PatchNet: A Simple Face Anti-Spoofing Framework via Fine-Grained Patch Recognition

Chien-Yi Wang[1] Yu-Ding Lu[2] Shang-Ta Yang[1] Shang-Hong Lai[1]
[1]Microsoft AI R&D Center, Taiwan [2]HTC
{chiwa, shanya, shlai}@microsoft.com jonlu.citi@gmail.com

Abstract

Face anti-spoofing (FAS) plays a critical role in securing face recognition systems from different presentation attacks. Previous works leverage auxiliary pixel-level supervision and domain generalization approaches to address unseen spoof types. However, the local characteristics of image captures, i.e., capturing devices and presenting materials, are ignored in existing works and we argue that such information is required for networks to discriminate between live and spoof images. In this work, we propose PatchNet which reformulates face anti-spoofing as a fine-grained patch-type recognition problem. To be specific, our framework recognizes the combination of capturing devices and presenting materials based on the patches cropped from non-distorted face images. This reformulation can largely improve the data variation and enforce the network to learn discriminative feature from local capture patterns. In addition, to further improve the generalization ability of the spoof feature, we propose the novel Asymmetric Margin-based Classification Loss and Self-supervised Similarity Loss to regularize the patch embedding space. Our experimental results verify our assumption and show that the model is capable of recognizing unseen spoof types robustly by only looking at local regions. Moreover, the fine-grained and patch-level reformulation of FAS outperforms the existing approaches on intra-dataset, cross-dataset, and domain generalization benchmarks. Furthermore, our PatchNet framework can enable practical applications like Few-Shot Reference-based FAS and facilitate future exploration of spoof-related intrinsic cues.

1. Introduction

Face anti-spoofing (FAS) is a crucial technique to prevent face recognition systems from security attacks. With the advance of deep neural network, several learning-based approaches were proposed to discriminate live faces from physical presentation attacks.

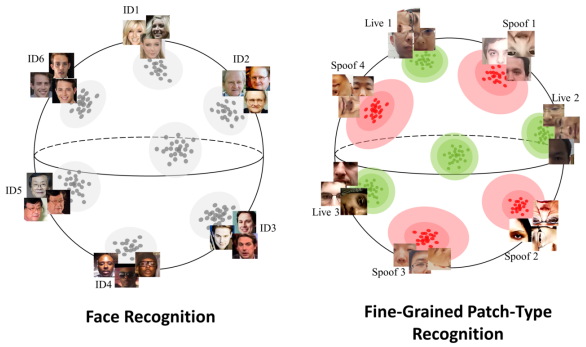


Figure 1. The face recognition model learns a face embedding space to discriminate between identities. Our **fine-grained patch-type recognition model** learns a patch embedding space to discriminate between patches with different capture characteristics.

Previous face anti-spoofing methods are highly limited by the scale and variation of the datasets. Commonly used datasets [1, 5, 18, 27, 34] contain less than 100 identities during training, and the spoof images are captured under limited variation. Based on our observation, training on such datasets with a binary classification model is prone to overfit to the biases introduced by the data collection, and the learned features are vulnerable in the unseen testing scenarios. Therefore, previous face anti-spoofing works [14, 15, 18, 21, 29, 30] leverage auxiliary pixel-wise supervision (e.g., the facial depth map and reflection map) as a strong prior knowledge to achieve better generalization ability under testing scenarios with unseen illumination or spoof types. The other FAS works [19, 33] propose to adopt Generative Adversarial Network (GAN) to disentangle the feature maps of live faces and spoof images by reconstructing new live and spoof facial images. Despite the effectiveness of these spoof-detecting techniques, it is still remained as an open question to describe the intrinsic cues learned from networks. Yu *et al*. [29] rephrase FAS as a structural material recognition problem, which assumes that the discrimination of the structural materials between human facial skin and physical spoofing carriers is the essence for FAS tasks. Following the similar motivation, we believe that the capability of recognizing and comparing different

fine-grained material types is the key to learn robust intrinsic cues for FAS.

In this paper, we propose PatchNet which learns discriminative features based on patches cropped from the entire face regions. Inspired by previous works [2, 6, 28], the patch-level inputs can enhance the data variation and enforce the network to learn spoof-specific features in the local region, and thus prevent the network from overfitting to the biases introduced by datasets. Instead of resizing the input face images into the same size as adopted by recent FAS works, we directly crop the fixed-size patches from raw facial captures to avoid the distortion of discriminative FAS cues. With the patch-level inputs, our PatchNet aims at classifying the corresponding fine-grained categories, i.e., the capturing devices and presenting materials, and we denote each category as a specific "patch-type". To enforce the network to learn robust spoof-related feature to recognize unseen patch types during testing, we adopt the angular margin-based softmax loss that is commonly used in face recognition tasks [8,23,25], which aims to optimize the face embedding on the normalized hypersphere (Figure 1). Moreover, since the patch type classes are not symmetric between live and spoof faces, we propose "asymmetric angular margin loss" and impose a larger margin on live type classes. Inspired by the recent works on self-supervised learning [3,4,10], and the fact that material patterns are presented spatially in the entire face region, we also propose "self-supervised similarity loss" to regularize the features with location and rotation invariance.

To demonstrate the effectiveness of PatchNet, we conduct extensive experiments on intra-dataset, cross-dataset, and domain generalization benchmark datasets, and PatchNet achieves the state-of-the-art performance under most testing scenarios. Moreover, we also conduct the ablation study to further investigate the proposed components.

Our contributions are summarized as follows:

- We reformulate face anti-spoofing as a fine-grained patch recognition problem, and design a simple framework called PatchNet to learn an embedding space to encode intrinsic cues from local patches to represent captures' characteristics.
- We propose novel Asymmetric Margin-based Softmax Loss and Self-supervised Similarity Loss to supervise the PatchNet training. While the former helps to learn a more generalized patch type embedding space to address the asymmetry between live and spoof, the latter can enforce the patch feature to be invariant within a single capture.
- The proposed framework could achieve state-of-the-art performance on intra-dataset, cross-dataset, and domain generalization benchmarks simultaneously without auxiliary pixel-wise supervision and domain generalization techniques. Moreover, the learned patch embedding space can enable applications like Few-Shot reference FAS and patch type retrieval, which can boost the FAS performance in certain deployment scenarios.

2. Related Works

Auxiliary-based Methods. Most of the recent works leverage auxiliary tasks as the prior knowledge to guide the feature learning toward more generalizable cues. Liu *et al*. [18] proposed to employ the depth map and rPPG as strong supervision signals for live samples to regularize the features. Kim *et al*. [13] further leveraged the reflection maps as the supervision signal for spoof samples. Many other FAS methods [9,14,15,20,21,29,30] also heavily rely on similar auxiliary pixel-wise supervision to improve their FAS model performance. Even though the feature learning can benefit from such supervision, the pseudo ground truths for those tasks are not accurate, and the generation of those supervision signals takes high computation resources.

Domain Generalization FAS Methods. In the face anti-spoofing community, domain generalization techniques are developed to address the domain shift between different anti-spoofing datasets. Shao *et al*. [21] employed meta-learning techniques to simulate the target domain shift during the training process to regularize the feature learning directions. Wang *et al*. [24] proposed to learn domain-independent features via a disentangled representation learning framework. The most related work to ours is [12], which treats live and spoof samples asymmetrically and applies adversarial loss and triplet loss to regularize the features in the normalized space. Actually, domains are hard to define in FAS tasks, as even within the same dataset, there are captures with very different capture devices. While people are using generalization methods to find common features across collections and spoof types, we aim to break the concept of domain and propose to learn a generic embedding space that encodes capture characteristics explicitly.

3. Proposed Method

3.1. Overview

As illustrated in Fig. 2, we reformulate face anti-spoofing as a fine-grained patch-type recognition problem and propose a simple training framework to learn the patch features efficiently. First, we apply certain transform on the original image to obtain the patch inputs, and the patch features are extracted by an encoder and then normalized in the feature space. Based on the meta-info from the training dataset, we split the categories finely based on the **presenting materials** and **capture devices**. For example, in

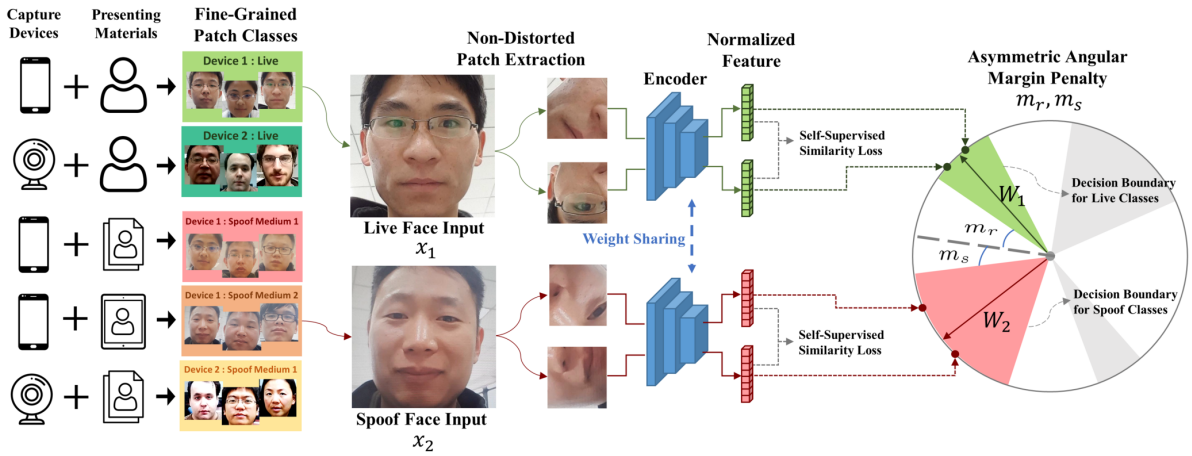


Figure 2. **Overview of our proposed PatchNet framework**. We address the face anti-spoofing with a fine-grained patch-type recognition model. The patch-type classes are pre-defined by the combination of the capture device and the presenting material, and the patch inputs are extracted from the face captures by non-distorted augmentation operations. **Asymmetric Angular Margin Softmax Loss** is employed in the last classification layer to impose larger angular margin on live classes. **Self-Supervised Similarity Loss** is applied to enforce the patch feature invariance within a single capture.

CASIA-FASD, there are two different spoof mediums and three different capture resolutions, so there are nine fine-grained patch types (three live and six spoof types).

Inspired by the latest face recognition approaches, during training we employ angular margin-based softmax loss, which can force feature cluster for each category to be compactly distributed and enable better generalization ability. Furthermore, as the distribution discrepancies between spoof samples are larger than live samples, we treat live and spoof samples asymmetrically: force the model to learn a more compact cluster within live samples while leaving spoof samples more dispersed in the feature space. We modify the angular margin-based softmax loss and apply asymmetric margin onto live and spoof patch types: imposing larger angular margin on live types to push more compact boundaries. Finally, the self-supervised similarity loss further regularizes the patch features by applying the positive part of the contrastive loss on two transformed patch views from a single whole face image. Given that spoof-specific discriminative information is present spatially in the entire face region, the features between two different patch views from the same face capture should be similar.

3.2. Patch Features Extraction

We want to avoid any transform which can lead to image distortion or the reduction of the important spoof-related information. Given the cropped face region x_i from the raw capture, the two augmented patch views from x_i are $x_i^{t_1} = t_1(x_i)$ and $x_i^{t_2} = t_2(x_i)$, where $t_1, t_2 \sim \mathcal{T}$. $\mathcal{T}$ is the sequence of non-distorted augmentation operations, which only have **random horizontal flip**, **random rotation**, and **fixed size cropping**. The two input patches are then passed into the encoder E_θ and normalization layer to get the final features: $f_i^{t_1} = Normalize(E_\theta(x_i^{t_1}))$, $f_i^{t_2} = Normalize(E_\theta(x_i^{t_2}))$.

3.3. Fine-Grained Patch Recognition

Assuming we have N patch type classes in the training dataset, which consists of k live and $N-k$ spoof classes. Each input patch $t(x_i)$ belongs to one fine-grained ground truth class $y_i \in \{L_1, L_2, ...L_k, S_1, S_2, ...S_{N-k}\}$, and the Angular-Margin Softmax Loss is applied to regularize the patch features. The Angular-Margin Softmax Loss has many variants [8, 16, 17, 23, 25] and is commonly used in face recognition to improve the generalization ability to open-set identities. In this work, we employ AM-Softmax [23] loss to optimize the fine-grained patch recognition model and modify it to address the asymmetric nature in face anti-spoofing.

3.3.1 Preliminaries

The formulation of the original Softmax loss is given by

$$\begin{aligned}\mathcal{L}_S &= -\frac{1}{n}\sum_{i=1}^{n} log\frac{e^{W_{y_i}^T \boldsymbol{f}_i}}{\sum_{j=1}^{c} e^{W_j^T \boldsymbol{f}_i}} \\ &= -\frac{1}{n}\sum_{i=1}^{n} log\frac{e^{\|W_{y_i}\|\|\boldsymbol{f}_i\|cos(\theta_{y_i})}}{\sum_{j=1}^{c} e^{\|W_j\|\|\boldsymbol{f}_i\|cos(\theta_j)}},\end{aligned} \tag{1}$$

where $\boldsymbol{f}$ is the input of the fully connected layer for classification ($\boldsymbol{f}_i$ denotes the i-th sample), W_j is the j-th column of the fully connected layer, and y_i is the ground truth label of the i-th sample. The term $W_{y_i}^T \boldsymbol{f}_i$ is also called the target logit of the i-th sample.

The large-margin property is introduced by Sphereface [16], which defines a general function $\psi(\theta)$

to impose the angular margin between feature and weight vectors. After applying feature and weight normalization ($\|W_{y_i}\| = \|\boldsymbol{f}_i\| = 1$), the loss function becomes

$$\mathcal{L}_S = -\frac{1}{n}\sum_{i=1}^{n} log\frac{e^{\psi(\theta_{y_i})}}{e^{\psi(\theta_{y_i})} + \sum_{j=1,j\neq y_i}^{c} e^{cos(\theta_j)}}, \quad (2)$$

where in AM-Softmax [23] the function $\psi(\theta)$ is defined as

$$\psi(\theta) = cos\theta - m \quad (3)$$

During implementation, the input after normalizing both the feature and the weight is actually $x = cos\theta_{y_i} = \frac{W_{y_i}^T f_i}{\|W_{y_i}\|\|f_i\|}$, so in the forward propagation it only needs to compute

$$\Psi(x) = x - m \quad (4)$$

Then it scales the cosine values using a hyper-parameter s and the final AM-Softmax loss function becomes

$$\begin{aligned}\mathcal{L}_{AMS} &= -\frac{1}{n}\sum_{i=1}^{n} log\frac{e^{s\cdot(cos\theta_{y_i}-m)}}{e^{s\cdot(cos\theta_{y_i}-m)} + \sum_{j=1,j\neq y_i}^{c} e^{s\cdot cos\theta_j}} \\ &= -\frac{1}{n}\sum_{i=1}^{n} log\frac{e^{s\cdot(W_{y_i}^T \boldsymbol{f}_i-m)}}{e^{s\cdot(W_{y_i}^T \boldsymbol{f}_i-m)} + \sum_{j=1,j\neq y_i}^{c} e^{sW_j^T \boldsymbol{f}_i}}.\end{aligned} \quad (5)$$

3.3.2 Asymmetric AM-Softmax Loss

We impose different angular margin m_l and m_s on live and spoof categories, respectively. Denote live category set as $L = \{L_1, L_2, ...L_k\}$ and spoof category set as $S = \{S_1, S_2, ...S_{N-k}\}$. The modified AM-Softmax Loss of one feature sample f_i becomes

$$\mathcal{L}_{AAMS}(\boldsymbol{f}_i) = \begin{cases} -log\frac{e^{s\cdot\left(W_{y_i}^T \boldsymbol{f}_i - m_l\right)}}{e^{s\cdot\left(W_{y_i}^T \boldsymbol{f}_i - m_l\right)} + \sum\limits_{j=1,j\neq y_i}^{N} e^{s\cdot W_j^T \boldsymbol{f}_i}} & y_i \in L \\ -log\frac{e^{s\cdot\left(W_{y_i}^T \boldsymbol{f}_i - m_s\right)}}{e^{s\cdot\left(W_{y_i}^T \boldsymbol{f}_i - m_s\right)} + \sum\limits_{j=1,j\neq y_i}^{N} e^{s\cdot W_j^T \boldsymbol{f}_i}} & y_i \in S \end{cases} \quad (6)$$

The final Asymmetric Recognition Loss on two augmented patch views from the image is formulated as

$$\mathcal{L}_{Asym} = -\frac{1}{n}\sum_{i=1}^{n}(\mathcal{L}_{AAMS}(\boldsymbol{f}_i^{t_1}) + \mathcal{L}_{AAMS}(\boldsymbol{f}_i^{t_2})) \quad (7)$$

3.4. Self-Supervised Similarity Loss

Given two different patch views from the same face image, the self-supervised similarity constraint is applied to enforce the features to be similar. Therefore, the spoof-related feature can be learned with patch location and rotation invariance.

$$\mathcal{L}_{Sim}(\boldsymbol{f}_i^{t_1}, \boldsymbol{f}_i^{t_2}) = \frac{1}{n}\sum_{i=1}^{n}\left\|\boldsymbol{f}_i^{t_1} - \boldsymbol{f}_i^{t_2}\right\|_2 \quad (8)$$

3.5. Training and Testing

3.5.1 Total Loss

The total loss L of the proposed framework during training is

$$\mathcal{L} = \alpha_1\mathcal{L}_{Asym} + \alpha_2\mathcal{L}_{Sim} \quad (9)$$

where α_1 and α_2 are the weights to balance the influence of loss components. In all of the experiments, we set $\alpha_1 = \alpha_2 = 1.0$.

3.5.2 Testing Strategy

Given a test face image, we uniformly crop patches from the whole image for the network inference, with the patch size the same as the one in the training process. Assuming we have P cropped patches features ($f^1, f^2, ...f^P$) from one face image, then the average live probability can be obtained by the sum of live class probabilities in the last fully connected layer:

$$LiveProb = \frac{1}{P}\sum_{i=1}^{P}\sum_{y\in L} Softmax(s\cdot W_y^T \boldsymbol{f^i}) \quad (10)$$

4. Experiments

4.1. Datasets and Protocols

Databases. Five databases OULU-NPU [1] (denoted as O), SiW [18] (denoted as S), CASIA-FASD [34] (denoted as C), Replay-Attack [5] (denoted as I), MSU-MFSD [27] (denoted as M) are used in the testing protocols. OULU-NPU and SiW are large-scale high-resolution databases containing four and three protocols to validate the generalization (e.g., unseen environment and spoof mediums) of models, respectively, which are utilized for intra-dataset testing. CASIA-MFSD, Replay-Attack, and MSU-MFSD are databases that contain low-resolution videos with much fewer video clips and are used for cross-dataset testing to validate the generalization ability to testing data with large distribution shift. There are three capture devices with quality ranging from low to high in CASIA-FASD, two devices in SiW and MSU-MFSD, and only one device in the other datasets. The fine-grained class number and the other statistics of databases are shown in Tab. 1. Note that in Oulu-NPU, even the collections are captured by six different types of phones, the quality and fine details are pretty similar, so we only split the patch type into five classes in total. More details and sample images can be found in the supplementary material.

Performance Metrics. In intra-dataset testing on OULU-NPU and SiW, we follow the original protocols and metrics, i.e., Attack Presentation Classification Error Rate (APCER), Bona Fide Presentation Classification Error Rate (BPCER), and Average Classification Error Rate

Dataset	# Subjects		# Clips		# Classes	
	Train	Test	Train	Test	Live	Spoof
OULU-NPU (O)	20	20	1800	1800	1	4
SiW (S)	90	75	2442	2036	2	12
CASIA-FASD (C)	20	30	480	720	3	6
ReplayAttack (I)	30	20	360	240	1	3
MSU-MFSD (M)	15	20	120	160	2	6

Table 1. Statistics of the face anti-spoofing datasets.

(ACER) for a fair comparison. Half Total Error Rate (HTER) and Area Under Curve (AUC) are adopted in the cross-dataset testing between OULU-NPU, CASIA-MFSD, Replay-Attack, and MSU-MFSD.

4.2. Implementation Details

All face anti-spoofing datasets above are stored in video format originally. We randomly select three frames from each video clip and use the state-of-the-art face detector RetinaFace [7] to crop the face for training. We set the fixed patch crop size as 160, and set the hyperparameter $s = 30.0, m_l = 0.4, m_s = 0.1$ in all protocols. We use ResNet18 [11] as the patch feature encoder, and we did not see much performance difference while using an encoder with larger capacity (as shown in the supplementary materials). Models are trained with SGD optimizer and the initial learning rate is 0.002. We train models with maximum 200 epochs while the learning rate halves every 90 epochs. During testing, we uniformly crop the fixed-size patches from the face input image: the minimum x and y coordinates are $size/2.0$, and the maximum x and y coordinates are $width - (size/2.0)$ and $height - (size/2.0)$, respectively. In all of the experiments during testing, we uniformly sample 3 patch anchors on each side, which results in $P = 9$ patches for score averaging.

4.3. Intra-Dataset Testing

We conduct experiments on Oulu-NPU [1] and SiW [18] for intra-dataset testing results. We compare the results with the most recent face anti-spoofing methods in the following.

4.3.1 Results on Oulu-NPU

Oulu-NPU [1] has four challenging protocols, which evaluate the model robustness against the unseen environment, unseen spoof mediums, unseen capture devices, and all of the above, respectively. The number of classes during training are 5, 3, 5, and 3, respectively. As shown in Tab. 2, our simple patch-based recognition approach achieves the best performance in all protocols. It clearly verifies the better generalization ability of the features learned through the patch recognition proxy tasks.

4.3.2 Results on SiW

SiW [18] is another commonly used high-quality dataset with more identities. The collection is captured by two different quality devices: Canon EOS T6 and Logitech C920. Compared to Oulu-NPU, it includes more environment variations and spoof mediums. The numbers of fine-grained patch type classes during training in protocol 1, 2, 3-1, and 3-2 are 14, 8, 6, and 10, respectively. As shown in Tab. 3, our method performs the best for the first two protocols and achieves competitive results in protocol 3.

Prot.	Method	APCER(%)	BPCER(%)	ACER(%)
1	Disentangle [33]	1.7	0.8	1.3
	SpoofTrace [19]	0.8	1.3	1.1
	BCN [29]	0.0	1.6	0.8
	CDCN [32]	0.4	1.7	1.0
	NAS-FAS [31]	0.4	0.0	0.2
	PatchNet (Ours)	0.0	0.0	**0.0**
2	Disentangle [33]	1.1	3.6	2.4
	SpoofTrace [19]	2.3	1.6	1.9
	BCN [29]	2.6	0.8	1.7
	CDCN [32]	1.5	1.4	1.5
	NAS-FAS [31]	1.5	0.8	**1.2**
	PatchNet (Ours)	1.1	1.2	**1.2**
3	Disentangle [33]	2.8±2.2	1.7±2.6	2.2±2.2
	SpoofTrace [19]	1.6±1.6	4.0±5.4	2.8±3.3
	BCN [29]	2.8±2.4	2.3±2.8	2.5±1.1
	CDCN [32]	2.4±1.3	2.2±2.0	2.3±1.4
	NAS-FAS [31]	2.1±1.3	1.4±1.1	1.7±0.6
	PatchNet (Ours)	1.8±1.47	0.56±1.24	**1.18±1.26**
4	Disentangle [33]	5.4±2.9	3.3±6.0	4.4±3.0
	SpoofTrace [19]	2.3±3.6	5.2±5.4	3.8±4.2
	BCN [29]	2.9±4.0	7.5±6.9	5.2±3.7
	CDCN [32]	4.6±4.6	9.2±8.0	6.9±2.9
	NAS-FAS [31]	4.2±5.3	1.7±2.6	**2.9±2.8**
	PatchNet (Ours)	2.5±3.81	3.33±3.73	2.9±3.0

Table 2. The results of testing on OULU-NPU protocols.

Prot.	Method	APCER(%)	BPCER(%)	ACER(%)
1	Disentangle [33]	0.07	0.50	0.28
	SpoofTrace [19]	0.00	0.00	**0.00**
	BCN [29]	0.55	0.17	0.36
	CDCN [32]	0.07	0.17	0.12
	DualStage [26]	0.00	0.00	**0.00**
	NAS-FAS [31]	0.07	0.17	0.12
	PatchNet (Ours)	0.00	0.00	**0.00**
2	Disentangle [33]	0.08±0.17	0.13±0.09	0.10±0.04
	SpoofTrace [19]	0.00±0.00	0.00±0.00	**0.00±0.00**
	BCN [29]	0.08±0.17	0.15±0.00	0.11±0.08
	CDCN [32]	0.00±0.00	0.13±0.09	0.06±0.04
	DualStage [26]	0.00±0.00	0.00±0.00	**0.00±0.00**
	NAS-FAS [31]	0.00±0.00	0.09±0.10	0.04±0.05
	PatchNet (Ours)	0.00±0.00	0.00±0.00	**0.00±0.00**
3	Disentangle [33]	9.35±6.14	1.84±2.60	5.59±4.37
	SpoofTrace [19]	8.3±3.3	7.5±3.3	7.9±3.3
	BCN [29]	2.55±0.89	2.34±0.47	2.45±0.68
	CDCN [32]	1.67±0.11	1.76±0.12	1.71±0.11
	DualStage [26]	4.77±5.04	2.44±2.74	3.58±3.93
	NAS-FAS [31]	1.58±0.23	1.46±0.08	**1.52±0.13**
	PatchNet (Ours)	3.06±1.1	1.83±0.83	2.45±0.45

Table 3. The results of testing on SiW protocols.

4.4. Ablation Study

In this subsection, all ablation studies are conducted on Protocol 1 (different illumination conditions and location between the train and test sets) of OULU-NPU [1] to explore the details of our patch-based recognition framework.

Output Class		Input Extraction		Loss Functions		ACER(%)
Binary	Fine	Resize	PatchCrop	$\mathcal{L}_{Asym}$	$\mathcal{L}_{Sim}$	
✓		✓				6.25
✓			✓			3.54
	✓	✓				5.63
	✓		✓			1.88
	✓		✓		✓	1.46
	✓		✓	✓		0.63
	✓		✓	✓	✓	0.0

Table 4. Ablation study of each component in PatchNet on OULU-NPU protocol 1.

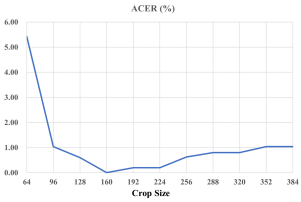


Figure 3. Comparison between choices of patch crop size.

m_l	m_s	ACER(%)
0.0	0.0	1.46
0.2	0.2	0.83
0.2	0.1	0.41
0.4	0.4	0.63
0.4	0.3	0.2
0.4	0.2	0.2
0.4	**0.1**	**0.0**
0.4	0.0	0.41

Table 5. Ablation study of margin choices in L_{Asym}.

Impact of Each Component. Tab. 4 shows the ablation study of each component in our proposed framework. The first row is the naive baseline (ACER: 6.25%) which formulates the face anti-spoofing as a binary classification problem (trained with the standard Cross Entropy loss) with the resized 256x256 face input. Surprisingly, by only adopting the fine-grained classes and raw frame cropping strategies, we can improve the performance significantly to 1.88% ACER. It shows that the naive baseline model could overfit to the high-level biases in the public FAS dataset, which only contains limited background and identities. Moreover, the fine details in cropped patches from raw frames are very critical to discriminate between different patch types in high-quality datasets like OULU-NPU. From the lower part of the table, we can observe the advantage of the proposed margin-based classification loss and the self-supervised similarity loss. It is clear that both regularization techniques can facilitate the encoder to learn more intrinsic features related to the capture device's characteristics and presenting materials.

Impact of Patch Crop Size. Fig. 3 demonstrates the ACER(%) on OULU-NPU protocol 1 between different crop sizes. We can observe that larger patch sizes during training might be prone to overfit to biases from the face capture. With the regularization of patch recognition loss and patch-based augmentation to increase training data variance, the overall performance does not differ much when enlarging the patch size. However, when the patch size is too small (e.g., 64), the performance degrades significantly as the capture characteristics can not be learned with very limited information.

Asymmetric Margin Choices. We conduct ablation experiments to verify the effectiveness of our asymmetric margin design in the angular margin softmax loss. From Tab. 5, we can observe that adding angular margins can significantly improve the generalization capability and outperform the model without any margin. Furthermore, adding a very large margin to spoof patch types which have more diverse appearances would hurt the discrimination power of learned features. We find that PatchNet works well on all testing protocols with the margin choice $m_l = 0.4$, $m_s = 0.1$.

Method	Train CASIA-MFSD(C)	Test Replay-Attack(I)	Train Replay-Attack(I)	Test CASIA-MFSD(C)
STASN [28]	31.5		30.9	
Disentangle [33]	22.4		30.3	
BCN [29]	16.6		36.4	
CDCN [32]	15.5		32.6	
DC-CDN [30]	**6.0**		<u>30.1</u>	
PatchNet (Ours)	<u>9.9</u>		**26.2**	

Table 6. The results of cross-dataset testing between CASIA-MFSD and Replay-Attack. The evaluation metric is HTER(%).

4.5. Cross-Dataset Testing

4.5.1 Experiments between C and I

First, following the related works, CASIA-MFSD (C) [34] and ReplayAttack (I) [5] are used for cross-dataset experiments, and the results are measured in HTER. During training, the numbers of fine-grained patch-type classes are 9 and 4 in $C \rightarrow I$ and $I \rightarrow C$ protocols, respectively. The results are shown in Tab. 6. Given the limited number of clips and low-quality videos from the ReplayAttack dataset, it is hard to learn generalizable features which can perform very well on other datasets, so the error rate in protocol $I \rightarrow C$ is still high compared with $C \rightarrow I$. Our proposed framework can achieve competitive performance compared to previous works in both protocols.

4.5.2 Domain Generalization Experiments

Some recent FAS works [12,21,24] consider each dataset as one domain and promote the domain generalization benchmark in FAS, which utilizes three datasets for training, and the remaining one as testing. As we aim to distinguish the patch type in the fine-grained manner, our proposed framework can be directly used to evaluate such benchmark without employing further generalization techniques (e.g., adversarial training or meta-learning). With access to more different patch types with more diverse capture devices, our framework is capable of learning more discriminative features through the patch recognition proxy task. There are four protocols in this benchmark: O&C&I to M, M&I&O to C, M&C&O to I, and M&C&I to O. During training, we directly combine the fine-grained patch classes from the three training datasets, which results in 18, 17, 22, and 21 classes, respectively.

The testing results are shown in Tab. 8. The proposed PatchNet achieves competitive results on all protocols. Due to the high variance of capture types in C dataset, it is hard

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(%)
PatchNet w/ coarse cls [12]	10.24	96.45	15.67	92.47	21.65	91.08	16.26	91.33
PatchNet w/o margin	10.0	96.61	18.0	91.57	17.25	90.47	15.04	92.42
PatchNet w/o $\mathcal{L}_{Sim}$	8.9	97.42	13.44	93.99	15.1	92.10	14.24	92.93
PatchNet (Ours)	**7.10**	**98.46**	**11.33**	**94.58**	**14.6**	**92.51**	**11.82**	**95.07**

Table 7. Evaluations of different components of the proposed method on four cross-dataset protocols.

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(%)
Auxiliary [18]	22.72	85.88	33.52	73.15	29.14	71.69	30.17	77.61
MADDG [20]	17.69	88.06	24.50	84.51	22.19	84.99	27.89	80.02
PAD-GAN [24]	17.02	90.10	19.68	87.43	20.87	86.72	25.02	81.47
RFM [21]	13.89	93.98	20.27	88.16	17.30	90.48	16.45	91.16
NAS-FAS [31]	16.85	90.42	15.21	92.64	**11.63**	**96.98**	13.16	94.18
SSDG-R [12]	7.38	97.17	10.44	95.94	11.71	96.59	15.61	91.54
ANRL [14]	16.03	91.04	**10.83**	**96.75**	17.85	89.26	15.67	91.90
DRDG [15]	15.56	91.79	12.43	95.81	19.05	88.79	15.63	91.75
PatchNet (Ours)	**7.10**	**98.46**	11.33	94.58	13.4	95.67	**11.82**	**95.07**

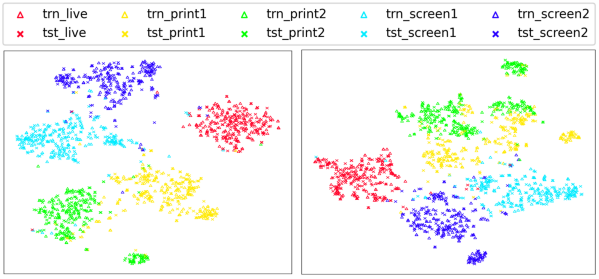
Table 8. Comparison results between the proposed PatchNet and state-of-the-art methods on four domain generalization protocols.

to learn robust local features to address both high and low resolution scenarios. We also conduct an ablation study in this benchmark to explore the influence of each component in our framework and show the results in Tab. 7. In the first ablation experiment, we split the patch classes using the strategy proposed by SSDG [12]: It aggregates the live samples as one class and treats spoof samples from each other dataset as one class, which results in 4 classes. The results verify that the proposed fine-grained class split, L_{Asym}, and L_{Sim} are all important to regularize the network to generalize better in challenging FAS tasks.

4.6. Visualizations

Patch Feature Distribution. In Fig. 4, we visualize the patch features by t-SNE [22] from OULU-NPU protocol 1, which both the training and testing sets consist of 5 patch types (live, print1, print2, screen1, screen2). We can observe in (b) that the model trained without margin cannot distinguish print1 and print2 types very well. The distribution for the live samples is more compact, and clusters are separated better for the model trained with the margin. The training and testing feature sets are aligned well in the feature space.

In Fig. 5, we visualize the feature distribution in the protocol M&I&O to C. We can observe that the features from the training set are separated well in the fine-grained manner. As dataset C contains three different quality capture devices, the corresponding three live classes are located separately in the embedding space. Aligning with the device characteristics, we observe that features from C3_L are close to O_L, which both types are captured from high-quality capture devices. Moreover, low-quality devices like C1_L exhibit similar appearance with that from similar quality device M1_L. It implies that the patch embedding space encodes the capture characteristic well.



(a) w/ margin (b) w/o margin

Figure 4. The t-SNE features visualizations on Oulu-NPU Protocol 1. (a) training with asymmetric margin (b) training without margin.

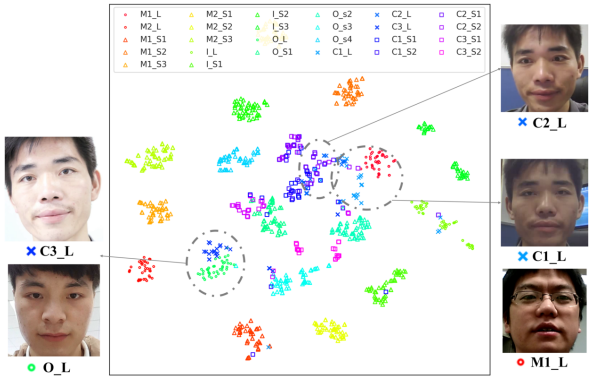


Figure 5. The t-SNE visualizations of normalized patch features in cross-dataset protocol M&I&O to C. The fine-grained patch type class is denoted by (Dataset)(SensorID)_(Liveness)(MediumID).

Patch Score Map. In Fig. 6, we compute the scores from patches across the whole image and visualize the live probability in the overlapped heat map. The samples are from the testing set of OULU-NPU protocol 1. The patch scores across the whole face input image are primarily consistent, except for some background and boundary parts, which is expected as the spoof cues should be learned from the face

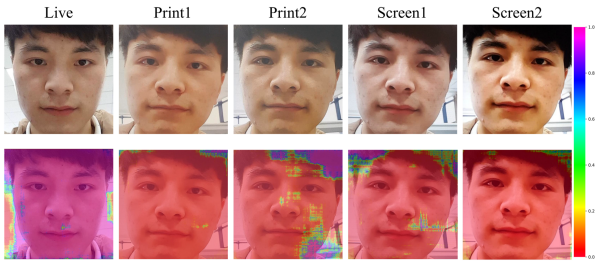


Figure 6. Patch score map of 5 different types in Oulu-NPU Protocol 1. From left to right: Live, Print1, Print2, Screen1, and Screen2. The number represents the live probability.

region but not the background's biases.

4.7. Applications

Testing on each Capture Device. Following the observation in Sec. 4.6, we argue that instead of testing on the whole dataset, it is more feasible to test the anti-spoofing performance on each device separately (e.g., in the CASIA-FASD dataset). We re-organize the testing dataset in O&C&I to M and O&M&I to C protocols, which is splitting the testing data by their capture device ID into 2 (M1, M2) and 3 (C1, C2, C3) groups, respectively. The cross-dataset anti-spoofing performance on the new split dataset of these two protocols is shown in Tab. 9. We observe that anti-spoofing performance on both devices in the M dataset are better than the average performance using the whole dataset, which is reasonable as the testing sets are smaller. However, only one device (the high-quality one C3) in the C dataset is better than the average performance, which aligns with the t-SNE visualization in Fig. 5. The capture images of C2 have many noises and image compression effects, which can lead to significant degradation of the discriminative power of features. With the detailed performance report on each device, we can further improve the anti-spoofing system or improve the quality of the problematic capture devices as well.

Few-Shot Reference Anti-Spoofing. With the learned patch embedding space which encodes intrinsic patch features to discriminate between patch types, the distance between features in the embedding space can be used to measure the similarity between patch types. As the features are already normalized in the space, we can compute the cosine distance between features to enable a new application: few-shot reference anti-spoofing. While there is a new capture device, it is easier to acquire some **live** face image samples, and those sample features can be used as the reference in the embedding space. We compute the distances between other testing features and live reference to obtain the similarity scores. Higher similarity scores mean the samples are more likely to be live. The 5-shot and 10-shot live reference anti-spoofing performance are reported in Tab. 9. We can conclude that with the live reference features and the learned patch embedding space, the performance can be boosted a lot in practical scenarios.

Method	O&C&I to M (AUC: 98.46%)		O&M&I to C (AUC: 94.58%)		
	M1	M2	C1	C2	C3
PatchNet	99.54	98.63	94.29	88.49	98.13
PatchNet w/ 5-shot	99.7	99.6	94.3	89.8	98.6
PatchNet w/ 10-shot	99.8	99.6	95.3	90.7	99.2

Table 9. Split testing and few-shot live reference testing on each capture device. The few-shot testing score is averaged by 10 experiment runs. AUC(%) score is reported.

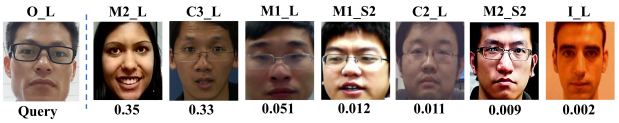


Figure 7. Patch type retrieval using a live sample from O dataset as the query. Top retrieval results are shown, and the numbers below are the cosine similarity.

Patch Type Retrieval. The normalized patch embedding space can be used for the patch type retrieval application, and can boost the FAS performance on certain capture devices. For example, after training the feature space with M&C&I datasets to recognize 21 patch types, we can measure the similarity between testing patch types and training patch types. As shown in Fig. 7, we take a live sample from O dataset as the query, and retrieve the training patch types by computing the cosine distance with each type weight vector. The top-7 retrieval patch types are (M2_L, C3_L, M1_L, M1_S2, C2_L, M2_S2, I_L). Some spoof types from M dataset have higher ranking than live types from C2 and I datasets, which have low quality captures. To optimize the performance while testing on O dataset, we can 1) remove ambiguous live types (C2_L, C1_L, I_L) during testing, or 2) re-define the class for (C2_L, C1_L, I_L) as "spoof" for training. Both strategies can boost the performance to 95.27% and 95.87% AUC, respectively.

5. Conclusions and Future Work

In this paper, we reformulate face anti-spoofing as a fine-grained patch type recognition task and present a simple training framework called PatchNet to efficiently learn the patch embedding space which encodes the spoof-related capture characteristics. The novel loss functions are designed to enhance the feature discrimination power. Extensive experiments on challenging FAS protocols verify the effectiveness of the proposed method. We note that exploration of a generic embedding space to discriminate different captures is still at an early stage. Future directions include: 1) Learning a more generalized embedding space by datasets with more variations or transferred from the material perception task, and 2) Investigation into the Few-Shot FAS protocols which have more practical values.

References

[1] Z. Boulkenafet, J. Komulainen, L. Li, X. Feng, and A. Hadid. Oulu-npu: A mobile face presentation attack database with real-world variations. In *IEEE International Conference on Automatic Face Gesture Recognition (FG)*, 2017. 1, 4, 5

[2] R. Cai, H. Li, S. Wang, C. Chen, and A. C. Kot. Drl-fas: A novel framework based on deep reinforcement learning for face anti-spoofing. *IEEE Transactions on Information Forensics and Security*, 2021. 2

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2

[4] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020. 2

[5] Ivana Chingovska, André Anjos, and Sébastien Marcel. On the effectiveness of local binary patterns in face anti-spoofing. In *BIOSIG-proceedings of the international conference of biometrics special interest group (BIOSIG)*. IEEE, 2012. 1, 4, 6

[6] Debayan Deb and Anil K Jain. Look locally infer globally: A generalizable face anti-spoofing approach. *IEEE Transactions on Information Forensics and Security*, 2020. 2

[7] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *CVPR*, 2020. 5

[8] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 2, 3

[9] Haocheng Feng, Zhibin Hong, Haixiao Yue, Yang Chen, Keyao Wang, Junyu Han, Jingtuo Liu, and Errui Ding. Learning generalized spoof cues for face anti-spoofing. *arXiv preprint arXiv:2005.03922*, 2020. 2

[10] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020. 2

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[12] Yunpei Jia, Jie Zhang, Shiguang Shan, and Xilin Chen. Single-side domain generalization for face anti-spoofing. In *CVPR*, 2020. 2, 6, 7

[13] Taewook Kim, YongHyun Kim, Inhan Kim, and Daijin Kim. Basn: Enriching feature representation using bipartite auxiliary supervisions for face anti-spoofing. In *CVPR Workshops*, 2019. 2

[14] 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 *Proceedings of the 29th ACM International Conference on Multimedia*, 2021. 1, 2, 7

[15] 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. *arXiv preprint arXiv:2106.16128*, 2021. 1, 2, 7

[16] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, 2017. 3

[17] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, 2016. 3

[18] Yaojie Liu, Amin Jourabloo, and Xiaoming Liu. Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In *CVPR*, 2018. 1, 2, 4, 5, 7

[19] Yaojie Liu, Joel Stehouwer, and Xiaoming Liu. On disentangling spoof trace for generic face anti-spoofing. In *ECCV*, 2020. 1, 5

[20] Rui Shao, Xiangyuan Lan, Jiawei Li, and Pong C Yuen. Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In *CVPR*, 2019. 2, 7

[21] Rui Shao, Xiangyuan Lan, and Pong C Yuen. Regularized fine-grained meta face anti-spoofing. In *AAAI*, 2020. 1, 2, 6, 7

[22] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 2008. 7

[23] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 2018. 2, 3, 4

[24] Guoqing Wang, Hu Han, Shiguang Shan, and Xilin Chen. Cross-domain face presentation attack detection via multi-domain disentangled representation learning. In *CVPR*, 2020. 2, 6, 7

[25] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *CVPR*, 2018. 2, 3

[26] Yu-Chun Wang, Chien-Yi Wang, and Shang-Hong Lai. Disentangled representation with dual-stage feature learning for face anti-spoofing. In *WACV*, 2022. 5

[27] D. Wen, H. Han, and A. K. Jain. Face spoof detection with image distortion analysis. *IEEE Transactions on Information Forensics and Security*, 2015. 1, 4

[28] Xiao Yang, Wenhan Luo, Linchao Bao, Yuan Gao, Dihong Gong, Shibao Zheng, Zhifeng Li, and Wei Liu. Face anti-spoofing: Model matters, so does data. In *CVPR*, 2019. 2, 6

[29] Zitong Yu, Xiaobai Li, Xuesong Niu, Jingang Shi, and Guoying Zhao. Face anti-spoofing with human material perception. In *ECCV*, 2020. 1, 2, 5, 6

[30] Zitong Yu, Yunxiao Qin, Hengshuang Zhao, Xiaobai Li, and Guoying Zhao. Dual-cross central difference network for face anti-spoofing. *arXiv preprint arXiv:2105.01290*, 2021. 1, 2, 6

[31] Z Yu, J Wan, Y Qin, X Li, SZ Li, and G Zhao. Nas-fas: Static-dynamic central difference network search for face anti-spoofing. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 5, 7

[32] 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*, 2020. 5, 6

[33] 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*, 2020. 1, 5, 6

[34] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li. A face antispoofing database with diverse attacks. In *IAPR International Conference on Biometrics (ICB)*, 2012. 1, 4, 6