Spoof Face Detection Via Semi-Supervised Adversarial Training

Chengwei Chen*, Wang Yuan*, Xuequan Lu*,**, Lizhuang Ma*

*a East China Normal University, North Zhongshan Road Campus: 3663 N, Shanghai and 200062, China
**Deakin University, 75 Pigdons Rd, Waurn Ponds VIC 3216, Australia

1. Introduction

Biometrics plays a key part in authentication and security applications. Access control using face, fingerprint or iris has been existed for quite a while in our daily life. Face recognition, one of the prevalent biometric applications, has achieved noticeable successes (Galbally et al., 2014). Face data has been a promising data type, due to its convenience, universality and acceptability for users. However, traditional face recognition systems can be easily fooled with common attacks like printed photographs. To obtain access to systems, criminals are already using some techniques to accurately simulate the biometric characteristics of valid users, such as faces. This process is known as face spoofing attack, which poses a great threat to face recognition systems (Patel et al., 2016). Presentation attacks (abbreviated as PA), including printed paper face, replaying a video and wearing a mask, are one of the most prevalent face spoofs. It has been demonstrated that traditional face recognition systems could be vulnerable to PA

*Corresponding author
e-mail: xuequan.lu@deakin.edu.au (Xuequan Lu)
on high-level learning features achieve more promising performance. However, these CNN-based methods [Li et al., 2016; Patel et al., 2016a] are adopted in spoof face detection with a softmax loss based on binary supervision. These supervised methods have the risk of overfitting on the training data and obtain low performance in the cross-dataset setting. In addition, temporal information is also a critical part in spoof face detection. For example, a liveness detection method [Bao et al., 2009] is proposed for spoofing face detection with using optical flow. It attempts to find the differences in motion patterns. The model tries to learn the concept of optical flow generated by 3D objects and 2D planes. The motion of an optical flow field consists of four basic movements: translation, rotation, moving, and swing. Previous methods either do not model the motion or use complicated modules such as modeling the motion. Based on these features which focus on representing specific characteristics, the trained model can make the real face images and spoof face images more separable. However, because of the specificity, these methods are hard to be generalized to other spoofing types.

Previous face anti-spoofing works focused on supervised methods, with the utilization of hand-crafted or learned features. Most approaches typically depend on binary or auxiliary supervision. Nevertheless, many previous works suffer from the following major limitations partially or wholly: (1) fully supervised setting—the utilization of both live and spoof face data (with labels), (2) the assumption of binary classification, and (3) the impracticality to take all types of spoof (maybe unknown spoof) into account. Furthermore, collecting spoof face data for training purpose is costly and time-consuming. Also, binary supervision could be insufficient to learn a good model and make desired predictions in cross dataset scenario. As a result, those face anti-spoofing techniques have limited robustness and generalization to various types of spoofing.

Motivated by the above limitations and analysis, we propose a novel adversarial network for anti-spoofing under the semi-supervised setting. We propose to train the live face data only, with a convolutional Encoder-Decoder network acting as a Generator. Besides, a second convolutional network is regarded as a Discriminator. The generator attempts to reconstruct the original input sample to fool the discriminator, while the discriminator tries to distinguish original images from generated images. In the process of training, both sub-networks compete with each other to achieve high-quality reconstructions for live faces data only.

While testing, the learned model has a lower reconstruction error of live face data than spoof face data. This is mainly because we train on live face data only, the model captures the real characteristic of live faces samples and the learned model can better describe the characteristics of live faces than those of spoof faces. We naturally take the optical flow maps converted from consecutive video frames as input, as the task of spoof face detection is video-based and involves temporal information. The semi-supervised setting significantly reduces the efforts in collecting spoof face data, thus making our method more robust and general to different types of face spoofing. As such, the proposed approach is practical in the real world. We validate our method on challenging datasets. We also compare our semi-supervised method with state-of-the-art supervised anti-spoofing techniques, showing that our method produces better or comparable results to those approaches.

In summary, the main contributions of this paper are:

- a novel semi-supervised approach training on live face data only for spoof face detection.
- we propose a framework trained by generator and discriminator adversarially while collaborating to understand the real underlying concept in the normal class and classifying the testing samples by pixel-wise reconstruction error.
- we design a domain adaption algorithm which tries to learn some transfer components across domains in a Reproducing Kernel Hilbert Space (RKHS) using Maximum Mean Discrepancy (MMD).
- validation on challenging datasets, and extensive comparisons (intra- and cross-dataset testing) with current supervised anti-spoofing techniques.

The rest the paper is organized as follows. Section 2 reviews the relevant research. We elaborate our approach in Section 3. Section 4 gives various experimental results, and Section 5 concludes our work.

2. Related Work

The previous face anti-spoofing methods (Boulkenafet et al., 2017) [de Freitas Pereira et al., 2012, 2013; Komulainen et al., 2013a; Määttä et al., 2011; Mirjalili and Ross, 2017; Patel et al., 2016b; Yang et al., 2013] can be generally divided into four categories: feature based methods, temporal information based methods, Hybrid methods as well as approaches based on other cues. Tab. 1 compares the characteristics of these previous spoof detection methods, including LBP (Määttä et al., 2011), DoG-SL (Peixoto et al., 2011), Color-texture (Boulkenafet et al., 2015), Optical flow field (Bao et al., 2009), Liveness optical flow (Smiatacz, 2012), Structure-tensor (Kollreider et al., 2005), Spatial-temporal domain (Sun et al., 2018), Patch-based CNN (Atoum et al., 2017), VGG [Li et al., 2016], Auxiliary (Li et al., 2018) and De-Spoof (Jourabloo et al., 2018).

2.1. Feature-based Methods

Most early face anti-spoofing works used handcrafted features of texture information for binary classification (e.g., SVM). They expected that differing feature descriptors such as LBP (de Freitas Pereira et al., 2012, 2013; Määttä et al., 2011), HOG (Komulainen et al., 2013a; Yang et al., 2013), DoG-SL (Komulainen et al., 2013a; Yang et al., 2013), SIFT (Patel et al., 2016b) and SURF (Chingovska et al., 2012) could be computed for live and spoof faces. Nonetheless, many feature descriptors are largely affected by illumination, imagery and other factors. Such feature-based methods often have poor generalization in cross-dataset testing (Liu et al., 2018).
CNN is good at extracting and learning deep features. Yang et al. (2014) treated CNN as a classifier for face anti-spoofing, and used different spatial scales of live and spoof face images for training. Xu et al. (2015) proposed a LSTM-CNN architecture to predict the frames of videos. Most previous CNN techniques for face anti-spoofing utilized a binary classification to predict live or spoof faces. Feng et al. (2016), Li et al. (2016), Patel et al. (2016a), and Yang et al. (2014). However, both live and spoof face data have to be considered in the training procedure. In worse cases, the test face data does not involve cues like printed page edges or digital replay devices while the trained model might use such cues to detect spoof faces. As a result, the classification ability for live and spoof faces is limited. Also, it is difficult to explain the final results.

Table 1. Characteristics of different face spoof detection methods.

| Methods       | Analysis type       | Strategy                     | Datasets                     | Algorithm type  |
|---------------|---------------------|------------------------------|------------------------------|-----------------|
| LBP           | Texture analysis    | Micro-texture analysis via LBP with SVM as a classifier | NUAA Photograph Imposter Database | Supervised     |
| DoG-SL        | Texture analysis    | Applying an adaptive histogram equalisation to the images | Yale Face Database and NUAA Photograph Imposter Database | Supervised     |
| Color-texture | Texture analysis    | Computing a half of Face with another half that is divided in two ways: horizontally and vertically | CASIA Face Anti-Spoofing and the Replay-Attack databases | Supervised     |
| Optical flow field | Motion analysis | Analyzing the optical flow field to detect real face | -                            | Supervised     |
| Structure tensor | Motion analysis | Face motion estimation based on the structure tensor and a few frames | XM2VTS database | Supervised     |
| Spatial-temporal Domain | Motion analysis + Texture analysis | A two-stream structure (spatial, temporal) | Replay-Attack, CASIA and 3DMAD | Supervised     |
| Patch-based CNN | Texture analysis + cue analysis | Extracting the local features and holistic depth maps from the face images | CASIA-FASD, MSU-USSA, and Replay-Attack | Supervised     |
| VGG           | Texture analysis    | Extracting the deep partial features from the convolutional neural network (CNN) | Replay-Attack and CASIA | Supervised     |
| Liveness optical flow | Motion analysis | Applying the Support Vector Machine to distinguish between the motion information of real faces and photographs | Regensburg university dataset | Supervised     |
| Auxiliary     | Texture analysis + cues analysis | Fusing the estimated depth and rPPG to distinguish live v.s. spoof faces | CASIA-MFSD and Replay-Attack | Supervised     |
| De-Spoof      | Texture analysis    | A CNN architecture with proper constraints and supervisions | Oulu-NPU, CASIA-MFSD and Replay-Attack | Supervised     |
| Our method    | Motion analysis     | A semi-supervised adversarial learning framework | Nuaa, CASIA-MFSD and Replay-Attack | Semi-Supervised |

2.3. Hybrid Methods

Hybrid techniques combining features and temporal information have also been proposed for spoof face detection. Schwartz et al. (2011) used multiple low-level features to create one high dimensional vector with the size of more than one million. They further adopted the partial least squares approach on this vector to distinct between live and spoof faces. Komulainen et al. (2013b) introduced the combination of computationally inexpensive linear classifiers for robust face anti-spoofing. They used the fusion of motion information and features. Both methods depend on the multi-block local binary pattern and motion estimation from input videos.

2.4. Methods Based on Other Cues

There have been considerable amount of works using other cues derived from the original video frames (Komulainen et al. 2013a, George et al. 2019). For example, rPPG signal, IR image (Zhang et al. 2011), depth image (Wang et al. 2013) and voice (Chetty 2010) are some common cues. Nevertheless, such cues have their own limitations. Taking rPPG-based methods as an instance, researchers often need to extract the rPPG signals from a long video, to achieve decent predictions (Liu et al. 2018). As a matter of fact, it is unfeasible for a face anti-spoofing system to detect spoof faces through analyzing a long video (e.g., 50 seconds).

2.5. Anomaly detection

Anomaly detection is a classical problem in computer vision. When samples are deviating from the expected behavior defined by “normal” samples of a training dataset, these samples are classified as the abnormal class.
Fig. 1. Our framework consists of a generator and a discriminator. The generator and discriminator are trained by competing and collaborating with each other to understand the underlying structure in the live faces data. The architecture layers of each component are described on the right.

Recently, deep learning based autoencoders are used to learn the pattern of normal behaviors and exploit the reconstruction loss to detect anomalies. For example, Baur et al. (2018) tackles the problem by learning a mapping to a lower dimensional representation, where the real distribution is modeled. The decoder upscales the latent feature vector to reconstruct the image. In recent research, a lot of abnormal detection methods (Zenati et al., 2018; Xia et al., 2019) based on the Generative Adversarial Networks (GANs) are proposed. For instance, Xia et al. (2019) proposed latent spatial features based on generative adversarial networks for face anti-spoofing with an additional feature classifier. The input of this framework extracts the appearance information from the original face image with different sizes. In our work, instead we use the motion information from the original face images. According to the ablation study (Section 4.2.2), the performance of using motion information is better than using appearance information. Moreover, each size of the input corresponds to one GAN model, which induces significantly higher costs. In addition, their framework only reports a high performance in the intra-dataset setting. It disregards the generalization issue by excluding the cross-dataset setting, which is critical for spoofing detection.

3. Proposed Approach

In this section, we present how to learn the intrinsic structure of live faces by using the proposed adversarial training framework. We start by describing the details of the overview network architecture, then depict each term in loss function, and finally give the description of the testing method.

3.1. Network Architecture

Our method consists of a data preprocessing step and a GAN-style architecture. The preprocessing step is to convert consecutive video frames into optical flow maps. Fig. 2 shows the visualization of optical flow map. The GAN-style architecture, inspired by the anomaly detection (Sabokrou et al., 2018), comprises of two components: the generation network and the discrimination network. Fig. 1 shows the overview of our framework.

Due to the outstanding performance of CNN (Krizhevsky et al., 2012; Lawrence et al., 1997; Kalchbrenner et al., 2014), we take a convolutional autoencoder as the Generator. The main idea is that we only consider the live face data for training. The learned model is therefore not good at depicting the characteristics of spoof face data, leading to high reconstruction errors. The reason why we employ Convolutional AutoEncoder (CAE) in the proposed framework can be concluded as...
follows: (1) Conventional Autoencoders (AEs) often ignore the structure of 2D images, and interpret the input as a single latent vector. (2) The network is constrained by the number of input images. The redundant parameters in AEs force each feature to be global by spanning the entire visual field. (3) The Convolutional AutoEncoder (CAE) can learn the optimal filters to minimize the reconstruction error. In fact, Convolutional Neural Networks are usually referred to supervised learning algorithms. CAE, instead, is trained only to learn filters to extract features that can be used to reconstruct the input.

To prevent being fooled by the generator, the discriminator learns the core characteristics in the original data during the period of training. The discriminator also assists the generator to get robust and stable parameters in the process of training. This part of parameters would increase the reconstruction gap between live faces and fake faces in the process of testing.

3.2. Overall Loss Function

To train our model, we define a loss function in Eq. (1) including two components, the adversarial loss and the pixel-wise image reconstruction loss.

\[ \mathcal{L} = w_i \mathcal{L}_{\text{rec}} + w_a \mathcal{L}_{\text{adv}}, \]

where \( w_i \) and \( w_a \) are the weighting parameters balancing the impact of individual item to the overall object function.

Adversarial loss. The Generative Adversarial Network (GAN) (Creswell et al., 2018) originates from a game between two players. One player is called the generator \( G(x) \). The generator creates samples that are intended to come from the same distribution as the training data. The other player is called the discriminator \( D(x) \). The discriminator would make a decision whether the samples are generated by the generator or taken from the training data. The generator attends to fool the discriminator by reconstructing fake samples similar to the true training data. This adversarial game between the generator and discriminator can be formulated as:

\[ \mathcal{L}_{\text{adv}} = \min_{G} \max_{D} \mathbb{E}_{x \sim p_{x}} [\log(D(x))] + \mathbb{E}_{x \sim p_{x}} [\log(1 - D(G(x)))] . \]

(2)

Image reconstruction loss: While the discriminator tries to differentiate between realistic images and generated images, and the generator trying to fool the discriminator. However, the generator is not optimized towards learning the real concept from input data only by adversarial loss. Some prior works have proposed that the distance between input images and generated images should be considered. Isola et al. (2017) shows that the use of L1 yields less blurry results than L2. Therefore, we use L1 loss function to penalize the generator by minimizing the distance between original input \( x \) and generated images \( G(x) \) as follows.

\[ \mathcal{L}_{\text{rec}} = \mathbb{E}_{x \sim p_{x}} \| x - G(x) \|_1 \]  

(3)

3.3. Data Preprocessing

We extract frames from each video with 30 frames per second. FlowNet2.0 (Ilg et al., 2017) is then employed to estimate the optical flow between frames, due to its effectiveness. Optical flow is the pattern of apparent motion, which is calculated based on two adjacent images. It defines both horizontal
and vertical displacements for each pixel, and reflects motion about objects and scene. The pre-trained FlowNet model estimates the optical flow between each pair of two adjacent frames and outputs the optical flow files. The horizontal and vertical components are included in optical flow files. The color-coding scheme (Lopez, 2017) allows us to visualize the horizontal and vertical displacements in one image, as illustrated in Fig. 5. Colors can be assigned to each pixel. We utilize the color-coding scheme to convert these optical flow files into images where the displacement vector is color.

The output flow maps are also RGB images with colors indicating the flow signal. The patches are generated from each flow map by a sliding window. The size of this window is set to $32 \times 32$. Fig. 2, 3 and 4 visualize optical flows of live faces, spoofing faces by holding the client biometry (i.e., fixed spoofing) and spoofing faces from the device held by the attacker’s hands (i.e., hand spoofing), respectively. It shows that the optical flows of live faces are more clear than the spoofing faces. Hand spoofing leads to considerable amount of noise on the attacker’s hands (i.e., hand spoofing), respectively. It shows that spoofing faces from the device held by the attacker leads to considerable amount of noise on the flow maps. This is because that the movement of spoofing faces and digital device screens is consistent. For fixed spoofing faces, spoofing faces by holding the client biometry (i.e., fixed spoofing) and spoofing faces by holding the client biometry (i.e., fixed spoofing) respectively. It shows that spoofing faces from the device held by the attacker’s hands (i.e., hand spoofing), respectively. It shows that spoofing faces from the device held by the attacker’s hands (i.e., hand spoofing), respectively. It shows that spoofing faces from the device held by the attacker’s hands (i.e., hand spoofing), respectively.

### 3.4. Testing method

To demonstrate the effectiveness of the proposed framework, we conduct two intra-testing experiments and one cross-testing experiment. For intra-testing experiments, the model is trained in the training dataset accordingly, as with the state-of-the-art methods (Yu and Jia, 2017). The testing dataset in the same domain is used to evaluate the performance of each method. Different from intra-testing experiments, cross-database experiments with different domains are more challenging. Domain adaptation (Finkel and Manning, 2009) is a field associated with machine learning and transfer learning. The aim of the domain adaptation problem is to train a well performing model from the source data distribution. The trained model could still perform well on a different (but related) target data distribution. As such, we attempt to extend the domain adaptation in our study. To our knowledge, we are the first to investigate the domain adaptation issue in the face anti-spoofing area.

In the cross-database situation, the labels of all target samples are unknown during training. Compared with the intra-database setting, it is more ubiquitous in real-world applications. Due to the unavailability of labels in the target domain, one commonly used strategy is to learn domain-invariant representations via minimizing the domain distribution discrepancy. In our cross-database scenario, the model is trained on dataset A and tested on dataset B. There exists some difference between the source domain and target domain, for example, image quality, reflection and environment. One intuitive solution is to consider mapping the reconstruction data to a high (possibly infinite) dimensional space and computing the sample means in this space using high-order statistics (up to infinity). As a result, we could achieve a better discrimination threshold for live and spoofing faces. By contrast, directly training a classifier on the source data and using the threshold set in the source data often leads to certain “overfitting” to the source distribution and reduced performance while testing on the target domain.

We consider a source domain $\mathcal{D}_s = \{x^s_i, y^s_i\}_{i=1}^{n_s}$ and a target domain $\mathcal{D}_t = \{x^t_i, y^t_i\}_{i=1}^{n_t}$. Here, $x^s_i \in \mathbb{R}^{n_s}, x^t_i \in \mathbb{R}^{n_t}$ are the reconstruction errors for each frame in the source domain and the target domain, respectively. $y^s_i \in C, y^t_i \in C$ are corresponding labels, where the target labels $\{y^t_i\}_{i=1}^{n_t}$ are not available for training. For domain adaption, we assume that the source and target domains are associated with the same label space, while $\mathcal{D}_s$ and $\mathcal{D}_t$ are drawn from distributions $P_s$ and $P_t$ which are assumed to be different. That is, the source and target distribution have different joint distributions of data $X$ and labels $Y$: $P_s(X, Y) \neq P_t(X, Y)$.

Maximum Mean Discrepancy (MMD) (Yan et al., 2017) is an effective non-parametric metric for comparing the distance between two distributions. Given two distributions $s$ and $t$, by mapping the data to a reproduced kernel Hilbert space (RKHS) using function $\phi(\cdot)$, the MMD between $s$ and $t$ is defined as,

$$
\text{MMD}(s, t) = \sup_{\|\phi\|_H \leq 1} \|E_{x^s}[\phi(x^s)] - E_{x^t}[\phi(x^t)]\|_H,
$$

where $E_{x^s}[\cdot]$ denotes expectation with respect to the distribution $s$, and $\|\phi\|_H \leq 1$ defines a set of functions in the unit ball of a RKHS. Based on the statistical tests defined by MMD, we have MMD($s, t$) = 0 $\iff$ $s = t$. Denote by $\mathcal{D}_s = \{x^s_i\}_{i=1}^{M}$ and $\mathcal{D}_t = \{x^t_i\}_{i=1}^{N}$, two sets of samples drawn i.i.d. from the distributions $s$ and $t$ respectively, the empirical estimation of MMD can be given by:

$$
\text{MMD}(\mathcal{D}_s, \mathcal{D}_t) = \left\| \frac{1}{M} \sum_{i=1}^{M} \phi(x^s_i) - \frac{1}{N} \sum_{j=1}^{N} \phi(x^t_j) \right\|_H,
$$

where $\phi(\cdot)$ denotes the feature map associated with the kernel map $k(x^s, x^t) = \langle \phi(x^s), \phi(x^t) \rangle$, which is usually defined as the convex combination of several basis kernels.

With the help of MMD, the statistical test method works in the following way. Based on the samples of two distributions, one distribution is the reference distribution formed by training live face samples, and another distribution is obtained in the same way from test samples. By finding the continuous function $\phi$ in the sample space, the mean value of the samples from different distributions on function $\phi$ is obtained. Dividing the

---

Fig. 5. Visualization of horizontal and vertical displacements in one RGB image.
two mean values yields an average difference between the two distributions. Finally, MMD is taken as the measurement to determine the category of the test videos. If the value of MMD is smaller than the predefined threshold $T$, the test samples distribution is considered to be the close to the live face reference distribution; otherwise they are spoof videos. The final testing scheme is summarized in Algorithm 1.

**Algorithm 1 Spoofing Detection in video $V$**

**input**: A video $V = \{F_1, F_2, ..., F_N\}$, trained models: Generator, Auxiliary encoder and decision threshold $T$  
**output**: report if $(\text{ScoreVideo} > T) \text{ 'spoof'}$ else 'non-spoof'

1: function $\text{DISTRIBUTION}(V)$  
2: $\text{FrameArray}[]$  
3: for $k = 1$ to $N$ do  
4: $\text{ScoreFrame} = \|G_e(F_k) - G_e(F'_k)\|_1$  
5: $\text{FrameArray} += \text{FrameArray} + \text{ScoreFrame}$  
6: end for  
7: return FrameArray  
8: end function  
9: $\text{ReferDis} = \text{DISTRIBUTION}(\text{TrainPosVideo})$  
10: $\text{TestDis} = \text{DISTRIBUTION}(\text{TestVideo})$  
11: $\text{ScoreVideo} = \text{MMD}(\text{ReferDis}, \text{TestDis})$

4. Experimental Results

In this section, we firstly present the experimental setting, including datasets, and more implementation details. Then, for ablation study, two experiments are conducted to analyze the proposed method in detail. Finally, We evaluate proposed method and the state-of-the-art techniques on both intra/inter-dataset settings.

4.1. Experimental Setup and Datasets

The proposed method is mainly implemented in the Tensorflow framework (Abadi et al. 2015). The experiments are carried out on a PC with a NVIDIA-1080 graphics card and a multi-core 2.1 GHz CPU. A good face anti-spoofing system must be robust to different types of attacks. We evaluate our method and the state-of-the-art techniques on three publicly available face spoofing detection databases: (i) NUAA Imposer Database (Tan et al. 2010a), (ii) Replay-Attack (Chin-govska et al. 2012) dataset, and (iii) CASIA MFSD (Zhang et al. 2012) dataset. These structures are kept fixed for all databases, and learning rate is set to 0.02.

According to the work (Akcay et al. 2018), CIFAR10 and MNIST datasets are used to construct the experiment to illustrate the superiority of our approach over the state-of-the-art one-class classifiers. One of the classes is regarded as normal class, while the rest ones belong to the abnormal class. In particular, we respectively get ten sets for MNIST and CIFAR10, and then detect the outlier anomalies by only training the model on the normal class data in ablation study.

The NUAA dataset is widely used for the evaluation of face liveness detection. This dataset consists of 15 different subjects captured in different places and illumination conditions, involving 12,614 real and photographed face images. Each subject was asked to look at the webcam frontally with a neutral expression and without noticeable movements such as eyeblink or head movement. For training data, it contains 1,743 real faces and 1,748 photographed faces. For testing, it includes 3,362 real faces and 5,761 photographed faces. Fig. 6 shows some samples from the NUAA dataset.

The Replay-Attack dataset is also a widely used and publicly available database. It has 360 videos (60 real faces videos and 300 spoof faces) as the training data. About validation data, it has the same number of videos as training videos. The validation data will be fully used, to calibrate the threshold to distinguish between real and spoof faces (explained in Section 3.4). The resolution of the Replay-Attack data is $320 \times 240$. The dataset considers different lighting conditions used in spoofing attacks. It consists of 80 videos of real faces and 400 videos of fake faces as the testing data. The fake faces are obtained by using the attackers’ bare hands or fixed support. Fig. 7 shows some samples from the Replay-Attack dataset.

The CASIA (Zhang et al. 2012) dataset involves 50 subjects, and each subject has 12 videos (3 real faces and 9 fake faces). The dataset is divided into the training set (20 subjects, 240 videos) and the test set (30 subjects, 360 videos). Compared with the Replay-Attack dataset, there is no validation data in this dataset. The CASIA dataset is more difficult in spoof face detection, in terms of image quality, resolution and video length. It consists of print and replay attacks using corresponding photos and replayed videos. Some of the print attack photos are manually cropped around the eyes to deter eye-blinking based techniques.

There exist a lot of face anti-spoofing approaches, and many CNN-based supervised works have achieved promising results in the intra-database setting. However, the high intra-database prediction accuracy does not guarantee a decent performance in the inter-database setting which is more common in real world. In fact, the cross-database performance better reflects the actual
capability of a system in real-world applications. Therefore, a good cross-database performance provides strong evidence that: i) features are generally invariant to different scenarios (i.e., camera and illuminations), ii) a spoof classifier trained in one scenario is generalizable to other scenarios, and iii) data captured in one scenario can be useful for developing effective spoof detectors in other scenarios. As such, to demonstrate the effectiveness of the proposed framework, we conduct two intra-database experiments and two cross-database experiments.

4.2. Ablation study

4.2.1. With/without the discriminator

To show the effectiveness of adversarial learning, it is necessary to conduct the experiment with or without the discriminator part. In the first scenario, we only use the common convolutional autoencoder with a simple image reconstruction error. During the inference, the reconstruction error of the test sample is regarded as the abnormality score. In the second scenario, all components are used in the proposed framework with the discriminator. Besides the image reconstruction error, the adversarial learning loss is also considered. The generator tries to generate a high-quality image to fool the discriminator. The discriminator attempts to distinguish the generated image from a realistic image. During the training process, the discriminator helps the generator to capture the underlying concept of normal samples. The test sample is detected in the same way as the first scenario (i.e., image reconstruction error).

Since we formulate the spoof face detection task as abnormal detection task, it is necessary to explore the effectiveness of the proposed method in both tasks. In the abnormal detection task, we conduct an experiment to demonstrate the superiority of our method over state-of-the-art one-class classifiers on MNIST and CIFAR10 datasets, shown in Tab. 2. For both MNIST and CIFAR10, we select one class as the normal class at each time, while leaving the rest to be the abnormal classes, leading to ten sets for abnormal detection. Normal data and abnormal data are to imitate live faces and spoof faces, respectively. Our method typically achieves improvements compared with other methods, including OCSVM (Schölkopf et al., 2001), KDE (Bishop, 2006), DAE (Hadsell et al., 2006), VAE (Kingma and Welling, 2013), Pix CNN (Kalchbrenner et al., 2016), AND (Abati et al., 2019) and DSVDD (Ruff et al., 2018).

In addition, it is clear that the proposed method with the discriminator achieves higher performance than the autoencoder without the support from discriminator in both datasets. In our spoofing face detection task, the cross-dataset experiment is also conducted by using the two scenarios (with and without the discriminator) described above. As shown in Tab. 3, the discriminator helps the generator (Autoencoder) to capture the concept of live faces. The trained generator is used to detect the spoof faces directly, and we obtain a better performance with the discriminator than the autoencoder strategy without the discriminator.

4.2.2. Impact on performance with optical flow

Table 4. Classification performance of the proposed approach in different types (motion or appearance) of input, in terms of HTER (%). The algorithm is trained using the CASIA-MFSD dataset and tested on the Replay-Attack dataset, and vice versa.

Table 3. Classification performance of autoencoder (without discriminator) and the proposed method (with discriminator), in terms of HTER (%). They are trained using the CASIA-MFSD dataset and tested on the Replay-Attack dataset, and vice versa.
original videos. A performance comparison between these two cases is presented in Tab. 2. Compared with appearance information, the motion information could better assist in distinguishing the spoofing faces from live faces.

4.3. Intra NUAA Database Experiment

We evaluate the performance of the proposed method and state-of-the-art techniques on the NUAA dataset, in an intra-database sense. The competitors include DoG and high frequency based (DoG-F) (Li et al., 2004), multiple difference of Gaussian (DoG-M) (Zhang et al., 2012), DoG-sparselogistic (DoG-SL) (Peixoto et al., 2011), diffused speed-local speed pattern (DS-LSP) (Kim et al., 2015), multiple local binary pattern (M-LBP) (Määttä et al., 2011), DoG-sparselow-rank bilinear logistic regression (DoG-LRBLR) (Tan et al., 2010b), DoG-sparselogistic (DoG-SL) (Peixoto et al., 2011), component-dependent descriptor (CDD) (Yang et al., 2013), ADKMM (Yu and Jia, 2017) and the nonlinear based convolution neural network (ND-CNN) (Alotaibi and Mahmood, 2017).

To evaluate the reconstruction performance for each epoch, we choose three face spoofing samples and three live samples from the train set randomly. Once the network are trained in each epoch, the trained model would output the reconstructed images of these live or spoofing samples. Fig. 8 reveals that the gap between the reconstruction losses of spoofing faces and those of live faces are increased until they become stable after a few epochs. This indicates that the proposed approach can quickly distinguish spoof face data from live samples, without requiring spoof face data for training. Tab. 2 shows the accuracies for all methods. Our semi-supervised approach achieves the best performance, which is the same as the supervised ADKMM (Yu and Jia, 2017) and supervised ND-CNN (Alotaibi and Mahmood, 2017).

4.4. Intra Replay-Attack Dataset Experiment

We also compared our method with state-of-the-art techniques on the Replay-Attack dataset, in the intra-database setting. Competitors include LBP, + SVM (Chingovska et al., 2012), LBP, + LDA (Chingovska et al., 2012), LBP, + SVM (Chingovska et al., 2012), LBP + SVM (Määttä et al., 2011), DS-LBP (Kim et al., 2015), ND-CNN (Alotaibi and Mahmood, 2017), VGG (Li et al., 2016), Color-texture (Boulkenafet et al., 2015), Fisher-vector-encoding (Boulkenafet et al., 2016), Depth-based-CNNs (Patch-based CNN, Depth-based CNN and Patch and depth CNN) (Atoum et al., 2017), D-K (Yu and Jia, 2017), DTCNN (Tu et al., 2019a), Hand-crafted + CNN (Rehman et al., 2020) and Generalized deep feature (Li et al., 2018a).

Table 5. Performance comparison using AUC on the NUAA dataset

| Methods                  | Accuracy |
|--------------------------|----------|
| Ours (semi-supervised)   | 99.3%    |
| ADKMM (’17)              | 99.3%    |
| ND-CNN (’17)             | 99.3%    |
| DS-LBP (’15)             | 98.5%    |
| CDD (’13)                | 97.7%    |
| DoG-SL (’11)             | 94.5%    |
| M-LBP (’11)              | 92.7%    |
| DoG-LRBLR (’10)          | 87.5%    |
| DoG-F (’04)              | 84.5%    |
| DoG-M (’12)              | 81.8%    |

Previous spoofing face detectors have achieved outstanding performance in the intra-dataset setting by supervised learning with both positive and negative labels. To a certain degree, these supervised methods have the risk of overfitting on the training data and obtain poor generalization in cross-dataset setting. In addition, in the real world, it is impossible for us to collect and cover all kinds of spoof faces. Some types of spoofing faces are even unknown. Based on these challenges, we formulate the spoofing faces detection task as an abnormal detection task by only training the normal samples (live faces), which obtain a comparable performance in intra dataset setting with strong generalization.

The proposed method obtains the best performance with only 40 epochs. Fig. 10 shows the reconstruction loss for live and spoof faces. Besides the training data and testing data, this dataset also provides the development data to evaluate the performance. We calculate the half total error rate (HER) (Bengo and Mariethoz, 2004) to measure the performance. The HER is half of the sum of the false rejection rate (FRR) and false acceptance rate (FAR). The half total error rate (HTER) would be also used in the metric of cross-database experiments.

\[ HTER = \frac{FRR + FAR}{2} \] (6)

The way to perform the attacks can be divided into two sub-
Table 7. Classification performance in terms of HTER (%). The models are trained using the CASIA-MFSD dataset and tested on the Replay-Attack dataset, and vice versa. 1: supervised method. 2: semi-supervised method.

| Methods                  | Train CASIA MFSD | Test Replay Attack | Train Replay Attack | Test CASIA MFSD | Average |
|--------------------------|------------------|--------------------|---------------------|-----------------|---------|
| 1-LBP ('13)              | 47.0%            | 39.6%              |                     |                 | 43.3%   |
| 1-LBP-TOP ('13)          | 49.7%            | 60.6%              | 55.2%               |                 |         |
| 1-Motion ('13)           | 50.2%            | 47.9%              | 49.1%               |                 |         |
| 1-CNN ('14)              | 48.5%            | 45.5%              | 47.0%               |                 |         |
| 1-Color LBP ('15)        | 37.9%            | 35.4%              | 36.7%               |                 |         |
| 1-Color Tex ('16)        | 30.3%            | 37.7%              | 34.0%               |                 |         |
| 1-Auxiliary ('18)        | 27.6%            | 28.4%              | 28.0%               |                 |         |
| 1-De-Spoof ('18)         | 28.5%            | 41.1%              | 34.8%               |                 |         |
| 1-DA ('18)               | 27.4%            | 36.0%              | 31.7%               |                 |         |
| 1-Dynamic texture ('18)  | 22.2%            | 35.0%              | 28.6%               |                 |         |
| 1-Of Domain ('18)        | 30.1%            | 36.8%              | 33.5%               |                 |         |
| 1-GFA-CNN ('19)          | 21.4%            | 34.3%              | 28.0%               |                 |         |
| 1-ADA ('19)              | 17.5%            | 41.6%              | 29.6%               |                 |         |

We first consider training on the training set of the CASIA-MFSD database and testing on the testing set of the Replay-Attack database. The quantitative results shown in Tab. 7 confirm that the proposed method achieves the best performance (HTER = 0.156) on the Replay-Attack test set which includes different types of spoofing attacks. The competitors consist of LBP (de Freitas Pereira et al., 2013), LBP-TOP (de Freitas Pereira et al., 2013), Motion (de Freitas Pereira et al., 2013), CNN (Yang et al., 2014), Color LBP (Boukenaët et al., 2015), Color Tex (Boukenaët et al., 2016), Auxiliary (Liu et al., 2018), De-Spoof (Jourabloo et al., 2018), DA (Li et al., 2018b), Dynamic texture (Shao et al., 2018), OF Domain (Sun et al., 2018), ADA (Wang et al., 2019) and GFA-CNN (Tu et al., 2019b). In Fig. 14 it is obvious that the trained model has strong generalization ability to make live faces and fake faces obtained by using the attackers’ bare hands separable. How-
Fig. 11. First row: original optical flow images of live faces. Second row: generated optical flow images of live faces. Third row: corresponding maps which display the differences between the original images (the first row) and generated images (the second row) from model by red points.

Fig. 12. First row: original optical flow images of spoofing faces. Second row: generated optical flow images of spoofing faces. Third row: corresponding maps which display the differences between original image (the first row) and generated images (the second row) by red point.

ever, some fake faces obtained by fixed support has the same distribution as live faces.

We then conduct the opposite experiment: training on the training set of the Replay-Attack dataset and testing on the testing set of the CASIA-MFSD dataset. Our method achieves a competitive performance ($HTER = 0.441$) for the cross testing on the testing set of the CASIA-MFSD dataset. From Tab. [7] we can see that the $HTER$ of our method is better than most binary supervision methods (Yang et al., 2014; Jourabloo et al., 2018; Wu et al., 2016; Boulkenafet et al., 2016, 2015). This demonstrates that the proposed approach can better identify the differences between live and fake faces.

As with all previous works (Wu et al. 2016; Jourabloo, Liu, and Liu 2018; Boulkenafet, Komulainen, and Hadid 2016), we observe that the models trained on CASIA-MFSD enables to generalize better than the model trained on the Replay Attack Database. We speculate as follows (1) It is probably because the resolution of the CASIA-MFSD data is significantly higher than that in the Replay-Attack dataset. The model trained with high resolution could generalize better than the model trained with low resolution. (2) Compared with Replay-Attack, the CASIA-MFSD contains more variations in collected database, For example, imaging quality, the distance between camera and face, background and attack types. Hence, the model optimized for Replay-Attack databases faces more challenge in the new acquisition conditions. This is one limitation of the our method and previous works, and worthy further research. As with previous works [Wu et al., 2016; Jourabloo et al., 2018; Boulkenafet et al., 2016], the two above cross-database experiments do not have the same $HTER$. We speculate that it is probably because the resolution of the CASIA-MFSD data is significantly higher than that in the Replay-Attack dataset. In Fig. [11] and [12] the first row and second row show the optical flow maps and the reconstructed optical flow maps, respectively. The third row visualizes the differences between the corresponding optical maps and reconstructed maps (red color). These visualization results clearly demonstrate that the reconstruction errors from live faces are lower than that of spoofing faces.
5. Conclusion
We have presented an adversarial framework for the detection of spoofing faces. Given an input face image, the trained model can automatically determine if it is a live or spoof face. Current face anti-spoofing techniques have to utilize both spoof data and live data for training, which can hardly cover every type of spoof faces. By contrast, our approach does not need spoof data for training, and is thus semi-supervised and robust to different types of spoof faces. Both the intra-/cross-database experiments show that our method achieves better or comparable results to state-of-the-art techniques. We believe our research will arouse some new insights in this field.

References
Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. URL: http://tensorflow.org/software available from tensorflow.org.
Abati, D., Porrello, A., Calderara, S., Cucchiara, R., 2019. Latent space autoregression for novelty detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 481–490.
Alpaydiner, S., Atapour-Abarghouei, A., Breckon, T.P., 2018. Ganomaly: Semi-supervised anomaly detection via adversarial training, in: Asian Conference on Computer Vision, Springer. pp. 622–637.
Alotaibi, A., Mahmood, A., 2017. Deep face liveness detection based on nonlinear diffusion using convolution neural network. Signal, Image and Video Processing 11, 713–720.
Atoun, Y., Liu, Y., Jourabloo, A., Liu, X., 2017. Face anti-spoofing using patch and depth-based cnns, in: 2017 IEEE International Joint Conference on Biometrics (IJCB), IEEE, pp. 319–328.
Bao, W., Li, H., Li, N., Jiang, W., 2009. A liveness detection method for face recognition based on optical flow field, in: 2009 International Conference on Image Analysis and Signal Processing, IEEE. pp. 233–236.
Baur, C., Wiestler, B., Albarqouni, S., Navab, N., 2018. Deep autoencoding models for unsupervised anomaly segmentation in brain mr images, in: International MICCAI Brainlesion Workshop, Springer. pp. 161–169.
Bengio, S., Mariethoz, J., 2004. A statistical significance test for person authentication, in: Proceedings of Odyssey 2004: The Speaker and Language Recognition Workshop.
Bishop, C.M., 2006. Pattern recognition and machine learning. springer.
Boulkenafet, Z., Komulainen, J., Hadid, A., 2015. Face anti-spoofing based on color texture analysis, in: 2015 IEEE international conference on image processing (ICIP), IEEE. pp. 2636–2640.
Boulkenafet, Z., Komulainen, J., Hadid, A., 2016. Face antispooing using speeded-up robust features and fisher vector encoding. IEEE Signal Processing Letters 24, 141–145.
Boulkenafet, Z., Komulainen, J., Hadid, A., 2017. Face anti-spoofing using speeded-up robust features and fisher vector encoding. IEEE Signal Processing Letters 24, 141–145.
Chetty, G., 2010. Biometric liveness checking using multimodal fuzzy fusion, in: International Conference on Fuzzy Systems, IEEE. pp. 1–8.
Chetty, G., Wagner, M., 2006. Multi-level liveness verification for face-voice biometric authentication, in: 2006 Biometrics Symposium: Special Session on Research at the Biometric Consortium Conference, IEEE. pp. 1–6.
Chingovska, I., Anjos, A., Marcel, S., 2012. On the effectiveness of local binary patterns in face anti-spooing, in: 2012 BIOSIG-proceedings of the international conference of biometrics special interest group (BIOSIG), IEEE. pp. 1–7.
Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., Bharath, A.A., 2018. Generative adversarial networks: An overview. IEEE Signal Processing Magazine 35, 53–65.
Feng, L., Po, L.M., Li, Y., Xu, X., Yuan, F., Cheung, T.C.H., Cheung, K.W., 2016. Integration of image quality and motion cues for face anti-spoofing: A neural network approach. Journal of Visual Communication and Image Representation 38, 451–460.
Finkel, J.R., Manning, C.D., 2009. Hierarchical bayesian domain adaptation, in: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics. pp. 602–610.
de Freitas Pereira, T., Anjos, A., De Martino, J.M., Marcel, S., 2012. Lbp- top based countermeasure against face spoofing attacks, in: Asian Conference on Computer Vision. Springer. pp. 121–132.
de Freitas Pereira, T., Anjos, A., De Martino, J.M., Marcel, S., 2013. Can face anti-spoofing countermeasures work in a real world scenario?, in: 2013 international conference on biometrics (ICB), IEEE. pp. 1–8.
Frishholz, R.W., Dieckmann, U., 2000. Biold: a multimodal biometric identification system. Computer 33, 64–68.
Frishholz, R.W., Werner, A., 2003. Avoiding replay-attacks in a face recognition system using head-pose estimation, in: Analysis and Modeling of Faces and Gestures, 2003. AMFG 2003. IEEE International Workshop on, IEEE. pp. 234–235.
Galbally, J., Marcel, S., Fierrez, J., 2014. Biometric anti-spoofing methods: A survey in face recognition. IEEE Access 2, 1530–1552.
Gan, J., Li, S., Zhai, Y., Liu, C., 2017. 3d convolutional neural network based on face anti-spoofing, in: 2017 2nd international conference on multimedia and image processing (ICMIP), IEEE. pp. 1–5.
George, A., Mostaani, Z., Geissenbuhler, D., Nikisins, O., Anjos, A., Marcel,
Zenati, H., Romain, M., Foo, C.S., Lecouat, B., Chandrasekhar, V., 2018. Adversarially learned anomaly detection, in: 2018 IEEE International Conference on Data Mining (ICDM), IEEE. pp. 727–736.
Zhang, Z., Yan, J., Liu, S., Lei, Z., Yi, D., Li, S.Z., 2012. A face antispoofing database with diverse attacks, in: 2012 5th IAPR international conference on Biometrics (ICB), IEEE. pp. 26–31.
Zhang, Z., Yi, D., Lei, Z., Li, S.Z., et al., 2011. Face liveness detection by learning multispectral reflectance distributions., in: FG, pp. 436–441.