

# Learning Deep Models for Face Anti-Spoofing: Binary or Auxiliary Supervision

Yaojie Liu\* Amin Jourabloo\* Xiaoming Liu  
 Department of Computer Science and Engineering  
 Michigan State University, East Lansing MI 48824  
 {liuyaoj1, jourablo, liuxm}@msu.edu

## Abstract

Face anti-spoofing is crucial to prevent face recognition systems from a security breach. Previous deep learning approaches formulate face anti-spoofing as a binary classification problem. Many of them struggle to grasp adequate spoofing cues and generalize poorly. In this paper, we argue the importance of auxiliary supervision to guide the learning toward discriminative and generalizable cues. A CNN-RNN model is learned to estimate the face depth with pixel-wise supervision, and to estimate rPPG signals with sequence-wise supervision. The estimated depth and rPPG are fused to distinguish live vs. spoof faces. Further, we introduce a new face anti-spoofing database that covers a large range of illumination, subject, and pose variations. Experiments show that our model achieves the state-of-the-art results on both intra- and cross-database testing.

## 1. Introduction

With applications in phone unlock, access control, and security, biometric systems are widely used in our daily lives, and face is one of the most popular biometric modalities. While face recognition systems gain popularity, attackers present face spoofs (i.e., presentation attacks, PA) to the system and attempt to be authenticated as the genuine user. The face PA include printing a face on paper (print attack), replaying a face video on a digital device (replay attack), wearing a mask (mask attack), etc. To counteract PA, face anti-spoofing techniques [16, 22, 23, 29] are developed to detect PA *prior to* a face image being recognized. Therefore, face anti-spoofing is vital to ensure that face recognition systems are robust to PA and safe to use.

RGB image and video are the standard input to face anti-spoofing systems, similar to face recognition systems. Researchers start the texture-based anti-spoofing approaches by feeding handcrafted features to binary classifiers [13, 18, 19, 27, 33, 34, 38, 49]. Later in the deep learning era, several Convolutional Neural Networks (CNN) approaches utilize

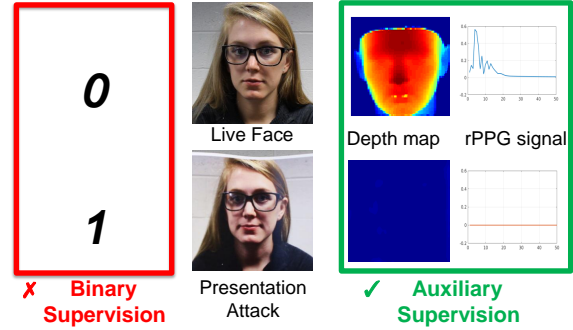


Figure 1. Conventional CNN-based face anti-spoof approaches utilize the binary supervision, which may lead to overfitting given the enormous solution space of CNN. This work designs a novel network architecture to leverage two auxiliary information as supervision: the depth map and rPPG signal, with the goals of improved generalization and explainable decisions during inference.

softmax loss as the supervision [21, 30, 37, 48]. It appears almost all prior work regard the face anti-spoofing problem as merely a *binary* (live vs. spoof) classification problem.

There are two main issues in learning deep anti-spoofing models with binary supervision. First, there are different levels of image degradation, namely *spoof patterns*, comparing a spoof face to a live one, which consist of skin detail loss, color distortion, moiré pattern, shape deformation and spoof artifacts (e.g., reflection) [29, 38]. A CNN with softmax loss might discover *arbitrary* cues that are able to separate the two classes, such as screen bezel, but not the *faithful* spoof patterns. When those cues disappear during testing, these models would fail to distinguish spoof vs. live faces and result in poor generalization. Second, during the testing, models learnt with binary supervision will only generate a binary decision without *explanation* or *rationale* for the decision. In the pursuit of Explainable Artificial Intelligence [1], it is desirable for the learnt model to generate the spoof patterns that support the final binary decision.

To address these issues, as shown in Fig. 1, we propose a deep model that uses the supervision from both the *spatial* and *temporal auxiliary information* rather than binary supervision, for the purpose of robustly detecting face PA

\*denotes equal contribution by the authors.

from a face video. These auxiliary information are acquired based on our domain knowledge about the key *differences* between live and spoof faces, which include two perspectives: spatial and temporal. From the spatial perspective, it is known that live faces have face-like depth, e.g., the nose is closer to the camera than the cheek in frontal-view faces, while faces in print or replay attacks have flat or planar depth, e.g., all pixels on the image of a paper have the same depth to the camera. Hence, depth can be utilized as auxiliary information to supervise both live and spoof faces. From the temporal perspective, it was shown that the normal rPPG signals (i.e., heart pulse signal) are detectable from live, but not spoof, face videos [31, 35]. Therefore, we provide different rPPG signals as auxiliary supervision, which guides the network to learn from live or spoof face videos respectively. To enable both supervisions, we design a network architecture with a short-cut connection to capture different scales and a novel non-rigid registration layer to handle the motion and pose change for rPPG estimation.

Furthermore, similar to many vision problems, data plays a significant role in training the anti-spoofing models. As we know, camera/screen quality is a critical factor to the quality of spoof faces. Existing face anti-spoofing databases, such as NUAA [43], CASIA [50], Replay-Attack [17], and MSU-MFSD [45], were collected 3 – 5 years ago. Given the fast advance of consumer electronics, the types of equipment (e.g., cameras and spoofing mediums) used in those data collection are outdated compared to the ones nowadays, regarding the resolution and imaging quality. More recent MSU-USSA [38] and OULU databases [14] have subjects with fewer variations in poses, illuminations, expressions (PIE). The lack of necessary variations would make it hard to learn an effective model. Given the clear need for more advanced databases, we collect a face anti-spoofing database, named Spoof in the Wild Database (SiW). SiW database consists of 165 subjects, 6 spoofing mediums, and 4 sessions covering variations such as PIE, distance-to-camera, etc. SiW covers much larger variations than previous databases, as detailed in Tab. 1 and Sec. 4. The main contributions of this work include:

- ◊ We propose to leverage novel auxiliary information (i.e., depth map and rPPG) to supervise the CNN learning for improved generalization.
- ◊ We propose a novel CNN-RNN architecture for end-to-end learning the depth map and rPPG signal.
- ◊ We release a new database that contains variations of PIE, and other practical factors. We achieve the state-of-the-art performance for face anti-spoofing.

## 2. Prior Work

We review the prior face anti-spoofing works in three groups: texture-based methods, temporal-based methods, and remote photoplethysmography methods.

**Texture-based Methods** Since most face recognition systems adopt only RGB cameras, using texture information has been a natural approach to tackling face anti-spoofing. Many prior works utilize hand-crafted features, such as LBP [18, 19, 33], HoG [27, 49], SIFT [38] and SURF [13], and adopt traditional classifiers such as SVM and LDA. To overcome the influence of illumination variation, they seek solutions in a different input domain, such as HSV and YCbCr color space [11, 12], and Fourier spectrum [29].

As deep learning has proven to be effective in many computer vision problems, there are many recent attempts of using CNN-based features or CNNs in face anti-spoofing [21, 30, 37, 48]. Most of the work treats face anti-spoofing as a simple *binary* classification problem by applying the softmax loss. For example, [30, 37] use CNN as feature extractor and fine-tune from ImageNet-pretrained CaffeNet and VGG-face. The work of [21, 30] feed different designs of the face images into CNN, such as multi-scale faces and hand-crafted features, and directly classify live vs. spoof. One prior work that shares the similarity with ours is [5], where Atoum *et al.* propose a two-stream CNN-based anti-spoofing method using texture and depth. We advance [5] in a number of aspects, including fusion with temporal supervision (i.e., rPPG), finer architecture design, novel non-rigid registration layer, and comprehensive experimental support.

**Temporal-based Methods** One of the earliest solutions for face anti-spoofing is based on temporal cues such as eye-blinking [36, 37]. Methods such as [26, 42] track the motion of mouth and lip to detect the face liveness. While these methods are effective to typical paper attacks, they become vulnerable when attackers present a replay attack or a paper attack with eye/mouth portion being cut.

There are also methods relying on more general temporal features, instead of the specific facial motion. The most common approach is frame concatenation. Many hand-crafted feature-based methods may improve intra-database testing performance by simply concatenating the features of consecutive frames to train the classifiers [11, 18, 28]. Additionally, there are some works proposing temporal-specific features, e.g., Haralick features [4], motion mag [7], and optical flow [6]. In the deep learning era, Feng *et al.* feed the optical flow map and Shearlet image feature to CNN [21]. In [47], Xu *et al.* propose an LSTM-CNN architecture to utilize temporal information for binary classification. Overall, all prior methods still regard face anti-spoofing as a binary classification problem, and thus they have a hard time to generalize well in the cross-database testing. In this work, we extract discriminative temporal information by learning the rPPG signal of the face video.

**Remote Photoplethysmography (rPPG)** Remote photoplethysmography (rPPG) is the technique to track vital signals, such as heart rate, without any contact with human skin [9, 20, 41, 44, 46]. Research starts with face videos with

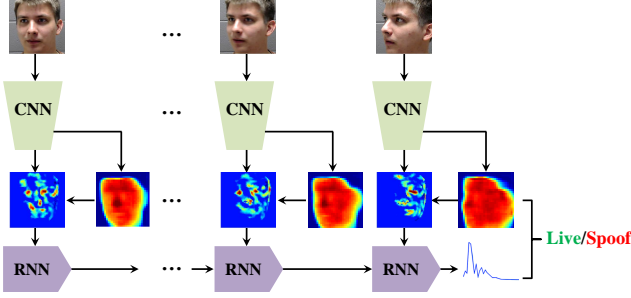


Figure 2. The overview of the proposed method.

no motion or illumination change to videos with multiple variations. In [20], Haan *et al.* estimate rPPG signals from RGB face videos with lighting and motion changes. It utilizes color difference to eliminate the specular reflection and estimate two orthogonal chrominance signals. After applying the Band Pass Filter (BPM), the ratio of the chrominance signals are used to compute the rPPG signal.

rPPG has previously been utilized to tackle face anti-spoofing [31, 35]. In [31], rPPG signals are used for detecting the 3D mask attack, where the live faces exhibit a pulse of heart rate unlike the 3D masks. They use rPPG signals extracted by [20] and compute the correlation features for classification. Similarly, Magdalena *et al.* [35] extract rPPG signals (also via [20]) from three face regions and two non-face regions, for detecting print and replay attacks. Although in replay attacks, the rPPG extractor might still capture the normal pulse, the combination of multiple regions can differentiate live vs. spoof faces. While the analytic solution to rPPG extraction [20] is easy to implement, we observe that it is sensitive to PIE variations. Hence, we employ a novel CNN-RNN architecture to *learn* a mapping from a face video to the rPPG signal, which is not only robust to PIE variations, but also discriminative to live vs. spoof.

### 3. Face Anti-Spoofing with Deep Network

The main idea of the proposed approach is to guide the deep network to focus on the *known spoof patterns* across spatial and temporal domains, rather than to extract any cues that could separate two classes but are not generalizable. As shown in Fig. 2, the proposed network combines CNN and RNN architectures in a coherent way. The CNN part utilizes the depth map supervision to discover subtle texture property that leads to distinct depths for live and spoof faces. Then, it feeds the estimated depth and the feature maps to a novel *non-rigid registration* layer to create aligned feature maps. The RNN part is trained with the aligned maps and the rPPG supervision, which examines temporal variability across video frames.

#### 3.1. Depth Map Supervision

Depth maps are a representation of the 3D shape of the face in a 2D image, which shows the face location and the

depth information of different facial areas. This representation is more informative than binary labels since it indicates one of the fundamental differences between live faces, and print and replay PA. We utilize the depth maps in the depth loss function to supervise the CNN part. The pixel-based depth loss guides the CNN to learn a mapping from the face area within a receptive field to a labeled depth value – a scale within  $[0, 1]$  for live faces and 0 for spoof faces.

To estimate the depth map for a 2D face image, given a face image, we utilize the state-of-the-art dense face alignment (DeFA) methods [25, 32] to estimate the 3D shape of the face. The frontal dense 3D shape  $\mathbf{S}_F \in \mathbb{R}^{3 \times Q}$ , with  $Q$  vertices, is represented as a linear combination of identity bases  $\{\mathbf{S}_{id}^i\}_{i=1}^{N_{id}}$  and expression bases  $\{\mathbf{S}_{exp}^i\}_{i=1}^{N_{exp}}$ ,

$$\mathbf{S}_F = \mathbf{S}_0 + \sum_{i=1}^{N_{id}} \alpha_{id}^i \mathbf{S}_{id}^i + \sum_{i=1}^{N_{exp}} \alpha_{exp}^i \mathbf{S}_{exp}^i, \quad (1)$$

where  $\alpha_{id} \in \mathbb{R}^{199}$  and  $\alpha_{ext} \in \mathbb{R}^{29}$  are the identity and expression parameters, and  $\alpha = [\alpha_{id}, \alpha_{exp}]$  are the shape parameters. We utilize the Basel 3D face model [39] and the facewarehouse [15] as the identity and expression bases.

With the estimated pose parameters  $\mathbf{P} = (s, \mathbf{R}, \mathbf{t})$ , where  $\mathbf{R}$  is a 3D rotation matrix,  $\mathbf{t}$  is a 3D translation, and  $s$  is a scale, we align the 3D shape  $\mathbf{S}$  to the 2D face image:

$$\mathbf{S} = s\mathbf{R}\mathbf{S}_F + \mathbf{t}. \quad (2)$$

Given the challenge of estimating the *absolute* depth from a 2D face, we normalize the  $z$  values of 3D vertices in  $\mathbf{S}$  to be within  $[0, 1]$ . That is, the vertex closest to the camera (e.g., nose) has a depth of one, and the vertex furthest away has the depth of zero. Then, we apply the Z-Buffer algorithm [51] to  $\mathbf{S}$  for projecting the normalized  $z$  values to a 2D plane, which results in an estimated “ground truth” 2D depth map  $\mathbf{D} \in \mathbb{R}^{32 \times 32}$  for a face image.

#### 3.2. rPPG Supervision

rPPG signals have recently been utilized for face anti-spoofing [31, 35]. The rPPG signal provides temporal information about face liveness, as it is related to the intensity changes of facial skin over time. These intensity changes are highly correlated with the blood flow. The traditional method [20] for extracting rPPG signals has three drawbacks. First, it is sensitive to pose and expression variation, as it becomes harder to *track* a specific face area for measuring intensity changes. Second, it is also sensitive to illumination changes, since the extra lighting affects the amount of reflected light from the skin. Third, for the purpose of anti-spoof, rPPG signals extracted from spoof videos might not be sufficiently *distinguishable* to signals of live videos.

One novelty aspect of our approach is that, instead of computing the rPPG signal via [20], our RNN part learns to estimate the rPPG signal. This eases the signal estimation

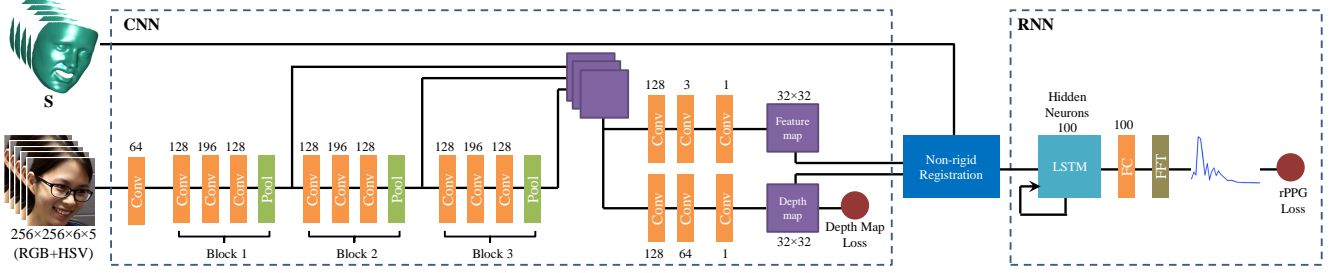


Figure 3. The proposed CNN-RNN architecture. The number of filters are shown on top of each layer, the size of all filters is  $3 \times 3$  with stride 1 for convolutional and 2 for pooling layers. Color code used: orange=convolution, green=pooling, purple=response map.

from face videos with PIE variations, and also leads to more discriminative rPPG signals, as different rPPG supervisions are provided to live vs. spoof videos. We assume that the videos of the same subject under different PIE conditions have the *same* ground truth rPPG signal. This assumption is valid since the heart beat is similar for the videos of the same subject that are captured in a short span of time ( $< 5$  minutes). The rPPG signal extracted from the constrained videos (i.e., no PIE variation) are used as the “ground truth” supervision in the rPPG loss function for *all* live videos of the same subject. This consistent supervision helps the CNN and RNN parts to be robust to the PIE changes.

In order to extract the rPPG signal from a face video without PIE, we apply the DeFA [32] to each frame and estimate the dense 3D face shape. We utilize the estimated 3D shape to track a face region. For a tracked region, we compute two orthogonal chrominance signals  $\mathbf{x}_f = 3\mathbf{r}_f - 2\mathbf{g}_f$ ,  $\mathbf{y}_f = 1.5\mathbf{r}_f + \mathbf{g}_f - 1.5\mathbf{b}_f$  where  $\mathbf{r}_f, \mathbf{g}_f, \mathbf{b}_f$  are the bandpass filtered versions of the  $\mathbf{r}, \mathbf{g}, \mathbf{b}$  channels with the skin-tone normalization. We utilize the ratio of the standard deviation of the chrominance signals  $\gamma = \frac{\sigma(\mathbf{x}_f)}{\sigma(\mathbf{y}_f)}$  to compute blood flow signals [20]. We calculate the signal  $\mathbf{p}$  as:

$$\mathbf{p} = 3(1 - \frac{\gamma}{2})\mathbf{r}_f - 2(1 + \frac{\gamma}{2})\mathbf{g}_f + \frac{3\gamma}{2}\mathbf{b}_f. \quad (3)$$

By applying FFT to  $\mathbf{p}$ , we obtain the rPPG signal  $\mathbf{f} \in \mathbb{R}^{50}$ , which shows the magnitude of each frequency.

### 3.3. Network Architecture

Our proposed network consists of two deep networks. First, a CNN part evaluates each frame separately and estimates the depth map and feature map of each frame. Second, a recurrent neural network (RNN) part evaluates the temporal variability across the feature maps of a sequence.

#### 3.3.1 CNN Network

We design a Fully Convolutional Network (FCN) as our CNN part, as shown in Fig. 3. The CNN part contains multiple blocks of three convolutional layers, pooling and resizing layers where each convolutional layer is followed by one exponential linear layer and batch normalization layer. Then, the resizing layers resize the response maps after each

block to a pre-defined size of  $64 \times 64$  and concatenate the response maps. The bypass connections help the network to utilize extracted features from layers with different depths similar to the ResNet structure [24]. After that, our CNN has two branches, one for estimating the depth map and the other for estimating the feature map.

The first output of the CNN is the estimated depth map of the input frame  $\mathbf{I} \in \mathbb{R}^{256 \times 256}$ , which is supervised by the estimated “ground truth” depth  $\mathbf{D}$ ,

$$\Theta_D = \arg \min_{\Theta_D} \sum_{i=1}^{N_d} \|\text{CNN}_D(\mathbf{I}_i; \Theta_D) - \mathbf{D}_i\|_1^2, \quad (4)$$

where  $\Theta_D$  is the CNN parameters and  $N_d$  is the number of training images. The second output of the CNN is the feature map, which is fed into the non-rigid registration layer.

#### 3.3.2 RNN Network

The RNN part aims to estimate the rPPG signal  $\mathbf{f}$  of an input sequence with  $N_f$  frames  $\{\mathbf{I}_j\}_{j=1}^{N_f}$ . As shown in Fig. 3, we utilize one LSTM layer with 100 hidden neurons, one fully connected layer, and an FFT layer that converts the response of fully connected layer into the Fourier domain. Given the input sequence  $\{\mathbf{I}_j\}_{j=1}^{N_f}$  and the “ground truth” rPPG signal  $\mathbf{f}$ , we train the RNN to minimize the  $\ell_1$  distance of the estimated rPPG signal to “ground truth”  $\mathbf{f}$ ,

$$\Theta_R = \arg \min_{\Theta_R} \sum_{i=1}^{N_s} \|\text{RNN}_R([\{\mathbf{F}_j\}_{j=1}^{N_f}]_i; \Theta_R) - \mathbf{f}_i\|_1^2, \quad (5)$$

where  $\Theta_R$  is the RNN parameters,  $\mathbf{F}_j \in \mathbb{R}^{32 \times 32}$  is the frontalized feature map (details in Sec. 3.4), and  $N_s$  is the number of sequences.

#### 3.3.3 Implementation Details

**Ground Truth Data** Given a set of live and spoof face videos, we provide the ground truth supervision for the depth map  $\mathbf{D}$  and rPPG signal  $\mathbf{f}$ , as in Fig. 4. We follow the procedure in Sec. 3.1 to compute “ground truth” data for live videos. For spoof videos, we set the ground truth depth maps to a plain surface, i.e., zero depth. Similarly, we follow the procedure in Sec. 3.2 to compute the “ground truth”



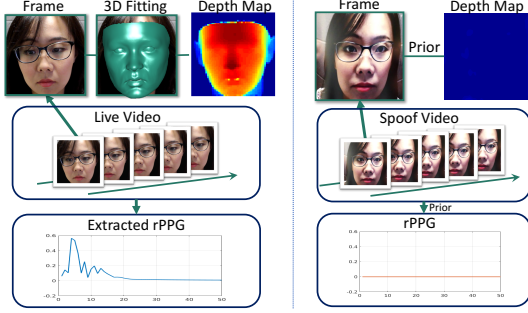


Figure 4. Example ground truth depth maps and rPPG signals.

rPPG signal from a patch on the forehead, for one live video of each subject without PIE variation. Also, we normalize the norm of estimated rPPG signal such that  $\|\mathbf{f}\|_2 = 1$ . For spoof videos, we consider the rPPG signals are zero.

Note that, while the term “depth” is used here, our estimated depth is different to the conventional depth map in computer vision. It can be viewed as a “pseudo-depth” and serves the purpose of providing discriminative auxiliary supervision to the learning process. The same perspective applies to the supervision based on pseudo-rPPG signal.

**Training Strategy** Our proposed network combines the CNN and RNN parts for end-to-end training. The desired training data for the CNN part should be from diverse subjects, so as to make the training procedure more stable and increase the generalizability of the learnt model. Meanwhile, the training data for the RNN part should be long sequences to leverage the temporal information across frames. These two preferences can be contradictory to each other, especially given the limited GPU memory. Hence, to satisfy both preferences, we design a two-stream training strategy. The first stream satisfies the preference of the CNN part, where the input includes face images  $\mathbf{I}$  and the ground truth depth maps  $\mathbf{D}$ . The second stream satisfies the RNN part, where the input includes face sequences  $\{\mathbf{I}_j\}_{j=1}^{N_f}$ , the ground truth depth maps  $\{\mathbf{D}_j\}_{j=1}^{N_f}$ , the estimated 3D shapes  $\{\mathbf{S}_j\}_{j=1}^{N_f}$ , and the corresponding ground truth rPPG signals  $\mathbf{f}$ . During training, our method alternates between these two streams to converge to a model that minimizes both the depth map and rPPG losses. Note that even though the first stream only updates the weights of the CNN part, the back propagation of the second stream updates the weights of both CNN and RNN parts in an end-to-end manner.

**Testing** To provide a classification score, we feed the testing sequence to our network and compute the depth map  $\hat{\mathbf{D}}$  of the last frame and the rPPG signal  $\hat{\mathbf{f}}$ . Instead of designing a classifier using  $\hat{\mathbf{D}}$  and  $\hat{\mathbf{f}}$ , we compute the final score as:

$$score = \|\hat{\mathbf{f}}\|_2^2 + \lambda \|\hat{\mathbf{D}}\|_2^2, \quad (6)$$

where  $\lambda$  is a constant weight for combining the two outputs of the network.

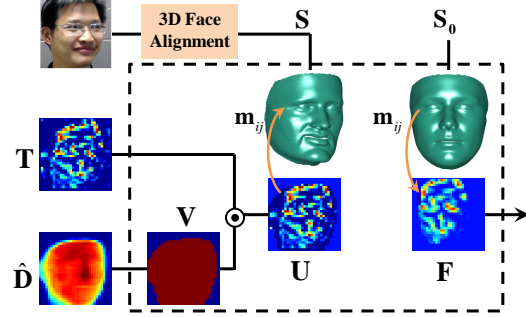


Figure 5. The non-rigid registration layer.

### 3.4. Non-rigid Registration Layer

We design a new non-rigid registration layer to prepare data for the RNN part. This layer utilizes the estimated dense 3D shape to align the activations or feature maps from the CNN part. This layer is important to ensure that the RNN tracks and learns the changes of the activations for the *same facial area* across time, as well as across all subjects.

As shown in Fig. 5, this layer has three inputs: the feature map  $\mathbf{T} \in \mathbb{R}^{32 \times 32}$ , the depth map  $\hat{\mathbf{D}}$  and the 3D shape  $\mathbf{S}$ . Within this layer, we first threshold the depth map and generate a binary mask  $\mathbf{V} \in \mathbb{R}^{32 \times 32}$ :

$$\mathbf{V} = \hat{\mathbf{D}} \geq threshold. \quad (7)$$

Then, we compute the inner product of the binary mask and the feature map  $\mathbf{U} = \mathbf{T} \odot \mathbf{V}$ , which essentially utilizes the depth map as a visibility indicator for each pixel in the feature map. If the depth value for one pixel is less than the threshold, we consider that pixel to be invisible. Finally, we frontalize  $\mathbf{U}$  by utilizing the estimated 3D shape  $\mathbf{S}$ ,

$$\mathbf{F}(i, j) = \mathbf{U}(\mathbf{S}(\mathbf{m}_{ij}, 1), \mathbf{S}(\mathbf{m}_{ij}, 2)), \quad (8)$$

where  $\mathbf{m} \in \mathbb{R}^K$  is the pre-defined list of  $K$  indexes of the face area in  $\mathbf{S}_0$ , and  $\mathbf{m}_{ij}$  is the corresponding index of pixel  $i, j$ . We utilize  $\mathbf{m}$  to project the masked activation map  $\mathbf{U}$  to the frontalized image  $\mathbf{F}$ . This proposed non-rigid registration layer has three contributions to our network:

- ◊ By applying the non-rigid registration, the input data are aligned and the RNN can compare the feature maps without concerning about the facial pose or expression. In other words, it can learn the temporal changes in the activations of the feature maps for the same facial area.

- ◊ The non-rigid registration removes the background area in the feature map. Hence the background area would not participate in RNN learning, although the background information is already utilized in the layers of the CNN part.

- ◊ For spoof faces, the depth maps are likely to be closer to zero. Hence, the inner product with the depth maps substantially weakens the activations in the feature maps, which makes it easier for the RNN to output zero rPPG signals. Likewise, the back propagation from the rPPG loss

Table 1. The comparison of our collected SiW dataset with existing datasets for face anti-spoofing.

Dataset	Year	# of subj.	# of sess.	# of live/attack vid. (V), ima. (I)	Pose range	Different expres.	Extra light.	Display devices	Spoof attacks
NUAA [43]	2010	15	3	5105/7509 (I)	Frontal	No	Yes	-	Print
CASIA-MFSD [50]	2012	50	3	150/450 (V)	Frontal	No	No	iPad	Print, Replay
Replay-Attack [17]	2012	50	1	200/1000 (V)	Frontal	No	Yes	iPhone 3GS, iPad	Print, 2 Replay
MSU-MFSD [45]	2015	35	1	110/330 (V)	Frontal	No	No	iPad Air, iPhone 5S	Print, 2 Replay
MSU-USSA [38]	2016	1140	1	1140/9120 (I)	$[-45^\circ, 45^\circ]$	Yes	Yes	MacBook, Nexus 5, Nvidia Shield Tablet	2 print, 6 Replay
Oulu-NPU [14]	2017	55	3	1980/3960 (V)	Frontal	No	Yes	Dell 1905FP, Macbook Retina	2 Print, 2 Replay
SiW	2018	165	4	1320/3300 (V)	$[-90^\circ, 90^\circ]$	Yes	Yes	iPad Pro, iPhone 7, Galaxy S8, Asus MB168B	2 Print, 4 Replay

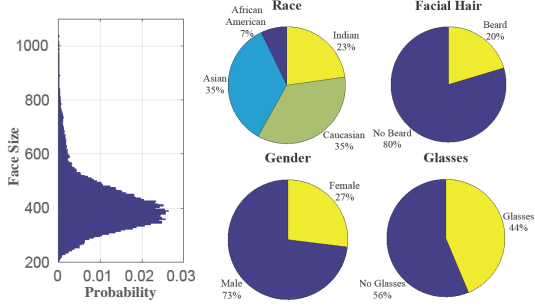


Figure 6. The statistics of the subjects in the SiW database. Left side: The histogram shows the distribution of the face sizes.

also encourages the CNN part to generate zero depth maps for either all frames, or one pixel location in majority of the frames within an input sequence.

#### 4. Collection of Face Anti-Spoofing Database

With the advance of sensor technology, existing anti-spoofing systems can be vulnerable to emerging high-quality spoof mediums. One way to make the system robust to these attacks is to collect new high-quality databases. In response to this need, we collect a new face anti-spoofing database named Spoof in the Wild (SiW) database, which has multiple advantages over previous datasets as in Tab. 1. First, it contains substantially more live subjects with diverse races, e.g., 3 times of the subjects of Oulu-NPU. Note that MSU-USSA is constructed using existing images of celebrities without capturing live faces. Second, live videos are captured with two high-quality cameras (Canon EOS T6, Logitech C920 webcam) with different PIE variations.

SiW provides live and spoof 30-fps videos from 165 subjects. For each subject, we have 8 live and 20 spoof videos, in total 4,620 videos. Some statistics of the subjects are shown in Fig. 6. The live videos are collected in four sessions. In Session 1, the subject moves his head with varying distances to the camera. In Session 2, the subject changes the yaw angle of the head within  $[-90^\circ, 90^\circ]$ , and makes different face expressions. In Sessions 3, 4, the subject repeats the Sessions 1, 2, while the collector moves the point light source around the face from different orientations.

The live videos captured by both cameras are of  $1,920 \times 1,080$  resolution. We provide two print and four replay video attacks for each subject, with examples shown in Fig. 7. To generate different qualities of print attacks, we

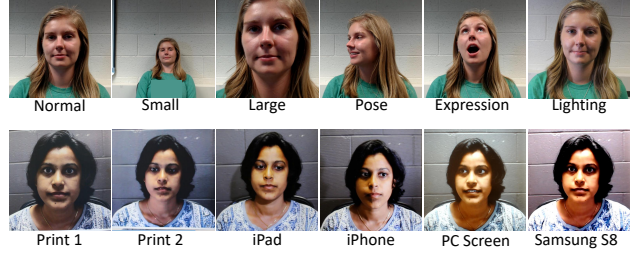


Figure 7. Example live (top) and spoof (bottom) videos in SiW.

capture a high-resolution image ( $5,184 \times 3,456$ ) for each subject and use it to make a high-quality print attack. Also, we extract a frontal-view frame from a live video for lower-quality print attack. We print the images with an HP color LaserJet M652 printer. The print attack videos are captured by holding printed papers still or warping them in front of the cameras. To generate high-quality replay attack videos, we select four spoof mediums: Samsung Galaxy S8, iPhone 7, iPad Pro, and PC (Asus MB168B) screens. For each subject, we randomly select two of the four high-quality live videos to display in the spoof mediums.

### 5. Experimental Results

#### 5.1. Experimental Setup

**Databases** We evaluate our method on multiple databases to demonstrate its generalizability. We utilize SiW and Oulu databases [14] as new high-resolution databases and perform intra and cross testing between them. Also, we use the CASIA-MFSD [50] and Replay-Attack [17] databases for cross testing and comparing with the state of the art.

**Parameter setting** The proposed method is implemented in TensorFlow [3] with a constant learning rate of  $3e-3$ , and 10 epochs of the training phase. The batch size of the CNN stream is 10 and that of the CNN-RNN stream is 2 with  $N_f$  being 5. We randomly initialize our network by using a normal distribution with zero mean and std of 0.02. We set  $\lambda$  in Eq. 6 to 0.015 and  $threshold$  in Eq. 7 to 0.1.

**Evaluation metrics** To compare with prior works, we report our results with the following metrics: Attack Presentation Classification Error Rate  $APCER$  [2], Bona Fide Presentation Classification Error Rate  $BPCER$  [2],  $ACER = \frac{APCER+BPCER}{2}$  [2], and Half Total Error Rate  $HTER$ . The  $HTER$  is half of the summation of the False Rejection Rate (FRR) and the False Acceptance Rate (FAR).

Table 2. TDR at different FDRs, cross testing on Oulu Protocol 1.

FDR	1%	2%	10%	20%
Model 1	8.5%	18.1%	71.4%	81.0%
Model 2	40.2%	46.9%	78.5%	93.5%
Model 3	39.4%	42.9%	67.5%	87.5%
Model 4	<b>45.8%</b>	<b>47.9%</b>	<b>81%</b>	<b>94.2%</b>

Table 3. ACER of our method at different  $N_f$ , on Oulu Protocol 2.

Test \ Train	5	10	20
	5	10	20
5	4.16%	4.16%	3.05%
10	4.02%	3.61%	2.78%
20	4.10%	3.67%	2.98%

## 5.2. Experimental Comparison

### 5.2.1 Ablation Study

**Advantage of proposed architecture** We compare four architectures to demonstrate the advantages of the proposed loss layers and non-rigid registration layer. *Model 1* has an architecture similar to the CNN part in our method (Fig. 3), except that it is extended with additional pooling layers, fully connected layers, and softmax loss for binary classification. *Model 2* is the CNN part in our method with a depth map loss function. We simply use  $||\hat{D}||_2$  for classification. *Model 3* contains the CNN and RNN parts without the non-rigid registration layer. Both of the depth map and rPPG loss functions are utilized in this model. However, the RNN part would process unregistered feature maps from the CNN. *Model 4* is the proposed architecture.

We train all four models with the live and spoof videos from 20 subjects of SiW. We compute the cross-testing performance of all models on Protocol 1 of Oulu database. The TDR at different FDR are reported in Tab. 2. *Model 1* has a poor performance due to the binary supervision. In comparison, by only using the depth map as supervision, *Model 2* achieves substantially better performance. However, after adding the RNN part with the rPPG supervision, our proposed *Model 4* can further the performance improvement. By comparing *Model 4* and 3, we can see the advantage of the non-rigid registration layer. It is clear that the RNN part cannot use feature maps directly for tracking the changes in the activations and estimating the rPPG signals.

**Advantage of longer sequences** To show the advantage of utilizing longer sequences for estimating the rPPG, we train and test our model when the sequence length  $N_f$  is 5, 10, or 20, using intra-testing on Oulu Protocol 2. From Tab. 3, we can see that by increasing the sequence length, the ACER decreases due to more reliable rPPG estimation. Despite the benefit of longer sequences, in practice, we are limited by the GPU memory size, and forced to decrease the image size to  $128 \times 128$  for all experiments in Tab. 3. Hence, we set  $N_f$  to be 5 with the image size of  $256 \times 256$  in subsequent experiments, due to importance of higher resolution (e.g, a lower ACER of 2.5% in Tab. 4 is achieved than 4.16%).

Table 4. The intra-testing results on four protocols of Oulu.

Prot.	Method	APCER (%)	BPCER (%)	ACER (%)
1	CPqD	2.9	10.8	6.9
	GRADIANT	<b>1.3</b>	12.5	6.9
	Proposed method	1.6	<b>1.6</b>	<b>1.6</b>
2	MixedFASNet	9.7	2.5	6.1
	Proposed method	<b>2.7</b>	2.7	2.7
	GRADIANT	3.1	<b>1.9</b>	<b>2.5</b>
3	MixedFASNet	$5.3 \pm 6.7$	$7.8 \pm 5.5$	$6.5 \pm 4.6$
	GRADIANT	<b><math>2.6 \pm 3.9</math></b>	$5.0 \pm 5.3$	$3.8 \pm 2.4$
	Proposed method	$2.7 \pm 1.3$	<b><math>3.1 \pm 1.7</math></b>	<b><math>2.9 \pm 1.5</math></b>
4	Massy_HNU	$35.8 \pm 35.3$	<b><math>8.3 \pm 4.1</math></b>	$22.1 \pm 17.6$
	GRADIANT	<b><math>5.0 \pm 4.5</math></b>	$15.0 \pm 7.1$	$10.0 \pm 5.0$
	Proposed method	$9.3 \pm 5.6$	$10.4 \pm 6.0$	<b><math>9.5 \pm 6.0</math></b>

Table 5. The intra-testing results on three protocols of SiW.

Prot.	Subset	Subject #	Attack	APCER (%)	BPCER (%)	ACER (%)
1	Train	90	First 60 Frames	3.58	3.58	3.58
	Test	75	All			
2	Train	90	3 display	$0.57 \pm 0.69$	$0.57 \pm 0.69$	$0.57 \pm 0.69$
	Test	75	1 display			
3	Train	90	print (display)	$8.31 \pm 3.81$	$8.31 \pm 3.80$	$8.31 \pm 3.81$
	Test	75	display (print)			

### 5.2.2 Intra Testing

We perform intra testing on Oulu and SiW databases. For Oulu, we follow the four protocols [10] and report their *APCER*, *BPCER* and *ACER*. Tab. 4 shows the comparison of our proposed method and the best two methods for *each* protocol respectively, in the face anti-spoofing competition [10]. Our method achieves the lowest *ACER* in 3 out of 4 protocols. We have slightly worse *ACER* on Protocol 2. To set a baseline for future study on SiW, we define three protocols for SiW. The Protocol 1 deals with variations in face pose and expression. We train using the first 60 frames of the training videos that are mainly frontal view faces, and test on all testing videos. The Protocol 2 evaluates the performance of cross spoof medium of replay attack. The Protocol 3 evaluates the performance of cross PA, i.e., from print attack to replay attack and vice versa. Tab. 5 shows the protocol definition and our performance of each protocol.

### 5.2.3 Cross Testing

To demonstrate the generalization of our method, we perform multiple cross-testing experiments. Our model is trained with live and spoof videos of 80 subjects in SiW, and test on all protocols of Oulu. The *ACER* on Protocol 1-4 are respectively: 10.0%, 14.1%,  $13.8 \pm 5.7\%$ , and  $10.0 \pm 8.8\%$ . Comparing these cross-testing results to the *intra-testing* results in [10], we are ranked sixth on the average *ACER* of four protocols, among the 15 participants of the face anti-spoofing competition. Especially on Protocol 4, the hardest one among all protocols, we achieve the *same ACER* of 10.0% as the top performer. This is a notable result since cross testing is known to be substantially harder than intra testing, and yet our cross-testing result is comparable with the top intra-testing performance. This demonstrates the generalization ability of our learnt model.

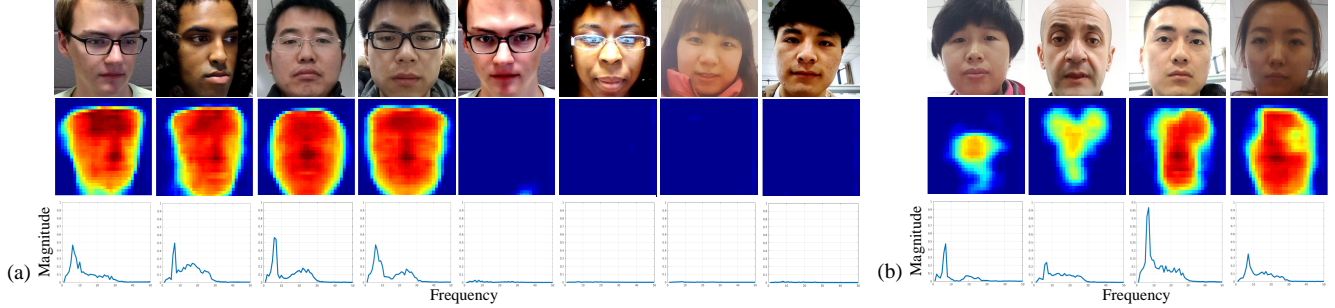


Figure 8. (a) 8 successful anti-spoofing examples and their estimated depth maps and rPPG signals. (b) 4 failure examples: the first two are live and the other two are spoof. Note our ability to estimate discriminative depth maps and rPPG signals.

Table 6. Cross testing on CASIA-MFSD vs. Replay-Attack.

Method	Train	Test	Train	Test
	CASIA-MFSD	Replay Attack	Replay Attack	CASIA-MFSD
Motion [19]		50.2%		47.9%
LBP [19]		55.9%		57.6%
LBP-TOP [19]		49.7%		60.6%
Motion-Mag [8]		50.1%		47.0%
Spectral cubes [40]		34.4%		50.0%
CNN [48]		48.5%		45.5%
LBP [11]		47.0%		39.6%
Colour Texture [12]		30.3%		37.7%
Proposed method		<b>27.6%</b>		<b>28.4%</b>

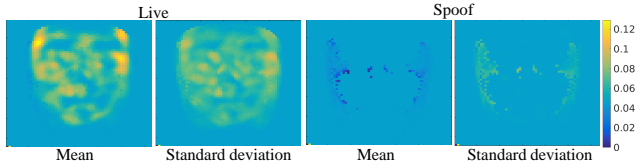


Figure 9. Mean/Std of frontalized feature maps for live and spoof.

Furthermore, we utilize the CASIA-MFSD and Replay-Attack databases to perform cross testing between them, which is widely used as a cross-testing benchmark. Tab. 6 compares the cross-testing *HTER* of different methods. Our proposed method reduces the cross-testing errors on the Replay-Attack and CASIA-MFSD databases by 8.9% and 24.6% respectively, relative to the previous SOTA.

#### 5.2.4 Visualization and Analysis

In the proposed architecture, the frontalized feature maps are utilized as input to the RNN part and are supervised by the rPPG loss function. The values of these maps can show the importance of different facial areas to rPPG estimation. Fig. 9 shows the mean and standard deviation of frontalized feature maps, computed from 1,080 live and spoof videos of Oulu. We can see that the side areas of forehead and cheek have higher influence for rPPG estimation.

While the goal of our system is to detect PAs, our model is trained to estimate the auxiliary information. Hence, in addition to anti-spoof, we also like to evaluate the accuracy of auxiliary information estimation. For this purpose, we calculate the accuracy of estimating depth maps and rPPG signals, for testing data in Protocol 2 of Oulu. As shown in Fig. 10, the accuracy for both estimation in spoof data

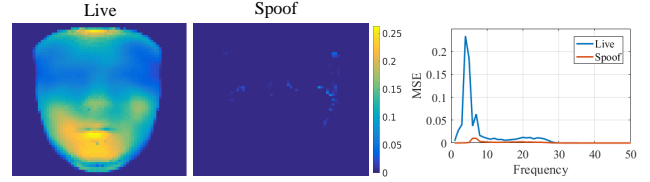


Figure 10. The MSE of estimating depth maps and rPPG signals.

is high, while that of the live data is relatively lower. Note that the depth estimation of the mouth area has more errors, which is consistent with the fewer activations of the same area in Fig. 9. Examples of successful and failure cases in estimating depth maps and rPPG signals are shown in Fig. 8.

Finally, we conduct statistical analysis on the failure cases, since our system can determine potential causes using the auxiliary information. With Protocol 2 of Oulu, we identify 31 failure cases (2.7% *ACER*). For each case, we calculate whether anti-spoofing using its depth map or rPPG signal would fail if that information alone is used. In total,  $\frac{29}{31}$ ,  $\frac{13}{31}$ , and  $\frac{11}{31}$  samples fail due to depth map, rPPG signals, or both. This indicates the future research direction.

## 6. Conclusions

This paper identifies the importance of auxiliary supervision to deep model-based face anti-spoofing. The proposed network combines CNN and RNN architectures to jointly estimate the depth of face images and rPPG signal of face video. We introduce the SiW database that contains more subjects and variations than prior databases. Finally, we experimentally demonstrate the superiority of our method.

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



## References

- [1] Explainable Artificial Intelligence (XAI). <https://www.darpa.mil/program/explainable-artificial-intelligence>. 1
- [2] ISO/IEC JTC 1/SC 37 Biometrics. information technology biometric presentation attack detection part 1: Framework. international organization for standardization, 2016. <https://www.iso.org/obp/ui/iso>. 6
- [3] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016. 6
- [4] A. Agarwal, R. Singh, and M. Vatsa. Face anti-spoofing using Haralick features. In *BTAS*, 2016. 2
- [5] Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu. Face anti-spoofing using patch and depth-based CNNs. In *IJCB*, 2017. 2
- [6] W. Bao, H. Li, N. Li, and W. Jiang. A liveness detection method for face recognition based on optical flow field. In *IASP*, pages 233–236, 2009. 2
- [7] S. Bharadwaj, T. Dhamecha, M. Vatsa, and R. Singh. Face anti-spoofing via motion magnification and multifeature videolet aggregation. 2014. 2
- [8] S. Bharadwaj, T. I. Dhamecha, M. Vatsa, and R. Singh. Computationally efficient face spoofing detection with motion magnification. In *CVPRW*, pages 105–110, 2013. 8
- [9] S. Bobbia, Y. Benezeth, and J. Dubois. Remote photoplethysmography based on implicit living skin tissue segmentation. In *ICPR*, pages 361–365, 2016. 2
- [10] Z. Boulkenafet. A competition on generalized software-based face presentation attack detection in mobile scenarios. In *IJCB*, 2017. 7
- [11] Z. Boulkenafet, J. Komulainen, and A. Hadid. Face anti-spoofing based on color texture analysis. In *ICIP*, pages 2636–2640, 2015. 2, 8
- [12] Z. Boulkenafet, J. Komulainen, and A. Hadid. Face spoofing detection using colour texture analysis. *IEEE Trans. Inf. Forens. Security*, 11(8):1818–1830, 2016. 2, 8
- [13] Z. Boulkenafet, J. Komulainen, and A. Hadid. Face anti-spoofing using speeded-up robust features and Fisher vector encoding. *IEEE Signal Process. Letters*, 24(2):141–145, 2017. 1, 2
- [14] Z. Boulkenafet, J. Komulainen, L. Li, X. Feng, and A. Hadid. OULU-NPU: A mobile face presentation attack database with real-world variations. In *FG*, 2017. 2, 6
- [15] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Face-warehouse: A 3D facial expression database for visual computing. *IEEE Trans. Vis. Comput. Graphics*, 20(3):413–425, 2014. 3
- [16] G. Chetty and M. Wagner. Multi-level liveness verification for face-voice biometric authentication. In *BC*, 2006. 1
- [17] I. Chingovska, A. Anjos, and S. Marcel. On the effectiveness of local binary patterns in face anti-spoofing. In *BIOSIG*, 2012. 2, 6
- [18] T. de Freitas Pereira, A. Anjos, J. M. De Martino, and S. Marcel. LBP-TOP based countermeasure against face spoofing attacks. In *ACCV*, pages 121–132, 2012. 1, 2
- [19] T. de Freitas Pereira, A. Anjos, J. M. De Martino, and S. Marcel. Can face anti-spoofing countermeasures work in a real world scenario? In *ICB*, 2013. 1, 2, 8
- [20] G. de Haan and V. Jeanne. Robust pulse rate from chrominance-based rPPG. *IEEE Trans. Biomedical Engineering*, 60(10):2878–2886, 2013. 2, 3, 4
- [21] L. Feng, L.-M. Po, Y. Li, X. Xu, F. Yuan, T. C.-H. Cheung, and K.-W. Cheung. Integration of image quality and motion cues for face anti-spoofing: A neural network approach. *J. Visual Communication and Image Representation*, 38:451–460, 2016. 1, 2
- [22] R. W. Frischholz and U. Dieckmann. BioID: a multimodal biometric identification system. *J. Computer*, 33(2):64–68, 2000. 1
- [23] R. W. Frischholz and A. Werner. Avoiding replay-attacks in a face recognition system using head-pose estimation. In *AMFGW*, pages 234–235, 2003. 1
- [24] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4
- [25] A. Jourabloo and X. Liu. Pose-invariant face alignment via CNN-based dense 3D model fitting. *Int. J. Comput. Vision*, 124(2):187–203, 2017. 3
- [26] K. Kollreider, H. Fronthaler, M. I. Faraj, and J. Bigun. Real-time face detection and motion analysis with application in liveness assessment. *IEEE Trans. Inf. Forens. Security*, 2(3):548–558, 2007. 2
- [27] J. Komulainen, A. Hadid, and M. Pietikainen. Context based face anti-spoofing. In *BTAS*, 2013. 1, 2
- [28] J. Komulainen, A. Hadid, M. Pietikainen, A. Anjos, and S. Marcel. Complementary countermeasures for detecting scenic face spoofing attacks. In *ICB*, 2013. 2
- [29] J. Li, Y. Wang, T. Tan, and A. K. Jain. Live face detection based on the analysis of fourier spectra. In *SPIE (BTHI)*, volume 5404, pages 296–304, 2004. 1, 2
- [30] 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 *IPTA*, 2016. 1, 2
- [31] S. Liu, P. C. Yuen, S. Zhang, and G. Zhao. 3D mask face anti-spoofing with remote photoplethysmography. In *ECCV*, pages 85–100, 2016. 2, 3
- [32] Y. Liu, A. Jourabloo, W. Ren, and X. Liu. Dense face alignment. In *ICCVW*, pages 1619–1628, 2017. 3, 4
- [33] J. Määtä, A. Hadid, and M. Pietikainen. Face spoofing detection from single images using micro-texture analysis. In *IJCB*, 2011. 1, 2
- [34] V. Mirjalili and A. Ross. Soft biometric privacy: Retaining biometric utility of face images while perturbing gender. In *ICB*, 2017. 1
- [35] E. M. Nowara, A. Sabharwal, and A. Veeraraghavan. Ppgsecure: Biometric presentation attack detection using photoplethysmograms. In *FG*, pages 56–62, 2017. 2, 3
- [36] G. Pan, L. Sun, Z. Wu, and S. Lao. Eyeblink-based anti-spoofing in face recognition from a generic webcam. In *ICCV*, 2007. 2

- [37] K. Patel, H. Han, and A. K. Jain. Cross-database face anti-spoofing with robust feature representation. In *CCBR*, pages 611–619, 2016. 1, 2
- [38] K. Patel, H. Han, and A. K. Jain. Secure face unlock: Spoof detection on smartphones. *IEEE Trans. Inf. Forens. Security*, 11(10):2268–2283, 2016. 1, 2, 6
- [39] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3D face model for pose and illumination invariant face recognition. In *AVSS*, pages 296–301, 2009. 3
- [40] A. Pinto, H. Pedrini, W. R. Schwartz, and A. Rocha. Face spoofing detection through visual codebooks of spectral temporal cubes. *IEEE Trans. Image Process.*, 24(12):4726–4740, 2015. 8
- [41] L.-M. Po, L. Feng, Y. Li, X. Xu, T. C.-H. Cheung, and K.-W. Cheung. Block-based adaptive ROI for remote photoplethysmography. *J. Multimedia Tools and Applications*, pages 1–27, 2017. 2
- [42] R. Shao, X. Lan, and P. C. Yuen. Deep convolutional dynamic texture learning with adaptive channel-discriminability for 3D mask face anti-spoofing. In *IJCB*, 2017. 2
- [43] X. Tan, Y. Li, J. Liu, and L. Jiang. Face liveness detection from a single image with sparse low rank bilinear discriminative model. In *ECCV*, pages 504–517, 2010. 2, 6
- [44] S. Tulyakov, X. Alameda-Pineda, E. Ricci, L. Yin, J. F. Cohn, and N. Sebe. Self-adaptive matrix completion for heart rate estimation from face videos under realistic conditions. In *CVPR*, pages 2396–2404, 2016. 2
- [45] D. Wen, H. Han, and A. Jain. Face Spoof Detection with Image Distortion Analysis. *IEEE Trans. Inf. Forens. Security*, 10(4):746–761, 2015. 2, 6
- [46] B.-F. Wu, Y.-W. Chu, P.-W. Huang, M.-L. Chung, and T.-M. Lin. A motion robust remote-PPG approach to driver’s health state monitoring. In *ACCV*, pages 463–476, 2016. 2
- [47] Z. Xu, S. Li, and W. Deng. Learning temporal features using LSTM-CNN architecture for face anti-spoofing. In *ACPR*, pages 141–145. IEEE, 2015. 2
- [48] J. Yang, Z. Lei, and S. Z. Li. Learn convolutional neural network for face anti-spoofing. *arXiv preprint arXiv:1408.5601*, 2014. 1, 2, 8
- [49] J. Yang, Z. Lei, S. Liao, and S. Z. Li. Face liveness detection with component dependent descriptor. In *ICB*, 2013. 1, 2
- [50] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li. A face antispoofing database with diverse attacks. In *ICB*, pages 26–31, 2012. 2, 6
- [51] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li. Face alignment across large poses: A 3D solution. In *CVPR*, pages 146–155, 2016. 3