Exploiting temporal and depth information for multi-frame face anti-spoofing

Zezheng Wang¹, Chenxu Zhao¹*, Yunxiao Qin¹, Qiusheng Zhou¹, Zhen Lei²
¹Jingdong Finance
²Chinese Academy of Sciences
{wangzezheng1, zhaochenxu1, qinyunxiao, zhouqiusheng}@jd.com, zlei@nlpr.ia.ac.cn

Abstract

Face anti-spoofing is significant to the security of face recognition systems. By utilizing pixel-wise supervision, depth supervised face anti-spoofing reasonably contains more generalization than binary classification does. Without considering the importance of sequential information in depth recovery, previous depth supervised methods only regard depth as an auxiliary supervision in the single frame. In this paper, we propose a depth supervised face anti-spoofing model in both spatial and temporal domains. The temporal information from multi-frames is exploited and incorporated to improve facial depth recovery, so that more robust and discriminative features can be extracted to classify the living and spoofing faces. Extensive experiments indicate that our approach achieves state-of-the-art results on standard benchmarks.

1. Introduction

Nowadays, face recognition system is well-known to the public for its convenience. However, most of existing face recognition systems are so vulnerable to the face spoofing that attackers can easily deceive face recognition system by using presentation attacks (PA), including printing a face on paper (print attack), replaying a face on a digital device (replay attack), or bringing a 3D-mask (3D-mask attack). Such PAs make face recognition unsafe, especially in the fields of money payment and privacy verification, which directly affects the nature interests of users. Therefore, face anti-spoofing turns to be quite important for face recognition system to confirm its security.

To counteract face spoofing attacks, researchers proposed a great number of face anti-spoofing methods [5, 13] to discriminate the living face and spoofing face. Previous approaches can be classified into two categories. The first is the traditional face anti-spoofing methods. This kind of method majors in training shallow classifier with hand-crafted local features, i.e., LBP, SIFT, and SURF. Due to only utilizing texture characteristic, it’s insufficient for these models to generalize to different kinds of replaying mediums, attack mediums, and various replaying environment. The second is CNN-based methods. As a number of face anti-spoofing databases, e.g. NUAA, CASIA, ReplayAttack, MSU-MFSD, OULU-NPU, and SiW, were collected, researchers start to focus on deep learning-based face anti-spoofing method. In contrast to texture-based methods, CNN-based methods can learn discriminative representations with aforementioned spoofing databases. However, common CNN-based methods treat face anti-spoofing as a matter of binary classification - spoofing as 0 and living as 1, and train the neural network supervised by softmax loss. However, binary classification with softmax loss can’t excavate the nature of spoof patterns [19], which consist of skin detail loss, color distortion, moiré pattern, shape deformation and spoof artifacts. Indeed, it’s hard for binary CNNs to obtain the explanation and rationality for the decision. To address the issue, the depth map of face (facial depth) is a better supervision. Atoum [2] and Liu [19] proposed depth supervised CNN architecture to obtain more information. Both of them utilize facial depth as supervision
and design an FCN structure to regress the facial depth. Obviously, using depth as auxiliary information improves the performance of face anti-spoofing.

In this paper, we emphasize the advantages of depth map in face anti-spoofing. To compare the real scenes of living scene with spoofing scene intuitively, images of living face contain face-like depth, whereas spoofing face images on print or replay carrier only have plat or planar depth. For example, when a person looks at frontal-view face, the nose is closer to the camera than the cheek, while attackers (except 3D-mask attackers) can just present faces with all pixels on the same depth. In the meantime, dense supervision offers more detailed guidance and robust supervision to obtain the spoof patterns, whereas binary label can only assist network to learn inexplicable representation. It’s worth noting that the task of depth supervised face anti-spoofing is different from the monocular depth estimation and 3D face reconstruction. Monocular depth estimation aims to obtain object distance to the camera by naturally seeking semantic information from the original image, while depth supervision in face anti-spoofing plays a supporting role to exploit the essential spoofing cues.

Depth in face anti-spoofing is indeed significant. Atoum [2] combines depth-based CNN and patch-based CNN for rich appearance features. Liu [19] designs a CNN-RNN model to estimate the facial depth with pixel-wise supervision and estimate rPPG signals with sequence-wise supervision. Both Atoum [2] and Liu [19] use facial depth as auxiliary supervision in a common style. However, they simply regard depth-based network as subpart of their architecture, by only utilizing single-frame information under depth supervision and ignoring the importance of temporal information in depth recovery. In reality, no matter whether the camera moves or the user moves or both move, sequential frames can be transformed into 3D space representation, due to multi-view points in the 3D space like SLAM. For example, presentation attackers including print attack and replay attack are distinct from live presentation. Under this circumstance, planar carrier can be easily detected in sequential dims. Specially in figure 1, we can see that there are apparently different temporal representations between living and spoofing scenes. In summary, discriminative temporal features are vital for network to grasp spoofing cues. In order to exploit significant temporal information under pixel-wise supervision for presentation attack detection, we propose a novel depth supervised neural network architecture with optical flow guided feature (OFF) block and convolution gated recurrent units (ConvGRU) module to extract the temporal information of adjacent frames and that of long-sequence frames, respectively.

The main contributions of this work include:

- We propose a novel depth supervised architecture with OFF block and ConvGRU module to cover significant temporal information and estimate the facial depth.
- We achieve the state-of-the-art performance on standard face anti-spoofing benchmarks.

2. Related Work

We review related face anti-spoofing work in three groups: binary supervised methods, depth supervised methods, and temporal-based methods.

**Binary supervised Methods** Since face anti-spoofing is essentially a binary classification problem, most of previous anti-spoofing methods purely train a classifier under binary supervision, i.e., spoofing face as 0 and living face as 1. In subdivided areas, binary classifiers contain traditional classifiers and neural network. Prior works usually combine hand-crafted features, such as LBP [9, 10, 21], SIFT [25], SURF [5], HoG [17, 34], DoG [26, 31] with traditional classifiers, such as SVM and Random Forest. Due to that aforementioned features might be sensitive to different environment, such as camera devices, lighting conditions and PAs, traditional methods don’t perform well in generalization.

As the development of hardware and the increase of the amount of data, CNN has proven to be a successful method in a number of computer vision problems. Recently, CNN is also popularly used in face anti-spoofing tasks [20, 22, 13, 11, 18, 24, 33]. However, most of the deep learning methods still consider face anti-spoofing as a binary classification problem with softmax loss. Both [18] and [24] fine-tune a pre-trained VGG-face model and take it as a feature extractor for subsequent classification. Even Nagpal et al. [22] exploits the influence of different network architecture and different hyperparameters on face anti-spoofing. Feng [11] and Li [18] also design different kinds of face images to feed into the CNN network for discriminatively learning on living faces and spoofing faces.

**Depth supervised Methods** Compared with binary supervised face anti-spoofing methods, depth supervised methods have a lot of advantages. Atoum [2] utilizes depth map of face as supervision signals for the first time. They propose a two-stream CNN-based approach for face anti-spoofing, by extracting the local features and holistic depth maps from the face images. That’s to say, they combine depth-based CNN and patch-based CNN from single frame to obtain discriminative representation to distinguish live vs. spoof. This work shows that depth estimation is beneficial for the model of face anti-spoofing to obtain promising results especially on higher-resolution images.

Liu [19] proposes a face anti-spoofing method with a combination of spatial perspective (depth) and temporal perspective (rPPG). They regard facial depth as an auxiliary supervision, as well as rPPG signals. As for temporal
information, they use a simple RNN to learn corresponding rPPG signals. Due to the simple pattern of sequence-processing, they have to implement a non-rigid registration layer to remove the influence of facial poses or expressions, which ignores that unnatural changes of facial poses or expressions are significant spoofing cues.

**Temporal-based Methods** Temporal-based information is vital to face anti-spoofing task. [23, 24, 28] major in the movement of key parts of the face. For example, [23, 24] make spoofing decisions based on eye-blinking. These methods are vulnerable to replay attack due to excessively relying on one aspect. Gan [13] proposes a 3D convolution network to distinguish the live vs. spoof. 3D convolution network is a stacked structure to learn the temporal features in a supervised pattern, which depends on significant amount of data and performs unsatisfactorily on small database. Xu [32] propose an architecture combining LSTM units with CNN for binary classification. Feng [11] presents a work which takes optical flow magnitude map and Shearlet feature as inputs of CNN. Their work shows that optical flow map presents obvious difference between living faces and different spoofing faces. All prior temporal-based methods are incapable to catch valid temporal information with a well-designed structure. In order to conquer this problem, we propose a neural network which combines the short-term motion module and long-term motion module to seek excellent facial motion to estimate facial depth and to detect PAs.

### 3. Approach

In our proposed approach, we make rational use of temporal information to recover facial depth, which is used as a supervision signal to catch spoof patterns for face anti-spoofing. As shown in figure 4, our proposed model mainly consists of two modules. One is single-frame part, majoring in seeking spoofing cues under the static depth supervision. The other is multi-frame part, consisting of optical flow guided feature (OFF) block as short-term motion module and convolutional gated recurrent units (ConvGRU) as long-term motion module. The model can exploit spoofing cues in both spatial and temporal domains under the depth supervision, successfully.

#### 3.1. Single-frame Part

Single-frame part is important to learn shallow spoofing information. In this work, we train a simple CNN to regress the depth map instead of traditional classification with softmax loss. In the following, we introduce our depth supervised single-frame architecture in two aspects: depth generation and network structure.

#### 3.1.1 Depth Generation

We supervise our single-frame network with facial depth map, which reflects face location and 3D shape of the face in the 2D plane image, and will reveal useful information in the judgement of real face or not. To differentiate living faces from spoofing faces, we define living depth map in a normalization of \([0, 1]\), and set spoofing depth map as 0 [19]. The definition of our depth map is originally beneficial to the testing inference by meaning the facial depth as living score. Simultaneously, we emphasize the essentiality of depth supervision again, that is, depth map provides pixel-wise supervision.

In this module, we adopt dense face alignment method PRNet [12] to estimate the 3D shape of the living face. PRNet can be used to project the 3D shape of a complete face into UV space. Through this method, we obtain a group of vertices \(V_{n×3}\) representing \(n\) 3D coordinates of facial keypoints. Since these coordinates are sparse when mapped to the plane 2D image, we tend to implement interpolation on \(V_{n×3}\) for dense face coordinates. As such, we specially calculate the min/max values of \(x\) and \(y\) in \(V_{n×3}\), and then build a meshgrid based on them. Afterwards, we interpolate the meshgrid for the smooth \(z\) value. By mapping the interpolated \(V_{n′×3}\) to a plane 2D image, and normalizing the face depth map to \([0, 1]\) by minimal and maximal \(z\) values, the generated facial depth map can be finally represented by \(D ∈ \mathbb{R}^{32×32}\). Figure 2 shows the labels and corresponding inputs of our proposed model.

#### 3.1.2 Network Structure

As shown in figure 4, There are three main cascaded blocks connected after one convolution in our single-frame part. Each block is composed of three convolution layers and one
pooling layer. We resize pooling features to the pre-defined size of $32 \times 32$ and concatenate them into one tensor, which is designed to regress the depth map followed by subsequent three convolutional groups.

Given an original RGB image $I^{256 \times 256 \times 3}$, we can obtain corresponding estimated depth map $D_{\text{single}} \in \mathbb{R}^{32 \times 32}$ by our single-frame part. Supervised by the generated “ground truth” depth $D \in \mathbb{R}^{32 \times 32}$, we design a novel depth loss consisting of the following two parts. One is squared euclidean norm loss between $D_{\text{single}}$ and $D$, which is an absolute depth regression:

$$L_{\text{absolute}} = ||D_{\text{single}} - D||_2^2,$$

and the other is a contrastive depth regression:

$$L_{\text{contrast}} = \sum_i ||K_i \odot D_{\text{single}} - K_i \odot D||_2^2,$$

where $\odot$ represents depthwise separable convolution operation [15]. $K_i^{\text{contrast}}$ represents the contrastive convolution kernel shown in figure 3, and index $i$ indicates the location of “1” around “-1”. As shown in equation 2 and figure 3, our contrastive depth loss aims to learn the topography of each pixel, which gives constraints to the contrast from the pixel to its neighbors.

Totally, our single-frame loss can be written as:

$$L_{\text{single}} = L_{\text{absolute}} + L_{\text{contrast}},$$

where $L_{\text{single}}$ is the final loss used in our single-frame part.

### 3.2. Multi-frame Part

Generally, face recognition systems are able to store consecutive frames in the manner of video. When it comes to the scene of videos, no matter whether the shooting device moves or the living/spoofing face moves, it can both be regarded as the scene that the shooting device moves around the face. In this way, we can derive more spoof patterns from temporal information according to the contrast movement between shooting device and face. The utilization of multi frames thereby is of great importance to face anti-spoofing.

In this work, we exploit the short-term and long-term motion in temporal domain. Specially, the short-term motion is extracted by optical flow guided feature (OFF) block and the long-term motion is obtained by ConvGRU module.

#### 3.2.1 Short-term Motion

![Figure 4. The pipeline of proposed architecture.](image)

The inputs are consecutive frames in a fixed interval. Our single-frame part aims to extract features at various levels and to output the single-frame estimated facial depth. OFF blocks take single-frame features from two consecutive frames as inputs and calculate short-term motion features. Then the final OFF features are fed into the ConvGRUs to obtain long-term motion information, and output the residual of single-frame facial depth. Finally, the combined estimated multi-frame depth maps are supervised by the depth loss and binary loss in respective manners.

We use OFF [30] block to extract short-term motion. The famous brightness constant constraint in traditional optical flow can be formulated as:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t),$$

![Figure 5. The architecture of our OFF block.](image)
where \( I(x, y, t) \) represents the brightness at the location \((x, y)\) at the frame of time \(t\). The equation denotes that when a pixel at \((x, y)\) moves \((\Delta x, \Delta y)\) in time span of \(\Delta t\), the brightness doesn't change. According to Taylor formula, equation 4 can be derived to:

\[
\frac{\partial I(p)}{\partial x} \Delta x + \frac{\partial I(p)}{\partial y} \Delta y + \frac{\partial I(p)}{\partial t} \Delta t = 0, 
\]

where \( p = (x, y, t) \). By dividing \(\Delta t\) in both sides of the equation, we obtain:

\[
\frac{\partial I(p)}{\partial x} v_x + \frac{\partial I(p)}{\partial y} v_y + \frac{\partial I(p)}{\partial t} = 0, 
\]

where \( v_x \) and \( v_y \) represent the two dimensional velocity of the pixel at \(p\). \( \frac{\partial I(p)}{\partial x}, \frac{\partial I(p)}{\partial y} \) and \( \frac{\partial I(p)}{\partial t} \) are gradients of \(I(p)\) at \(x, y\) and time axes, respectively. \((v_x, v_y)\) is exactly called optical flow. In equation 6, we can see that \(\vec{F}\) is orthogonal to \((v_x, v_y, 1)\). Obviously, \(\vec{F}\) is guided by the optical flow. As optical flow methods usually apply low-level or high-level features to match the flow, we replace image \(I\) by features in equations above. \(\vec{F}(p)\) can be reformulated as:

\[
\vec{F}(I; w; t)(p) = \frac{\partial F(I; w; t)(p)}{\partial x}, \frac{\partial F(I; w; t)(p)}{\partial y}, \frac{\partial F(I; w; t)(p)}{\partial t}, 
\]

where \(F\) is the extracted feature from image \(I\), and \(w\) is parameters of the feature extraction function. \(\vec{F}(I; w; t)(p)\) is called optical flow guided features (OFF) [30], which encodes spatial gradients and temporal gradient guided by the feature-level optical flow.

Since the computation of original optical flow is too slow to meet the demand of real time, we adopt OFF [30] in this work in consideration of its efficiency. Besides, in order to well adapt to the task of face anti-spoofing and depth, we involve the reduced original feature \(F_l(t)\) in the sub-modules. To get more spatial information, spatial gradient \(F_l(t + \Delta t)\) at frame \(t + \Delta t\) is also included. Then, by merging the lower-level OFF feature \(OFF_{l-1}(t)\) and subsequent operation, we figure out the set of sub-modules and current OFF feature \(OFF_l(t)\). In the following of this paper, we represent the output of the final OFF block as \(OFF(t)\).

### 3.2.2 Long-term Motion

As a short-term motion feature, OFF primarily captures the motion information between two consecutive frames, whereas has difficulty in fusing long-sequence motion. In this regard, we resort to convolution gated recurrent unit (ConvGRU) module to gain the long-term motion.

GRU [8] derives from Long Short-term Memory (LSTM) [14], which has simpler structure and fewer trainable parameters. Similar to LSTM, GRU aims to process long sequence information as well.

However, normal GRU discards the spatial information when processing the hidden units, so that we take into account the convolution operation in the hidden layer to deal with the spatiotemporal sequences, named Convolution Gated Recurrent Units (ConvGRU), related to ConvLSTM [29]. The key equations of ConvGRU are shown as follows:

\[
R_t = \sigma(K_r \otimes [H_{t-1}, X_t]), \\
U_t = \sigma(K_u \otimes [H_{t-1}, X_t]), \\
\hat{H}_t = \tanh(K_h \otimes [R_t \ast H_{t-1}, X_t]), \\
H_t = (1 - U_t) \ast H_{t-1} + U_t \ast \hat{H}_t, 
\]

where \(X_t, H_t, U_t\) and \(R_t\) are the matrix of input, output, update gate and reset gate, respectively, \(K_r, K_u, K_h\) are the
kernels in the convolution layer, \( \otimes \) is convolution operation, * denotes element wise product, and \( \sigma \) is the sigmoid activation function. By feeding \( \{O,F(t)\}^{N_f-1} \) into \( \{X_t\}^{N_f-1} \), we set up depth maps \( \{D_{\text{single}}^t\}^{N_f-1} \), where \( D_{\text{multi}}^t = H_t \) and \( N_f \) denotes the number of input frames. Referring to the residual idea, we integrate the single depth map and multi depth map:

\[
D_{\text{fusion}}^t = \alpha \cdot D_{\text{single}}^t + (1 - \alpha) \cdot D_{\text{multi}}^t, \quad \alpha \in [0, 1] \quad (11)
\]

where \( \alpha \) is the weight of \( D_{\text{single}}^t \) in \( D_{\text{fusion}}^t \). Finally, we build up the set of multi-frame depth maps \( \{D_{\text{fusion}}^t\}^{N_f-1} \).

### 3.2.3 Multi-frame Loss

We make the final decision to judge live vs. spoof in the multi-frame part, nevertheless, in view of the potential unclear depth map for discrimination, we hereby insert a binary loss when looking for the difference between living and spoofing depth map. Please note that the depth supervision is decisive, whereas the binary supervision takes assistant role to discriminate the indistinct depth map. On this ground, we establish the multi-frame loss:

\[
L_{\text{multi}}^\text{absolute}(t) = ||D_{\text{fusion}}^t - D^t||^2_2, \quad (12)
\]

\[
L_{\text{multi}}^\text{contrast}(t) = \sum_i ||K_i^\text{contrast} \odot D_{\text{fusion}}^t - K_i^\text{contrast} \odot D^t||^2_2, \quad (13)
\]

\[
L_{\text{multi}}^\text{depth} = \sum_{t=1}^{N_f-1} (L_{\text{multi}}^\text{absolute}(t) + L_{\text{multi}}^\text{contrast}(t)), \quad (14)
\]

\[
L_{\text{multi}}^\text{binary} = -B^t \ast \log(fcs([D_{\text{fusion}}^t]^{N_f-1})), \quad (15)
\]

\[
L_{\text{multi}} = \beta \cdot L_{\text{multi}}^\text{binary} + (1 - \beta) \cdot L_{\text{multi}}^\text{depth}, \quad (16)
\]

where \( D^t \) and \( B^t \) are depth label and binary label at time \( t \), respectively, \( [D_{\text{fusion}}^t]^{N_f-1} \) is the concatenated depth maps of \( N_f - 1 \) frames, \( fcs \) denotes two fully connected layers and one softmax function after the concatenated depth maps, which outputs the logits of two classes, \( \beta \) is the weight of binary loss in the final multi-frame loss \( L_{\text{multi}} \). In equation 15, we use cross entropy loss to calculate the binary loss. In equation 16, we combine the binary loss and depth loss by a simple sum operation.

### 4. Experiment And Evaluations

#### 4.1. Databases and Metrics

##### 4.1.1 Databases

Three databases - OULU-NPU [6], CASIA-MFSD [35], Replay-Attack [7] are used in our experiment. OULU-NPU [6] is a high-resolution database, consisting of 4950 real access and spoofing videos. The ratio of real videos to attack videos in OULU-NPU [6] is 1:4. And this database contains four protocols to validate the generalization of models. CASIA-MFSD [35] and Replay-Attack [7] are databases which contain low-resolution videos. We use these two databases for cross testing.

##### 4.1.2 Metrics

In OULU-NPU dataset, we obey its original protocols and metrics. OULU-NPU utilizes Attack Presentation Classification Error Rate \( APCER \), which evaluates the highest error among all PAIs; Bona Fide Presentation Classification Error Rate \( BPCR \), which evaluates error of real access data; and \( ACER \) [16], which evaluates the mean of \( APCER \) and \( BPCR \):

\[
ACER = \frac{APCER + BPCR}{2}. \quad (17)
\]

HTER is adopted in the cross testing between CASIA-MFSD and Repaly-Attack, which evaluates the mean of False Rejection Rate (FRR) and the False Acceptance Rate (FAR):

\[
HTER = \frac{FRR + FAR}{2}. \quad (18)
\]

#### 4.2. Implementation Details

##### 4.2.1 Training Strategy

Our proposed method combines the single-frame part and multi-frame part. Two-stage strategy is applied in the training process. **Stage 1:** We train the single-frame part by the single-frame depth loss, in order to learn a fundamental representation. **Stage 2:** We fix the parameters of the single-frame part, and finetune the parameters of multi-frame part by the depth loss and binary loss. Note that the diverse data should be adequately shuffled for the stability of training and generalization of the learnt model. The network is fed by \( N_f \) frames, which are sampled by an interval of three frames. This sampling interval makes sampled frames maintain enough temporal information in the limitation of GPU memory.

##### 4.2.2 Testing Strategy

For the final classification score, we feed the sequential frames into the network and obtain depth maps
\{D^t_{\text{fusion}}\}_{N_f-1}^0 \) and the living logits \( \hat{b} \) in \( fcs(D^t_{\text{fusion}}) \). The final living score can be obtained by:

\[
\text{score} = \beta \cdot \hat{b} + (1 - \beta) \cdot \frac{\sum_{t=1}^{N_f-1} ||D^t_{\text{fusion}} \ast M^t_{\text{fusion}}||_1}{N_f - 1},
\]

where \( \beta \) is the same as that in equation 16, \( M^t_{\text{fusion}} \) is the mask of face at frame \( t \), which can be generated by the dense face landmarks in PRNet [12], and the second module denotes that we compute the mean of depth values in the facial areas as one part of the score.

### 4.2.3 Hyperparameter Setting

We implement our proposed method in Tensorflow [1], with learning rate 3e-3 of single-frame part and 1e-2 of multi-frame part. The batch size of single-frame part is 10 and that of multi-frame part is 2 with \( N_f \) being 5 in most of our experiment, except that batch size being 4 and \( N_f \) being 3 in protocol 3 of OULU-NPU. Adadaelte optimizer is used in our training procedure, with \( \rho \) as 0.95 and \( \epsilon \) as 1e-8. We set \( \alpha \) and \( \beta \) to optimal values by our experimental experience, and according to the analysis of the following section that protocol 4 in OULU-NPU is most challenging to the generalization, we recommend that the parameters \( \alpha = 0.8 \) and \( \beta = 0.9 \) are suitable for the realistic scenes.

### 4.3. Experimental Comparison

#### 4.3.1 Intra Testing

We compare the performance of intra testing on OULU-NPU dataset. OULU-NPU proposes four protocols to evaluate the generalization of the developed face presentation attack detection (PAD) methods. Protocol 1 is designed to evaluate the generalization of PAD methods under previously unseen illumination and background scene. Protocol 2 is designed to evaluate the generalization of PAD methods under unseen attack medium, such as unseen printers or displays. Protocol 3 utilizes a Leave One Camera Out (LOCO) protocol, in order to study the effect of the input camera variation. Protocol 4 considers all above factors and integrates all the constraints from protocols 1 to 3, so protocol 4 is the most challenging. Table 1 shows that our proposed method ranks first on three protocols - protocol 1, 2, 4, and ranks third on protocol 3. We can see that our model performs well at the generalization of external environment and attack mediums, and is slightly worse when it comes to the input camera variation. It’s worth noting that our proposed method has the lowest mean and std of ACER in protocol 4, which is most suitable for the real-life scenarios.

### 4.3.2 Cross Testing

We utilize CASIA-MFSD and Replay-Attack dataset to perform cross testing. This can be regarded as two testing protocols. One is training on the CASIA-MFSD and testing on Replay-Attack, which we name protocol CR; the other is training on the Replay-Attack and testing on CASIA-MFSD, which we name protocol RC. In Table 3, we see that our proposed method ranks first on three protocols - protocol 1, 2, 4, and 24.0 on protocol RC, reducing 36.6% and 15.5% respectively compared with the previous state of the art. The improvement of performance on cross testing demonstrates the generalization and superiority of our proposed method.
### 4.3.3 Ablation Study

We implement experiment on five architectures to demonstrate the advantages of our proposed sequential structure under the supervision of depth. As shown in table 4, Model 1 is the single-frame part of our method. Model 2 combines single-frame CNN with OFF blocks under binary supervision and depth supervision. Model 3 combines single-frame CNN with ConvGRU under binary supervision and depth supervision. Model 5 is our complete architecture, integrating all modules. Model 4 discards binary supervision compared with model 5. Comparing ACER of model 2 and model 3 with that of model 1, we see that our OFF module and ConvGRU module both improve the performance of face anti-spoofing. And the ACER of model 5 shows that the combination of OFF module and ConvGRU module has more positive effects. Via discarding the binary supervision, we test the effect of binary supervision on our model 4. In this model, we find that multi-frame model with simple depth supervision can also outperform the single-frame model and binary supervision indeed assists the model to distinguish live vs. spoof.

In table 5, we study the influence of contrastive depth loss. Model 1 is our single-frame model supervised by both the euclidean depth loss and the contrastive depth loss, while model 1* is supervised only by euclidean depth loss. Comparing model 1 with model 1*, we can see that contrastive depth loss can improve the generalization of our model.

Moreover, the inference of the model 1 costs around 18 ms and that of model 5 costs around 96 ms, which indicates that our method is efficient enough to be applied in reality.

### 4.3.4 Qualitative Analysis

Figure 6 presents our generated depth maps in OULU-NPU. D-score denotes the living score calculated by the mean of depth values in facial area, which is the depth subpart in equation 19. Though the multi-frame maps in spoofing scenes are visually noiser than those in the single-frame maps, the discrimination is obvious when only considering the multi-frame maps themselves. Specially, in single-frame maps, the D-score of real scene is 0.368, which is lower than that of replay2 scene. By contrast, in multi-frame maps, the D-score of real scene is higher than the D-scores in all of the attack scenes. Visually, the multi-frame maps are also more distinguishable between the real scene and attack scenes than single-frame maps.

### 5. Conclusions

In this paper, we propose a novel face anti-spoofing method, which is depth supervised and consists of adequate temporal information. To seek the spatiotemporal information, we take OFF block as short-term motion module and ConvGRU as long-term motion module, and then combine them into our architecture. Our proposed method can discover the nature of spoof patterns efficiently and accurately under depth-supervision. Extensive experimental results demonstrate the superiority of our method.
[35] 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. 6