Learning Generalized Deep Feature Representation for Face Anti-Spoofing

Article *in* IEEE Transactions on Information Forensics and Security · April 2018

DOI: 10.1109/TIFS.2018.2825949

6 authors, including:

Haoliang Li
City University of Hong Kong

Peisong He
Sichuan University

Shiqi Wang
City University of Hong Kong

Anderson Rocha
State University of Campinas (UNICAMP)

Learning Generalized Deep Feature Representation for Face Anti-spoofing

Haoliang Li, *Student Member, IEEE,* Peisong He, *Student Member, IEEE,* Shiqi Wang, *Member, IEEE,* Anderson Rocha, *Senior Member, IEEE,* Xinghao Jiang, *Member, IEEE,* and Alex C. Kot, *Fellow, IEEE*

***Abstract*—In this work, we propose a novel framework leveraging the advantages of the representational ability of deep learning and domain generalization for face spoofing detection. In particular, the generalized deep feature representation is achieved by taking both spatial and temporal information into consideration, and a 3D Convolutional Neural Network (3D CNN) architecture tailored for the spatial-temporal input is proposed. The network is first initialized by training with augmented facial samples based on cross-entropy loss and further enhanced with a specifically designed generalization loss, which coherently serve as the regularization term to train the network in an end-to-end fashion. The training samples from different domains can seamlessly work together for learning the generalized feature representation by manipulating their feature distribution distances. We evaluate the proposed framework with different experimental setups using various databases. Experimental results indicate that our method can learn more discriminative and generalized information compared with the state-of-the-art methods.**

***Index Terms*—Face spoofing, deep learning, 3D CNN, domain generalization.**

I. Introduction

Biometrics offers a powerful and practical solution to authentication-required applications. Due to the breakthrough of biometrics authentication via deep learning and its better security capability compared with traditional authentication methods (e.g., password, secret question, token code), more and more attention has been attracted from both academia and industry nowadays. Typical biometric modalities include fingerprints, iris, face and voice prints, among which "face" is the most popular one as it does not require any additional hardware infrastructure and almost all mobile phones are equipped with a front-facing camera. Despite the success of face recognition, it is still vulnerable to the presentation attacks due to the popularity of social media from which facial images are easy to acquire [1]. For instance, a presentation attack can record the face information of a person by printing (printing attack), replaying on screen (replay attack) or even counterfeiting the face via 3D masking [2] and VR [3], which brings extremely challenging security issues.

Security concerns of face recognition systems have motivated a number of studies for face spoofing detection. From the perspective of evaluating the disturbance information injected into the spoofing media, a series of approaches aim at extracting the distortion information, which may appear on spoofed face samples. Typical spoofing artifacts include texture artifacts [4], motion artifacts [5] and image quality relevant artifacts [6]. Other approaches focus on the system level in which specific sensors (e.g., gravity sensor) can be utilized for auxiliary assistance [7] or additional hardware can be incorporated into the verification system (e.g., infrared sensor [8]). Moreover, human-computer interaction may also be required for spoofing detection (head moving, eye blinking, etc.) [9], [10].

With numerous approaches proposed to deal with the artifacts within a single image, there are still two important issues in face anti-spoofing. On one hand, how to generalize well to the "unseen data" becomes pivotal, as obtaining enough data with sufficient variability in the training process is not always practical. On the other hand, much less work has been dedicated to extracting information along the temporal direction, which can also provide valuable cues (liveness information, unexpected motion [9], [10], temporal aliasing, etc.). More importantly, learning spatial plus temporal features would become more difficult as more training data would be necessary and the lack of generalization could be even more pronounced. All these issues cast challenges on the generalization capability of robust feature representation. In view of this, we focus on deep feature representation in a generalized way by exploiting the information from both spatial and temporal dimensions. In particular, 3D convolutional neural networks (3D CNN), which have been proved to be efficient for action recognition task [11], are employed to learn spoofing-specific information based on typical printed and replay video attacks. The solution incorporates 2D and 3D features related to the presentation attack problem, and learns not only spatial variations associated with attacks but also artifacts that take place over time. More specifically, we employ the 3D CNN architecture with a data augmentation strategy for the spoofing detection task. To obtain a more robust and generalized 3D CNN model, the lack of generalization is dealt with by introducing a regularization mechanism, which focuses on improving classification accuracy during training as well as generalizing to unknown conditions by minimizing the feature distribution dissimilarity across domains. These capabilities allow us to make a further step regarding the detection of attacks under unknown or different conditions.

The main contributions of our work are as follows.

- we apply a 3D CNN network which take both spatial and temporal information into consideration with a specifically designed data augmentation method for face spoofing detection.
- To further improve the generalization performance, we employ a generalization regularization by minimizing the Maximum Mean Discrepancy distance among different domains.

- We conduct extensive experimental analysis on four different datasets as well as our proposed cross-camera based protocol. The results show that our proposed framework can achieve significantly better performance compared with other state-of-the-art methods.

II. Related Work

A. Face Anti-spoofing

In terms of various application scenarios, we roughly categorize existing face spoofing detection methods into three categories, including motion analysis based [5] (which may require user cooperation), texture analysis based [4][12], and sensor-assisted detection [7]. The first two categories can be generally applied to face verification/registration task with personal computers and mobile phones, while the last one requires extra hardwares. To further enhance the robustness of biometric spoofing detection, some other biometrics information can be incorporated into the face antispoofing system [13], [14], [15], [16].

Motion analysis relies on extracting liveness information (e.g., eye blinking, lips movement, head rotation) for distinguishing between genuine and spoofed ones. For instance, such liveness information can be obtained via optical flow. In [5], the authors reported that even subtle movement can be regarded as motion cues. For this kind of method, the user assistance is usually required. Though motion analysis based methods are effective to counter printed photo attacks, they may suffer performance drops when the spoofing attack is conducted by video replay.

The idea of facial texture and distortion analysis originates from the assumption that the spoofed medium is likely to lack high-frequency information, due to the face media reproduction process. By analyzing the texture artifacts left behind during an attack, we can extract useful information such that the genuine and spoofed faces can be properly distinguished. In [17], a texture analysis method based on two dimensional Fourier spectrum is conducted. In [18], Tan *et al.* proposed a total-variation based decomposition method and extracted the different-of-Gaussian (DoG) information on the high-frequency part. The final model is learned in a bilinear sparse low-rank regression manner. Texture features designed for object detection/recognition tasks have also been proved to be effective for face spoofing detection. In [4], multi-scale Local Binary Pattern (LBP) with Support Vector Machine (SVM) classifier was proposed, achieving superior performance on NUAA [18] and Idiap REPLAY-ATTACK databases [19]. The multi-scale LBP feature was further extended to facial component based method followed by fisher vector [20], such that more discriminative information can be extracted. Other texture features, such as Scale-invariant feature transform (SIFT) and Speed-up Robust feature (SURF) [21], can also be applied to the face anti-spoofing task. As the high-frequency information can also be discarded in the temporal domain, the texture features based on 2-D plane can be extended to 3-D plane [22]. By jointly exploring color and texture information, the face anti-spoofing performance can be largely improved [12][23]. Recently, a dynamic texture face spoofing was proposed [24] by considering volume local binary count patterns. Moreover, by incorporating flash light, the texture pattern can be detected more readily [25]. Another stream of feature design is based on image quality methods. In [6], 25 quality assessment based metrics were employed as the discriminative features for face spoofing detection. In [26], the authors extended the method in a regression manner to tackle the problem whereby samples were taken from multiple camera models. In [27], a feature concatenation based method was proposed by considering specular, blurriness and color distortion. However, both texture-based and distortion-based features are likely to be overfitted to one particular setup, which may limit their application for practical scenarios when confronting diverse image/video capturing conditions.

In addition to motion analysis and texture analysis methods, additional sensors can also be leveraged for face spoofing detection. Compared with face images directly captured by the popular camera models, 3D depth information [28], [29], multi-spectrum and infrared images [8], and even vein flow information [30] can be obtained if additional sensors are deployed. Such methods can be enhanced by audio information [31], which can further improve the robustness of face spoofing detection. However, as additional equipments are required in such methods, they are usually more expensive.

Deep learning based methods have also been proved to be effective for biometric spoofing detection tasks. Yang *et al.* [32] first proposed to use Convolutional Neural Network (CNN) for face spoofing detection. Some other works [33], [34], [35], [36] have been proposed to modify the network architecture directly, which can further improve the detection accuracy. In [37], a CNN has been proved to be effective for face, fingerprint, and iris spoofing detection. Nogueira *et al.* [38] further showed that a pre-trained CNN model based on ImageNet [39] can be transferred to fingerprint spoofing detection without any fine-tuning process. In [2], a deep dictionary learning based method was proposed for mask attacking detection. Additional information (e.g., eye blinking) can also be considered as auxiliary information by associating it with deep learning [40], which further improves the face spoofing detection performance. More recently, Atoum *et al.* [41] proposed a depth-based CNN for face spoofing detection to extract depth information based on RGB face images. Gan *et al.* [42] proposed a 3D CNN based framework to jointly capture the the spatial and temporal information. As [42] also deal with CNNs for the PAD problem, it is important to highlight the differences between their method and the one we propose herein. In summary, our technique prioritizes 3×3 convolutions (for efficiency), it has a streamlined strategy for temporal feature learning and also different pre-preprocessing and augmentation mechanisms. In general, deep learning methods can achieve desirable performance when the training and testing samples are acquired in very similar conditions (e.g., captured with the same type of phone). However, such environment cannot be always ensured due to the diverse capturing devices, illumination conditions and shooting angles and this is what our method sets forth to address.

B. Multimedia Recapturing Detection

Multimedia recapturing aims at reproducing the content illegally from the perspective of security. During the multimedia content reproduction process, the camera, display screen as well as the lighting condition are carefully tuned to obtain the reproduced content with the best quality. To the best of our knowledge, the first work addressing the problem of image recapturing detection on LCD screens was proposed in [43], whereby three distortion types, including the texture pattern caused by aliasing, the loss-of-detail pattern caused by the low resolution of LCD screens and the color distortion caused by the device gamut were analyzed. To address this problem, LBP, multi-scale wavelet statistics as well as color channel statistics were combined as a single feature vector for classification. As claimed in [44], although the texture pattern can be eliminated by setting the recapturing condition properly, the loss-of-detail artifact cannot be avoided during recapturing, which can be further employed as discriminative features for image reproduction detection. Recently, Li *et al.* [45] proposed a CNN+RNN framework to exploit the deep representation of recapturing artifacts, which was proved to be effective when using 32×32 image block as the input of the network. For video reproduction, Wang and Farid [46] proposed to explore geometry principles based on the motivation that the recaptured scene is constrained to a planar surface, while the original video was taken by projecting objects from the real world to the camera. In [46], both mathematical analysis and experimental results showed that the reproduction process can cause "non-zero" skew in the projection matrix by assuming that the skew value of camera for the original capturing was zero. Along this vein, the algorithm proposed in [47] detected the radial lens distortion based on the geometry principle. A mathematical model was built for lens distortion and distorted line based on the edge of video frame, which was regarded as discriminative cue for reproduction identification. In [47], the characteristic ghosting artifact, which is generated due to the lack of synchronization between the camera and the projected screen, could be detected by a designed filter composed by two Dirac pulses as the discriminative information.

III. Methodology

Generally speaking, both spatial and temporal artifacts (e.g., unexpected texture patterns, color distortions and blurring [43], [48]) may occur during the face spoofing process. Regarding the texture pattern, such pattern appearing in spatial dimension is caused by the mismatch of the replay device resolution and the capturing device resolution, while in temporal domain it is derived from the divergence between flash frequency of display device (e.g., 120 Hz) and the video frequency (e.g., 25 Hz). The color distortion is due to the mismatch of color gamut between the display medium and the recapturing model. Besides the texture pattern and color distortion, the unexpected motion such as display device shaking along the temporal dimension can also be beneficial for spoofing detection. Instead of using the hand-crafted features in inferring the distinctive information, applying Convolutional Neural Network (CNN) to spoofing detection has shown promising results for different spoofing setups. However, as the current adopted CNN models for spoofing detection are all based on 2D images trained in a label-guided manner [38], [37], there are two outstanding limitations:

- Due to the limitation of the 2D CNN structure, the temporal statistics encoded in contiguous frames are ignored.
- Directly applying the classification loss with label information can lead to overfitting problem to a certain database collection. In this scenario, the trained model cannot generalize well to the unseen data.

In view of these limitations, we develop a 3D CNN architecture such that discriminative information can be learned from both spatial and temporal dimensions. In particular, when training and testing samples are captured under similar environments, our model can achieve lower error rate compared with 2D CNN models as well as other handcrafted features used in prior art. More importantly, when training a CNN by considering face samples collected from different cameras under diverse illumination conditions, the extracted features across domains are expected to lie in a similar manifold such that a classifier trained with such features will have better generalization ability. In view of this, we also take advantage of domain generalization in network training by introducing a regularization term, which forces the learned features to share similar distributions. The pipeline of our proposed scheme is shown in Fig. 1.

A. 3D Convolutional Neural Network

In the 2D convolutional neural network, the convolution process is only applied on the 2D feature maps to compute the response in the spatial dimension, which has largely ignored the temporal information. In contrast with 2D CNN, the 3D CNN is conducted by convolving an input cube, which is stacked by multiple contiguous frames with a 3D kernel. We refer to the 3D convolution kernel size in the $l-$th layer by $W_l \times H_l \times T_l$, where T_l denotes the temporal depth and $W_l \times H_l$ represents the spatial size of the kernel. As such, the temporal information can also be preserved in the feature map. By jointly considering the temporal information, we can achieve better feature learning capability for face spoofing detection. In particular, each convolution operation is performed followed by a non-linear activation function such as ReLU. Mathematically, such process can be formulated as

$$
\begin{aligned}
y_{d_2,l}^{ijk} &= \sum_{d_1} \sum_{m=0}^{W_l-1} \sum_{n=0}^{H_l-1} \sum_{p=0}^{T_l-1} w_{d_1,d_2,l}^{mnp} x_{d_1,l-1}^{(i+m)(j+n)(k+p)} + b_{d_2,l} \\
x_{d_2,l}^{ijk} &= \sigma\left(y_{d_2,l}^{ijk}\right)
\end{aligned}
\tag{1}
$$

where $x_{d_1,l-1}^{ijk}$ is the value of a unit at position (i,j,k) in the d_1-th feature map from the $(l-1)$-th layer, $w_{d_1,d_2,l}^{mnp}$ is the value of the element at position (m,n,p) of the 3D convolution kernel connected to the d_2-th feature map in the $l-$th layer, $b_{d_2,l}$ is the bias term, and $\sigma(\cdot)$ denotes a non-linear activation layer. Subsequently, a 3D pooling layer is applied to reduce the resolution of feature maps and enhance

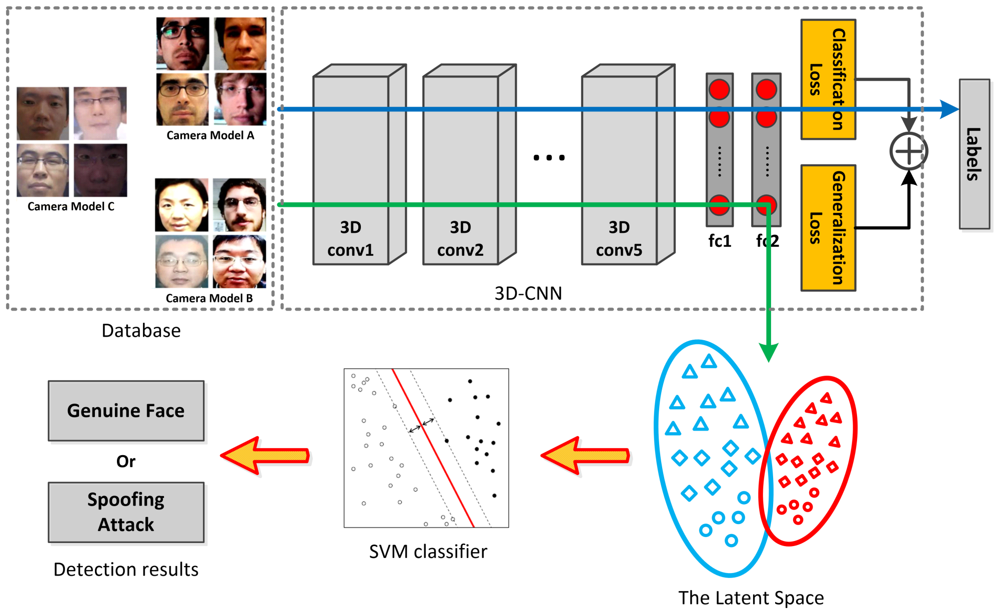


Fig. 1. The pipeline of the proposed scheme for face spoofing detection. The final objective function is determined by both classification loss and generalization loss. The output from FC2 layer is employed as latent discriminative feature for classification. The 3D conv model contains the 3D convolution layer, BatchNormalization, LeakyReLU, MaxPooling and Dropout layer. The second fully connected layer (FC2) is used for latent discriminative feature extraction.

the invariance of the input signals to distortions. According to the research in [49], smaller receptive fields of 3D convolution kernels with deeper architectures can yield better performance for video classification. Although our problem is different from [49], we found out that adopting a smaller receptive field leads to better results for face spoofing detection as well. Therefore, in the 3D CNN architecture, we only consider the spatial-temporal receptive field as $3 \times 3 \times 3$. The proposed 3D CNN model is detailed in Table I. This architecture has five convolutional layers followed by the fully connected layer. The study regarding the appropriate number of convolutional layers is presented in Section IV-D.

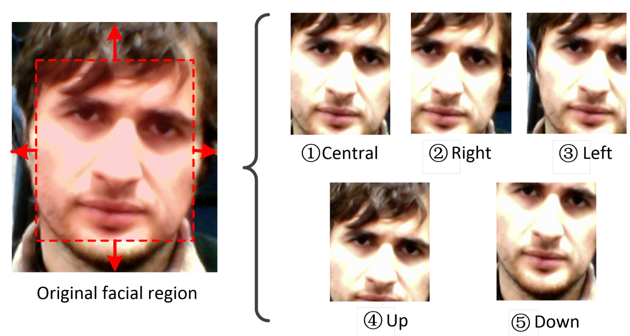


Fig. 2. Illustrations of generating spatially augmented data.

B. Data Augmentation

As it can be observed from Table I, our proposed 3D CNN model has more than 4M parameters to be optimized. However, existing samples in public databases are not enough to train such model. Therefore, the underfitting problem can not be avoided due to the large number of parameters in the model and the sparsity of training samples. To address this issue, we propose a data augmentation method based on video cubes to increase the number of training data. It should be noted that traditional augmentation methods such as injecting additional noise may not be feasible for the spoofing detection problem, given that the distortion information plays a key role in face spoofing detection. Therefore, the strategy of augmenting the video cubes is developed concerning this task.

1) Spatial Augmentation: To mitigate the variation of background for face spoofing detection, face detection is usually conducted as a pre-processing step [19]. However, variations of background near face regions can even be beneficial to face spoofing detection when considering deep learning approaches, as spoofing artifacts can be from the background region or the bezel of spoofing medium. Therefore, we propose to shift the bounding box in four different directions (up, down, right and left) by $\alpha \cdot l$, where l is equal to the width/height of bounding box. The parameter α is a predefined scaling factor, which is empirically set to 0.2 in our work. We stop the spatial augmentation if the bounding box moves out of the image boundary. We show an example of spatial augmentation in Fig. 2.

2) Color Augmentation: Color information can be useful for face spoofing detection due to the color gamut mismatch of the spoofing medium and the camera model. However, it is generally acknowledged that the color gamut induced by different camera models can also be different [50], [51]. To improve the diversity of color information, we conduct a gamma correction on each individual frame of a given video cube. Considering the face captured by a certain camera model with gamma value γ_1, the gamma correction process to γ_2 can be represented as

$$I_{aug} = \lfloor ((I/255)^{\gamma_2/\gamma_1}) * 255 \rfloor \qquad (2)$$

TABLE I
OUR PROPOSED 3D CNN ARCHITECTURE

Layer	Type/Module	Output Size	Filter/Pooling Size	Setting	# Parameter
1	3D Convolution	128 × 8 ×128 × 128	3 × 3 × 3		10K
2	3D BatchNormalization				
3	LeakyReLU			Leaky Factor: 0.1	
4	3D MaxPooling	128 × 8 × 64 × 64	1 × 2 × 2		
5	3D Convolution	128 × 8 ×64 × 64	3 × 3 × 3		10K
6	3D BatchNormalization				
7	LeakyReLU			Leaky Factor: 0.1	
8	3D MaxPooling	128 × 8 × 32 × 32	1 × 2 × 2		
9	3D Convolution	128 × 8 ×32 × 32	3 × 3 × 3		10K
10	3D BatchNormalization				
11	LeakyReLU			Leaky Factor: 0.1	
12	3D MaxPooling	128 × 8 × 16 × 16	1 × 2 × 2		
13	3D Convolution	128 × 8 ×16 × 16	3 × 3 × 3		10K
14	3D BatchNormalization				
15	LeakyReLU			Leaky Factor: 0.1	
16	3D MaxPooling	128 × 4 × 8 × 8	2 × 2 × 2		
17	3D Convolution	128 × 4 ×8 × 8	3 × 3 × 3		10K
18	3D BatchNormalization				
19	LeakyReLU			Leaky Factor: 0.1	
20	3D MaxPooling	128 × 2 × 4 × 4	2 × 2 × 2		
21	Linear	1024			4M
22	BatchNormalization				
23	ReLU				
24	Dropout			Dropout Rate: 0.5	
25	Linear	2			2K
26	LogSoftMax	2			

† For 4D Tensor, the dimension is denoted as "Feature Map × Time × Width × Height". For 3D Tensor, the dimension is denoted as "Time × Width × Height".

where I and I_{aug} are the original pixel and augmented pixel, respectively, in RGB space. '$\lfloor\cdot\rfloor$' denotes the round and truncation operations, where the output value is truncated into the range [0,255]. Since the camera performs linear correction ($\gamma = 1.0$) and exponential gamma correction (e.g. $\gamma = 2.2$) before display[1], we choose the ratio γ_2/γ_1 to be $1.0/2.2$ and $2.2/1.0$ for color augmentation in our work. We show an example of color augmentation in Fig. 3.

(a) (b) (c)

Fig. 3. Illustrations of color-augmented data generation. (a) Original Face; (b) Face with the gamma correction ratio $1.0/2.2$; (c) Face with the gamma correction ratio $2.2/1.0$.

C. Model Generalization

Although deep learning is powerful in learning representative information when training data are diverse, it may still suffer from performance degradation when test data are "unseen", such as the test samples obtained from a totally different environment from the training data. Generally speaking, it is impossible to involve face samples captured by all types of cameras from every potential scenario. In view of this, we leverage the advantage of domain generalization [52] to solve this problem. More specifically, given face samples from a few different capturing conditions, by partitioning the face samples into different domains based on the capturing conditions, we aim at learning a robust representation across different domains for face spoofing detection by introducing the generalization loss as the regularization term. As such, the generalization capability of the network can be better enhanced.

Assume that there are face samples from L domains for training, which are denoted by $X = \{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_L\}$, with $\mathbf{X}_i$ representing the samples from the domain i. The total number of samples in X is $N_1 + N_2 + ... + N_L$, where N_1, N_2, ..., N_L are the number of samples from each domain. Moreover, the extracted features from the $f-$th fully-connected layer of network is assumed to be $\mathbf{Y}_f = [\mathbf{Y}_{f,1}^\top, \mathbf{Y}_{f,2}^\top, ..., \mathbf{Y}_{f,L}^\top]^\top$, $\mathbf{Y}_f \in \mathbb{R}^{(N_1+N_2+...+N_L)\times D}$ where $\mathbf{Y}_{f,i} \in \mathbb{R}^{N_i \times D}$ refers to the features of $f-$th fully connected layer from domain i. We further denote $\mathbf{Y}_{f,i,k} \in \mathbb{R}^D$ as the feature of $k-$th sample from $\mathbf{Y}_{f,i}$. To align the feature distributions from different domains, we adopt the Maximum Mean Discrepancy (MMD) [53], a popular metric to measure the similarity between two distributions, to minimize the feature distribution divergence across domains. As such, given two distributions, they are identical if the MMD distance between them equals to zero. To learn the generalized feature representation, we aim at optimizing the network, which embeds the input samples X to $\mathbf{Y}_f$, such that the MMD distances among different domains can be minimized [53].

[1]http://www.cambridgeincolour.com/tutorials/gamma-correction.htm

The MMD distance among multiple domains is given by,

$$d(\mathbf{Y}_f) = \frac{1}{L(L-1)} \sum_{i \neq j} \left\| \frac{1}{N_i} \sum_{k_1=1}^{N_i} \mathbf{Y}_{f,i,k_1} - \frac{1}{N_j} \sum_{k_2=1}^{N_j} \mathbf{Y}_{f,j,k_2} \right\|^2 \tag{3}$$

which can be further rewritten as,

$$d(\mathbf{Y}_f) = \mathrm{Tr}(\mathbf{K}_f \mathbf{Q}), \tag{4}$$

where $\mathbf{K}_f$ is the Gram matrix defined based on $\mathbf{Y}_f$, $\mathbf{K}_f = \mathbf{Y}_f \mathbf{Y}_f^T$, and $\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{1,1} & \mathbf{Q}_{1,2} & \cdots & \mathbf{Q}_{1,l} \\ \mathbf{Q}_{2,1} & \mathbf{Q}_{2,2} & \cdots & \mathbf{Q}_{2,l} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Q}_{l,1} & \mathbf{Q}_{l,2} & \cdots & \mathbf{Q}_{l,l} \end{bmatrix}$ is the coefficient matrix defined based on the samples from domain pairs. In particular, the matrix block $\mathbf{Q}_{i,j}$ from domains i and j is given by

$$\mathbf{Q}_{i,j} = \begin{cases} \frac{1}{LN_iN_j}\mathbf{1}_{N_i \times N_j} & if \quad i = j \\ -\frac{1}{L(L-1)N_iN_j}\mathbf{1}_{N_i \times N_j} & otherwise, \end{cases} \tag{5}$$

where $\mathbf{1}_{N_i \times N_j}$ denotes the all-ones matrix with dimension $N_i \times N_j$. The gradient of the generalization loss with respect to the network parameter Θ can be computed as

$$\frac{\partial d(\mathbf{Y}_f)}{\partial \Theta} = \frac{\partial d(\mathbf{Y}_f)}{\partial \mathbf{Y}_f} \frac{\partial \mathbf{Y}_f}{\partial \Theta} = 2\mathbf{Q}\mathbf{Y}_f \frac{\partial \mathbf{Y}_f}{\partial \Theta}, \tag{6}$$

where $\frac{\partial \mathbf{Y}_f}{\partial \Theta}$ can be obtained via back-propagation method [54].

To learn the generalized feature representation with our proposed 3D CNN network, we train the network from scratch on the given face samples collected from multiple domains with cross-entropy loss ($\mathcal{L}$) [55]. Moreover, the MMD distance among the domains is required to be minimized simultaneously. As such, the network parameters can be learned as

$$\Theta^* = \arg\min_\Theta \quad \mathcal{L} + \lambda \mathcal{R}, \tag{7}$$

where Θ is the network parameters and $\mathcal{R}$ is represented by

$$\mathcal{R} = \sum_{f=1}^{F} \mathrm{Tr}(\mathbf{Y}_f \mathbf{Y}_f^\top \mathbf{Q}). \tag{8}$$

Here F is the number of fully connected layer in the network, which is set to be 2 in our work since we have two fully connected layers in our proposed network.

IV. Experimental Results

A. Databases

We adopt four face spoofing detection databases, ① Idiap REPLAY-ATTACK [19], ② CASIA Face AntiSpoofing [56], ③ MSU mobile face spoofing database [27], and ④ Youtu Liveness database [57] for the face anti-spoofing task.

The Idiap REPLAY-ATTACK database consists of 1200 face videos with 50 different subjects in total. The videos were captured by only the front-facing camera of a Macbook with the resolution 320×480 pixels. Two environments are considered when taking the videos. One is the controlled environment with uniform background and illumination condition. The other is more complex with natural lighting and reflection in the background. For the spoofing medium, iPad 1 (with the size 1024×768 pixels), iPhone 3GS (with the size 480×320 pixels) and A4 printed paper are considered to display the face for spoofing purposes.

The CASIA Face AntiSpoofing database has 600 face videos in total from 50 subjects. Compared with the Idiap database, the acquisition camera models are more diverse. The quality levels of capturing devices range from low resolution, medium resolution (two different USB cameras with the resolution of 480×640 pixels) and high resolution (Sony NEX-5 camera with the resolution 1280×720 pixels). The CASIA database has diverse attack types, including warping, cutting and video-replay attacks. Though the camera models and attacking types are more diverse, compared with Idiap REPLAY-ATTACK, the capturing background and the ethnicity of subjects (all Chinese) are limited.

The MSU mobile face spoofing database has 280 videos with 35 subjects. Both Laptop camera (with the resolution 640×480 pixels) and Android phone camera (with the resolution 720×480 pixels) are considered for face sample collection. The videos were taken under various illumination conditions with different human ethnicities. Two different spoofing attacks, printed photo attack and replay video attack, were considered in MSU database. Recently, another face spoofing database, MSU unconstrained smartphone spoof attack database [58] was constructed, which has more than 10k images with around 1k subjects. However, since we only focus on face spoofing with videos, this database was not adopted in our experiment.

We also consider Youtu Liveness Database (Youtu) [57], which has much larger scale in terms of the number of video clips, camera models, capturing environments compared with the other three. In Youtu database, there are 3500 videos with 20 subjects for public-research purpose. Five mobile devices (Hasee Smart-Phone, Huawei Smart-Phone, iPad 4, iPhone 5s and ZTE Smart-Phone) were employed for face video acquisition. For spoofing medium, printed paper attack, video display attack, mask attack and video replay attack were considered.

B. Evaluation Protocol

For Idiap Replay-Attack database, it is divided into three sub-folds, including training fold, development fold and testing fold. We report the Equal Error Rate (EER) on the development fold and use the threshold determined by EER on the development fold to obtain the Half Total Error Rate (HTER) on the testing fold. For CASIA, MSU and Youtu databases, a classifier is trained with the training fold and then EER rate is evaluated on the testing fold following the protocols defined in [56], [27], [57].

To further evaluate the generalization capability of our proposed method, we conduct experiments to evaluate the performance in the scenario of cross-camera based face spoofing detection. In particular, we first train a 3D CNN model by employing genuine and spoofed face samples captured by multiple camera models. Then we evaluate the performance by testing with another camera model which was not involved

TABLE II
CROSS-CAMERA EXPERIMENTAL PROTOCOL.

Protocol	Training Camera Models	Testing Camera Model
1	Long-time-used USB camera New USB camera Macbook Sony NEX-5	Macbook Air
2	Long-time-used USB camera New USB camera Macbook Sony NEX-5	Google Nexus 5
3	Long-time-used USB camera New USB camera Macbook Air Google Nexus 5 Sony NEX-5	Macbook
4	Macbook Macbook Air Google Nexus 5	Long-time-used USB camera
5	Macbook Macbook Air Google Nexus 5	New USB camera
6	Macbook Macbook Air Google Nexus 5	Sony NEX-5
7	Macbook Macbook Air New USB camera	Google Nexus 5
8	Long-time-used USB camera New USB camera Macbook	Sony NEX-5
9	Macbook New USB camera Sony NEX-5	Long-time-used USB camera
10	Long-time-used USB camera Sony NEX-5 Macbook Air	Google Nexus 5

in the training phase. To conduct such cross-camera based experiments, we merge the CASIA, Idiap REPLAY-ATTACK and MSU databases together and re-arrange the training, validation and testing sets based on camera models. The samples for training and validation are from the same camera model while the samples for testing are from another camera model. Here, we have six camera models in hand including "Long-time-used USB camera","New USB camera", "Macbook", "Macbook Air", "Google Nexus 5", and "Sony NEX-5". By considering the generalization capability where camera models are different between training samples and testing samples[2], we randomly create ten different cross-camera scenarios to evaluate the performance of our 3D CNN framework. The details of experimental protocols are listed in Table II. During training, we randomly divide the training data as training fold and development fold. The average Half Total Error Rate is reported by repeating the process for five times.

[2]The medium of spoofing attack (e.g., paper printed attack and video replay attack) can also be taken into consideration in cross-domain face spoofing detection. However, compared with the diversity of camera models, the conditions of spoofing medium are easier to be controlled. Therefore, in this work, we only consider the influence of camera models.

C. Experimental Setup

For the image-based face anti-spoofing detection method, Viola-Jones face detection algorithm [59] was employed for face localization in each video frame. As stated in [19], localizing the face region can effectively mitigate the noise information induced by background. However, for our temporal-based method, directly employing face detection based on individual frames and concatenating each frame into a temporal cube is not practical, as this may destroy the temporal consistency. To extract reliable face region and preserve useful temporal information, we propose a "Max-Min" strategy to choose the largest face detection bounding box among the frames in a face video as the final face region. To be more specific, considering the upper-left point $(x_{1,i}, y_{1,i})$ and the bottom-right point $(x_{2,i}, y_{2,i})$ of the bounding box in the $i-$th frame of a given video with T frames ($i = 1, 2, ..., T$), the final bounding box is located by $(x_{1,min}, y_{1,min})$ and $(x_{2,max}, y_{2,max})$, where $x_{1,min} = \min\{x_{1,1}, ..., x_{1,T}\}$, $y_{1,min} = \min\{y_{1,1}, ..., y_{1,T}\}$, $x_{2,max} = \max\{x_{2,1}, ..., x_{2,T}\}$, and $y_{2,max} = \max\{y_{2,1}, ..., y_{2,T}\}$. Frames without detectable face region are not considered. Based on our experiment, we found that it is a simple but effective way to crop face regions and maintain temporal information simultaneously. After obtaining the bounding box for each video, the face region is resized to 128×128 pixels. Basically, the temporal size can be determined based on the memory of GPU card. We set the temporal size to be 8 frames in our experiments. The total number of extracted cubes from the database (without data augmentation) are summarized in Table III.

TABLE III
NUMBER OF SAMPLES FOR DIFFERENT DATABASES.

Database	Idiap	CASIA	MSU	Youtu
# Cubes	36500	13587	9588	125159

For the learning process, we first initialize the parameters of CNN according to [60] and train the network only with cross-entropy loss. Then, the last convolutional layer as well as the fully-connected layer are fine-tuned with both cross-entropy loss and generalization loss. The idea behind such training strategy is that shallow layers are more likely to be generalized [61]. By fine-tuning the deeper layers, more discriminative information is expected to be extracted, which can be better generalized by minimizing MMD distance. Here, the domain is determined by the number of camera models for training. (The generalization regularization is omitted if there is only one camera model for training.) The weight λ of the regularization term is set in the way that at the end of training, the classification loss and regularization term loss are approximately the same. Such setting is reasonable since the feature representation which has both discrimination and generalization ability can be learned. More specifically, the weight is selected in a range {0.001, 0.01, 0.1, 1, 10}. For the learning parameter setting, we experimentally set the two different momentum values as $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and the learning rate is set to 0.001 for training the network from scratch and 0.0001 for fine-tuning. Moreover, the weight decay is set to 0.00005 and learning rate decay is set to 10^{-7}. The network is trained with the adaptive moment estimation (Adam) method [62] in a mini-batch manner with the size 8 during network initialization step and 100 for each domain during the fine-tuning step. It is observed that the objective function can converge after 50 epochs for the initialization step and 10 epochs for the fine-tuning step. The GPU card employed for our task is Tesla K40 and the framework is implemented by Torch library[3]. After training the network, we employ the second fully connected layer (with dimension 1024 elements) as our latent discriminative feature. The Support Vector Machine [63] with linear kernel is used for classifier training. Finally, the output scores of 3D cubes belonging to the same video is combined with average operation to generate the final detection result.

D. Experimental Results

1) Analysis of the Network Structure: We first analyze the proposed 3D CNN structure by comparing with other 3D CNN structures with different number of 3D convolutional layer to analyze the relationship between the depth of network and the final performance. We employ the Youtu Liveness Database in this analysis due to its diversity. In particular, a 3D convolutional layer contains a 3D convolution model, a 3D BatchNormalization model, a LeakyReLU model and a Maxpooling model. The dimensions of the second fully-connected layer are all set to 1024 for the model with different number of convolutional layers. The results are listed in Table IV in terms of Equal Error Rate (EER).

TABLE IV
PERFORMANCE COMPARISONS BY DIFFERENT 3D CNN STRUCTURES WITH YOUTU LIVENESS DATABASE. THE RESULTS ARE MEASURED BY EQUAL ERROR RATE (EER).

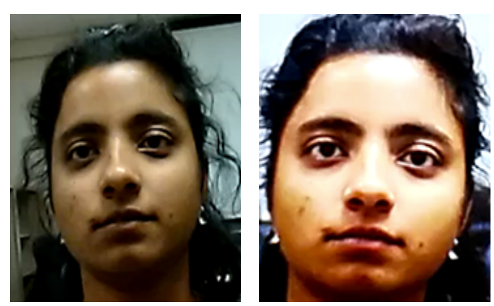
Structure	EER
#Conv Layer=3	10.8
#Conv Layer=4	8.4
#Conv Layer=5	**7.0**
#Conv Layer=6	7.6

Based on the results, we observe that we can achieve better performance with the increase of network depth. Such observation is also consistent with other computer vision tasks [64], [60]. However, we also notice that the performance drops when we further increase the depth of network to be more than five Conv layers, which may originate from the overfitting of network.

2) Intra-Database Evaluation: We then evaluate our algorithm by assuming the training and testing face samples are all from the same camera models and capturing conditions, which is referred to intra dataset validation in the literature. The performance of the proposed algorithm is compared with the state-of-the-art algorithms on different databases.

[3]http://torch.ch/

We adopt both frame- and sequence-based methods as our baselines, including texture based ([4], [22], [65], [48], [66], [12], [24]), image quality assessment based ([6],[27]) as well as data-driven based ([32]) methods. The comparison results are shown in Table V. Generally speaking, our proposed algorithm outperforms other baseline methods in most of the cases, which demonstrates the effectiveness of our method by considering learning both spatial and temporal information in a data-driven fashion. For Idiap REPLAY-ATTACK and MSU databases, we can observe that both hand-crafted features (spatial and temporal based) and deep learning based methods can achieve satisfactory performance. The reason lies in that the face capturing conditions are relatively simple in Idiap REPLAY-ATTACK and MSU databases. In particular, only one camera model is adopted in Idiap REPLAY-ATTACK and two for MSU database, and moreover, there is almost no motion information in videos, which makes the extracted features less influenced by camera and unexpected movements. For the CASIA database, it is observed that the sequence-based methods are generally worse than frame-based methods, which is reasonable since CASIA database contains diverse motion such as "paper wrapping" and "eye blinking". Such information can deteriorate the performance when considering temporal information. On the other hand, the quality-based method [6] cannot achieve satisfactory performance compared with texture-based methods (e.g. [12], [4]). This is due to the fact that the diversity of the camera models (Long-time-used USB camera, New USB camera and Sony NEX-5) can significantly influence the quality distortion information from spoofing artifacts. Again, by applying our proposed framework, we can achieve good performances when camera models and motion information are diverse and complicated. Compared with CASIA database, Youtu database is even more diverse not only in terms of camera models and motion, but also the illumination conditions. For Youtu database, our method outperforms the state-of-the-art color texture-based method [12]. Compared with other databases, for which our proposed method can achieve very low error rate, the error rate based on Youtu with our 3D CNN framework is still relatively higher.



(a) Genuine Sample (b) Spoofed Sample

Fig. 4. Genuine and spoofed face samples for visualization.

Besides objectively comparing the detection accuracy, we are also interested in the information extracted from the proposed 3D CNN network. To have a better understanding of our trained neural network, we feed both genuine and spoofed

TABLE V
PERFORMANCE COMPARISONS WITH THE STATE-OF-THE-ART METHODS.

	Method	Idiap		CASIA	MSU	Youtu
		EER (%)	HTER (%)	EER (%)	EER (%)	EER (%)
Sequence-based	de Freitas Pereira *et.al.* (2014) [22]	7.9	7.6	10.0	14.2*	25.0*
	Bharadwaj *et.al.* (2013) [65]	0.2	**0.0**	14.4	–	–
	Wen *et.al.* (2015) [27]	7.4	7.6*	26.5*	5.8	39.5*
	Tirunagari *et.al.* (2015) [66]	5.3	3.8	21.8	–	–
	Pinto *et.al.* (2015) [48]	–	2.8	14.0	–	–
	Boulkenafet *et.al.* (2016) [12]	**0.0**	3.5	3.2	3.5	9.6*
	Zhao *et.al.* (2017) [24]	1.7	0.8	6.5	–	–
	Gan *et.al.* (2017) [42]	0.2	**0.0**	6.4*	4.8*	11.3*
	Proposed Method	0.3	1.2	**1.4**	**0.0**	**7.0**
Frame-based	Galbally *et.al.* (2014) [6]	–	15.2	32.4	–	–
	Matta *et.al.* (2011) [4]	13.9	13.8	18.2	10.9	27.1*
	Yang *et.al.* (2014) [32]	6.1	2.1	7.6*	5.8*	8.0*
	Boulkenafet *et.al.* (2016) [12]	0.4	2.8	2.1	4.9	12.5*
	Liu *et.al.* (2017) [41]	0.8	0.7	2.7	–	–

'–' represents that the results were not available. '$*$' represents that the results were achieved with the implementation by ourselves.

face sample as shown in Fig. 4 to the proposed network, the outputs of Conv1 layer in both spatial domain and temporal domain, which can be regarded as a set of 3D cubes (feature maps), are visualized. For the spatial domain, we average the output in terms of temporal domain and obtain 128 spatial feature maps with size 128×128. Then, all elements in spatial feature maps are normalized to [0,1]. For the temporal domain, we compute the Discrete Fourier Transform based on the temporal domain by considering the direct current (DC) component of spatial information, which leads to 128 frequency spectra. Spatial and temporal information obtained from the Conv1 layer regarding genuine and spoofed face samples are visualized in Fig. 5 and 6, respectively.

Regarding the visualization of spatial domain, it can be observed that the obtained spatial feature maps are similar to the output by employing high-pass filter, which demonstrates that the high frequency component can be used to distinguish between the genuine and spoofed face samples. Spatial feature maps of the genuine sample tend to be darker in some cases compared with the spoofing one. It indicates that the trained network preserves the lighting and color information which play important roles in face spoofing detection (e.g. unexpected reflection and color gamut distortion present in the spoofing medium).

Regarding the visualization of temporal domain, it can be observed that frequency spectra belonging to the spoofed sample tend to contain high-frequency components with a larger magnitude in some cases (marked with the red box). Such results indicate that spoofed samples are more likely to suffer from temporal aliasing caused by re-sampling in the recapturing process, which is the key feature for face spoofing detection in the temporal domain.

3) Cross-Camera-Model Evaluation: Furthermore, we analyze the performance when camera models and illumination conditions are different for training and testing, and the protocols are illustrated in Table II. The methods adopting $LBP - TOP_{8,8,8,1,1,1}$ feature [22] and color texture [12] are employed as our texture-based baselines, image distortion statistics [27] is employed as the quality-based baseline and the 3D CNN framework without domain generalization regularization is employed as another deep learning based baseline. Considering that we can apply domain generalization based on the hand-crafted feature as well, the baseline hand-crafted features are also enhanced with domain generalization as proposed in [67]. The parameters for domain generalization [67] are determined by cross-validation, as explained in [67], [68]. The average HTER results are compared in Table VI.

Firstly, we can observe that the LBP-TOP and the distortion-based features cannot achieve desired performance under cross-camera settings for most of the cases. Such observation indicates that the trained classifiers with these two features are overfitted to the training data. The performance can be

(a) Spatial visualization of a genuine sample.

(b) Spatial visualization of a spoofed sample.

Fig. 5. Spatial visualization of genuine and spoofed face samples based on 128 feature maps.

(a) Temporal visualization of genuine sample.

(b) Temporal visualization of spoofed sample.

Fig. 6. Temporal visualization of genuine and spoofed face samples based on frequency spectrum. The horizontal axis ranges in $[-\pi, \pi]$

TABLE VI
HTER PERFORMANCE (%) FOR CROSS-CAMERA EXPERIMENTS (THE PROTOCOLS ARE DEFINED IN TABLE II).

Protocol	1	2	3	4	5	6	7	8	9	10	Average
de Freitas Pereira *et.al.* [22]	43.3	51.4	44.3	40.3	35.7	61.8	40.5	54.0	23.4	42.4	43.7
de Freitas Pereira *et.al.* [22] with DG [67]	42.7	43.6	40.5	**36.4**	46.8	51.3	46.7	46.1	20.1	37.2	41.1
Wen *et.al.* [27]	43.2	45.3	55.2	54.5	55.8	57.2	42.0	51.3	33.8	53.3	49.2
Wen *et.al.* [27] with DG [67]	39.5	49.6	50.0	48.4	47.0	44.2	44.2	35.9	24.9	37.2	42.1
Boulkenafet *et.al.* [12]	25.5	42.5	27.6	47.8	49.4	**34.6**	47.2	49.1	41.6	35.1	40.0
Boulkenafet *et.al.* [12] with DG [67]	22.8	42.3	35.4	39.7	45.5	44.7	39.1	27.8	21.4	33.0	35.2
3D CNN	28.1	46.7	37.0	53.6	41.3	49.0	44.9	34.4	15.9	31.4	38.2
Proposed Method	**19.0**	**32.7**	**26.0**	38.1	**34.3**	36.9	**38.1**	**21.9**	**11.5**	**28.6**	**28.7**

"DG" refers to domain generalization.

improved to some extent by applying domain generalization, which is reasonable since domain generalization can encode discriminative information that is shared among different domains. Moreover, it is observed that the performance varies from 25% to 49% by employing color texture feature [12]. When the training camera models are diverse, color texture can achieve the relatively good performance. By adopting domain generalization based on color texture, we can observe that the performance can also be improved in some scenarios. However, since the parameters for training samples adopted by domain generalization may not be able to encode the variation scale for test data (e.g., in Protocol 3, the training samples are much more diverse compared with those in Protocol 6), there is a performance drop. Therefore, it is reasonable that simply applying domain generalization on hand-crafted features may not work. We also observe that simply employing 3D CNN framework without generalization will not achieve desired performance expect for some special cases (e.g. protocol 9) where similar temporal information can be shared between training and testing samples. This can be explained by the following two reasons.

- Although the data augmentation process is applied to enlarge the size of training samples, such augmentation process is still limited due to the capturing devices and illumination condition. When the testing data are captured from a totally different environment, we still have the overfitting problem by simply applying deep learning for face spoofing detection.
- Different domain may introduce different artifacts. Since deep learning is conducted in a data-driven manner, it is likely that the neural network model may learn spoofing information which only dominate in a specific domain of training data. Therefore, when applying the neural network to extract features on testing data acquired under different conditions, we may not be able to extract the discriminative information.

Finally, the 3D CNN model with generalization regularization term learned in an end-to-end manner can significantly improve the performance for all protocols, as the generalization term can force the network to learn generalized spoofing features, which are less influenced by camera models and illumination conditions. On the other hand, it is worth mentioning that the error rate is still relatively high when the training samples are not diverse enough (protocols 4-7 in which the motions are consistent for training samples while different types of motion such as paper wrapping exist in testing samples).

4) Empirical Analysis of the 3D CNN Convergence: An empirical analysis of the convergence of our proposed 3D CNN network is conducted regarding the classification loss as well as the generalization loss. As indicated in Table I, we adopt several LeakyReLU and pooling modules, which make the network highly nonlinear. However, it is shown that we can still reach a local minima which leads to good performances. For a better understanding of the convergence, we visualize both the average classification loss and the generalization loss (MMD distance) for each mini-batch by Youtu database in Fig. 7. As we can see, in the beginning, the classification loss is relatively smaller compared with the generalization loss. The reason is due to the initialization of 3D CNN network by only classification loss. However, after training for a few epochs, the classification loss becomes larger since the network is more fitted to the generalization term. Finally, after training the network for more than 5 epochs, the classification loss and the generalization loss are close to each other.

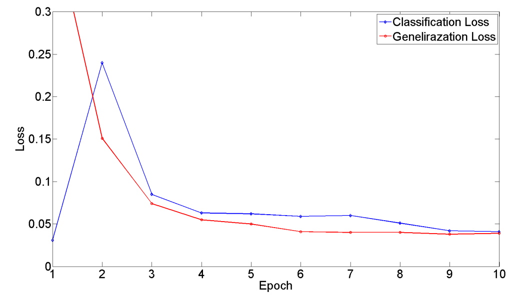


Fig. 7. Convergence visualization for Youtu liveness database.

E. Computational Time Analysis

Table VII summarizes the computational time of different methods based on a 360-frame video with the resolution 320×480. For baseline methods, the algorithm is implemented by using Matlab on an Intel Core i5 CPU@3.2GHz machine. For our own method, we compute the time on GPU Tesla K40. As we can observe from Table VII, our method can achieve a competitive time consumption compared with other methods. Considering that we analyze the time comsumption based on a 360-frame video, our proposed algorithm can process in a "real-time" fashion. It should be noted that leveraging GPU is not a disadvantage since such computational resources have become easier to obtain with the development of hardware technology nowadays. Although the method in [22] can be computed faster, the error rate of [22] is much higher than our results. Our framework will be further optimized (e.g., network compression) to meet the computational requirement of face spoofing application on mobile devices.

TABLE VII
COMPUTATIONAL TIME (SECONDS) ANALYSIS BASED ON A 360-FRAMES VIDEO WITH DIFFERENT METHODS

Method	Time per video
de Freitas Pereira *et.al.* [22]	7.35
Wen *et.al.* [27]	43.2
Boulkenafet *et.al.* [12]	27.8
Proposed Method	10.4

F. Discussions

Learning a robust and generalized feature representation for face spoofing detection is a challenging task. Since the capturing of face samples is totally independent, it is difficult to collect a large database containing all possible camera models, illumination conditions and facial appearances. Our proposed framework takes advantage of both deep learning and domain generalization technique to significantly improve the performance of intra and cross-condition setups, including the case that both training and testing samples are taken from similar condition (intra), and the case that training and testing samples are not well aligned (cross). As the first attempt to tackle the face spoofing detection task with 3D CNN, there are still some limitations. Though our proposed 3D CNN can achieve good performance when training and testing samples are from similar conditions, how to design a more robust network which can be generalized better should be further investigated, as error rates for cross-condition evaluation are still higher than intra setup. Moreover, how to apply a more reasonable distance measure in order to generalize the network better will also be studied in the future.

V. Conclusion and Future Work

In this work, we propose a 3D CNN framework to tackle the face spoofing detection problem. Compared with other deep learning based biometric spoofing detection method, the novelty of our paper lies in twofold. First, we apply a 3D CNN network which takes both spatial and temporal information into consideration with a specifically designed data augmentation method for face spoofing detection, which shows better classification capability compared with simply training a 3D CNN model from scratch. Secondly, to further improve the generalization performance, we employ a generalization regularization by minimizing the Maximum Mean Discrepancy distance among different domains. Our framework can be efficiently trained in an end-to-end manner and the experimental results show better generalization ability compared with state-of-the-art methods.

Future work might be dedicated to applying the concepts we propose herein to different domains other than just face spoofing PAD given that the generalization regularization through minimizing the Maximum Mean Discrepancy might be useful to decrease the impacts of different datasets (different domains) in a given problem. In this way, a generalized feature representation might be learned through the manipulation of feature distribution distances of the different sources of training data.

Acknowledgment

This research was carried out at the Rapid-Rich Object Search (ROSE) Lab at the Nanyang Technological University, Singapore. The ROSE Lab is supported by the National Research Foundation, Singapore, under its Interactive Digital Media (IDM) Strategic Research Program. The authors acknowledge with thanks the financial support of the Tan Chin Tuan Foundation, the São Paulo Research Foundation, Fapesp (DéjàVu Grant #2017/12646-3), and the Coordination for the Improvement of Higher Level Education Personnel, CAPES (DeepEyes Grant). P. He received financial support from China Scholarship Council (CSC) for the Scholarship Fund No. [2016] 3100 program.

References

[1] S. Marcel, M. S. Nixon, and S. Z. Li, *Handbook of Biometric Anti-Spoofing*. Springer, 2014, vol. 1.

[2] I. Manjani, S. Tariyal, M. Vatsa, R. Singh, and A. Majumdar, "Detecting silicone mask based presentation attack via deep dictionary learning," *IEEE Transactions on Information Forensics and Security*, 2017.

[3] Y. Xu, T. Price, J.-M. Frahm, and F. Monrose, "Virtual u: Defeating face liveness detection by building virtual models from your public photos." in *USENIX security symposium*, 2016, pp. 497–512.

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

[5] K. Kollreider, H. Fronthaler, M. I. Faraj, and J. Bigun, "Real-time face detection and motion analysis with application in liveness assessment," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 548–558, Sep 2007.

[6] J. Galbally, S. Marcel, and J. Fierrez, "Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 710–724, Feb 2014.

[7] S. Chen, A. Pande, and P. Mohapatra, "Sensor-assisted facial recognition: an enhanced biometric authentication system for smartphones," in *Annual International Conference on Mobile systems, Applications, and Services*. ACM, 2014, pp. 109–122.

[8] Z. Zhang, D. Yi, Z. Lei, and S. Z. Li, "Face liveness detection by learning multispectral reflectance distributions," in *IEEE International Conference on Automatic Face & Gesture Recognition and Workshops*, 2011, pp. 436–441.

[9] G. Pan, L. Sun, Z. Wu, and S. Lao, "Eyeblink-based anti-spoofing in face recognition from a generic webcamera," in *IEEE International Conference on Computer Vision*, 2007, pp. 1–8.

[10] G. Pan, L. Sun, and Z. Wu, *Liveness detection for face recognition*. INTECH Open Access Publisher, 2008.

[11] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[12] 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, Aug 2016.

[13] R. N. Rodrigues, L. L. Ling, and V. Govindaraju, "Robustness of multimodal biometric fusion methods against spoof attacks," *Journal of Visual Languages & Computing*, vol. 20, no. 3, pp. 169–179, 2009.

[14] B. Biggio, Z. Akhtar, G. Fumera, G. L. Marcialis, and F. Roli, "Security evaluation of biometric authentication systems under real spoofing attacks," *IET biometrics*, vol. 1, no. 1, pp. 11–24, 2012.

[15] R. N. Rodrigues, N. Kamat, and V. Govindaraju, "Evaluation of biometric spoofing in a multimodal system," in *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*. IEEE, 2010, pp. 1–5.

[16] M. Farmanbar and Ö. Toygar, "Spoof detection on face and palmprint biometrics," *Signal, Image and Video Processing*, vol. 11, no. 7, pp. 1253–1260, 2017.

[17] J. Li, Y. Wang, T. Tan, and A. K. Jain, "Live face detection based on the analysis of fourier spectra," in *Defense and Security*. International Society for Optics and Photonics, 2004, pp. 296–303.

[18] X. Tan, Y. Li, J. Liu, and L. Jiang, "Face liveness detection from a single image with sparse low rank bilinear discriminative model," in *European Conference on Computer Vision*. Springer, 2010, pp. 504–517.

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

[20] J. Yang, Z. Lei, S. Liao, and S. Z. Li, "Face liveness detection with component dependent descriptor," in *IEEE International Conference on Biometrics (ICB)*, 2013, pp. 1–6.

[21] D. Gragnaniello, G. Poggi, C. Sansone, and L. Verdoliva, "An investigation of local descriptors for biometric spoofing detection," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 849–863, Apr 2015.

[22] 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, pp. 1–15, 2014.

[23] 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.

[24] X. Zhao, Y. Lin, and J. Heikkilä, "Dynamic texture recognition using volume local binary count patterns with an application to 2d face spoofing detection," *IEEE Transactions on Multimedia*, 2017.

[25] P. P. Chan, W. Liu, D. Chen, D. S. Yeung, F. Zhang, X. Wang, and C.-C. Hsu, "Face liveness detection using a flash against 2d spoofing attack," *IEEE Transactions on Information Forensics and Security*, 2017.

[26] H. Li, S. Wang, and A. C. Kot, "Face spoofing detection with image quality regression," in *IEEE International Conference on Image Processing Theory, Tools and Application*, 2016, pp. 1–6.

[27] 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, Apr 2015.

[28] T. Wang, J. Yang, Z. Lei, S. Liao, and S. Z. Li, "Face liveness detection using 3d structure recovered from a single camera," in *IEEE International Conference on Biometrics (ICB)*, 2013, pp. 1–6.

[29] Y. Wang, F. Nian, T. Li, Z. Meng, and K. Wang, "Robust face anti-spoofing with depth information," *Journal of Visual Communication and Image Representation*, vol. 49, pp. 332–337, 2017.

[30] V. Conotter, E. Bodnari, G. Boato, and H. Farid, "Physiologically-based detection of computer generated faces in video," in *IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 248–252.

[31] G. Chetty, "Biometric liveness checking using multimodal fuzzy fusion," in *IEEE International Conference on Fuzzy Systems*, 2010, pp. 1–8.

[32] J. Yang, Z. Lei, and S. Z. Li, "Learn convolutional neural network for face anti-spoofing," *arXiv preprint arXiv:1408.5601*, 2014.

[33] N. N. Lakshminarayana, N. Narayan, N. Napp, S. Setlur, and V. Govindaraju, "A discriminative spatio-temporal mapping of face for liveness detection," in *Identity, Security and Behavior Analysis (ISBA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–7.

[34] X. Tu and Y. Fang, "Ultra-deep neural network for face anti-spoofing," in *International Conference on Neural Information Processing*. Springer, 2017, pp. 686–695.

[35] 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*. Springer, 2017, pp. 27–34.

[36] 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 *Image Processing Theory Tools and Applications (IPTA), 2016 6th International Conference on*. IEEE, 2016, pp. 1–6.

[37] D. Menotti, G. Chiachia, A. Pinto, W. Robson Schwartz, H. Pedrini, A. Xavier Falcao, and A. Rocha, "Deep representations for iris, face, and fingerprint spoofing detection," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 864–879, Apr 2015.

[38] R. F. Nogueira, R. de Alencar Lotufo, and R. C. Machado, "Fingerprint liveness detection using convolutional neural networks," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1206–1213, Jun 2016.

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[40] 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.

[41] Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu, "Face anti-spoofing using patch and depth-based cnns," in *In Proceeding of International Joint Conference on Biometrics*, Denver, CO, October 2017.

[42] J. Gan, S. Li, Y. Zhai, and C. Liu, "3d convolutional neural network based on face anti-spoofing," in *Multimedia and Image Processing (ICMIP), 2017 2nd International Conference on*. IEEE, 2017, pp. 1–5.

[43] H. Cao and A. C. Kot, "Identification of recaptured photographs on lcd screens," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2010, pp. 1790–1793.

[44] T. Thongkamwitoon, H. Muammar, and P.-L. Dragotti, "An image recapture detection algorithm based on learning dictionaries of edge profiles," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 5, pp. 953–968, May 2015.

[45] H. Li, S. Wang, and A. C. Kot, "Image recapture detection with convolutional and recurrent neural networks," *Electronic Imaging*, vol. 2017, no. 7, pp. 87–91, 2017.

[46] W. Wang and H. Farid, "Detecting re-projected video." in *Information Hiding*, vol. 5284. Springer, 2008, pp. 72–86.

[47] P. Bestagini, M. Visentini-Scarzanella, M. Tagliasacchi, P. L. Dragotti, and S. Tubaro, "Video recapture detection based on ghosting artifact analysis," in *IEEE International Conference on Image Processing (ICIP)*, 2013, pp. 4457–4461.

[48] A. Pinto, H. Pedrini, W. R. Schwartz, and A. Rocha, "Face spoofing detection through visual codebooks of spectral temporal cubes," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4726–4740, 2015.

[49] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[50] M. C. Stone, W. B. Cowan, and J. C. Beatty, "Color gamut mapping and the printing of digital color images," *ACM Transactions on Graphics (TOG)*, vol. 7, no. 4, pp. 249–292, 1988.

[51] H. Li, A. C. Kot, and L. Li, "Color space identification from single images," in *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 1774–1777.

[52] H. Li, S. J. 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.

[53] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.

[54] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in neural information processing systems*, 1990, pp. 396–404.

[55] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of operations research*, vol. 134, no. 1, pp. 19–67, 2005.

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

[57] H. Li, W. Li, H. Cao, S. Wang, F. Huang, and A. C. Kot, "Unsupervised domain adaptation for face anti-spoofing," *IEEE Transactions on Information Forensics and Security (in press)*, 2018.

[58] K. Patel, H. Han, and A. Jain, " Secure Face Unlock: Spoof Detection on Smartphones," *IEEE Trans. Information Forensic and Security*, vol. 11, no. 10, pp. 2268 – 2283, Oct 2016.

[59] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2001, pp. I–511.

[60] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[61] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems*, 2014, pp. 3320–3328.

[62] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[63] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[64] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[65] S. Bharadwaj, T. I. Dhamecha, M. Vatsa, and R. Singh, "Computationally efficient face spoofing detection with motion magnification," in *The IEEE International Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2013, pp. 105–110.

[66] S. Tirunagari, N. Poh, D. Windridge, A. Iorliam, N. Suki, and A. T. Ho, "Detection of face spoofing using visual dynamics," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 762–777, Apr 2015.

[67] K. Muandet, D. Balduzzi, and B. Schölkopf, "Domain generalization via invariant feature representation," in *International Conference on Machine Learning (ICML)*, 2013, pp. 10–18.

[68] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, Feb 2011.