Meta-Teacher For Face Anti-Spoofing

Yunxiao Qin, Zitong Yu, Longbin Yan, Zezheng Wang, Chenxu Zhao, Zhen Lei, *Senior Member, IEEE*

Abstract—Face anti-spoofing (FAS) secures face recognition from presentation attacks (PAs). Existing FAS methods usually supervise PA detectors with handcrafted binary or pixel-wise labels. However, handcrafted labels may are not the most adequate way to supervise PA detectors learning sufficient and intrinsic spoofing cues. Instead of using the handcrafted labels, we propose a novel Meta-Teacher FAS (MT-FAS) method to train a meta-teacher for supervising PA detectors more effectively. The meta-teacher is trained in a bi-level optimization manner to learn the ability to supervise the PA detectors learning rich spoofing cues. The bi-level optimization contains two key components: 1) a lower-level training in which the meta-teacher supervises the detector's learning process on the training set; and 2) a higher-level training in which the meta-teacher's teaching performance is optimized by minimizing the detector's validation loss. Our meta-teacher differs significantly from existing teacher-student models because the meta-teacher is explicitly trained for better teaching the detector (student), whereas existing teachers are trained for outstanding accuracy neglecting teaching ability. Extensive experiments on five FAS benchmarks show that with the proposed MT-FAS, the trained meta-teacher 1) provides better-suited supervision than both handcrafted labels and existing teacher-student models; and 2) significantly improves the performances of PA detectors.

Index Terms—Face anti-spoofing, meta-teacher, pixel-wise supervision.

1 Introduction

Face recognition [1]–[4] has been widely utilized in identity authentication products. However, face recognition is vulnerable to realistic presentation attacks (PAs), including faces printed on paper (print attack), faces replayed on digital devices (replay attack), *etc.*. Aiming to secure face recognition systems from PAs, face anti-spoofing (FAS) [5]–[9] technology has attracted increasing attention from both academia and industry.

In the past two decades, both traditional handcrafted feature-based [5], [6], [10], [11] and deep learning-based [7]–[9], [12], [13] methods have been shown to be effective for FAS. On the one hand, classical handcrafted descriptors [5], [6], [10], [11] extract discrimination between live and spoof faces based on human prior knowledge. These approaches are efficient but unreliable in complex and unseen scenarios. On the other hand, deep learning-based methods [7], [12], [13] usually train robust presentation attack (PA) detectors to mine intrinsic spoofing patterns in an end-to-end fashion. Compared with handcrafted feature-based detectors, deep learning-based PA detectors usually have stronger representation capacity to detect spoof faces due to both deep networks and large-scale training data.

Generally, FAS can be treated as a binary classification problem (i.e., live as '0' vs. spoof as '1'); thus, binary label with binary cross-entropy loss is widely used for supervising the PA detector. However, deep models with binary loss might discover arbitrary cues [14] that can separate the two classes (e.g., screen bezel) but not the faithful spoofing patterns.

Recently, several hand-designed pixel-wise labels, including facial depth label [14]–[16], facial reflection label [17], [18], and pixel-wise binary label [19]–[21], have become more popular than binary classification label in FAS. These pixel-wise labels utilize human's prior-knowledge about spoof faces to supervise the PA detector to learn the spoofing cues of the global distinction between 1) live and spoof facial depths; 2) live and spoof facial light reflections; and 3) live and spoof facial skin and materials (*e.g.*, paper, screen), respectively. The facial depth label is the most popularly employed pixel-wise label. Because obtaining the real facial depths of all live and spoof faces is impractical, the facial depth-based FAS methods [14], [15], [22], [23] commonly train the PA detector to regress spoof faces as zero-map (each pixel value is zero) and to regress live faces as pseudo facial depths. When testing, they classify each face by comparing the average value of the detector's prediction map with the threshold.

Although existing handcrafted pixel-wise labels provide reasonable supervision signals for the PA detector, they might still be sub-optimal in two aspects: 1) the design of these labels is empirical, and the human prior-knowledge about FAS applied in these labels may block the detector from exploring a broad range of spoofing cues; and 2) it is difficult for a specific human-defined label to be effective against all possible spoof types. As increasingly challenging attack manners are developing, the inherent global spoofing cues that these labels provide to the detector may lose effectiveness. For instance, as shown in Fig. 1, SiW-M [19] proposed several novel spoof types, such as funny eye, paper mask, and transparent mask. The spoofing cues are mainly located on the eyeglasses and mask; as a result, existing pixel-wise labels (facial depth label and pixel-wise

- *Y. Qin, L. Yan are with Northwestern Polytechnical University, Xian 710072, China. E-mail: qyxqyx@mail.nwpu.edu.cn, yanlongbin@mail.nwpu.edu.cn.*
- *Z. Yu is with Center for Machine Vision and Signal Analysis, University of Oulu, Oulu 90014, Finland. E-mail: zitong.yu@oulu.fi.*
- *Z. Wang is with Beijing Kwai Technology Co., Ltd, Beijing 102600, China E-mail: wangzezheng@kuaishou.com.*
- *C. Zhao is with MiningLamp Technology, Beijing 100000, China. E-mail: zhaochenxu@mininglamp.com.*
- *Z. Lei is with National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. E-mail: zlei@nlpr.ia.ac.cn.*

Manuscript received November 21, 2020. Corresponding Author: Zhen Lei.

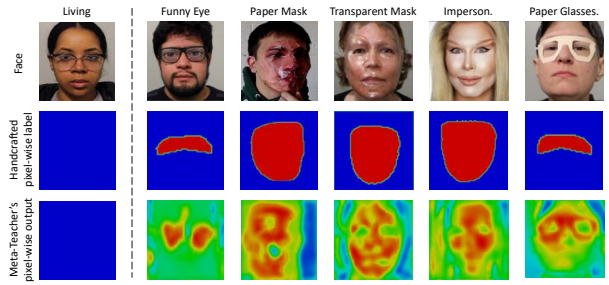


Fig. 1. The first row: some examples of the live and spoof (funny eye, paper mask, transparent mask, impersonate, and paper glasses) faces from SiW-M [19]. The second row: the exhausted annotated pixel-wise mask labels for the live and spoof faces in the first row. The third row: the normalized pixel-wise supervision provided by the proposed meta-teacher. The colors ranging from blue to red denote the float numbers ranging from 0 to 1. Compared with handcrafted labels, the meta-teacher can explore reasonable pixel-wise supervision to train the PA detector without using human effort.

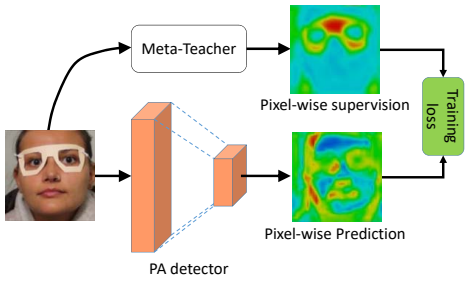


Fig. 2. Using meta-teacher to supervise the PA detector's learning.

binary label) may lose their effectiveness in supervising the PA detector to discriminate between live faces and these spoof faces. To better guide the detector in capturing the new local spoofing cues, SiW-M provides novel expensive human-annotated pixel-wise mask labels, as shown in Fig. 1.

Considering the aforementioned two underlying weaknesses of human-defined labels, **we explore better pixel-wise supervision for PA detectors from a novel perspective**. In the field of computer vision, as an alternative to handcrafted labels, another popular and effective supervision approach is to use a well-trained teacher model to supervise the training of another deep model (student) [24]–[31]. This kind of method is commonly referred to as teacher-student method. They usually first train a powerful and larger teacher model to learn the training data. Then, they use the well-performing teacher model to supervise the learning of a shallower and lightweight student model.

In this paper, inspired by teacher-student methods, we develop a novel **Meta-Teacher FAS (MT-FAS)** method to train a novel teacher called meta-teacher, to explore better pixel-wise supervision for PA detectors. In this work, we use the meta-teacher's prediction to supervise PA detectors. Our meta-teacher differs considerably from existing teachers in the following two aspects:

First, existing teacher-student methods [25]–[28] do not use explicit supervision to optimize the teacher's teaching ability. Thus, we cannot ensure that the teacher who well matches the training data can always perform well in supervising the student [32] because matching the training data and supervising the student are two different tasks. In contrast, the proposed MT-FAS trains the meta-teacher **exploring how to better teach (supervise) the detector (student) instead of learning the training data**. In other words, the optimizing objective of MT-FAS is the meta-teacher's pixel-wise supervision towards the detector (student) but not the meta-teacher's accuracy on the training set. **Second**, to effectively supervise students, existing teachers are usually built with deeper and heavier models to ensure their excellent performances [24], [25]. In contrast, our meta-teacher does not have to be deep and heavy because it is explicitly trained to learn how to better supervise the PA detector. In terms of performance, we guarantee it to be perfect by manually separating its pixel-wise predictions for live and spoof faces into two different scopes. Specifically, we constrain each pixel value of the meta-teacher's output map for live faces to zero and constrain that for spoof faces to the range from zero to a certain positive number. Fig. 1 visualizes the meta-teacher's pixel-wise predictions for live and spoof faces. Clearly, for spoof faces, the proposed meta-teacher has learned how to provide efficient pixel-wise supervision to train the PA detector learning the effective spoofing cues for the novel spoof types.

We utilize a bi-level optimization manner [33]–[36] to train the meta-teacher. In the lower-level, the meta-teacher supervises the PA detector to learn the training set. Normally, the more precisely the meta-teacher supervises the PA detector, the better the trained detector performs. Therefore, in the higher-level, we validate the meta-teacher's teaching performance by evaluating the trained detector on the validation set. In this work, we use the detector's validation performance to represent the meta-teacher's teaching performance. Finally, we use the detector's validation loss to optimize the meta-teacher to learn how to provide superior supervision to improve the detector's study of spoofing cues. After optimizing the meta-teacher, we use it to supervise existing PA detectors' learning without using human-defined labels, as illustrated in Fig. 2. To sum up, the main contributions of this paper are listed:

- Instead of supervising the PA detector with handcrafted labels, we propose a novel Meta-Teacher FAS (MT-FAS) method for training a meta-teacher model to provide better-suited supervision for the PA detector. In contrast to existing teacher-student methods, we develop a bi-level optimization manner to explicitly optimize the meta-teacher's teaching ability, which is also novel for the field of teacher-student learning.
- Comprehensive experiments are conducted to verify the effectiveness of the proposed MT-FAS and meta-teacher. Benefiting from the bi-level optimization objective of learning how to precisely supervise the student's learning, the proposed meta-teacher has the following advantage that compared with handcrafted labels and existing teachers, it supervises existing PA detectors more effectively and improves the performances of existing PA detectors substantially. With the help of the trained meta-teacher, we update state-of-the-art performances on five FAS benchmarks.

In the remainder of the paper, Section 2 provides the

related works of face anti-spoofing and teacher-student. Section 3 introduces the Meta-Teacher and gives details about its implementation for the FAS task. Section 4 provides rigorous ablation studies and evaluates the performance of the proposed MT-FAS on five benchmark datasets. Finally, a conclusion is given in Section 5.

2 Background

2.1 Face Anti-Spoofing

In recent decades, FAS technology, which protects face recognition systems from PAs, has becomes a research hotspot. Researchers traditionally utilize well-designed feature extractors, such as local binary patterns (LBPs) [5], [37], SIFT [38], speeded-up robust features (SURF) [11], histogram of oriented gradients (HOG) [6], difference of Gaussians (DoG) [10] and remote photoplethysmography (rPPG) [39]–[41] to extract discriminative features between live faces and spoof faces. Based on the extracted discriminative features, spoof faces can be detected with a classifier.

Recently, deep learning-based FAS methods [8], [12], [21], [23], [42]–[49] outperform traditional FAS methods on several large-scale benchmarks and have become the mainstream technology in the field of FAS. Specifically, deep learning-based methods usually train a deep network-based PA detector to learn discriminative features between live faces and spoof faces in a data-driven manner. The binary classification label (*e.g.*, live as '0' and spoof as '1', or vice versa) is the most widely employed label to supervise a PA detector's training. Inspired by the discriminations between the facial depths of live and spoof faces, the facial depth label [14], [15], [50], [51] was recently proposed. The facial depth label provides fine-grained local supervision for PA detectors, and facial depth-based methods usually supervise the detectors to regress live faces as facial depths and to regress the spoof faces as zero-maps.

The reflection discrepancy between real facial skin and the surface of PAs is another inherent discrimination between live faces and spoof faces. The methods of [18] and [17] adopt off-the-shelf generated reflection map as the supervision signal and validate the effectiveness of the reflection supervision. However, this pseudo reflection label is easily influenced by environmental illumination.

In addition to the above-mentioned labels, some methods [19], [20] have shown that pixel-wise binary label also works promisingly. Pixel-wise binary labels assume that the materials of physical carriers of spoof faces are consistent and can be used to discriminate between live faces and spoof faces. These methods usually set the live face label to a zero-map and set the spoof face label to a one-map (each pixel value of the map is one) or vice versa.

Although existing human-defined labels are capable of supervising PA detectors to learn reasonable spoofing cues, the best-suited form of supervision remains an open question. In this paper, different from existing FAS methods using human-defined labels, we explore adaptive and learnable supervision signals specifically for supervising the PA detector more precisely.

2.2 Teacher-Student Methods

Traditionally, the training of neural network models is supervised by handcrafted labels. Several recent studies [24]–[30], [32], [52] show that using one or more well-trained deep and wide models to supervise another lighter model's training would benefit the lighter model's performance. This kind of training is commonly referred to as knowledge distillation (KD), as the lighter model distills knowledge from the cumbersome models. Because these methods simulate teachers' teaching process, where the cumbersome models act as teachers and the lighter models act as students, these methods are also called as teacher-student methods.

Hinton et al. [24] propose the earliest teacher-student method. The authors utilize the teacher's output logits to supervise the student's training. In addition to the teacher's logits, FitNets [25] demonstrates that intermediate features of the teacher supervises the student more efficiently. Researchers often make the teacher's network much larger than the student's network to guarantee the teacher's superior capacity and performance. BANs [26], however, shows that the teacher network does not have to be larger than the student model. A teacher that has the same network as the student can still improve the student's learning. Furthermore, [29] and [32] demonstrate that improving the teacher's performance does not always enable the student to learn better. Teacher-student methods will lose efficacy when the representation ability gap between the teacher models and the student turns too large.

Although the existing teacher-student methods are promising, they train the teacher to focus on learning how to perform better but not how to reliably teach the student. In this paper, we propose to train a FAS meta-teacher to focus on providing better-suited supervision signals for the student (PA detector).

3 Methodology

The goal of MT-FAS is to train a meta-teacher to teach (supervise) PA detectors more precisely rather than learn the training data. In other words, the optimizing objective of MT-FAS is the meta-teacher's teaching ability. MT-FAS solves this objective via a bi-level optimizing manner that consists of lower-level learning and higher-level learning in each training iteration. In the lower-level learning, the meta-teacher supervises the detector's learning process on the training set. In the higher-level learning, the trained detector is evaluated on the validation set, and the meta-teacher is optimized by minimizing the detector's validation loss.

In this section, we detail the proposed MT-FAS with the following steps. First, we show how the meta-teacher supervises the detector's learning process in the lower-level learning. Second, we introduce how we evaluate the meta-teacher's teaching performance. Third, we detail the optimization of the meta-teacher in each training iteration. Finally, we consider more detailed problems in the implementation of MT-FAS.

3.1 Using the Meta-Teacher to Supervise the Detector

In the lower-level learning of each bi-level training iteration, we use the meta-teacher to supervise the PA detector (stu-

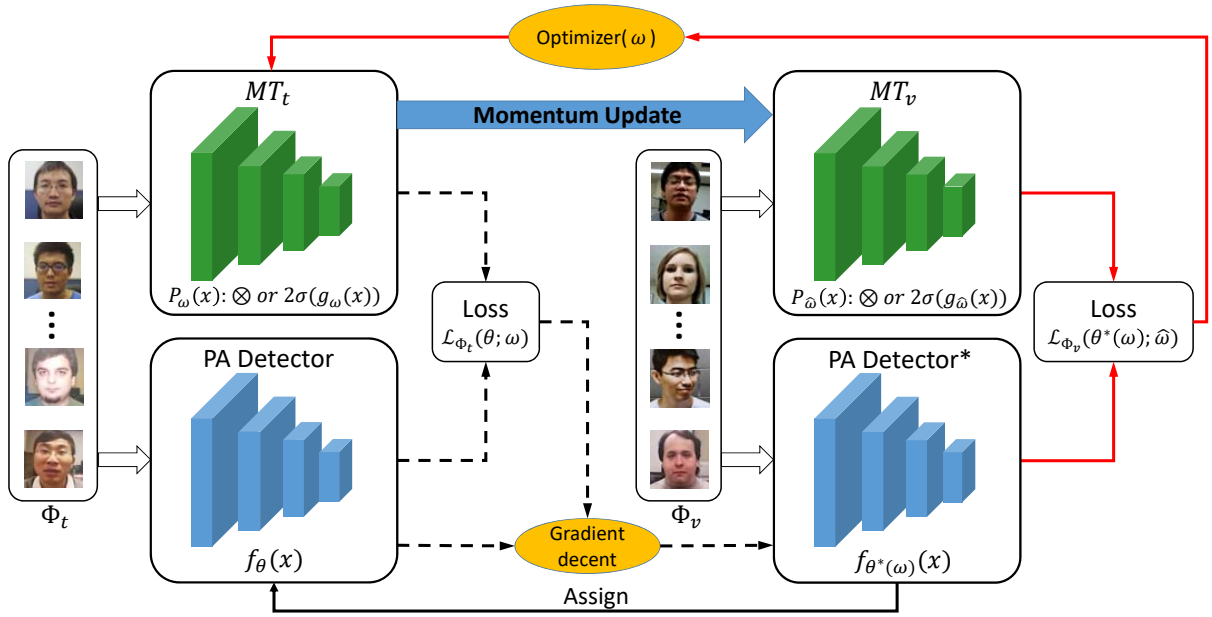


Fig. 3. The training framework of MT-FAS. Each training iteration contains a lower-level and a higher-level learning. In the lower-level (black dotted arrow), the mini training data batch is simultaneously fed into the meta-teacher MT_t and the PA detector. Given training data, MT_t provides pixel-wise supervision for the detector. Be supervised by MT_t, the detector with weight θ is optimized to turn to newly detector with weight $\theta^*(\omega)$. The optimizer in the lower-level learning is gradient descent with learning rate α. In the higher-level (the red arrow), the detector is evaluated on the mini validation data batch using the pixel-wise answer provided by another meta-teacher MT_v. Then, the meta-teacher MT_t is optimized using the detector's validation loss $\mathcal{L}_{\Phi_v}(\theta^*(\omega)$. MT_v is momentum updated via Eq. 11. The PA detector's weight θ is updated by copying the weight $\theta^*(\omega)$, as the black arrow shows.

dent) to learn the training set. The learning of the detector can be formulated as

$$\begin{aligned} \mathcal{L}_{\Phi_t}(\theta;\omega) &= \frac{1}{N_t}\sum_{k}^{N_t} \|f_\theta(x_k) - P_\omega(x_k)\|_2, \\ \theta^*(\omega) &= \underset{\theta}{argmin}\, \mathcal{L}_{\Phi_t}(\theta;\omega), \end{aligned} \tag{1}$$

where Φ_t is the training set and x_k is the k-th training face; N_t is the number of faces in the training set; f_θ is the detector parameterized by θ; P_ω is the meta-teacher parameterized by ω. In this work, either the meta-teacher or the detector predicts each input face as a map since numerous existing works [14]–[18], [21] demonstrate that pixel-wise regression supervision outperforms binary cross-entropy supervision. $f_\theta(x_k)$ and $P_\omega(x_k)$ are the meta-teacher and the detector's output maps, respectively, for face x_k. $\mathcal{L}_{\Phi_t}(\theta)$ is the detector's mean-square error (MSE) loss on the training set.

In the lower-level learning, MT-FAS uses the meta-teacher's output map to supervise the detector's training on the training set. As θ is optimized with the supervision provided by the meta-teacher, we use $\theta^*(\omega)$ to denote the weight of the optimized detector. θ and ω can also be understood as the lower-level and upper-level weights [53], [54], respectively.

To guarantee perfect meta-teacher performance in distinguishing live and spoof faces, we manually set the output of the meta-teacher $P^t(x_k)$ to

$$P_\omega(x_k) = \begin{cases} \otimes, & if\ x_k \in live, \\ 2\sigma(g_\omega(x_k)), & if\ x_k \in spoof, \end{cases} \tag{2}$$

where $\otimes \in \mathcal{R}^{32\times32}$ is the zero-map for live faces and $2\sigma(g_\omega(x_k)) \in \mathcal{R}^{32\times32}$ is the output pixel-wise map for spoof faces. g_ω is the network of the meta-teacher, and σ is the sigmoid function such that each pixel value in the map $2\sigma(g_\omega(x_k))$ is constrained to the range of 0 to 2.

3.2 Evaluating the Meta-Teacher's Teaching Quality

The higher-level learning of each bi-level training iteration evaluates and optimizes the meta-teacher's teaching performance. Here, we introduce how we evaluate the meta-teacher's teaching performance. The optimization details of the meta-teacher will be introduced in Section 3.3.

Theoretically, the meta-teacher's teaching quality determines the performance of the trained student (detector); that is, better teaching quality of the meta-teacher will lead to better performance of the detector on the validation set. Therefore, the most straightforward way to evaluate the quality of the teacher's teaching is to evaluate the performance of the trained student on the validation/testing set. In this work, we use the student's performance on the validation set to represent the teacher's teaching ability. The detector's validation loss can be formulated as

$$\mathcal{L}_{\Phi_v}(\theta^*(\omega);\widehat{\omega}) = \frac{1}{N_v}\sum_{k}^{N_v} \|f_{\theta^*(\omega)}(x_k) - P_{\widehat{\omega}}(x_k)\|_2, \tag{3}$$

where Φ_v denotes the validation set and N_v is the number of examples in Φ_v. x_k is the k-th example in Φ_v. $P_{\widehat{\omega}}$ is another meta-teacher on the validation set, and $\widehat{\omega}$ is the corresponding weight. For clarity, we denote the meta-teacher P_ω on the training set as MT_t and denote the meta-teacher $P_{\widehat{\omega}}$ on the validation set as MT_v. The reason we use two different meta-teachers is that we use the detector's validation loss to represent MT_t's teaching ability. If the detector's validation loss is calculated with MT_t, then the situation in which MT_t

is evaluated by itself will occur. This situation may result in misestimation of the meta-teacher and further damage the training of the meta-teacher. To avoid this issue, a new MT_v is needed to calculate the validation loss $\mathcal{L}_{\Phi_v}(\theta^*(\omega);\widehat{\omega})$ and evaluate MT_t. We will present experiments in Section 4.4.2 to demonstrate the indispensability of MT_v. Similar to MT_t, we formulate MT_v as

$$P_{\widehat{\omega}}(x_k) = \begin{cases} \otimes, & if\ x_k \in live, \\ 2\sigma(g_{\widehat{\omega}}(x_k)), & if\ x_k \in spoof, \end{cases} \tag{4}$$

where $g_{\widehat{\omega}}$ is the network of MT_v. The main difference between MT_t and MT_v is that they predict pixel-wise maps for spoof faces using different weights: the weight of MT_v is $\widehat{\omega}$, while that of MT_t is ω.

To ensure that the meta-teacher MT_t's teaching ability is correctly evaluated, we propose the **premise** that *a better MT_t can teach the detector to achieve a lower validation loss*. To satisfy this premise, MT_t and MT_v should closely correlate with each other despite their differences. The relationship between MT_t and MT_v will be discussed in greater detail in Section 3.4.

Typically, if the premise is perfectly satisfied, the best MT_t should lead the detector to achieve the minimal validation loss. Therefore, the best weight ω^* for MT_t can be formulated as

$$\omega^* = \underset{\omega}{argmin}\, \mathcal{L}_{\Phi_v}(\theta^*(\omega);\widehat{\omega}). \tag{5}$$

3.3 Optimization of the Meta-Teacher

The purpose of MT-FAS is to train the meta-teacher MT_t to provide better pixel-wise supervision to train the detector f_θ, as Eq. 5 shows. Here, we detail how MT_t is optimized.

First, we approximate the optimization of θ^* with one gradient descent step, formulated as

$$\theta^*(\omega) \approx \theta - \alpha \cdot \nabla_\theta \mathcal{L}_{\Phi_t}(\theta;\omega), \tag{6}$$

where α is the detector's learning rate on the training set. Since we wish the detector trained with P_ω to achieve the minimal validation loss, we optimize P_ω by minimizing the detector's validation loss. The formulation of the optimization is

$$\omega' = \omega - \beta \cdot \nabla_\omega \mathcal{L}_{\Phi_v}(\theta^*(\omega);\widehat{\omega}), \tag{7}$$

where β denotes the learning rate of the meta-teacher. By using the chain rule, we reformulate $\nabla_\omega \mathcal{L}_{\Phi_v}(\theta^*(\omega);\widehat{\omega})$ as

$$\nabla_\omega \mathcal{L}_{\Phi_v}(\theta^*(\omega);\widehat{\omega}) = \nabla_\omega \theta^*(\omega) \cdot \nabla_{\theta^*(\omega)} \mathcal{L}_{\Phi_v}(\theta^*(\omega);\widehat{\omega}). \tag{8}$$

We replace $\theta^*(\omega)$ using Eq. 6 and write the first item $\nabla_\omega \theta^*(\omega)$ in detail as

$$\begin{aligned} \nabla_\omega \theta^*(\omega) &\approx \nabla_\omega(\theta - \alpha \cdot \nabla_\theta \mathcal{L}_{\Phi_t}(\theta;\omega)) \\ &= -\alpha \cdot \nabla_\omega \nabla_\theta \mathcal{L}_{\Phi_t}(\theta;\omega) \\ &= -\alpha \cdot \nabla_\omega \nabla_\theta \frac{1}{N_t} \sum_k^{N_t} \|g_\theta(x_k) - P_\omega(x_k)\|_2. \end{aligned} \tag{9}$$

$\frac{1}{N_t}\sum_k^{N_t} \|g_\theta(x_k) - P_\omega(x_k)\|_2$ is the validation loss and is differentiable with respect to $P_\omega(x_k)$. $P_\omega(x_k)$ is also differentiable with respect to ω. Therefore, we can calculate the gradient $\nabla_\omega \nabla_\theta \frac{1}{N_t}\sum_k^{N_t} \|g_\theta(x_k) - P_\omega(x_k)\|_2$. Finally, we rewrite the optimization of MT_t as

$$\begin{aligned} \omega' &= \omega - \beta \cdot \nabla_\omega \mathcal{L}_{\Phi_v}(\theta^*(\omega);\widehat{\omega}) \\ &= \omega + \beta \cdot \alpha \cdot \nabla_\omega \nabla_\theta \mathcal{L}_{\Phi_t}(\theta;\omega) \cdot \nabla_{\theta^*(\omega)} \mathcal{L}_{\Phi_v}(\theta^*(\omega);\widehat{\omega}). \end{aligned} \tag{10}$$

Algorithm 1 Training of Meta-teacher

input: FAS training set Ψ_t, learning rates β and α, update interval T, momentum update parameter γ, two batch size values M and N.
output: The meta-teacher's weight ω.
1 : Initialize ω and θ by pretraining MT_t and the detector on the training set, initialize $\widehat{\omega}$ to ω.
2 : Iter = 0
3 : while not done **do**
4 : sample $M+N$ live and $M+N$ spoof faces from Ψ_t.
5 : build mini training data batch Φ_t with $2M$ live and spoof faces; build mini validation data batch Φ_v with the other $2N$ faces.
6 : $\mathcal{L}_{\Phi_t}(\theta;\omega) = \frac{1}{2M}\sum_k^{2M} \|f_\theta(x_k) - P_\omega(x_k)\|_2$
7 : $\theta^*(\omega) = \theta - \alpha \cdot \nabla_\theta \mathcal{L}_{\Phi_t}(\theta;\omega)$
8 : $\mathcal{L}_{\Phi_v}(\theta^*(\omega);\widehat{\omega}) = \frac{1}{2N}\sum_k^{2N} \|f_{\theta^*(\omega)}(x_k) - P_{\widehat{\omega}}(x_k)\|_2$
9 : $\omega = \omega + \beta \cdot [\, \alpha \cdot \nabla_\omega \nabla_\theta \mathcal{L}_{\Phi_t}(\theta;\omega) \cdot \nabla_{\theta^*(\omega)} \mathcal{L}_{\Phi_v}(\theta^*(\omega);\widehat{\omega}) + \nabla_\omega \frac{\mu}{2M}\sum_k^{2M} mean(\sigma(g_\omega(x_k)))\,]$
10: $\widehat{\omega} = \gamma \cdot \widehat{\omega} + (1-\gamma)\cdot\omega$
11: **if** Iter **%** $T == 0$ **do**
12: $\theta = \theta^*(\omega)$
13: Iter $\leftarrow$ Iter + 1
14: end

3.4 Other Details of the Meta-Teacher

From Eq. 10, we can see how MT_t is optimized. We present the full training procedure of MT-FAS in Algorithm 1. Here, we discuss some implementation details.

3.4.1 Initialization of MT_t, MT_v, and the detector

Before optimizing MT_t, we first initialize MT_t, MT_v, and the detector to appropriate weights by pretraining them on the training set according to the following reasons: i) randomly initialized MT_t and MT_v output noisy maps for spoof faces; ii) a randomly initialized detector predicts noisy maps for all input faces. These factors may lead to a failure to satisfy the aforementioned premise. In the pretraining step, we train MT_t and the detector to regress all spoof faces as one-maps (each pixel value is one) and to regress live faces as zero-maps. After pretraining of MT_t, we initialize MT_v with the pretrained MT_t. Line 1 of Algorithm 1 shows the pretraining. Experiments in Section 4.4.1 verify the importance of pretraining.

3.4.2 Relationship between MT_t and MT_v

According to Eq. 3, we use another meta-teacher MT_v to calculate the trained detector's validation loss and use the validation loss to represent the meta-teacher MT_t's teaching performance. Therefore, MT_v greatly affects the evaluation of MT_t's teaching performance. An improper MT_v may destroy the premise defined in Section 3.2 and further causes MT_v to mis-evaluate MT_t. For example, assume the worst situation, where the outputs of MT_v are opposite to those

of MT_t. The detector's validation loss evaluated based on MT_v will incorrectly assess MT_t's teaching ability. To avoid this situation, we have manually restricted the predictions of both MT_t and MT_v for live faces to zero-map $\otimes$ and constrained the pixel-wise predictions for spoof faces into the range (0, 2). Nevertheless, we still need to constrain the difference between the outputs of MT_v and MT_t for assessing MT_t's teaching performance more precisely. To this end, in MT-FAS, we use a momentum update manner to update MT_v in every iteration, which can be formulated as

$$\widehat{\omega} = \gamma \cdot \widehat{\omega} + (1 - \gamma) \cdot \omega, \tag{11}$$

where γ is a hyper-parameter that is set to 0.999 by default. We will study the effect of γ on the meta-teacher MT_t's performance in Section 4.4.3. Another benefit of the momentum update manner of MT_v is that in the training process, the optimization of MT_t is probably rough and not smooth. Updating MT_v with Eq. 11 smooths the updating, filters out the noise from MT_t, and further stabilizes the evaluation and training of MT_t.

3.4.3 Updating the detector

According to Section 3.1-3.3, the higher-level learning only optimizes ω without updating θ. However, ignoring updating θ will cause the representation gap between the detector and MT_t to turn larger and larger with the progressively optimizing of MT_t. This may harm the evaluation and optimization of the meta-teacher. Therefore, for the detector to adapt to the updating MT_t, we periodically update θ by assigning it the weight $\theta^*(\omega)$ optimized in the lower-level learning. Lines 11 and 12 of Algorithm 1 describe the periodic updating of the detector with the interval T. T is set to 10 by default. Experiments described in Section 4.4.4 will show the importance of updating the detector.

3.4.4 Avoiding minor prediction

When minimizing $\mathcal{L}_{\Phi_v}(\theta^*(\omega); \widehat{\omega})$, we should avoid the possible meaningless optima where the meta-teacher outputs zero-map $\otimes$ for all faces. This existing of the meaningless optima is caused by 1) we manually let MT_t outputting zero-map $\otimes$ for all live faces; 2) we use momentum updating defined in Eq. 11 to update MT_v. Thus, MT_t may learn to make its prediction map $2\sigma(g_\omega(x_k))$ be closer and closer to zero-map, since the smaller the pixel values of $2\sigma(g_\omega(x_k))$ are, the smaller the pixel-values of MT_v's output $2\sigma(g_{\widehat{\omega}}(x_k))$, and finally the smaller the loss $\mathcal{L}_{\Phi_v}(\theta^*(\omega); \widehat{\omega})$.

To avoid this meaningless optimization, we encourage MT_t and MT_v to output maps $2\sigma(g_\omega(x_k))$ and $2\sigma(g_{\widehat{\omega}}(x_k))$ containing appropriate large pixel values by adding another item $\beta \cdot \nabla_\omega \frac{\mu}{N_t} \sum_k^{N_t} mean(\sigma(g_\omega(x_k)))$ to Eq. 10. Finally, ω is optimized with

$$\begin{aligned} \omega^{'} = \omega + \beta \cdot [\alpha \cdot \nabla_\omega \nabla_\theta \mathcal{L}_{\Phi_t}(\theta; \omega) \cdot \nabla_{\theta^*(\omega)} \mathcal{L}_{\Phi_v}(\theta^*(\omega); \widehat{\omega}) \\ + \nabla_\omega \frac{\mu}{N_t} \sum_k^{N_t} mean(\sigma(g_\omega(x_k)))]. \end{aligned} \tag{12}$$

$mean(\sigma(g_\omega(x_k)))$ is the average value of all pixels in $g_\omega(x_k)$. μ with the default value of 0.001 is a hyper-parameter that controls how strongly we encourage MT_t to output larger map.

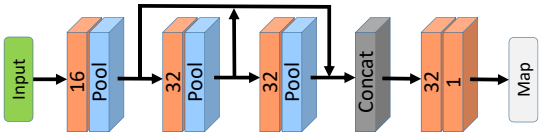


Fig. 4. Network structure of FAS-DR-Light. Each orange cube is the convolution layer and the number on it means the number of filters. The Pool layer is Max-pooling.

3.4.5 Avoiding out-of-memory

The training of the meta-teacher MT_t entails a high memory cost according to Eq. 10 because i) the losses $\mathcal{L}_{\Phi_t}$ and $\mathcal{L}_{\Phi_v}$ are calculated on all the faces from the training set Φ_t and validation set Φ_v, respectively; and ii) second-order gradient is needed to optimize MT_t. We use the following two solutions to reduce memory consumption. First, we employ randomly sampled data batch as Φ_t and use another randomly sampled data batch as Φ_v. Lines 4 and 5 in Algorithm1 describe the data sampling process. According to the data sampling process, we can replace N_t and N_v in Eq. 1, Eq. 3, and Eq. 12 with $2M$ and $2N$, respectively. **Note that both the data batches Φ_t and Φ_v are sampled from the training set, and there are no overlapping examples between Φ_t and Φ_v.**

Second, we set all network structures of MT_t, MT_v, and the detector to a light version of FAS-DR [16] and name this version FAS-DR-Light. The network structure of FAS-DR-Light is shown in Fig. 4. We also show FAS-DR in Fig. 6 for clarity. Note that our goal is to train a meta-teacher MT_t to provide better pixel-wise supervision to train the detector. Thus, the detector here is a surrogate detector because it is only used to assist the meta-teacher's training. After training the meta-teacher MT_t, we use the trained MT_t to supervise the existing state-of-the-art [16], [19], [50], [55] detectors for better FAS performances.

4 Experiments

4.1 Experimental Setup

Performance Metrics. We use the following metrics in our experiment. 1) Attack Presentation Classification Error Rate ($APCER$), which denotes the ratio that spoof faces are misclassified into live faces. 2) Bona Fide Presentation Classification Error Rate ($BPCER$), which denotes the ratio that live faces are misclassified into spoof faces. 3) Average Classification Error Rate ($ACER$) [56], which evaluates the mean of $APCER$ and $BPCER$. 4) Area Under Curve (AUC) , which denotes the area under the Receiver Operating Characteristic (ROC) curve. 5) Half Total Error Rate ($HTER$), which denotes the mean of the False Acceptance Rate (FAR) and False Rejection Rate (FRR) [57].

Experimental Datasets. We evaluate the developed MT-FAS on several popular FAS datasets, including OULU-NPU [58], SiW-M [19], CASIA-MFSD [59], Idiap Replay-Attack [60], and MSU-MFSD [61]. We show some examples of these datasets in Fig. 5 for a better understanding of live and spoof faces.

OULU-NPU [58] is one of the most commonly used FAS dataset. It contains several face capture conditions (six cameras and three sessions) and two kinds of printed

spoof face and two kinds of replayed spoof face. Four protocols are used to evaluate PA detector's performance. Protocols 1, 2, and 3 evaluates the detector's performance on cross-camera, cross-session, cross-spoof-type scenarios, respectively. Protocol 4 is the more challenging because it evaluates the detector on the scenario of simultaneously cross-camera, cross-session, and cross-spoof-type.

SiW-M [19] is a recently proposed zero-shot FAS dataset. It contains 13 kinds of spoof attacks, such as print attack, 3D-Mask attack, Impersonation, Mannequin, and *etc.* 13 leave-one-attack-out sub-protocols are used to evaluate PA detectors' performance on novel spoof types. In each sub-protocol, the test set is formed with a part of live faces and one spoof attack, and the training set is formed with the other live faces and the other 12 spoof attacks.

CASIA-MFSD [59] contains live and spoof faces captured with 50 genuine subjects. Three kinds of attack manners (warped photo-attack, cut photo attack, and video attack) are used to create spoof faces, and each facial image is recorded with three kinds of imaging qualities (low quality, normal quality, and high quality). Therefore, each subject has 3 kinds of live faces captured with different imaging qualities and has 3×3=9 kinds of spoof faces captured with different attack manners and different imaging qualities.

Idiap Replay-Attack [60] captures all live and spoof faces from 50 clients under two different lighting conditions. Five attack manners including four kinds of replayed faces and one kind of printed face are used to capture spoof faces.

MSU-MFSD [61] uses two different cameras to record all live and spoof faces from 35 genuine subjects. Three kinds of spoof faces are considered, including two kinds of replayed faces and one kind of printed face. Therefore, each subject has 2 kinds of live faces and has 2×3=6 kinds of spoof faces captured with the two cameras.

Cross-domain FAS is a popular problem for practical FAS deployment. A domain-generalization benchmark [62] is commonly used to evaluate PA detector on this problem. This benchmark contains four datasets (OULU-NPU, CASIA-MFSD, Idiap Replay-Attack, and MSU-MFSD) and four cross-domain protocols. Each protocol uses one dataset as the testing domain while the other three datasets as the training domain.

Hyper-parameter Setup. We train MT_t for 30,000 iterations (20,000 pretraining (Line 1 of Algorithm 1) + 10,000 bi-level-training iterations (Lines 4-14 of Algorithm 1)). The two hyper-parameters M and N in Algorithm 1 are set to 20 and 10, respectively. Both the learning rates β and α are set to 0.001. On each mini training data batch Φ_t, the meta-teacher MT_t supervises the surrogate detector to do two gradient descent steps. μ, γ, and T are set to 0.001, 0.999, and 10, respectively. Both the network structures of MT_t, MT_v is FAS-DR-Light illustrated in Fig. 4. FAS-DR-Light's output for each face is a single-channel pixel-wise map with 32×32 resolution. All facial images used in this work are RGB images with 256×256 resolution. After training the meta-teacher MT_t, we use Eq. 13 as the final pixel-wise supervision to train PA detectors.

$$P_\omega(x_k)=\begin{cases}\otimes, & if\ x_k \in live\\ \frac{\sigma(g_\omega(x_k))-min(\sigma(g_\omega(x_k)))}{max(\sigma(g_\omega(x_k)))-min(\sigma(g_\omega(x_k)))}, & if\ x_k \in spoof\end{cases} \tag{13}$$

TABLE 1
Experimental Results on OULU-NPU [58].

Protocol	Method	APCER(%)	BPCER(%)	ACER(%)
1	GRADIANT [63]	1.3	12.5	6.9
	STASN [64]	1.2	2.5	1.9
	Auxiliary [14]	1.6	1.6	1.6
	FaceDs [65]	1.2	1.7	1.5
	FAS-SGTD [15]	2.0	0.0	1.0
	Disentangled [47]	1.7	0.8	1.3
	DeepPixBiS [20]	0.8	0.0	**0.4**
	FAS-DR(Depth)	0.7	2.3	1.5
	FAS-DR(MT)	0.0	1.2	0.6
	CDCN(Depth) [55]	0.4	1.7	1.0
	CDCN(MT)	0.0	0.8	**0.4**
2	DeepPixBiS [20]	11.4	0.6	6.0
	FaceDs [65]	4.2	4.4	4.3
	Auxiliary [14]	2.7	2.7	2.7
	GRADIANT [63]	3.1	1.9	2.5
	STASN [64]	4.2	0.3	2.2
	FAS-SGTD [15]	2.5	1.3	1.9
	Disentangled [47]	1.1	3.6	2.4
	FAS-DR(Depth)	1.6	3.4	2.5
	FAS-DR(MT)	0.9	2.7	1.8
	CDCN(Depth) [55]	1.5	1.4	1.5
	CDCN(MT)	1.3	1.4	**1.4**
3	DeepPixBiS [20]	11.7±19.6	10.6±14.1	11.1±9.4
	GRADIANT [63]	2.6±3.9	5.0±5.3	3.8±2.4
	FaceDs [65]	4.0±1.8	3.8±1.2	3.6±1.6
	Auxiliary [14]	2.7±1.3	3.1±1.7	2.9±1.5
	STASN [64]	4.7±3.9	0.9±1.2	2.8±1.6
	FAS-SGTD [15]	3.2±2.0	2.2±1.4	2.7±0.6
	Disentangled [47]	2.8±2.2	1.7±2.6	2.2±2.2
	FAS-DR(Depth)	1.9±1.4	5.8±7.5	3.8±3.5
	FAS-DR(MT)	1.0±0.8	3.8±4.1	2.4±2.1
	CDCN(Depth) [55]	2.4±1.3	2.2±2.0	2.3±1.4
	CDCN(MT)	2.3±1.5	1.9±1.8	**2.1±1.7**
4	DeepPixBiS [20]	36.7±29.7	13.3±14.1	25.0±12.7
	GRADIANT [63]	5.0±4.5	15.0±7.1	10.0±5.0
	Auxiliary [14]	9.3±5.6	10.4±6.0	9.5±6.0
	STASN [64]	6.7±10.6	8.3±8.4	7.5±4.7
	FaceDs [65]	1.2±6.3	6.1±5.1	5.6±5.7
	FAS-SGTD [15]	6.7±7.5	3.3±4.1	5.0±2.2
	Disentangled [47]	5.4±2.9	3.3±6.0	4.4±3.0
	FAS-DR(Depth)	5.4±4.9	8.2±7.8	6.8±5.2
	FAS-DR(MT)	2.0±2.2	6.6±5.7	4.3±4.0
	CDCN(Depth) [55]	4.6±4.6	9.2±8.0	6.9±2.9
	CDCN(MT)	0.9±2.0	6.4±4.9	**3.7±2.9**

Detector Nomenclature. Our work aims to train a meta-teacher providing better pixel-wise supervision for PA detectors. To evaluate the trained meta-teacher, we use the meta-teacher to supervise the existing PA detectors' training. We call the detector that is supervised by the meta-teacher Detector(MT). For example, we call CDCN [55] supervised by the meta-teacher CDCN(MT). For comparing the meta-teacher with existing handcrafted pixel-wise supervision, we also perform experiments in which we use existing handcrafted pixel-wise supervisions to train existing detectors. We name these trained detectors using the fashion of Detector(Label). For example, FAS-DR(Depth) denotes the detector FAS-DR [16] supervised by facial depth label.

4.2 Comparison with Handcrafted Labels

In this subsection, we verify the advantage of the proposed MT-FAS by comparing the performances of the detectors supervised by the meta-teacher with those of the counterpart detectors supervised by handcrafted labels.

Fig. 5. Examples from OULU-NPU [58], SiW-M [19], CASIA-MFSD [59], Idiap Replay-Attack [60], and MSU-MFSD [61]. In each dataset, the faces in the leftmost column are live faces while all the other faces are spoof faces.

TABLE 2
Experimental Results on SiW-M [19] with leave-one-attack-out protocol.

Method	Metrics(%)	Replay	Print	Mask Attacks					Makeup Attacks			Partial Attacks			Average
				Half	Silicone	Trans.	Paper	Manne.	Obfusc.	Imperson.	Cosmetic	Funny Eye	Paper Glasses	Partial Paper	
SVM+LBP [58]	APCER	19.1	15.4	40.8	20.3	70.3	0.0	4.6	96.9	35.3	11.3	53.3	58.5	0.6	32.8±29.8
	BPCER	22.1	21.5	21.9	21.4	20.7	23.1	22.9	21.7	12.5	22.2	18.4	20.0	22.9	21.0±2.9
	ACER	20.6	18.4	31.3	21.4	45.5	11.6	13.8	59.3	23.9	16.7	35.9	39.2	11.7	26.9±14.5
	EER	20.8	18.6	36.3	21.4	37.2	7.5	14.1	51.2	19.8	16.1	34.4	33.0	7.9	24.5±12.9
Auxiliary [14]	APCER	23.7	7.3	27.7	18.2	97.8	8.3	16.2	100.0	18.0	16.3	91.8	72.2	0.4	38.3±37.4
	BPCER	10.1	6.5	10.9	11.6	6.2	7.8	9.3	11.6	9.3	7.1	6.2	8.8	10.3	8.9±2.0
	ACER	16.8	6.9	19.3	14.9	52.1	8.0	12.8	55.8	13.7	11.7	49.0	40.5	5.3	23.6±18.5
	EER	14.0	4.3	11.6	12.4	24.6	7.8	10.0	72.3	10.1	**9.4**	21.4	18.6	4.0	17.0±17.7
CDCN++ [55]	APCER	9.2	6.0	4.2	7.4	18.2	0.0	5.0	39.1	0.0	14.0	23.3	14.3	0.0	10.8±11.2
	BPCER	12.4	8.5	14.0	13.2	19.4	7.0	6.2	45.0	1.6	14.0	24.8	20.9	3.9	14.6±11.4
	ACER	10.8	7.3	9.1	10.3	18.8	3.5	5.6	42.1	0.8	14.0	24.0	17.6	1.9	12.7±11.2
	EER	9.2	5.6	**4.2**	11.1	19.3	5.9	5.0	43.5	**0.0**	14.0	23.3	**14.3**	**0.0**	11.9±11.8
BCN [18]	APCER	12.4	5.2	8.3	9.7	13.6	0.0	2.5	30.4	0.0	12.0	22.6	15.9	1.2	10.3±9.1
	BPCER	13.2	6.2	13.1	10.8	16.3	3.9	2.3	34.1	1.6	13.9	23.2	17.1	2.3	12.2±9.4
	ACER	12.8	5.7	10.7	10.3	14.9	**1.9**	**2.4**	**32.3**	0.8	12.9	22.9	**16.5**	1.7	11.2±9.2
	EER	13.4	5.2	8.3	9.7	13.6	5.8	**2.5**	**33.8**	**0.0**	14.0	23.3	16.6	1.2	11.3±9.5
STDN [66]	APCER	1.6	0.0	0.5	7.2	9.7	0.5	0.0	96.1	0.0	21.8	14.4	6.5	0.0	12.2±26.1
	BPCER	14.0	14.6	13.6	18.6	18.1	8.1	13.4	10.3	9.2	17.2	27.0	35.5	11.2	16.2±7.6
	ACER	7.8	7.3	**7.1**	12.9	13.9	4.3	6.7	53.2	4.6	19.5	20.7	21.0	5.6	14.2±13.2
	EER	**7.6**	**3.8**	8.4	13.8	14.5	5.3	4.4	35.4	**0.0**	19.3	21.0	20.8	1.6	12.0±10.0
DTN [19]	APCER	1.0	0.0	0.7	24.5	58.6	0.5	3.8	73.2	13.2	12.4	17.0	17.0	0.2	17.1±23.3
	BPCER	18.6	11.9	29.3	12.8	13.4	8.5	23.0	11.5	9.6	16.0	21.5	22.6	16.8	16.6±6.2
	ACER	9.8	6.0	15.0	18.7	36.0	4.5	7.7	48.1	11.4	14.2	19.3	19.8	8.5	16.8±11.1
	EER	10.0	2.1	14.4	18.6	26.5	5.7	9.6	50.2	10.1	13.2	19.8	20.5	8.8	16.1±12.2
DTN(MT)	APCER	6.1	5.2	8.3	14.8	24.3	5.9	5.0	39.4	5.1	10.0	17.1	19.8	1.1	12.5±10.2
	BPCER	10.9	10.1	17.8	18.6	16.9	0.0	6.2	28.9	2.4	14.7	20.9	21.7	6.7	13.5±8.0
	ACER	9.5	7.6	13.1	16.7	20.6	2.9	5.6	34.2	3.8	12.4	**19.0**	20.8	3.9	13.1±8.7
	EER	9.1	7.8	14.5	14.1	18.7	3.6	6.9	35.2	3.2	11.3	18.1	17.9	3.5	12.6±8.5
FAS-DR(Depth)	APCER	9.3	6.0	15.9	11.9	19.6	7.2	8.9	38.2	2.9	14.9	20.0	20.1	1.1	13.5±9.4
	BPCER	6.3	5.8	10.9	11.6	15.2	3.5	6.0	39.7	1.8	10.4	19.3	16.7	3.8	11.6±9.6
	ACER	7.8	5.9	13.4	11.7	17.4	5.4	7.4	39.0	2.3	12.6	19.6	18.4	2.4	12.6±9.5
	EER	8.0	4.9	10.8	10.2	14.3	3.9	8.6	45.8	1.0	13.3	**16.1**	15.6	1.2	11.8±11.0
FAS-DR(MT)	APCER	4.1	4.3	6.5	3.7	9.2	5.9	5.0	36.4	0.0	10.0	20.3	17.5	0.0	9.5±9.7
	BPCER	8.5	5.4	12.0	10.9	14.7	0.8	1.6	42.6	0.4	10.9	21.7	19.4	2.2	11.6±11.1
	ACER	**6.3**	**4.9**	9.3	**7.3**	**12.0**	3.3	3.3	39.5	**0.2**	**10.4**	21.0	18.4	**1.1**	**10.5±10.3**
	EER	7.8	4.4	11.2	**5.8**	**11.2**	**2.8**	2.7	38.9	0.2	10.1	20.5	18.9	1.3	**10.4±10.2**

4.2.1 Experiment on OULU-NPU

Here, we evaluate MT-FAS on OULU-NPU [58]. First, we train the meta-teacher MT_t on protocol 1. Second, on all protocols, we use the trained MT_t to supervise the learning of two existing PA detectors (FAS-DR and CDCN [55]). The detailed network structure of FAS-DR [16] is shown in Fig. 6. FAS-DR uses four convolution neural network-based blocks to regress the input face as a pixel-wise map. The bottom three blocks extract low-level, middle-level, and high-level image features. The features are then concatenated and fed into the top block to regress the pixel-wise map. CDCN processes the input face in a similar way. The most difference between FAS-DR and CDCN is that FAS-DR uses vanilla convolution layers to process images while CDCN uses central difference-based convolution layers.

We denote the two trained detectors as FAS-DR(MT) and CDCN(MT) and report their performances in Table 1. Both of them perform well on all protocols, especially CDCN(MT). Compared with other existing PA detectors, CDCN(MT) achieves the best performances on all protocols. For instance, CDCN(MT) decreases the $ACER$ by approximately 16% on protocol 4.

Note that, since the official CDCN is trained with facial depth supervision, we denote it as CDCN(Depth) for clarity. Moreover, we use facial depth label to train another detector FAS-DR(depth). FAS-DR(depth) and CDCN(Depth) can be treated as baselines for FAS-DR(MT) and CDCN(MT), respectively. The comparison between FAS-DR(depth) and FAS-DR(MT), and the comparison between CDCN(depth) and CDCN(MT) can verify the advantage of meta-teacher over the handcrafted facial depth label in supervising the two PA detectors. The experimental results reported in Table 1 show that on protocols 1-4, CDCN(MT) and FAS-DR(MT) achieve lower $ACER$ than CDCN(Depth) and FAS-DR(depth), respectively. For instance, compared with FAS-DR(depth), FAS-DR(MT) decreases $ACER$ by approximately 60% on protocol 1. The comparisons demonstrate that compared with handcrafted facial depth label, the meta-teacher supervises the two detectors (FAS-DR and CDCN) more accurately.

4.2.2 Experiment on SiW-M

On each protocol of SiW-M [19], we first train the meta-teacher MT_t and then use the trained meta-teacher to train the FAS-DR detector. We report the experimental results in Table 2. FAS-DR(MT) outperforms the other state-of-the-art methods by a large margin. One highlight is that compared with DTN [19], FAS-DR(MT) decreases $ACER$ by approximately 37.5%.

TABLE 3
Experimental Results on the Domain-generalization Benchmark. The Metrics Used in This Experiment are $HTER$ and AUC.

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(%)
MS_LBP [67]	29.76	78.50	54.28	44.98	50.30	51.64	50.29	49.31
CNN [68]	29.25	82.87	34.88	71.95	34.47	65.88	29.61	77.54
IDA [69]	66.67	27.86	55.17	39.05	28.35	78.25	54.20	44.59
LBPTOP [70]	36.90	70.80	42.60	61.05	49.45	49.54	53.15	44.09
Color Texture [71]	28.09	78.47	30.58	76.89	40.40	62.78	63.59	32.71
Auxiliary(Depth only) [14]	22.72	85.88	33.52	73.15	29.14	71.69	30.17	66.61
MMD-AAE [72]	27.08	83.19	44.59	58.29	31.58	75.18	40.98	63.08
MADDG [73]	17.69	88.06	24.5	84.51	22.19	84.99	27.98	80.02
DR_MD [74]	17.02	90.10	19.68	87.43	20.87	86.72	25.02	81.47
MA-Net [49]	20.80	/	25.60	/	24.70	/	26.30	/
RFMetaFAS [50]	13.89	93.98	20.27	88.16	17.30	90.48	16.45	91.16
RFMetaFAS*	11.90	94.65	24.76	84.29	18.89	88.71	21.83	85.56
RFMetaFAS(MT)*	12.31	**94.89**	22.91	85.63	12.77	94.02	18.16	89.40
FAS-DR-BC(Depth)	13.81	91.61	19.67	89.36	19.14	87.85	19.56	88.28
FAS-DR-BC(MT)	**11.67**	93.09	**18.44**	**89.67**	**11.93**	**94.95**	**16.23**	**91.18**

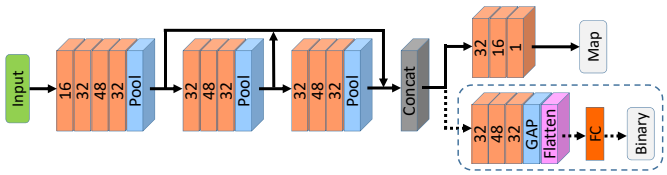


Fig. 6. Network structure of FAS-DR and FAS-DR-BC. FAS-DR is the network without the layers in the dashed box while FAS-DR-BC contains all layers in this figure. Each orange cube is the convolution layer and the number on it means the number of filters. The convolution layers in the dashed box use 2x2 stride while the other convolution layers use 1x1 stride. The Pool layer is Max-pooling with 2x2 stride and GAP is the global average pooling.

To compare the supervision of the meta-teacher MT_t with that of facial depth and pixel-wise binary mask labels, we further train the detector FAS-DR with facial depth label and train the detector DTN with the meta-teacher. The corresponding experimental results are shown in Table 2. Compared with FAS-DR(Depth), FAS-DR(MT) decreases the average $ACER$ by approximately 17%, and compared with DTN, DTN(MT) decreases the average $ACER$ by approximately 22%. As the official DTN is trained using pixel-wise binary mask label (shown in Fig. 1), the comparison between the official DTN and DTN(MT) vividly reveals the advantage of the trained meta-teacher MT_t over handcrafted pixel-wise binary mask label. The comparison between the official FAS-DR(Depth) and FAS-DR(MT) further reveals the advantage of the meta-teacher over facial depth supervision.

The possible underlining reasons why the meta-teacher outperforms handcrafted pixel-wise regression labels in supervising PA detectors are 1) the handcrafted labels are not the most suitable labels for the FAS problem; and 2) the meta-teacher is task-oriented trained to produce more suitable pixel-wise labels for FAS.

4.2.3 *Experiment on domain-generalization benchmark*

In the domain-generalization benchmark [62], four protocols are utilized to evaluate the PA detector's domain-generalization performance. In this experiment, we revise the FAS-DR network by adding a binary classification branch to it. We name the revised network FAS-DR-BC and illustrate its structure in Fig. 6. When testing, the prediction of FAS-DR-BC is the average score of the pixel-wise regression map and binary classification branches. Specifically, $\hat{y} = 0.5 * mean(\hat{y}_{map}) + 0.5 * \hat{y}_{binary}$, where $\hat{y}_{map}$ is the map regressed by the regression branch and $\hat{y}_{binary}$ is the score predicted by the classification branch. The score $\hat{y}_{binary}$ denotes the probability that the input face is a spoof face.

On each protocol, we first train the meta-teacher on the corresponding training set and then use the meta-teacher to supervise FAS-DR-BC's regression output. We further assess the meta-teacher by using it to supervise the recent state-of-the-art detector RFMetaFAS [50]. The official RFMetaFAS is trained using both facial depth label and binary classification label. In this experiment, we replace the facial depth supervision with the trained meta-teacher and keep all the other settings the same as those in the official RFMetaFAS.

The experimental results are shown in Table 3. O, C, I, and M denote OULU-NPU [58], CASIA-MFSD [59], Idiap Replay-Attack [60], and MSU-MFSD [61], respectively. "I&C&M to O" denotes the protocol where the PA detector is trained on I, C, and M, and tested on O; the same explanation holds for the other protocols.

In this experiment, FAS-DR-BC(MT) achieves state-of-the-art performances in most cases. Compared with FAS-DR-BC(Depth), FAS-DR-BC(MT) decreases $HTER$ by approximately 15%, 6%, 38%, and 17%, on the four protocols. Besides, RFMetaFAS(MT)* also outperforms RFMetaFAS*. Note that RFMetaFAS* is our reimplementation of RFMetaFAS using the published official code with carefully tuned hyper-parameters.

This experiment demonstrates 1) the meta-teacher's superiority over facial depth supervision when training PA detectors, 2) the generalization of the MT-FAS method on the domain-generalization benchmark, and 3) the meta-teacher's generalizability on supervising different PA detectors.

4.3 Comparison with existing teacher-student methods

In this subsection, we evaluate whether the proposed MT-FAS outperforms existing teacher-student methods in improving PA detectors' learning. The compared teacher-student methods include Distill [24], BANs [26], and Fitnet [25].

TABLE 4
Comparison Between the Meta-teacher and Other Teachers on Protocol 4 of OULU-NPU [58]. Time$_T$ denotes the total training time (hours) of the teacher.

Method	ACER(%)	FLOPs$_T$	Time$_T$	FLOPs$_S$
FAS-DR(Depth)	6.8±5.2	/	/	280G
FAS-DR(Distill) [24]	5.7±3.6	453G	13.6H	431G
FAS-DR(BANs) [26]	6.1±5.4	280G	9.0H	379G
FAS-DR(Fitnet) [25]	5.5±3.9	453G	13.6H	436G
FAS-DR(Distill_Light)	7.5±4.4	5.3G	1.5H	283G
FAS-DR(OKDDip) [31]	5.4±3.6	/	/	862G
FAS-DR(MT)	**4.3±4.0**	781G	8.9H	283G

We implement the compared teachers on FAS, and use the pixel-wise binary label to train them regressing live faces as zero-maps and regressing spoof faces as one-maps. For the Distill and Fitnet teachers, we set their network to be deeper than FAS-DR by repeating each convolution layer containing 48 filters for 2 times in FAS-DR. For the BANs teacher, we set its network to be the same as FAS-DR. As our meta-teacher uses a shallow backbone FAS-DR-Light, another light version teacher Distill-Light which also uses the FAS-DR-Light backbone is trained as the baseline of our meta-teacher.

For all the compared teachers, we train them on the training set for 10 epochs with the learning rate of 0.001. Adam is selected as the optimizer. After training the teachers, we apply them to supervise the FAS-DR detector's training. Note that to ensure a fair comparison between the proposed meta-teacher and the compared teachers, we modify the compared teachers' output with Eq. 14 when using them to supervise the FAS-DR detector. In Eq. 14, x is the input face image, and $\otimes$ is zero-map. ω is the teacher's weight and $g_\omega(x)$ is the teacher's output for spoof faces.

$$T_\omega(x) = \begin{cases} \otimes, & if\ x \in live, \\ \frac{g_\omega(x) - min(g_\omega(x))}{max(g_\omega(x)) - min(g_\omega(x))}, & if\ x \in spoof \end{cases} \quad (14)$$

With this modification, these teachers output pixel-wise predictions for spoof faces and output zero-maps for all live faces, which is consistent with the predictions of the proposed meta-teacher. We denote the trained detectors supervised by these teachers with the nomenclature fashion of Detector(Teacher). For example, FAS-DR(BANs) denotes the FAS-DR detector supervised by the BANs teacher.

OKDDip [31], a teacher-free online knowledge-distillation method, is also considered as a baseline to MT-FAS. OKDDip simultaneously uses m (3 in our re-implementation) students where $m-1$ are auxiliary peers and one is group-leader to do knowledge distillation. It trains each auxiliary peer in the first-level distillation and uses the ensemble of auxiliary peers together with ground-truth to supervise the group-leader in the second-level distillation. We treat the ensemble of auxiliary peers as another 'teacher' because it supervises the group-leader.

4.3.1 Experiment on OULU-NPU

As protocol 4 is the most challenging protocol in OULU-NPU, we implement the compared teachers on protocol 4. Table 4 reports the performances of FAS-DR trained using the compared teachers. Note that the network of all the students in OKDDip is set to FAS-DR and FAS-DR(OKDDip) denotes the trained group-leader. The experimental results show that meta-teacher outperforms the compared teachers in supervising FAS-DR's learning. Compared with the other teachers, the meta-teacher decreases FAS-DR's $ACER$ by at least 20%.

We also compare the training costs of all teachers in Table 4. FLOPs$_T$ and FLOPs$_S$ denote the training cost of the teacher model and the student model, respectively, in each training iteration. Time$_T$ denotes the total training time (hours) of the teacher. Note that all teachers are trained on one NVIDIA Tesla P40 GPU. FAS-DR(Depth) is also listed in Table 4 as a baseline. As FAS-DR(Depth) is the detector FAS-DR trained using facial depth label without using teacher model, both its FLOPs$_T$ and Time$_T$ are zero (denoted as /). Both FLOPSs$_T$ and Time$_T$ of OKDDip [31] are zero too because OKDDip is an online knowledge distillation method that simultaneously trains all models without separately training the teacher and student. Therefore all FLOPs of OKDDip in each training iteration are summed into FLOPs$_S$.

Table 4 shows that compared with the other teachers, the proposed MT-FAS costs more FLOPs to train the meta-teacher in each bi-level training iteration. The reason is that MT-FAS needs to calculate the second-order gradient shown in Eq. 12 to optimize the meta-teacher. But overall, compared with the other teachers, the meta-teacher does not cost more training time. This is because the meta-teacher needs fewer training iterations than the other methods to converge. The compared teachers cost about 40,000 iterations to converge while the meta-teacher needs fewer iterations (20,000 pretraining iterations (Line 1 of Algorithm 1) + 10,000 bi-level optimization iterations (Lines 4-14 of Algorithm 1)).

Table 4 does not show the training time of each detector (student) because we use the same training iterations to train all detectors. In other words, the training cost of each detector is mainly reflected by FLOPs$_S$. We can see that the training of FAS-DR(MT) costs fewer FLOPs than most of the other detectors including FAS-DR(Distill), FAS-DR(BANs), and *etc.* The reason is that benefiting from the light-weight network FAS-DR-Light, the meta-teacher costs fewer computation resources than most of the other compared teachers in inference.

4.3.2 Experiment on domain-generalization benchmark

On the domain-generalization benchmark, we utilize the compared teachers to supervise the detector FAS-DR-BC's regression prediction. Note that the teacher of OKDDip means the ensemble of auxiliary peers. The network of each auxiliary peer in OKDDip is set to FAS-DR and the network of the group-leader is set to FAS-DR-BC in this experiment. We use the ensemble of auxiliary peers together with the pixel-wise binary label to supervise the group-leader's regression prediction in the second-level distillation of OKDDip. Table 5 reports the corresponding experimental results of FAS-DR-BC supervised by these teachers. Obviously, FAS-DR-BC(MT) outperforms the other trained FAS-DR-BC counterparts, which verifies the advantage of the proposed MT-FAS over the teacher-student methods and OKDDip.

TABLE 5
Comparison Between the Meta-teacher and Other Teachers on the Domain-generalization Benchmark.

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(%)
FAS-DR-BC(Depth)	13.81	91.61	19.67	89.36	19.14	87.85	19.56	88.28
FAS-DR-BC(Distill) [24]	15.24	90.46	23.22	86.71	23.86	83.65	17.60	89.51
FAS-DR-BC(BANs) [26]	**11.31**	**93.57**	19.12	88.43	24.31	78.29	18.53	89.17
FAS-DR-BC(Fitnet) [25]	12.53	91.79	18.65	89.58	21.14	85.17	17.19	90.25
FAS-DR-BC(Distill_Light)	11.90	92.86	20.03	86.48	25.65	79.10	18.30	88.76
FAS-DR-BC(OKDDip) [31]	12.25	92.36	19.81	88.52	19.75	86.29	17.22	90.06
FAS-DR-BC(MT)	11.67	93.09	**18.44**	**89.67**	**11.93**	**94.95**	**16.23**	**91.18**

All the aforementioned experiments validate that the proposed meta-teacher outperforms not only the widely employed human-designed labels but also existing teachers, in supervising PA detectors. The possible reason for these experimental results is that existing teachers are trained to match the training data but not to improve their teaching ability. In contrast, the proposed meta-teacher is trained to learn how to supervise the student to perform better.

4.4 Ablation Study

In this subsection, we evaluate how crucial components or settings affect the meta-teacher's performance. All ablation experiments are conducted with FAS-DR on protocol 1 of OULU-NPU.

4.4.1 *Effect of pretraining*

In our implementation of MT-FAS, before optimizing the meta-teacher MT_t, we initialize MT_t, MT_v, and the surrogate detector by pretraining them on the training set. In this experiment, we optimize MT_t from scratch without pretraining. We denote the meta-teacher trained without pretraining as MT_w/o_pre and denote the trained FAS-DR detector supervised by MT_w/o_pre as FAS-DR(MT_w/o_pre). The corresponding experimental result is shown in Table 6. Without pretraining, the detector FAS-DR's $ACER$ rises significantly from 0.6% to 6.3%, which reveals the importance of pretraining for the meta-teacher.

4.4.2 *Indispensability of MT_v*

When training the meta-teacher MT_t, as shown in Eq. 3, we use another MT_v to evaluate MT_t's teaching quality. In this ablation experiment, we verify whether MT_v is indispensable to evaluate MT_t. In other words, if we use MT_t to calculate $\mathcal{L}_{\Phi_v}$ (realized by copying ω to $\widehat{\omega}$ at every training iteration), then can we still stably optimize MT_t with Eq. 12?

We denote the meta-learner trained without MT_v as MT_w/o_MT_v and denote the trained detector supervised by MT_w/o_MT_v as FAS-DR(MT_w/o_MT_v). The experimental result shown in Table 6 apparently indicates that MT_v is indispensable for evaluating MT_t's teaching quality. Without MT_v, the FAS-DR detector's $ACER$ greatly deteriorates from 0.6% to 7.1%.

4.4.3 *Momentum update hyper-parameter γ*

In this ablation experiment, we aim to verify how MT_v's momentum update hyper-parameter γ affects the trained meta-teacher MT_t. γ is set to 0.999 by default in this work.

TABLE 6
Ablation Experimental Results on Protocol 1 of OULU-NPU [58].

Method	APCER(%)	BPCER(%)	ACER(%)
FAS-DR(MT_w/o_MT_v)	0.4	13.8	7.1
FAS-DR(MT_w/o_pre)	1.0	11.5	6.3
FAS-DR(MT_w/o_adapt)	2.8	5.6	4.2
FAS-DR(Depth)	0.7	2.3	1.5
FAS-DR(MT_resnet)	0.2	1.4	0.8
FAS-DR(MT)$_{64}$	0.2	1.2	0.7
FAS-DR(MT)$_{32}$	0.4	0.8	**0.6**
FAS-DR(MT)	0.0	1.2	**0.6**

Fig. 7. The impact of the parameter γ towards the meta-teacher and the trained FAS-DR detector. The x-axis denotes the value of γ and the y-axis denotes $ACER$ on protocol 1 of OULU-NPU.

Here, we set γ to other values of 0.9, 0.99, 0.995, 0.9995, 0.9999, and 1.0. According to Eq. 11, the larger γ is, the slower the update of MT_v. When $\gamma = 1.0$, MT_v will be frozen in the optimization procedure of MT_t. $\gamma = 0.9$, 0.99, and 0.995 will update MT_v faster than $\gamma = 0.999$.

Fig. 7 shows how γ affects the performance of the meta-teacher and further affects the trained detector. When $\gamma = 0.999$, the detector achieves the best performance with the lowest $ACER$. Either smaller or larger γ damages the meta-teacher and consequently decreases the FAS-DR detector's performance. For instance, compared with $\gamma = 0.999$, $\gamma = 0.9$ harms the meta-teacher and greatly rises FAS-DR(MT)'s $ACER$ to 5.3%. This experiment indicates that updating MT_v either too slow or too fast is unfriendly for the meta-teacher MT_t's training.

4.4.4 *Adapt the surrogate detector to the changing MT_t*

Within the bi-level optimizing progress of the meta-teacher MT_t, we also update the surrogate PA detector to adapt it to the changing MT_t. Line 12 in Algorithm 1 shows that we use the weight $\theta^*(\omega)$ to update the surrogate PA detector's weight θ. In this ablation experiment, we remove

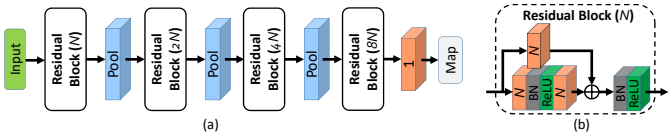


Fig. 8. (a) The architecture of the ResNet-8 backbone. (b) The inner structure of the residual block. Pool: Maxpooling with stride of 2 and pooling size of 2. Orange cube: convolution layer. BN: batch-normalization. Note that the number on orange cube denotes the number of filters of the convolution layer. The convolution layer on the shortcut path uses 1×1 stride while all the other convolution layers use 3×3 stride. N is set to 16 in this work.

Line 12 of Algorithm1 to evaluate whether the update of the surrogate PA detector is necessary to improve the meta-teacher's optimization. We denote the newly trained meta-teacher as MT_w/o_adapt and denote the FAS-DR detector supervised by MT_w/o_adapt as FAS-DR(MT_w/o_adapt).

Table 6 reports the performance of FAS-DR(MT_w/o_adapt). Clearly, FAS-DR(MT) outperforms FAS-DR(MT_w/o_adapt), revealing the importance of adapting the surrogate PA detector to the changing MT_t. One possible underlying reason is that ignoring updating the surrogate detector (without Line 12 of Algorithm 1) results in a larger representation gap between MT_t and the surrogate detector in the meta-teacher MT_t's training procedure. An excessively large representation gap between MT_t and the surrogate detector hinders MT_t from effectively teaching the surrogate detector and results in incorrect evaluation and optimization of MT_t.

4.4.5 Backbone of MT_t

We set the backbone of MT_t to FAS-DR-Light by default. In this experiment, we verify whether the meta-teacher is generalizable to other backbones by replacing its backbone with another backbone called ResNet-8. The architecture of ResNet-8 is shown in Fig. 8. It uses four residual blocks [75] to extract features and one convolution layer to regress the map with 32x32 resolution.

We denote the meta-teacher using ResNet-8 backbone as MT_resnet and denote the FAS-DR detector supervised by MT_resnet as FAS-DR(MT_resnet). When training MT_resnet, we keep all the other experimental settings the same as those of the default meta-teacher. Table 6 reports the performance of FAS-DR(MT_resnet). Although FAS-DR(MT_resnet) performs slightly worse than FAS-DR(MT), it is still evident that FAS-DR(MT_resnet) outperforms FAS-DR(Depth). Compared with FAS-DR(Depth), FAS-DR(MT_resnet) decreases $ACER$ by approximately 47%, validating the generalizability of the proposed MT-FAS method across different backbones.

4.4.6 Supervising different PA detectors

Tables 1, 2, and 3 demonstrate that the meta-teacher provides superior pixel-wise supervision and improves the performances of different PA detectors including FAS-DR [16], CDCN [55], DTN [19], and RFMetaFAS [50]. Here we further use the meta-teacher to supervise the other two modified FAS-DR backbones who have different numbers of parameters with the default FAS-DR. The default FAS-DR backbone used in the work is shown in Fig. 6. Its first (the bottom) convolution layer contains 16 filters. In this experiment, except for the last (the topmost) layer, we modify the FAS-DR backbone by doubling or quadrupling the number of filters of all the other convolution layers. We denote the new FAS-DR backbone as FAS-DR$_{32}$ or FAS-DR$_{64}$. Either FAS-DR$_{32}$ or FAS-DR$_{64}$ has much more parameters than the default FAS-DR backbone. We use the meta-teacher to train them and denote the trained detectors FAS-DR$_{32}$(MT) and FAS-DR$_{64}$(MT). The corresponding experimental results shown in Table 6 demonstrate that the meta-teacher's teaching performance is insensitive to the number of parameters in PA detectors.

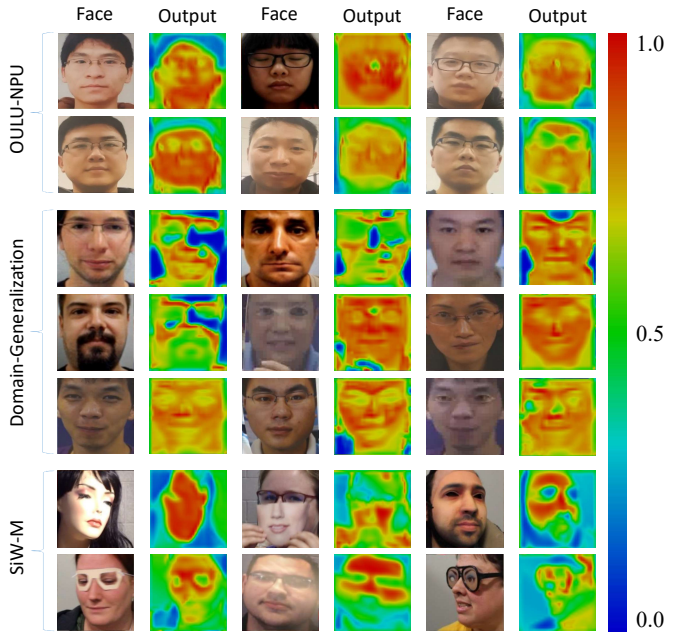


Fig. 9. The outputs of the meta-teacher for spoof faces from the three benchmarks. The colors ranging from blue to red denote the float values from zero to one.

4.5 Visualization

4.5.1 Prediction of the meta-teacher

Fig. 9 visualizes the predictions of the meta-teacher for spoof faces in all three benchmarks. The visualization shows that the predictions of the meta-teacher contain facial structure and spoofing cues, especially on SiW-M. For example, the prediction for the face in the 6-th row and 3-rd column focuses greatly on the partial mask. The prediction for the face in the 7-th row and 5-th column pays more attention to the camouflage glasses, a particular spoof category in SiW-M. Fig. 9 demonstrates that the meta-teacher provides PA detectors with effective pixel-wise supervision that contains rich and intrinsic spoofing cues. This finding may explain why the detectors supervised by our meta-teacher outperforms existing vanilla detectors.

4.5.2 Prediction of ablated meta-teachers

We also want to know the outputs of the aforementioned ablated meta-teachers. Fig. 10 visualizes these meta-teachers' outputs for spoof faces from OULU-NPU. We note that MT_w/o_MT_v, MT_w/o_pre, MT$_{\gamma=0.99}$, and

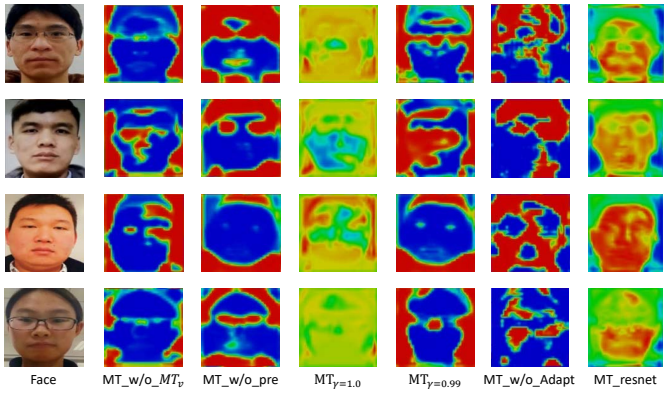


Fig. 10. The ablation meta-teachers' predictions for spoof faces from OULU-NPU. The colors ranging from blue to red denote the float values from zero to one.

MT_w/o_adapt seem to overfit to some facial regions, which may explain why they could not supervise the detector learning the most precise spoofing cues. Setting γ to 1.0 will freeze MT_v when training MT_t. Therefore, $\text{MT}_{\gamma=1.0}$ may have difficulty in thoroughly learn the teaching ability. Among all the ablated meta-teachers shown in Fig. 10, the output map of MT_resnet contains the richest information. This may explain why MT_resnet is more efficient than the other ablated meta-teachers in guiding the PA detector to learn spoofing cues.

4.5.3 Correctly and wrongly classified faces

Fig. 11 visualizes some testing faces that are correctly and wrongly classified by FAS-DR(MT). All visualized faces are sampled from the testing set of protocol 2 in OULU-NPU. Live $\rightarrow$ Spoof denotes the live faces that are wrongly classified as spoof faces. Spoof $\rightarrow$ Live denotes the spoof faces that are wrongly classified as live faces. Live $\rightarrow$ Live and Spoof $\rightarrow$ Spoof denote the correctly classified live and spoof faces, respectively. Fig. 11 also visualizes the corresponding pixel-wise maps predicted by FAS-DR(MT) for understanding why the faces are correctly or wrongly classified. In our work, FAS-DR(MT) classifies each testing face by comparing the average value of the predicted map with the threshold. Given the predicted map $\hat{y}_{map}$, the testing face will be classified as a spoof face if $mean(\hat{y}_{map})>$threshold, otherwise, it will be classified as a live face. For instance, the three live faces in the left upper corner are misclassified as spoof faces because the averages of the predicted maps are larger than the threshold. Besides, for illustrating how the meta-teacher teaching the PA detector, we also show the pixel-wise maps outputted by the meta-teacher in Fig. 11.

5 Conclusion and Future Work

Existing deep learning-based FAS methods use handcrafted labels to supervise the PA detector's learning. Although handcrafted labels perform satisfactorily in supervising existing PA detectors, these labels rely heavily on human's prior-knowledge about FAS and may lose effectiveness against newly developed spoof types. In this paper, we aim to explore better supervision towards PA detectors from a

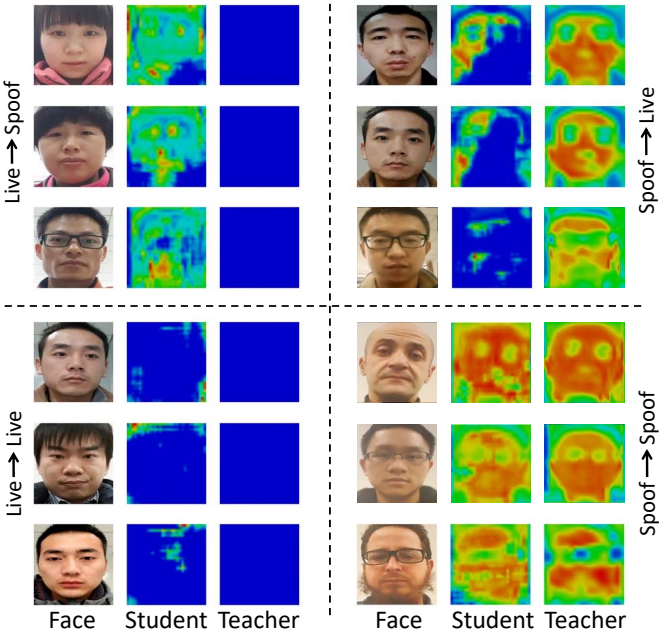


Fig. 11. Some faces that are correctly and wrongly classified by FAS-DR(MT). The 'Student' column denotes the detector FAS-DR(MT)'s output for the corresponding face. The 'Teacher' column denotes the meta-teacher's output. The colors ranging from blue to red denote the float numbers from zero to one. Blue denotes 0 and Red denotes 1.

novel perspective. To this end, we propose a novel meta-teacher face anti-spoofing (MT-FAS) method, in which a meta-teacher is trained to learn how to provide better-suited supervision to the PA detector. Once the meta-teacher is trained, we use it to supervise existing PA detectors' training and improve these detectors' performance. By extensive experiments, we demonstrate that the meta-teacher outperforms not only the most widely employed manually-designed labels but also existing teacher-student methods in training PA detectors. Moreover, with the advantage of the meta-teacher, we improve upon the state-of-the-art performances on several popular FAS benchmarks.

Despite the demonstrated advantages of the trained meta-teacher, we should note that the training of the meta-teacher needs complex second-order gradient, as Eq. 12 shows. In the future, we will dive deeper into MT-FAS and try to reduce the requirement of calculating second-order gradient. Moreover, extending MT-FAS to other computer vision tasks (*e.g.*, classification, noisy-label problems) is also an interesting direction worthy to be explored.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (No. 2020YFC2003901). This work was also supported in part by the National Natural Science Foundation of China (No. 61876178, 61872367, and 61806196).

References

[1] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[2] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.

[3] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, "The megaface benchmark: 1 million faces for recognition at scale," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4873–4882.

[4] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.

[5] T. de Freitas Pereira, A. Anjos, J. M. De Martino, and S. Marcel, "Lbp- top based countermeasure against face spoofing attacks," in *Asian Conference on Computer Vision*. Springer, 2012, pp. 121–132.

[6] J. Komulainen, A. Hadid, and M. Pietikäinen, "Context based face anti-spoofing," in *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. IEEE, 2013, pp. 1–8.

[7] R. Shao, X. Lan, and P. C. Yuen, "Deep convolutional dynamic texture learning with adaptive channel-discriminability for 3d mask face anti-spoofing," in *IEEE International Joint Conference on Biometrics (IJCB)*, 2017, pp. 748–755.

[8] Y. Zhang, Z. Yin, Y. Li, G. Yin, J. Yan, J. Shao, and Z. Liu, "Celeba-spoof: Large-scale face anti-spoofing dataset with rich annotations," in *European Conference on Computer Vision*. Springer, 2020, pp. 70–85.

[9] X. Wu, J. Zhou, J. Liu, F. Ni, and H. Fan, "Single-shot face anti-spoofing for dual pixel camera," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1440–1451, 2020.

[10] B. Peixoto, C. Michelassi, and A. Rocha, "Face liveness detection under bad illumination conditions," in *2011 18th IEEE International Conference on Image Processing*. IEEE, 2011, pp. 3557–3560.

[11] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face antispoofing using speeded-up robust features and fisher vector encoding," *IEEE Signal Processing Letters*, vol. 24, no. 2, pp. 141–145, 2017.

[12] O. Lucena, A. Junior, V. Moia, R. Souza, E. Valle, and R. Lotufo, "Transfer learning using convolutional neural networks for face anti-spoofing," in *International Conference Image Analysis and Recognition*, 2017, pp. 27–34.

[13] Z. Xu, S. Li, and W. Deng, "Learning temporal features using lstm-cnn architecture for face anti-spoofing," in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, 2015, pp. 141–145.

[14] Y. Liu, A. Jourabloo, and X. Liu, "Learning deep models for face anti-spoofing: Binary or auxiliary supervision," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 389–398.

[15] W. Zezheng, Y. Zitong, Z. Chenxu, Z. Xiangyu, Q. Yunxiao, Z. Qiusheng, Z. Feng, and L. Zhen, "Deep spatial gradient and temporal depth learning for face anti-spoofing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[16] Y. Qin, C. Zhao, X. Zhu, Z. Wang, Z. Yu, T. Fu, F. Zhou, J. Shi, and Z. Lei, "Learning meta model for zero-and few-shot face anti-spoofing," *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 11 916–11 923, Apr. 2020.

[17] T. Kim, Y. Kim, I. Kim, and D. Kim, "Basn: Enriching feature representation using bipartite auxiliary supervisions for face anti-spoofing," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[18] Z. Yu, X. Li, X. Niu, J. Shi, and G. Zhao, "Face anti-spoofing with human material perception," in *European Conference on Computer Vision*, 2020.

[19] Y. Liu, J. Stehouwer, A. Jourabloo, and X. Liu, "Deep tree learning for zero-shot face anti-spoofing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4680–4689.

[20] A. George and S. Marcel, "Deep pixel-wise binary supervision for face presentation attack detection," in *International Conference on Biometrics (ICB)*, no. CONF, 2019.

[21] Y. Qin, W. Zhang, J. Shi, Z. Wang, and L. Yan, "One-class adaptation face anti-spoofing with loss function search," *Neurocomputing*, 2020.

[22] Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu, "Face anti-spoofing using patch and depth-based cnns," in *2017 IEEE International Joint Conference on Biometrics (IJCB)*, 2017, pp. 319–328.

[23] Z. Yu, X. Li, J. Shi, Z. Xia, and G. Zhao, "Revisiting pixel-wise supervision for face anti-spoofing," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2021.

[24] E. G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *CoRR*, 2015.

[25] A. Romero, N. Ballas, E. S. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," *international conference on learning representations*, 2015.

[26] T. Furlanello, C. Z. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born again neural networks," *international conference on machine learning*, pp. 1602–1611, 2018.

[27] X. Jin, B. Peng, Y. Wu, Y. Liu, J. Liu, D. Liang, J. Yan, and X. Hu, "Knowledge distillation via route constrained optimization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1345–1354.

[28] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Advances in neural information processing systems*, 2014, pp. 2654–2662.

[29] S.-I. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh, "Improved knowledge distillation via teacher assistant," *national conference on artificial intelligence*, 2020.

[30] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Advances in Neural Information Processing Systems*, 2017, pp. 742–751.

[31] D. Chen, J.-P. Mei, C. Wang, Y. Feng, and C. Chen, "Online knowledge distillation with diverse peers," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 3430–3437, Apr. 2020. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/5746

[32] J. H. Cho and B. Hariharan, "On the efficacy of knowledge distillation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4794–4802.

[33] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," *ICLR*, 2019.

[34] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," *arXiv preprint arXiv:1703.03400*, 2017.

[35] Y. Qin, W. Zhang, Z. Wang, C. Zhao, and J. Shi, "Layer-wise adaptive updating for few-shot image classification," *IEEE Signal Processing Letters*, 2020.

[36] Z. Li, F. Zhou, F. Chen, and H. Li, "Meta-sgd: Learning to learn quickly for few shot learning," *arXiv preprint arXiv:1707.09835*, 2017.

[37] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face anti-spoofing based on color texture analysis," in *2011 18th IEEE International Conference on Image Processing*, 2015, pp. 2636–2640.

[38] K. Patel, H. Han, and A. K. Jain, "Secure face unlock: Spoof detection on smartphones," *IEEE transactions on information forensics and security*, vol. 11, no. 10, pp. 2268–2283, 2016.

[39] X. Li, J. Komulainen, G. Zhao, P.-C. Yuen, and M. Pietikäinen, "Generalized face anti-spoofing by detecting pulse from face videos," in *23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 4244–4249.

[40] S. Liu, P. C. Yuen, S. Zhang, and G. Zhao, "3d mask face anti-spoofing with remote photoplethysmography," in *European Conference on Computer Vision*. Springer, 2016, pp. 85–100.

[41] B. Lin, X. Li, Z. Yu, and G. Zhao, "Face liveness detection by rppg features and contextual patch-based cnn," in *Proceedings of the 2019 3rd International Conference on Biometric Engineering and Applications*. ACM, 2019, pp. 61–68.

[42] C. Nagpal and S. R. Dubey, "A performance evaluation of convolutional neural networks for face anti spoofing," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[43] L. Li, X. Feng, Z. Boulkenafet, Z. Xia, M. Li, and A. Hadid, "An original face anti-spoofing approach using partial convolutional neural network," in *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE, 2016, pp. 1–6.

[44] K. Patel, H. Han, and A. K. Jain, "Cross-database face antispoofing with robust feature representation," in *Chinese Conference on Biometric Recognition*. Springer, 2016, pp. 611–619.

[45] Z. Yu, Y. Qin, X. Xu, C. Zhao, Z. Wang, Z. Lei, and G. Zhao, "Autofas: Searching lightweight networks for face anti-spoofing," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 996–1000.

[46] Y. Jia, J. Zhang, S. Shan, and X. Chen, "Single-side domain generalization for face anti-spoofing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[47] K.-Y. Zhang, T. Yao, J. Zhang, Y. Tai, S. Ding, J. Li, F. Huang, H. Song, and L. Ma, "Face anti-spoofing via disentangled representation learning," in *European Conference on Computer Vision*. Springer, 2020, pp. 641–657.

[48] 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*, vol. 16, pp. 937–951, 2020.

[49] A. Liu, Z. Tan, J. Wan, Y. Liang, Z. Lei, G. Guo, and S. Z. Li, "Face anti-spoofing via adversarial cross-modality translation," *IEEE Transactions on Information Forensics and Security*, 2021.

[50] R. Shao, X. Lan, and P. C. Yuen, "Regularized fine-grained meta face anti-spoofing," *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[51] Z. Yu, Y. Qin, X. Li, Z. Wang, C. Zhao, Z. Lei, and G. Zhao, "Multi-modal face anti-spoofing based on central difference networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 650–651.

[52] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 535–541.

[53] G. Anandalingam and l. t. friesz, "Hierarchical optimization: an introduction," *Annals of Operations Research*, pp. 1–11, 1992.

[54] B. Colson, P. Marcotte, and G. Savard, "An overview of bilevel optimization," *Annals OR*, pp. 235–256, 2007.

[55] Z. Yu, C. Zhao, Z. Wang, Y. Qin, Z. Su, X. Li, F. Zhou, and G. Zhao, "Searching central difference convolutional networks for face anti-spoofing," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 5295–5305.

[56] international organization for standardization, "Iso/iec jtc 1/sc 37 biometrics: Information technology biometric presentation attack detection part 1: Framework." in *https://www.iso.org/obp/ui/iso*, 2016.

[57] D. Umphress and G. Williams, "Identity verification through keyboard characteristics," *International journal of man-machine studies*, vol. 23, no. 3, pp. 263–273, 1985.

[58] Z. Boulkenafet, J. Komulainen, L. Li, X. Feng, and A. Hadid, "Oulu-npu: A mobile face presentation attack database with real-world variations," in *FGR*, 2017, pp. 612–618.

[59] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li, "A face anti-spoofing database with diverse attacks," in *International Conference on Biometrics (ICB)*, 2012, pp. 26–31.

[60] I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in *Biometrics Special Interest Group*, 2012, pp. 1–7.

[61] D. Wen, H. Han, and A. K. Jain, "Face spoof detection with image distortion analysis," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 746–761, 2015.

[62] R. Shao, X. Lan, J. Li, and P. C. Yuen, "Multi-adversarial discriminative deep domain generalization for face presentation attack detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 023–10 031.

[63] Z. Boulkenafet, J. Komulainen, Z. Akhtar, A. Benlamoudi, D. Samai, S. E. Bekhouche, A. Ouafi, F. Dornaika, A. Taleb-Ahmed, L. Qin *et al.*, "A competition on generalized software-based face presentation attack detection in mobile scenarios," in *2017 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2017, pp. 688–696.

[64] X. Yang, W. Luo, L. Bao, Y. Gao, D. Gong, S. Zheng, Z. Li, and W. Liu, "Face anti-spoofing: Model matters, so does data," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[65] A. Jourabloo, Y. Liu, and X. Liu, "Face de-spoofing: Anti-spoofing via noise modeling," in *European Conference on Computer Vision*, 2018, pp. 290–306.

[66] Y. Liu, J. Stehouwer, and X. Liu, "On disentangling spoof trace for generic face anti-spoofing," in *European Conference on Computer Vision*. Springer, 2020, pp. 406–422.

[67] J. Määttä, A. Hadid, and M. Pietikäinen, "Face spoofing detection from single images using micro-texture analysis," in *IEEE International Joint Conference on Biometrics (IJCB)*, 2011, pp. 1–7.

[68] J. Yang, Z. Lei, and S. Z. Li, "Learn convolutional neural network for face anti-spoofing," *Computer Science*, vol. 9218, pp. 373–384, 2014.

[69] D. Wen, H. Han, and A. K. Jain, "Face spoof detection with image distortion analysis," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 746–761, 2015.

[70] T. de Freitas Pereira, J. Komulainen, A. Anjos, J. M. De Martino, A. Hadid, M. Pietikäinen, and S. Marcel, "Face liveness detection using dynamic texture," *EURASIP Journal on Image and Video Processing*, vol. 2014, no. 1, p. 2, 2014.

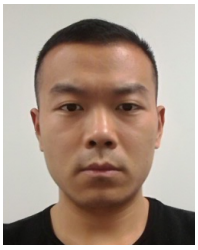
[71] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face spoofing detection using colour texture analysis," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1818–1830, 2016.

[72] H. Li, S. Jialin Pan, S. Wang, and A. C. Kot, "Domain generalization with adversarial feature learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5400–5409.

[73] R. Shao, X. Lan, J. Li, and P. C. Yuen, "Multi-adversarial discriminative deep domain generalization for face presentation attack detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

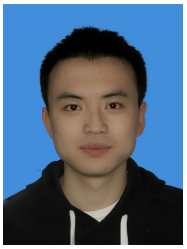
[74] G. Wang, H. Han, S. Shan, and X. Chen, "Cross-domain face presentation attack detection via multi-domain disentangled representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6678–6687.

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

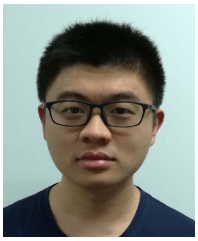


Yunxiao Qin received the M.S. degree in Control Science and Engineering from Northwestern Polytechnical University, Xian, China, in 2015, where he is currently pursuing the Ph.D. degree. His current research interests include meta-learning, face anti-spoofing and deep reinforcement learning.

Zitong Yu received the M.S. degree from University of Nantes, France, in 2016, and he is currently a Ph.D. candidate in the Center for Machine Vision and Signal Analysis, University of Oulu, Finland. His research interests focus on remote photoplethysmograph measurement, face anti-spoofing and video understanding.



Longbin Yan received the B.S. and M.S. degrees from the School of Marine Science and Technology, Northwestern Polytechnical University,Xi'an, China, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree in the Laboratory of Centre of Intelligent Acoustics and Immersive Communications, School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, China. His research interests include hyperspectral image analysis, object detection, and image super-resolution.



Zezheng Wang received the B.S. and M.S. degrees from Tianjin University, Tianjin, China, in 2015 and 2018, respectively. He is currently an Artificial Intelligence Engineer at Beijing Kwai Technology Co., Ltd, Beijing, China. His current research interests include, machine learning, signal processing, and face anti-spoofing.

Chenxu Zhao received M.S. Degree from Beihang University, Beijing, China, in 2016, and was in the joint programme with National Laboratory of Pattern Recognition (NLPR) Laboratory of Institute of Automation, Chinese Academy of Sciences, from 2014 to 2016. From 2016 to 2017, he worked as research scientist in SenseTime, Beijing, China. He is currently a research scientist in MiningLamp Technology, Beijing, China. His major research areas include face anti-spoofing, face recognition and meta-learning.

Zhen Lei received the B.S. degree in automation from the University of Science and Technology of China, in 2005, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, in 2010, where he is currently a Professor. He has published over 130 papers in international journals and conferences. His research interests are in computer vision, pattern recognition, image processing, and face recognition in particular. He served as an Area Chair of the International Joint Conference on Biometrics in 2014, the IEEE International Conference on Automatic Face and Gesture Recognition in 2015, the IAPR/IEEE International Conference on Biometric in 2015, 2016, 2018, and the IEEE International Conference on Biometrics: Theory, Applications and Systems in 2018.