aggregation, we collect multi-layer features along with the hierarchical structure in a pyramid-like [26] approach, due to the different scales of style characteristics. For example, the brightness of scenes is mainly implicated in broad-scale features, while the texture of presentation materials usually focuses on local-scale regions.

3.2. Shuffled Style Assembly

Adaptive Instance Normalization (AdaIN) [18] is an adaptive style transfer method, which can assemble a content input x and a style input y, as follows:

$$\text{AdaIN}(x, \gamma, \beta) = \gamma \left(\frac{x - \mu(x)}{\sigma(x)} \right) + \beta, \tag{2}$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ represent channel-wise mean and standard deviation respectively, γ and β are affine parameters generated from the style input y.

In this work, to combine content feature f_c and style feature f_s, style assembly layers (SAL) are built up by using AdaIN layers and convolution operators with residual mapping, described as below:

$$\begin{aligned} \gamma, \beta &= \text{MLP}[\text{GAP}(f_s)], \\ z &= \text{ReLU}[\text{AdaIN}(K_1 \otimes f_c, \gamma, \beta)], \\ \text{SAL}(f_c, f_s) &= \text{AdaIN}(K_2 \otimes z, \gamma, \beta) + f_c, \end{aligned} \tag{3}$$

where K_1 and K_2 are 3×3 convolution kernels, $\otimes$ is the convolution operation, and z is the intermediate variable.

However, f_s contains not only liveness-related information, but also domain-specific one that may cause domain bias during network optimization. To alleviate this problem, the shuffled style assembly method is proposed to form auxiliary stylized features for domain generalization.

Given an input sequence of length N in a mini-batch, x_i represents the input sample, where $i \in \{1, 2 \ldots N\}$. Its content feature can be expressed as $f_c(x_i)$ while the style feature as $f_s(x_i)$. Thus, the corresponding assembled feature space $S(x_i, x_i)$ can be formulated as follows:

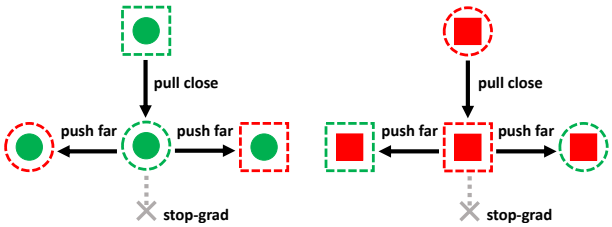

Figure 3. The illustration of the contrastive learning between self-assembly and shuffle-assembly features. Different shapes represent data from different domains: $round$=domain 1, $square$=domain 2. Different colors represent different liveness information: $green$=living, red=spoofing. Lastly, the dotted line represents style information while the interior solid represents content information.

$$S(x_i, x_i) = \text{SAL}(f_c(x_i), f_s(x_i)), \tag{4}$$

which represents the process of style assembly using paired content and style features of input sample x_i. Therefore, $S(x_i, x_i)$ can be denoted as self-assembly features.

Furthermore, to exploit liveness-related style features, we synthesize an auxiliary feature space by shuffling the original pairs of $f_c(x_i)$ and $f_s(x_i)$ randomly, as follows:

$$\begin{aligned} S(x_i, x_{i^*}) &= \text{SAL}(f_c(x_i), f_s(x_{i^*})), \\ i^* &\in random\{1, 2, \ldots, N\}, \end{aligned} \tag{5}$$

where $random$ means a uniformly chosen permutation. $S(x_i, x_{i^*})$ can be denoted as shuffle-assembly features.

3.3. Contrastive Learning for Stylized Features

From the view of style features, a major obstacle is that domain-specific style features may conceal liveness-related ones in cross-domain scenarios, which may cause mistakes in judgment. To solve this problem, we propose a contrastive learning approach to emphasize liveness-related style features as well as suppress domain-specific ones.

After combining content and style features, we obtain self-assembly features $S(x_i, x_i)$ and shuffle-assembly features $S(x_i, x_{i^*})$. For $S(x_i, x_i)$, we input them to the classifier and supervise them using our binary ground-truth signals with the loss function L_{cls}. For $S(x_i, x_{i^*})$, we measure their difference with $S(x_i, x_i)$ by using cosine similarity:

$$Sim(a, b) = -\frac{a}{\|a\|_2} \cdot \frac{b}{\|b\|_2}, \tag{6}$$

where $\|\cdot\|_2$ is l_2-norm, a and b represent two compared features. This is equivalent to the mean squared error of l_2-normalized vectors [16].

As shown in Fig. 3, self-assembly features $S(x_i, x_i)$ are set as anchors in the stylized features space. Inspired by

[5], a stop-gradient (stopgrad) operation is implemented on $S(x_i, x_i)$ to fix their position in the feature space. Then, the shuffle-assembly features $S(x_i, x_{i^*})$ are guided to go close or far toward their corresponding anchors $S(x_i, x_i)$ according to the liveness information. During the process, back-propagation is applied through the shuffle-assembly features but not through self-assembly ones, and the liveness-intensive style information is further aggregated. Thus, the contrastive loss L_{contra} can be formulated as follows:

$$L_{contra} = \sum_{i=1}^{N} Eq\left(x_i, x_{i^*}\right) \cdot Sim\left(\text{stopgrad}(a), b\right), \quad (7)$$

where $a = S(x_i, x_i)$ and $b = S(x_i, x_{i^*})$. $Eq(x_i, x_{i^*})$ measures the consistency of the liveness labels between x_i and x_{i^*}, which can be formulated as follows:

$$Eq\left(x_i, x_{i^*}\right) = \begin{cases} +1, & \text{label}(x_i) == \text{label}(x_{i^*}), \\ -1, & \text{otherwise.} \end{cases} \quad (8)$$

Finally, The whole process of our framework can be described in Algorithm 1 in detail.

Algorithm 1 The optimization strategy of SSAN.

Input: Mixture domain dataset $D_s = \{x_i^s, y_i^s\}_{i=1}^{n_s}$, initialized CNN model $\Phi_0(\cdot)$.

Output: Final CNN model parameter $\Phi_T(\cdot)$.

1: **while** not end of iteration **do**
2: Shuffle the input sequence for the permuted sequence $\{x_{i^*} \mid i^* = random\,[1, 2, \ldots, N]\}$.
3: Input x_i for content feature $f_c(x_i)$ and style feature $f_s(x_i)$. Input x_{i^*} for style feature $f_s(x_{i^*})$.
4: Input $f_c(x_i)$ to the discriminator and compute the adversarial loss L_{adv} based on Eqn. (1).
5: Assemble $f_c(x_i)$ and $f_s(x_i)$ to get self-assembly features $S(x_i, x_i)$. Assemble $f_c(x_i)$ and $f_s(x_{i^*})$ to get shuffle-assembly features $S(x_i, x_{i^*})$.
6: Input $S(x_i, x_i)$ to the classifier and compute the classification loss L_{cls}.
7: Utilize $S(x_i, x_i)$ and $S(x_i, x_{i^*})$ to compute the contrastive loss L_{contra} based on Eqn. (7).
8: Compute $L_{overall} = L_{cls} + \lambda_1 \cdot L_{adv} + \lambda_2 \cdot L_{contra}$. Make gradient back propagation and update the model parameters $\Phi(\cdot)$.
9: **end while**
10: Evaluate $\Phi_T(\cdot)$ on the testing data D_t.

3.4. Loss Function

After describing the operating of our network, we collect the overall loss function $L_{overall}$ for stable and reliable training, which can be formulated as follows:

$$L_{overall} = L_{cls} + \lambda_1 \cdot L_{adv} + \lambda_2 \cdot L_{contra}, \quad (9)$$

where λ_1 and λ_2 are two hyper-parameters to balance the proportion of different loss functions.

Table 1. The datasets and their corresponding numbers we use in the large-scale benchmark.

Dataset	Number	Dataset	Number
CASIA-MFSD [66]	D1	Rose-Youtu [23]	D7
REPLAY-ATTACK [7]	D2	WFFD [20]	D8
MSU-MFSD [53]	D3	CelebA-Spoof [65]	D9
HKBU-MARs V2 [29]	D4	CASIA-SURF [64]	D10
OULU-NPU [4]	D5	WMCA [14]	D11
SiW [32]	D6	CeFA [27]	D12

4. Large-Scale FAS Benchmarks

There exists a gap between academia and industry, which can be summarized as the following two aspects.

Data Quantity. Compared with the authentic scenarios, the amount of data in academia is still too small, which may cause overfitting of the model and limit the development of the algorithm. To overcome this problem, we merge twelve datasets then design corresponding intra- and inter- dataset testing protocols to further evaluate our method.

Data Distribution and Evaluation Metrics. In real-world data distribution, live faces usually account for the majority. However, most existing evaluation protocols collect almost equivalent live and spoof faces as the testing set to calculate their average error rate for evaluation, which may disagree with the reality. Besides, data in reality usually consists of multiple fields with different distributions. Nevertheless, academic datasets usually contain fewer data domains. To reduce the above inconsistencies, multiple datasets are used as training and testing sets simultaneously in our protocols. Specifically, in the training stages, all of the training data are used to optimize our models. In the testing stages, due to the similar distribution of live faces [3, 21], we gather all live data from each testing dataset as the negative cases, then partial spoof data in the current testing dataset is arranged as positive cases. Lastly, the mean and variance of true positive rate (TPR) of false-positive rate (FPR) are computed along with each testing dataset for an overall evaluation.

Twelve datasets are used in the large-scale FAS Benchmarks, which are numbered as shown in Table 1. The evaluation protocols are designed as follows:

• **Protocol 1.** This protocol is implemented in an intra-dataset evaluation scenario. Specifically, all datasets are used as training and testing sets, simultaneously.

• **Protocol 2.** This protocol is implemented in a cross-domain evaluation scenario by dividing these datasets into two piles: $P1$: {D3, D4, D5, D10, D11, D12}, $P2$: {D1, D2, D6, D7, D8, D9}. Thus, there contain two sub-protocols: **protocol 2_1**: training on $P1$ and testing on $P2$; **Protocol 2_2**: training on $P2$ and testing on $P1$. Note that the cross-domain protocols are more challenging as the testing set covers more unseen datasets and more complex unknown attacks, which are correlated to real-world scenarios.

More details are provided in supplementary materials.

Table 2. The results of cross-dataset testing on OULU-NPU, CASIA-MFSD, Replay-Attack, and MSU-MFSD.

Method	O&C&I to M		O&M&I to C		O&C&M to I		I&C&M to O	
	HTER(%)	AUC(%)	HTER(%)	AUC(%)	HTER(%)	AUC(%)	HTER(%)	AUC(%)
MMD-AAE [24]	27.08	83.19	44.59	58.29	31.58	75.18	40.98	63.08
MADDG [41]	17.69	88.06	24.50	84.51	22.19	84.99	27.98	80.02
SSDG-M [21]	16.67	90.47	23.11	85.45	18.21	94.61	25.17	81.83
DR-MD-Net [48]	17.02	90.10	19.68	87.43	20.87	86.72	25.02	81.47
RFMeta [42]	13.89	93.98	20.27	88.16	17.30	90.48	16.45	91.16
NAS-FAS [60]	19.53	88.63	16.54	90.18	14.51	93.84	**13.80**	**93.43**
D^2AM [6]	12.70	95.66	20.98	85.58	15.43	91.22	15.27	90.87
SDA [50]	15.40	91.80	24.50	84.40	15.60	90.10	23.10	84.30
DRDG [31]	12.43	95.81	19.05	88.79	15.56	91.79	15.63	91.75
ANRL [30]	10.83	**96.75**	17.83	89.26	16.03	91.04	15.67	91.90
SSAN-M (Ours)	**10.42**	94.76	**16.47**	**90.81**	**14.00**	**94.58**	19.51	88.17
SSDG-R [21]	7.38	97.17	10.44	95.94	11.71	96.59	15.61	91.54
SSAN-R (Ours)	**6.67**	**98.75**	**10.00**	**96.67**	**8.88**	**96.79**	**13.72**	**93.63**

5. Experiments

5.1. Implementation Details

Data Preparation. The datasets shown in Table 1 contain image and video data. For image data, we utilize all images of them. For video data, we extract frames of them at specific intervals. After obtaining data in image format, we adopt MTCNN [62] for face detection, then crop and resize faces to 256×256 as RGB input. Moreover, a dense face alignment approach ($i.e.$, PRNet [12]) is used to generate the ground-truth depth maps with size 32×32 for genuine faces, while spoof depth maps are set to zeros.

Network Setting. Similar to [21], two structures are established, denoted as SSAN-M and SSAN-R. Specifically, SSAN-M adopts the embedding part of DepthNet [32] while SSAN-R adopts that of ResNet-18 [17] for feature generation. More details are in supplementary materials.

Training Setting. Due to the limit of the GPU memory size, the batch size is set to 16 for SSAN-M and set to 256 for SSAN-R. Different ground-truth are used as supervision signals: depth maps for SSAN-M and binary labels for SSAN-R. Therefore, their corresponding L_{cls} are mean-squared and cross-entropy loss, respectively. λ_1 and λ_2 are set to 1 in training. The Adam optimizer with the learning rate (lr) of 1e-4 and weight decay of 5e-5 is used in the experiments on OCIM. The SGD optimizer with the momentum of 0.9 and weight decay of 5e-4 is used in the experiments on proposed protocols. Its initial lr is 0.01 and decreases by 0.2 times every two epochs until the 30^{th} epoch.

Testing Setting. In testing, we calculate the final classification score to separate bona fide and attack presentations. Specifically, the mean value of the predicted depth map is the final score for SSAN-M, while the value of the sigmoid function on living is the final score for SSAN-R.

5.2. Experiment on OCIM

Four datasets are used to evaluate the performance of SSAN in different cross-domain scenarios following the implementation of [41]: OULU-NPU [4] (O), CASIA-MFSD [66] (C), Replay-Attack [7] (I), and MSU-MFSD [53] (M).

Experiment in Leave-One-Out (LOO) Setting. For an overall evaluation, we conduct cross-dataset testing by using the LOO strategy: three datasets are selected for training, and the rest one for testing. We compare our models with the recent SOTA methods, as shown in Table 2. It can be observed that our SSAN-M shows the best performance on protocols of O&C&I to M, O&M&I to C, O&C&M to I, and the competitive performance on the protocol of I&C&M to O. These results demonstrate the domain generalization capacity of our method. Moreover, when we adopt the ResNet18-based network denoted as SSAN-R, its performance obtains an excellent improvement and exceeds the model SSDG-R proposed in [21] with similar settings. The above phenomenon indicates our network SSAN-R is more effective in the cross-dataset scenario, thus will be further measured in the large-scale protocols we propose.

Table 3. Comparison results on limited source domains.

Method	M&I to C		M&I to O	
	HTER(%)	AUC(%)	HTER(%)	AUC(%)
MS-LBP [34]	51.16	52.09	43.63	58.07
IDA [53]	45.16	58.80	54.52	42.17
LBP-TOP [10]	45.27	54.88	47.26	50.21
MADDG [41]	41.02	64.33	39.35	65.10
SSDG-M [21]	31.89	71.29	36.01	66.88
DR-MD-Net [48]	31.67	75.23	34.02	72.65
ANRL [30]	31.06	72.12	30.73	74.10
SSAN-M (Ours)	**30.00**	**76.20**	**29.44**	**76.62**

Experiment on Limited Source Domains. We also evaluate our method when extremely limited source domains are available. Specifically, MSU-MFSD and Replay-Attack are selected as the source domains for training and the remaining two ($i.e.$, CASIA-MFSD and OULU-NPU) will be used as the target domains for testing respectively. As shown in Table 3, our method achieves the lowest HTER and the highest AUC despite limited source data, which proves the modeling efficiency and generalization capability of our network in a challenging task.

5.3. Experiment on Proposed Benchmarks

To further evaluate the performance of our method in reality, we conduct the experiments on the large-scale FAS

benchmark we proposed, as shown in Table 4. Different network structures ($i.e.$, CNN [17] and Transformer [44]) and some recent SOTA methods ($i.e.$, CDCN [61] and SSDG [21]) are also conducted in their default settings for comparison. From the evaluation results, we can observe that our method achieves the best performance, exceeding that of other compared methods, which proves the effectiveness of our SSAN in real-world data distribution. It is worth noting that some methods have achieved excellent performance on existing protocols, but may suffer an acute degeneration in the large-scale benchmarks. This phenomenon further reveals the mismatch between academia and industry in FAS. More detailed analyses are in supplementary materials.

Table 4. The results on the large-scale FAS benchmarks.

Prot.	Method	TPR@FPR(%)		
		10%	1%	0.1%
1	ResNet18 [17]	96.04±11.96	89.32±26.08	69.10±34.34
	Deit-T [44]	97.75±5.70	90.38±16.08	73.42±30.00
	CDCN [61]	92.59±15.99	84.40±31.93	71.54±32.05
	SSDG-R [21]	96.48±10.37	89.13±25.59	68.12±39.12
	SSAN-R (Ours)	**98.31±4.19**	**90.51±22.31**	**78.45±31.98**
2_1	ResNet18 [17]	55.64±22.05	17.53±13.44	3.64±3.93
	Deit-T [44]	44.03±17.77	10.15±6.08	1.25±1.04
	CDCN [61]	55.92±21.45	11.07±8.21	0.69±0.74
	SSDG-R [21]	53.44±19.23	3.27±3.09	0.06±0.06
	SSAN-R (Ours)	**63.61±21.69**	**25.56±18.07**	**6.58±5.56**
2_2	ResNet18 [17]	63.38±27.54	41.53±30.41	19.00±14.79
	Deit-T [44]	63.29±13.39	30.46±19.15	11.30±9.45
	CDCN [61]	20.97±25.23	3.58±4.83	0.58±0.88
	SSDG-R [21]	41.13±28.45	7.19±8.73	1.94±2.35
	SSAN-R (Ours)	**64.54±28.36**	**47.07±33.71**	**31.61±23.33**

5.4. Ablation Study

To verify the superiority of our SSAN as well as the contributions of each component, multiple incomplete models are built up by controlling different variables. All results are measured in the same manner, as shown in Table 5.

Effectiveness of Different Components. To verify the effectiveness of generalized content feature space, we conduct the experiments of SSAN w/o L_{adv}. Specifically, content features usually record some common patterns in FAS, which is easier to reduce their domain difference, compared to directly operating on the complete features. Besides, to make assembly between arbitrary combinations of content and style features for domain generalization, stripping domain distinction from content information is indispensable.

On the other hand, to prove the importance of contrastive learning for shuffled stylized features, the experiments of SSAN w/o L_{contra} are implemented for comparison. The quantitative results indicate that the style assembly guided by liveness-intensive cues is beneficial to improve the performance for cross-domain FAS tasks.

Impact of the Stop-Gradient Operation. In contrastive learning for stylized features, the self-assembly features adopt the approach of stop-gradient to fix their position in the feature space as an anchor. Then, their corresponding shuffle-assembly features obeying on the liveness information to go close or far toward them. The ablation experiment of SSAN w/o stop-grad shows its effectiveness of feature aggregation in contrastive learning for emphasizing liveness-related style information and suppressing domain-specific ones. Besides, from the continuous evaluation curves shown in Fig. 4, it can be summarized that the stop-gradient operation will contribute to stable training.

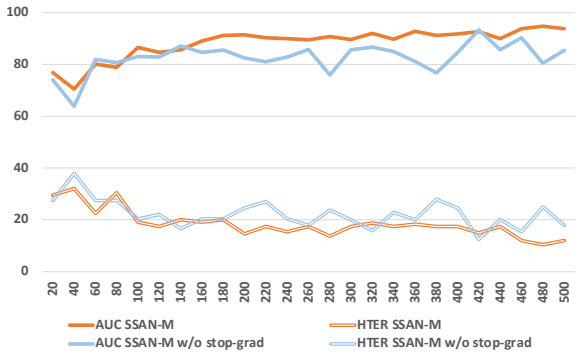

Figure 4. The comparison curves between SSAN-M and SSAN-M w/o stop-grad under protocol O&C&I to M. The x-axis represents the number of epochs while the y-axis records the value of AUC(%) the HTER(%), as shown in the legend.

Comparison Between the Hard and Soft Supervision. The relative movement approach in contrastive learning we adopt can be regarded as soft supervision in stylized feature space, compared to the direct supervision using the ground-truth. To investigate their different efficiency, we conduct the experiment of w/ hard-sup for an ablation study between them, as shown in Table 5. The declining performance shows the soft supervision method is more suitable for our networks under the cross-domain testing scenarios.

Analysis of Contrastive Learning. Existing works [28, 36] implement classical supervised contrastive learning (SCL) on the complete representation in FAS. Differently, our method conducts contrastive learning between self-assembly and shuffle-assembly features. To make a comparison between them, the experiment of w/ SCL is conducted by implementing contrastive learning on self-assembly directly. The final results demonstrate the efficiency of the auxiliary features in contrastive learning, which are built in a shuffle-then-assembly approach.

5.5. Visualization and Analysis

Features Visualization. To analyze the feature space learned by our SSAN method, we visualize the distribution of different features using t-SNE [46], as shown in Fig. 5. For content features, it can be observed that their distribution is more compact and mixed, though they may belong to multiple databases or various liveness attributions. For style features, there exists a coarse boundary between living and spoofing along with a narrow distribution, despite no direct supervision on them. This phenomenon indicates that our

Table 5. Evaluations of different components of the proposed method with different architectures.

Method	O&C&I to M		O&M&I to C		O&C&M to I		I&C&M to O	
	HTER(%)	AUC(%)	HTER(%)	AUC(%)	HTER(%)	AUC(%)	HTER(%)	AUC(%)
SSAN-M w/o L_{adv}	10.42	**94.83**	24.44	81.60	24.75	83.01	27.11	80.41
SSAN-M w/o L_{contra}	12.50	93.59	17.59	89.33	14.75	92.67	22.47	85.79
SSAN-M w/o stop-grad	12.50	93.33	20.93	85.02	16.38	89.78	23.65	83.14
SSAN-M w/ hard-sup	12.08	93.42	28.89	77.70	20.61	86.46	24.83	82.39
SSAN-M w/ SCL	12.92	92.50	23.70	84.67	18.75	87.28	25.45	82.03
SSAN-M (Ours)	**10.42**	94.76	**16.47**	**90.81**	**14.00**	**94.58**	**19.51**	**88.17**
SSAN-R w/o L_{adv}	10.83	94.08	14.26	94.48	12.25	94.93	14.27	92.83
SSAN-R w/o L_{contra}	12.08	95.62	12.59	94.97	10.75	95.01	15.31	92.31
SSAN-R w/o stop-grad	11.25	93.46	11.30	95.11	9.00	96.03	14.06	93.14
SSAN-R w/ hard-sup	11.67	96.04	14.63	94.65	11.38	94.61	15.21	92.97
SSAN-R w/ SCL	11.25	94.00	12.04	94.91	12.50	95.34	15.80	92.95
SSAN-R (Ours)	**6.67**	**98.75**	**10.00**	**96.67**	**8.88**	**96.79**	**13.72**	**93.63**

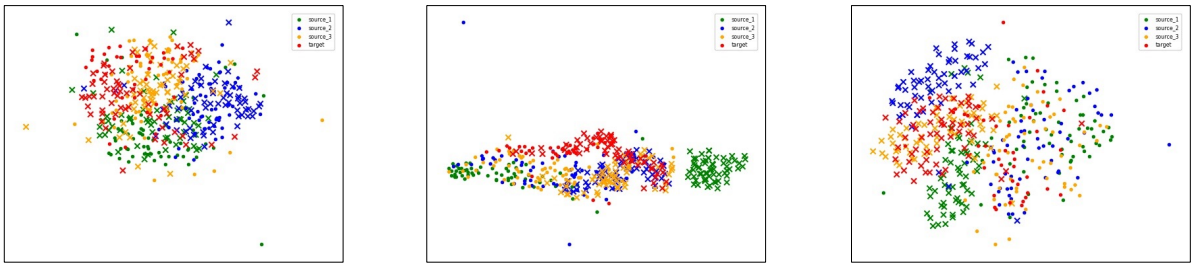

Figure 5. The t-SNE [46] visualization of different features under protocol O&C&I to M. The graphs of (a), (b), and (c) describe the feature distribution of content features, style features, and stylized features, respectively. Different colors indicate features from different domains: $green$=O, $blue$=C, $yellow$=I, red=M. Different shapes represent different liveness information: $point$=living, $cross$=spoofing.

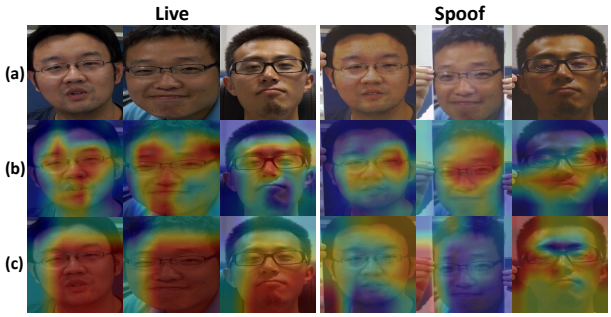

Figure 6. Grad-CAM [67] visualizations of activation areas under protocol O&M&I to C. (a): Original images. (b): Visualizations for content features generation. (c): Visualizations for assembled features (content + style) generation.

contrastive learning for stylized features is effective to emphasize liveness-related style features as well as suppress other irrelevant ones, such as domain-specific information. For stylized features, we combine the content and style information for the classification between living and spoofing. The visualization results show that even though encountering an unknown distribution, our method still can generalize well to the target domain.

Attention Visualization. To find the regions that led to content feature extraction and liveness detection, we adopt the Grad-CAM [67] to describe their activation maps upon the original images, as shown in Fig. 6. It can be observed that despite living and spoofing, their content features both mainly focus on the landmark areas in faces that contain abundant semantic features and physical attributes. Then, after combined with the style information, the stylized features for classification show different activation properties: (1) For the live faces, our model lays the stress on the face regions to seek cues for judgment; (2) For the spoofing faces, some spoofing cues will be concentrated by our method, such as the moire phenomenon in replay attacks and the photo cut position in print attacks.

6. Conclusion

In this paper, we have proposed a novel shuffled style assembly network (SSAN) for generalizable face anti-spoofing (FAS). Different from the previous methods implemented on the complete features, we operate on content and style features separately due to their various properties. For content features, adversarial learning is adopted to make them domain-indistinguishable. For style features, a contrastive learning strategy is used to emphasize liveness-related style information while suppress domain-specific one. Then, the correct pairs of content and style features are reassembled for classification. Moreover, to bridge the gap between academia and industry, a large-scale benchmark for FAS is built up by aggregating existing datasets. Experimental results on existing and proposed benchmarks have demonstrated the superiority of our methods.

Acknowledgements This work was partially supported by the National Natural Science Foundation of China under Grants No. 61871052 and 62192784.

References

[1] Shervin Rahimzadeh Arashloo, Josef Kittler, and William Christmas. An anomaly detection approach to face spoofing detection: A new formulation and evaluation protocol. *IEEE access*, 5:13868–13882, 2017. 3

[2] Yousef Atoum, Yaojie Liu, Amin Jourabloo, and Xiaoming Liu. Face anti-spoofing using patch and depth-based cnns. In *IJCB*, pages 319–328, 2017. 2

[3] Davide Belli, Debasmit Das, Bence Major, and Fatih Porikli. A personalized benchmark for face anti-spoofing. In *WACV*, pages 338–348, 2022. 5

[4] Zinelabinde Boulkenafet, Jukka Komulainen, Lei Li, Xiaoyi Feng, and Abdenour Hadid. Oulu-npu: A mobile face presentation attack database with real-world variations. In *FG*, pages 612–618, 2017. 3, 5, 6

[5] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, pages 15750–15758, 2021. 5

[6] Zhihong Chen, Taiping Yao, Kekai Sheng, Shouhong Ding, Ying Tai, Jilin Li, Feiyue Huang, and Xinyu Jin. Generalizable representation learning for mixture domain face anti-spoofing. In *AAAI*, pages 1132–1139, 2021. 2, 6

[7] Ivana Chingovska, André Anjos, and Sébastien Marcel. On the effectiveness of local binary patterns in face anti-spoofing. In *BIOSIG*, pages 1–7, 2012. 5, 6

[8] Artur Costa-Pazo, David Jiménez-Cabello, Esteban Vázquez-Fernández, José Luis Alba-Castro, and Roberto J López-Sastre. Generalized presentation attack detection: a face anti-spoofing evaluation proposal. In *ICB*, pages 1–8, 2019. 3

[9] Tiago de Freitas Pereira, André Anjos, José Mario De Martino, and Sébastien Marcel. Lbp- top based countermeasure against face spoofing attacks. In *ACCV*, pages 121–132, 2012. 1, 2

[10] Tiago de Freitas Pereira, Jukka Komulainen, André Anjos, José Mario De Martino, Abdenour Hadid, Matti Pietikäinen, and Sébastien Marcel. Face liveness detection using dynamic texture. *EURASIP Journal on Image and Video Processing*, 2014(1):1–15, 2014. 6

[11] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, pages 4690–4699, 2019. 1

[12] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *ECCV*, pages 534–551, 2018. 6

[13] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, pages 1180–1189, 2015. 4

[14] Anjith George, Zohreh Mostaani, David Geissenbuhler, Olegs Nikisins, André Anjos, and Sébastien Marcel. Biometric face presentation attack detection with multi-channel convolutional neural network. *IEEE TIFS*, 15:42–55, 2019. 3, 5

[15] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012. 2

[16] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, 33:21271–21284, 2020. 4

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 6, 7

[18] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, pages 1501–1510, 2017. 1, 2, 4

[19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015. 1, 2

[20] Shan Jia, Xin Li, Chuanbo Hu, Guodong Guo, and Zhengquan Xu. 3d face anti-spoofing with factorized bilinear coding. *IEEE TCSVT*, 31(10):4031–4045, 2020. 5

[21] Yunpei Jia, Jie Zhang, Shiguang Shan, and Xilin Chen. Single-side domain generalization for face anti-spoofing. In *CVPR*, pages 8484–8493, 2020. 1, 2, 5, 6, 7

[22] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 2

[23] Haoliang Li, Wen Li, Hong Cao, Shiqi Wang, Feiyue Huang, and Alex C Kot. Unsupervised domain adaptation for face anti-spoofing. *IEEE TIFS*, 13(7):1794–1809, 2018. 1, 2, 3, 5

[24] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *CVPR*, pages 5400–5409, 2018. 6

[25] Bofan Lin, Xiaobai Li, Zitong Yu, and Guoying Zhao. Face liveness detection by rppg features and contextual patch-based cnn. In *ACM ICBEA*, pages 61–68, 2019. 2

[26] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 4

[27] Ajian Liu, Zichang Tan, Jun Wan, Sergio Escalera, Guodong Guo, and Stan Z Li. Casia-surf cefa: A benchmark for multi-modal cross-ethnicity face anti-spoofing. In *WACV*, pages 1179–1187, 2021. 5

[28] Ajian Liu, Chenxu Zhao, Zitong Yu, Jun Wan, Anyang Su, Xing Liu, Zichang Tan, Sergio Escalera, Junliang Xing, Yanyan Liang, et al. Contrastive context-aware learning for 3d high-fidelity mask face presentation attack detection. *arXiv preprint arXiv:2104.06148*, 2021. 7

[29] Siqi Liu, Pong C Yuen, Shengping Zhang, and Guoying Zhao. 3d mask face anti-spoofing with remote photoplethysmography. In *ECCV*, pages 85–100, 2016. 5

[30] Shubao Liu, Ke-Yue Zhang, Taiping Yao, Mingwei Bi, Shouhong Ding, Jilin Li, Feiyue Huang, and Lizhuang Ma. Adaptive normalized representation learning for generalizable face anti-spoofing. In *ACM MM*, pages 1469–1477, 2021. 2, 6

[31] Shubao Liu, Ke-Yue Zhang, Taiping Yao, Kekai Sheng, Shouhong Ding, Ying Tai, Jilin Li, Yuan Xie, and Lizhuang

Ma. Dual reweighting domain generalization for face presentation attack detection. In *IJCAI*, pages 867–873, 2021. 6

[32] Yaojie Liu, Amin Jourabloo, and Xiaoming Liu. Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In *CVPR*, pages 389–398, 2018. 1, 3, 5, 6

[33] Yaojie Liu, Joel Stehouwer, Amin Jourabloo, and Xiaoming Liu. Deep tree learning for zero-shot face anti-spoofing. In *CVPR*, pages 4680–4689, 2019. 3

[34] Jukka Määttä, Abdenour Hadid, and Matti Pietikäinen. Face spoofing detection from single images using micro-texture analysis. In *IJCB*, pages 1–7, 2011. 6

[35] Luciana T Menon, Alessandro L Koerich, Alceu S Britto Jr, et al. Style transfer applied to face liveness detection with user-centered models. *arXiv preprint arXiv:1907.07270*, 2019. 2

[36] Shlok Kumar Mishra, Kuntal Sengupta, Max Horowitz-Gelb, Wen-Sheng Chu, Sofien Bouaziz, and David Jacobs. Improved detection of face presentation attacks using image decomposition. *arXiv preprint arXiv:2103.12201*, 2021. 7

[37] Oren Nuriel, Sagie Benaim, and Lior Wolf. Permuted adain: Reducing the bias towards global statistics in image classification. In *CVPR*, pages 9482–9491, 2021. 2

[38] Keyurkumar Patel, Hu Han, and Anil K Jain. Secure face unlock: Spoof detection on smartphones. *IEEE TIFS*, 11(10):2268–2283, 2016. 1, 2

[39] Yunxiao Qin, Zitong Yu, Longbin Yan, Zezheng Wang, Chenxu Zhao, and Zhen Lei. Meta-teacher for face anti-spoofing. *IEEE TPAMI*, 2021. 2

[40] Yunxiao Qin, Chenxu Zhao, Xiangyu Zhu, Zezheng Wang, Zitong Yu, Tianyu Fu, Feng Zhou, Jingping Shi, and Zhen Lei. Learning meta model for zero-and few-shot face anti-spoofing. In *AAAI*, pages 11916–11923, 2020. 1, 2

[41] Rui Shao, Xiangyuan Lan, Jiawei Li, and Pong C Yuen. Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In *CVPR*, pages 10023–10031, 2019. 1, 2, 3, 6

[42] Rui Shao, Xiangyuan Lan, and Pong C Yuen. Regularized fine-grained meta face anti-spoofing. In *AAAI*, pages 11974–11981, 2020. 1, 2, 3, 6

[43] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR*, pages 1521–1528, 2011. 1

[44] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357, 2021. 7

[45] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 2

[46] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 7, 8

[47] Guoqing Wang, Hu Han, Shiguang Shan, and Xilin Chen. Improving cross-database face presentation attack detection via adversarial domain adaptation. In *ICB*, pages 1–8, 2019. 2

[48] Guoqing Wang, Hu Han, Shiguang Shan, and Xilin Chen. Cross-domain face presentation attack detection via multi-domain disentangled representation learning. In *CVPR*, pages 6678–6687, 2020. 1, 2, 6

[49] Guoqing Wang, Hu Han, Shiguang Shan, and Xilin Chen. Unsupervised adversarial domain adaptation for cross-domain face presentation attack detection. *IEEE TIFS*, 16:56–69, 2020. 1

[50] Jingjing Wang, Jingyi Zhang, Ying Bian, Youyi Cai, Chunmao Wang, and Shiliang Pu. Self-domain adaptation for face anti-spoofing. In *AAAI*, pages 2746–2754, 2021. 2, 6

[51] Mei Wang and Weihong Deng. Deep face recognition: A survey. *Neurocomputing*, 429:215–244, 2021. 1

[52] Zhuo Wang, Qiangchang Wang, Weihong Deng, and Guodong Guo. Learning multi-granularity temporal characteristics for face anti-spoofing. *IEEE TIFS*, 17:1254–1269, 2022. 1

[53] Di Wen, Hu Han, and Anil K Jain. Face spoof detection with image distortion analysis. *IEEE TIFS*, 10(4):746–761, 2015. 5, 6

[54] Bowen Yang, Jing Zhang, Zhenfei Yin, and Jing Shao. Few-shot domain expansion for face anti-spoofing. *arXiv preprint arXiv:2106.14162*, 2021. 2

[55] Jianwei Yang, Zhen Lei, and Stan Z Li. Learn convolutional neural network for face anti-spoofing. *arXiv preprint arXiv:1408.5601*, 2014. 1, 2

[56] Zitong Yu, Xiaobai Li, Xuesong Niu, Jingang Shi, and Guoying Zhao. Face anti-spoofing with human material perception. In *ECCV*, pages 557–575, 2020. 2

[57] Zitong Yu, Yunxiao Qin, Xiaobai Li, Chenxu Zhao, Zhen Lei, and Guoying Zhao. Deep learning for face anti-spoofing: A survey. *arXiv preprint arXiv:2106.14948*, 2021. 1

[58] Zitong Yu, Yunxiao Qin, Xiaqing Xu, Chenxu Zhao, Zezheng Wang, Zhen Lei, and Guoying Zhao. Auto-fas: Searching lightweight networks for face anti-spoofing. In *ICASSP*, pages 996–1000, 2020. 1

[59] Zitong Yu, Yunxiao Qin, Hengshuang Zhao, Xiaobai Li, and Guoying Zhao. Dual-cross central difference network for face anti-spoofing. In *IJCAI*, pages 1281–1287, 2021. 1

[60] Zitong Yu, Jun Wan, Yunxiao Qin, Xiaobai Li, Stan Z Li, and Guoying Zhao. Nas-fas: Static-dynamic central difference network search for face anti-spoofing. *IEEE TPAMI*, 43(9):3005–3023, 2020. 3, 6

[61] Zitong Yu, Chenxu Zhao, Zezheng Wang, Yunxiao Qin, Zhuo Su, Xiaobai Li, Feng Zhou, and Guoying Zhao. Searching central difference convolutional networks for face anti-spoofing. In *CVPR*, pages 5295–5305, 2020. 1, 7

[62] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE SPL*, 23(10):1499–1503, 2016. 6

[63] Ke-Yue Zhang, Taiping Yao, Jian Zhang, Ying Tai, Shouhong Ding, Jilin Li, Feiyue Huang, Haichuan Song, and Lizhuang Ma. Face anti-spoofing via disentangled representation learning. In *ECCV*, pages 641–657, 2020. 2

[64] Shifeng Zhang, Xiaobo Wang, Ajian Liu, Chenxu Zhao, Jun Wan, Sergio Escalera, Hailin Shi, Zezheng Wang, and Stan Z Li. A dataset and benchmark for large-scale multi-modal face anti-spoofing. In *CVPR*, pages 919–928, 2019. 5

[65] Yuanhan Zhang, ZhenFei Yin, Yidong Li, Guojun Yin, Junjie Yan, Jing Shao, and Ziwei Liu. Celeba-spoof: Large-scale face anti-spoofing dataset with rich annotations. In *ECCV*, pages 70–85, 2020. 5

[66] Zhiwei Zhang, Junjie Yan, Sifei Liu, Zhen Lei, Dong Yi, and Stan Z Li. A face antispoofing database with diverse attacks. In *ICB*, pages 26–31, 2012. 5, 6

[67] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016. 8