J. Vis. Commun. Image R. 59 (2019) 574–582

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Face liveness detection using convolutional-features fusion of real and deep network generated face images ☆

Yasar Abbas Ur Rehman *, Lai-Man Po, Mengyang Liu, Zijie Zou, Weifeng Ou, Yuzhi Zhao

Department of Electronic Engineering, City University of Hong Kong, Hong Kong Special Administrative Region

ARTICLE INFO

Article history:
Received 3 January 2019
Revised 9 February 2019
Accepted 12 February 2019
Available online 13 February 2019

Keywords:
Convolution neural networks
Face anti-spoofing
Face liveness detection
Adaptive fusion
Auto-encoder
DNG face images

ABSTRACT

Conventionally, classifiers designed for face liveness detection are trained on real-world images, where real-face images and corresponding face presentation attacks (PA) are very much overlapped. However, a little research has been carried out in utilization of the combination of real-world face images and face images generated by deep convolutional neural networks (CNN) for face liveness detection. In this paper, we evaluate the adaptive fusion of convolutional-features learned by convolutional layers from real-world face images and deep CNN generated face images for face liveness detection. Additionally, we propose an adaptive convolutional-features fusion layer that adaptively balance the fusion of convolutional-features of real-world face images and face images generated by deep CNN during training. Our extensive experiments on the state-of-the-art face anti-spoofing databases, i.e., CASIA, OULU and Replay-Attack face anti-spoofing databases with both intra-database and cross-database scenarios indicate promising performance of the proposed method on face liveness detection compared to state-of-the-art methods.

1. Introduction

Face recognition utilizing state-of-the-art feature learning algorithms [1–8] have gained remarkable accuracy in recent years. However, the security of face recognition based systems has been a major concern particularly in face biometric based user authentication and security applications. While face recognition based systems can classify different users based on their facial appearance with remarkable accuracy; however, face recognition based systems are unable to classify whether a given face represents a genuine user's face also known as live face, or an imposter is impersonating as a genuine user by presenting a fake face also known as spoof face of a genuine user in order to get an illegal access to the genuine user's assets. Thus, face liveness detection has been an indispensable and key design requirement in modern face recognition based systems used for access control.

Face liveness detection or face anti-spoofing algorithms and systems that deal mainly with the detection of face spoofing attacks are capable of efficiently classifying face presentation attacks (PAs) that are captured under the similar conditions as PA samples that the face liveness detection system is trained on. However, these face liveness detection algorithms and systems have low performance in unconstrained or cross-database face liveness detection scenarios [9]. As a result, the realization of low cost and reliable face liveness detection system is still an open research issue.

In recent years, various algorithmic development for face liveness detection systems have been reported. These developments can be broadly classified into two domains: fixed features based face anti-spoofing systems and deep features based face anti-spoofing systems [10]. Fixed features based face anti-spoofing systems exploit hand-crafted features to perform classification between live face and PA. On the other hand, deep features based face anti-spoofing systems utilize deep neural networks, such as convolutional neural networks (CNN), to classify a live face and PA [11]. Since, features learned by deep neural networks are dynamic, they are currently the most preferred choice for most face anti-spoofing systems [11–13]. However, most of these studies either utilized a transfer learning approach or trained CNN networks with binary supervision. As a result, the performance of these systems in general degrades in unconstrained face anti-spoofing scenarios.

Conventional face liveness detection systems are trained on real-world live face images and their corresponding PA in existing color spaces, such as RGB, YcbCr and HSV and their combination

☆ This paper has been recommended for acceptance by 'Zicheng Liu'.

* Corresponding author.
E-mail addresses: yaurehman2-c@my.cityu.edu.hk (Y.A.U. Rehman), eelmpo@-cityu.edu.hk (L.-M. Po), mengyaliu7-c@my.cityu.edu.hk (M. Liu), zijiezou2-c@my.cityu.edu.hk (Z. Zou), weifengou2-c@my.cityu.edu.hk (W. Ou), yzzhao2-c@my.cityu.edu.hk (Y. Zhao).

https://doi.org/10.1016/j.jvcir.2019.02.014
1047-3203/

[14–16]. However, live face images and corresponding PA are very much overlapped (because of the constraints imposed during color image generation) in these existing color spaces. Therefore, it is still an open research problem to find a color space or combination of color spaces which can be best for face liveness detection in general. Additionally, the problem of face liveness detection has been studied as a binary-class classification problem. However, a binary classification approach to face liveness detection is unable to explain the latent capabilities of the classification model, like CNN, in classifying a real face and a PA. Thus, when an extra supervision or additional information is available to the face liveness detection classification model, the decision regarding the classification of a face image being considered as live or PA can be explained better. Further, the CNN models with additional level of supervision have been shown to be effective for face liveness detection in general [17].

Recently, it has been shown that the images generated by deep neural networks like auto-encoder and generative adversarial networks (GAN) do not consider the constraints that are imposed during real-world image generation [18]. Additionally, the work in [19] proposed to supervise a CNN model for PA detection by learning a color generator that transform the existing color space to a learned color space. By training a CNN model in the learned color space, the authors achieved better performance than using RGB, HSV and YCbCr color space for face liveness detection in intra-database scenarios. While these results are encouraging for face liveness detection; a robust and reliable face liveness detection system that can perform well in unconstrained environment and that can effectively detect unknown PA in general are still open research problems [20].

To this end, we propose to exploit the adaptive fusion of convolutional features of real-world face images and deep CNN based auto-encoder generated (DNG) face images for face liveness detection as shown in Fig. 1. Particularly, we utilize the adaptive disparity and blending between the convolutional features of real-world face images and DNG face images learned by convolutional layers with shared weights in CNN for face liveness detection. As depicted in Fig. 1, rather than using conventional fusion, we propose to construct a special layer called adaptive convolutional-features fusion layer, in the CNN, that adaptively learn to weight the disparity and blending between the convolutional features of real-world face images and DNG face images. The layers in CNN following the proposed adaptive convolutional-features fusion layer are supervised by the adaptive fusion of convolutional features of real-world face images and DNG face images for face liveness detection. Our extensive experimental analyses indicate that the proposed supervision improves the performance of face liveness detection.

The main contribution of this paper can be summarized as follows:

- We utilize a deep CNN network with an adaptive convolutional-feature fusion layer that perform the weighted fusion between convolutional features learned by the convolutional layers from real-world face images and DNG face images.
- We evaluate the choice of adaptive kernel window for finding suitable weights matrix for fusing convolutional features of real-world face images and DNG face images for face liveness detection. Additionally, we also evaluate the effect of weighted blending and disparity (with different kernel window), of real-world face images and DNG face images, on face liveness detection performance.
- We provide detail performance analyses of the proposed system on face anti-spoofing problem in both intra-database and cross-database scenarios. Furthermore, we also discuss the placement of adaptive convolutional-feature fusion layer in a CNN network and its effects in general on face liveness detection performance.

The rest of this paper has been organized as follow: In Section 2 we review the state-of-the-art works done in face liveness detection. In Section 3, we provide the methodology for the proposed approach for face liveness detection. In Section 4, we provide experimental results and discussions. Finally, we concluded the paper with conclusion and future work in Section 5.

2. Literature review

The recent success of deep neural networks in face anti-spoofing applications has motivated us to group face anti-spoofing techniques into three broad categories, i.e. fixed features-based face anti-spoofing techniques, CNN features based face anti-spoofing techniques and combined fixed features and CNN features based face anti-spoofing techniques. Fixed features based face anti-spoofing techniques utilized liveness cues in the

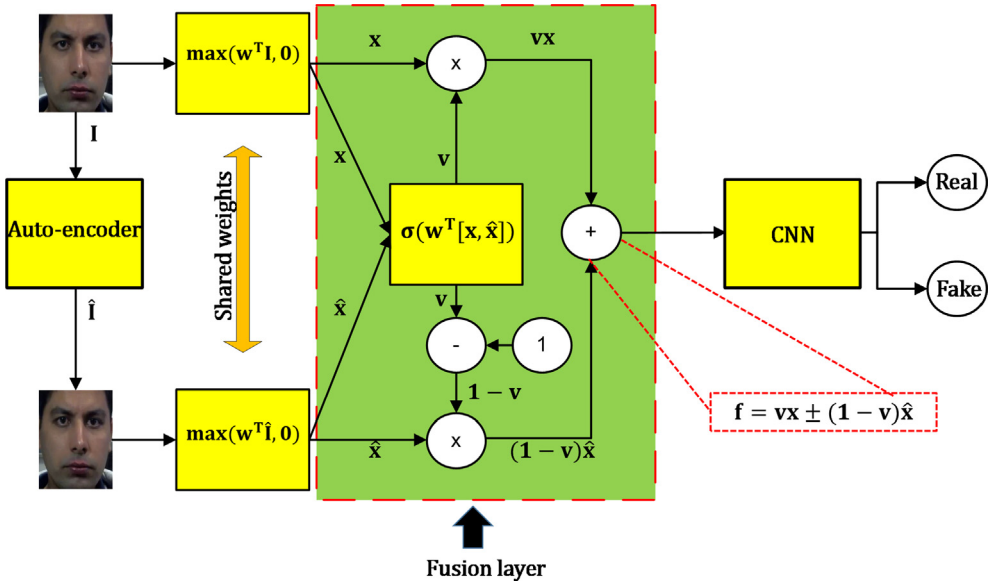

Fig. 1. Proposed approach for face liveness detection. The original face image I and corresponding DNG face image $\hat{I}$ are passed through the convolution layers with shared weights, followed by adaptive convolutional-features fusion layer that adaptively weight the incoming convolutional-feature maps and passed it to later layers in CNN.

face images to differentiate between a live face and PA. In these systems, features are calculated using texture [21,22], spectral properties [23] or motion cues in the face image [24]. However, in CNN features based face anti-spoofing techniques, features are dynamically calculated from the input data using back-propagation techniques [25]. In combined fixed features based and CNN features based face anti-spoofing techniques, a CNN network is being trained on the fixed features extracted from the face data [26]. Commonly used features in these techniques are HOG [27] and LBP [28] and their variants. In the following paragraphs, we review some of the state-of-the-art works done in recent years in CNN based face anti-spoofing systems.

In [11], the authors proposed 3-layer and 2-layer CNN networks for learning deep representation for iris, face and fingerprint spoofing detection. They proposed to optimize CNN architectures and weights that are suitable for detecting spoofing attacks in a particular modality. However, for face liveness detection in particular, the authors only reported intra-database analysis. In a similar work in [29], the authors proposed a 2-layer CNN network with a single LSTM layer for face anti-spoofing application. The work reported in [12] utilized a pre-trained AlexNet [30] for face liveness detection with varying frame lengths and data augmentation techniques. Similarly, the work in [13], utilized various architecture and layers of pre-trained deep VGG network [31] with PCA for selecting useful face signatures and an SVM classifier for classifying live face and PA. In [32], the authors utilized a pre-trained ResNet-50 [33] with an LSTM layer for PA detection.

Recently, the use of hand-crafted features, data-augmentation, transfer learning and feature fusion techniques in CNN for face liveness detection have been researched. For example, the work in [34] used the integration of various liveness cues like image quality, optical flow map and Shearlet based feature descriptors to train a CNN network. In a similar work in [35], Shearlet based feature descriptors were utilized for face liveness detection. In [36], the authors utilized a combination of multi-level LBP (MLBP) and CNN features to train a CNN for face liveness detection. In [37], the authors utilized the energy contents of the pixels in a face image to produce a spatial-temporal mapping that is used to train a CNN network for PA detection. In [38], the authors utilized combination of depth information and deep features learned from a CNN to train an SVM classifier for face liveness detection. In [39], the authors utilized LBP features with a CNN to classify a live face and a PA. In [40], the authors utilized the domain adaption techniques with CNN for face liveness detection. Similarly in [41] and [42], the authors proposed a 3D-CNN for learning spatial-temporal representation from face images for face liveness detection. In [43], the authors utilized various regions of face images along with depth information to detect the liveness of the face. Similarly in [17], the authors utilized a CNN for learning depth feature maps and an RNN network for estimating the rPPG signal to discriminate between a live face and a fake face. In [44], the authors used deep dictionary learning approach to detect 3D silicon mask based face attacks.

While these methods were all based on face liveness detection in existing color spaces; recently in [19], the authors proposed to learn a generator for transforming the existing color space into a new learned color like space. Further, they used triplet loss to supervise a three Siamese CNN architectures for face liveness detection. However, three CNN architectures might not be efficient, particularly in mobile applications.

3. Methodology

In this section, we first analyze the DNG face images generated by deep CNN auto-encoder followed by our proposed pipeline, as shown in Fig. 1, of utilizing a combination of these DNG face images and real-world face images for face liveness detection.

3.1. Evalution of real images and DNG images

Let the real-world face image is represented by I, and a deep CNN auto-encoder is represented by M. Then, the resultant DNG face image $\widehat{I}$ and the corresponding residual Res can be defined by using Eq. (1).

$$\widehat{I} = (M * I) + Res \tag{1}$$

We utilize the deep CNN auto-encoder defined in [45] for the proposed work. However compared to [45], we train the auto-encoder with only RGB face images and mean square error loss as opposed to a combination of RGB and HSV face images with $l1$ loss. This is done in order to examine the RGB color space of both real-world face images and DNG face images.

Further, the real-world images and DNG images possess disparities in each R, G and B components. This is because the real-world images are captured, decomposed and digitized from the real world, whereas DNG images are produced from the auto-encoder like structures that do not consider such constraints [18]. Therefore, the auto-encoder provide a mapping of the real-world color space into a generated color space [19]. We first examine the disparities in the color space between the original face image I and the DNG face image $\widehat{I}$ generated by the auto-encoder using similarity index histogram [18], followed by utilizing these disparities for face liveness detection.

Let suppose the histogram of the color component c of i^{th} real-world face image is represented as $h^c_{o,i}$, and the corresponding histogram of color component c in generated color space, by the auto-encoder, is represented as $h^c_{g,i}$. Then the mean histograms of all real-world face images and the corresponding auto-encoder generated DNG face images, in c color component, are represented as:

$$\mu_{h^c_o} = \frac{1}{N}\sum\nolimits_i h^c_{o,i}, \quad c = \{R, G, B\} \tag{2}$$

$$\mu_{h^c_g} = \frac{1}{N}\sum\nolimits_i h^c_{g,i}, \quad c = \{R, G, B\} \tag{3}$$

The similarity index SI^c_i for the i^{th} histogram for each color component of the original image and the corresponding auto-encoder generated image is represented by the following equations.

$$SI^c_i = \frac{D\left(h^c_{o,i}, \mu_{h^c_o}\right)}{D\left(h^c_{o,i}, \mu_{h^c_g}\right)} \tag{4}$$

$$D(h_p, h_q) = \frac{1}{2}\sum\nolimits_x \frac{\left(h_p(x) - h_q(x)\right)^2}{\left(h_p(x) + h_q(x)\right)} \tag{5}$$

The D in Eq. (5) represents the chi-square distance between the two histograms. After obtaining the similarity index SI^c_i for each c color component of each i^{th} real-world face image I and the corresponding auto-encoder generated DNG face image $\widehat{I}$, we form a similarity index histogram P^c_o and P^c_g for each color component c of the original face images and the auto-encoder generated DNG face images. Fig. 2 shows the similarity index histograms for live images and corresponding PA in each original and deep CNN auto-encoder generated R, G and B color spaces for CASIA database [46]. As shown in the Fig. 2, there is a clear disparity between the three-color components of RGB face images and the images generated by the auto-encoder. Further, we tested the deep CNN auto-encoder trained on CASIA database and tested on Replay-Attack database [47]. Fig. 3 shows the similarity index histogram for these results. It can be seen in the Fig. 3 that the disparity among each color component increases by a larger margin. From these results, we conclude that the DNG face images in cross-database testing

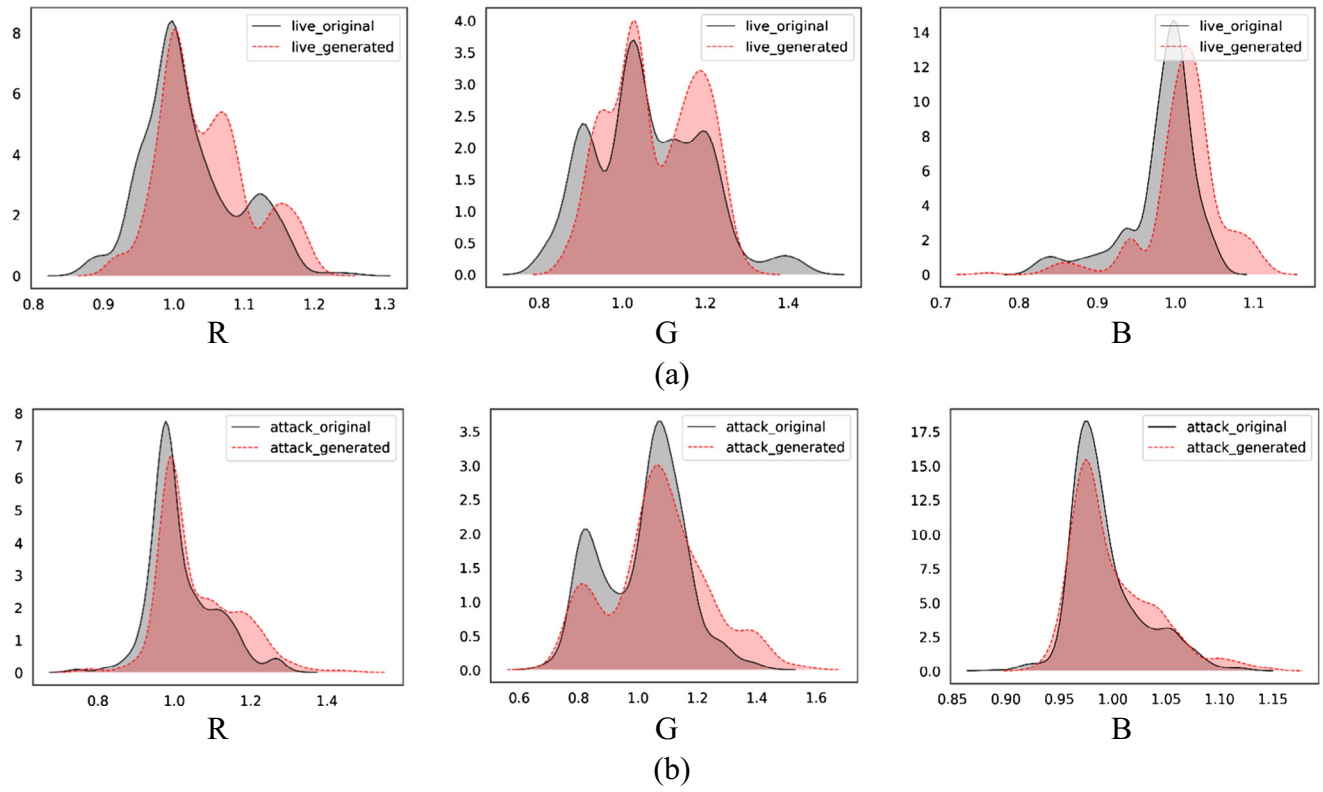

Fig. 2. (a) SI histogram for individual color component of live face image and corresponding auto-encoder generated image. (b) SI histogram for individual color component of PA face image and corresponding auto-encoder generated image.

show clear disparity (in the color components) between its original counter parts. Further, we can utilize these disparities to supervise a CNN for face liveness detection.

3.2. Adaptive convolutional-feature fusion layer

The aim of adaptive convolutional-feature fusion layer is to learn suitable weights for appropriate fusion of features learned by convolution layers from input real world face images and DNG face images. The proposed adaptive convolutional-feature fusion layer in the proposed work either blend the convolutional-feature maps or compute the disparity between the convolutional-feature maps. The blending and disparity are collectively defined by the following equations.

$$f = vx \pm (1 - v)\widehat{x} \tag{6}$$

$$v = \sigma\left(w^T[x, \widehat{x}]\right) \tag{7}$$

In Eq. (6), x represents the feature maps learned by convolution layer from real-world face images, while $\widehat{x}$ represents the feature maps learned by the same convolution layer from DNG face images. The $[.]$ in Eq. (7) represents the concatenation operation, and w^T represents the weight matrix followed by sigmoid activation σ for learning suitable weights from concatenated convolutional feature maps of both real-world face images and DNG face images. Thus, we provide an extra level of supervision to the later convolution layers (following the adaptive convolutional-features fusion layer) in the CNN to focus on the blending/disparity of the feature maps of the preceding layers rather than the feature maps itself. This is particularly useful for face liveness detection as we need discriminative clues to distinguish a live face from a fake one.

The gradient of the error function E with respect to the weight matrix w^T and input x and $\widehat{x}$ can be easily computed for both blending and disparity operations by using Eqs. (8)–(10).

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial f} \times \frac{\partial f}{\partial w} = \delta_f \times v(1 - v)\left(x \pm \widehat{x}\right) \tag{8}$$

$$\frac{\partial E}{\partial x} = \frac{\partial E}{\partial f} \times \frac{\partial f}{\partial x} = \delta_f \times v\left(1 + xw^T[1, 0](1 - v)\right) \tag{9}$$

$$\frac{\partial E}{\partial \widehat{x}} = \frac{\partial E}{\partial f} \times \frac{\partial f}{\partial \widehat{x}} = \delta_f \times 1 \pm \left(v\left(1 + \widehat{x}w^T[0, 1](1 - v)\right)\right) \tag{10}$$

$\delta_f = \frac{\partial E}{\partial f}$ in Eqs. (8)–(10) represents the error propagated back from the following layer.

3.3. CNN architecture

The proposed CNN has been shown in Table 1. The input to the proposed CNN is a real-world face image I and corresponding DNG face image $\widehat{I}$ from the auto-encoder. We utilized the auto-encoder structure proposed in [45]. As shown in Table 1, the first two convolutional layer share their weights followed by a fusion layer that either blend or calculate the disparity among the feature maps of the preceding convolutional layers. Further, to ease the flow of gradient across the CNN architecture, we further map the 3 convolutional layers, following the fusion layer, to output 2 feature maps using 1×1 convolutional layers. At the end of the CNN architecture, all the feature maps from convolutional layers 4, 5, 6 and 7 are concatenated followed by global average pooling layer that average all the feature maps and provide an output vector, which is then fed to the fully-connected layer with 2-way soft-max activation. Since global average pooling has no parameter to learn, a direct relationship can be established between the convolution layers and output of soft-max.

We further used a dropout of 0.2 after each max-pooling layer and regularization factor of 0.0005 in each convolution layer except for the adaptive convolutional-feature fusion layer. For

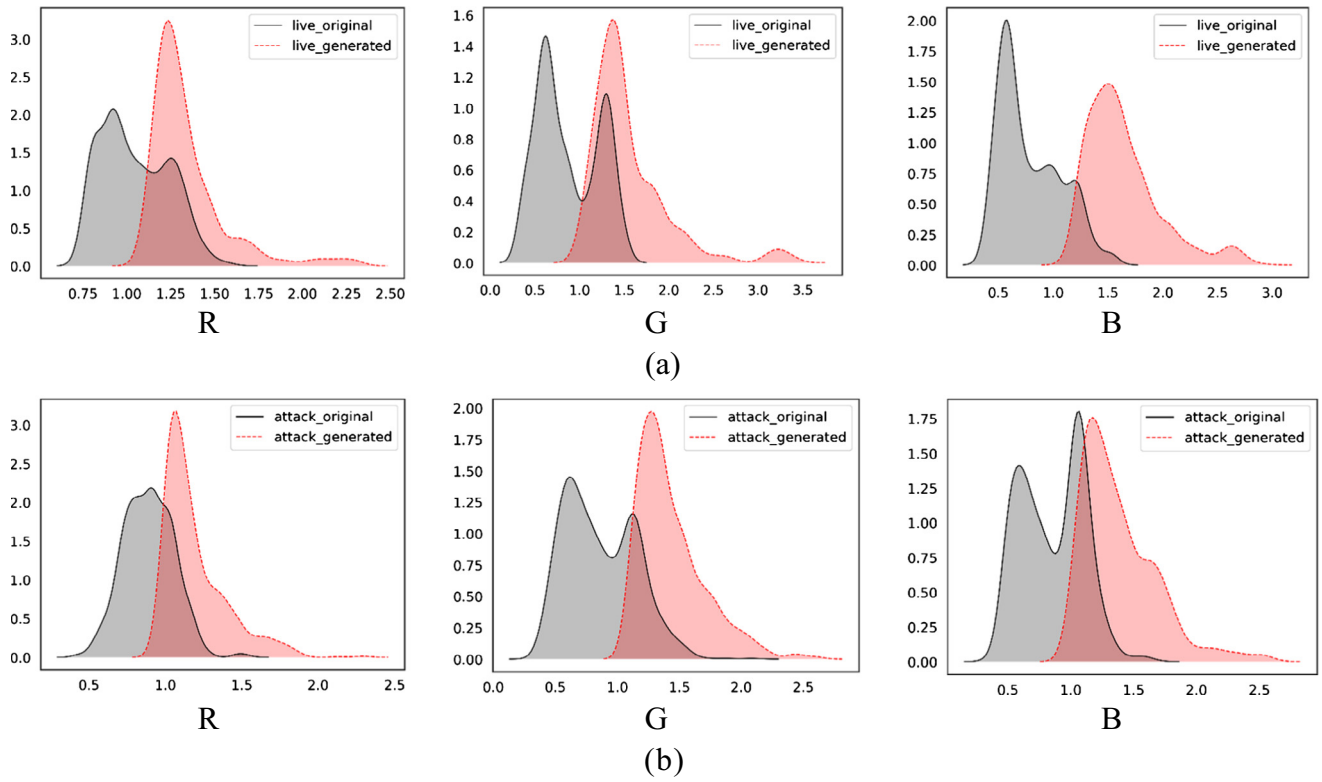

Fig. 3. (a) SI histogram for individual color component of live face image and corresponding auto-encoder generated image. (b) SI histogram for individual color component of PA face image and corresponding auto-encoder generated image.

Table 1
Configuration of proposed cnn architecture.

Layer name	Kernel size	Output channel	Input
Conv_d1	3×3	16	$I(u,v)$
Conv_d1	3×3	16	$\hat{I}(u,v)$
Fusion layer	–	32	[Conv_d1, Conv_d1]
CNN			
Conv_1	3×3	32	**Fusion Layer**
Max-pool_1	2×2	32	Conv_1
Conv_2	3×3	64	Max-pool_1
Max-pool_2	2×2	64	Conv_2
Conv_3	3×3	128	Max-pool_2
Max-pool_3	2×2	128	Conv_3
Conv_4	1×1	2	Max-pool_3
Conv_5	1×1	2	Conv_1
Conv_6	1×1	2	Conv_2
Conv_7	1×1	2	Conv_3
F1 = concatenate *[Conv_4, Conv_5, Conv_6, Conv_7]*			
Global Average Pooling	–	8	F1
Fc1	10	2	Global Average Pooling
2 way soft-max			

the case of using only RGB input data, we remove the fusion layer, and the duplicate of conv_d1.

3.4. Training

We first train the auto-encoder only on live face images for each face anti-spoofing database. Afterwards, we use the pre-trained auto-encoder to produce the DNG face images during training of the rest of the CNN. We train the proposed CNN for a total of 20 epochs. The initial learning rate was set to 0.01, which is reduced by a factor of 0.1 after 10th and 15th epoch. The batch-size was set to 32. Before feeding the training data to the proposed CNN, samples in the training data were randomly shuffled. The proposed network took approximately 40 min to train on GTX 1080 GPU. Each epoch took approximately between 120 s to 128 s depending on the size of the input image.

4. Experiments and discussions

We first perform a comprehensive analysis of the proposed CNN in intra-database scenarios followed by a cross-database analysis. Further, we provide an ablation study and discussion on analyzing the placement of fusion layer in a CNN network. For analysis of the proposed system, we utilize three state-of-the-art face anti-spoofing databases, CASIA-FASD [46], Replay-Attack [47] and OULU-NPU [48]. A brief introduction of these databases is given in the following sub-sections.

4.1. Face Anti-spoofing database

(1) ***CASIA-FASD***

This video face anti-spoofing database contain 50 subjects with three PA types, i.e. photo-attack, cut photo-attack and display medium attack. Further, each category of PA are produced in three different imaging quality, i.e. low quality, normal quality, and high qualtiy. The training set consist of 20 subjects, while the testing set consist of 30 subjects.

(2) ***Idiap Replay-Attack database***

This video face anti-spoofing database also contain 50 subjects with three PA types, i.e. mobile attack, ipad attack and printed photo attack. Further, two different illumination conditions were

provided, i.e. controlled and adverse. The training set and development set contain 60 real access and 300 attack videos while the testing set contain 80 real access and 400 attack videos.

(3) ***OULU***

This video face anti-spoofing database contain 55 subjects with 2 PA types, i.e. printed and display that are captured under two different illumination conditions and background scenes. The training set and test set contain 20 subjects, while the development set contain 15 subjects.

4.2. Evalution protocol

We evaluated the performance of the proposed system using Half Total Error Rate (HTER), Equal Error Rate (EER), Bona Fide Presentation Classification Error Rate (BPCER), Attack Presentation Classification Error Rate (APCER) and Average Classification Error Rate (ACER). For intra-database evaluation, we utilized the BPCER, APCER and their average ACER metric. For cross-database evaluation, we utilized HTER value. Since HTER is threshold dependent, the threshold computed at EER point on the development set is used to calculate HTER on the database under consideration.

4.3. Preliminary comparison: Choice of $\boldsymbol{w}^T$ and blending vs disparity

Our preliminary experiments on state-of-the-art OULU face anti-spoofing database indicated the effectiveness of using the adaptive convolutional-features fusion of features learned by convolution layers from real world and deep generated face images as compared to using RGB face images alone. From Table 2, it can be noticed that the APCER using only RGB face image data is much higher than as compared to using blending/disparity of the convolutional-features learned by RGB face image data and corresponding DNG face image data. Fig. 4 shows the sample of real-world live face image and corresponding samples of real-world PA along with the feature maps produced by convolution layer from real-world RGB face image, and the disparity feature maps produced by the proposed fusion layer from RGB and DNG input face images. It can be seen from Fig. 4(c) that the proposed fusion layer focus on the most discriminative regions in the face images. We further noticed that the disparity with a window size of 3×3 provided better performance as compared to performing blending/disparity of convolutional-features with a higher window size.

A quick comment on the choice of using OULU-NPU database is that, this database contains an equal proportion of samples of different PA and it is more challenging as compared to other state-of-the-art face anti-spoofing databases.

4.4. Intra-database face liveness detection

For the intra-database face liveness detection analysis, we utilized the BPCER, APCER and their average ACER metric. The operating threshold value for the test set was determined using the EER point on the development set. Since, CASIA dataset has no development set, therefore the BPCER, APCER and ACER values have been reported at EER point.

Table 3 shows the performance of the proposed method in intra-database face liveness detection scenarios. It can be observed from Table 3, that the proposed method achieved the best lower ACER of 5.72%, 0.29% and 0.30% on OULU, CASIA and Replay-Attack databases. Additionally, from Table 3, it can be observed that the proposed method performed better on OULU dataset as compared to Replay-Attack and CASIA face anti-spoofing datasets. The reason is because the OULU database has more variations and capturing conditions and is more challenging as compared CASIA and Replay-Attack database. Further, the intra-database analysis only provides an upper bound on the performance of face anti-spoofing system. Therefore, a cross-database analysis is needed to further verify the performance in un-constrained scenarios and to check stability of the face anti-spoofing system.

4.5. Cross-database face liveness detection

For the cross-database analysis, we trained the face anti-spoofing system on one database and test it on the other database. The operating threshold was determined by development set or the testing set of the face anti-spoofing database on which the face anti-spoofing system was trained on.

Table 4 shows the cross-database results on all three databases. From Table 4, we can see that the proposed system trained with OULU database achieved much better performance on Replay-Attack with an all-time lower HTER of 9.03% and CASIA database with an HTER of 12.81% which are consistent with the intra-database results of Table 3. On the other hand, the proposed system trained on CASIA database and Replay-Attack database whether on RGB data or using convolutional-features fusion provide inconsistent results. This suggested that there is a certain bias in these databases toward a specific attack type. Further analysis of Replay-Attack and CASIA face anti-spoofing databases revealed the imbalance between the attack samples in these databases. While CASIA database utilize more printed attack, i.e. print-photo and cut-photo,[1] the Replay-Attack database is more biased toward the screen-based attacks. On contrary, the OULU database contains an equal number of samples of both printed attacks and display attacks. Therefore, the proposed method trained on OULU database provide consistent performance in both intra-database as well as cross-database scenarios.

4.6. Ablation study on placement of fusion layer in CNN

We further investigated the placement of the fusion layer in the CNN network. Particularly, we investigated the placement of convolution-features fusion layer after convolution layer 1, 2, and 3. This placement effectively created Siamese convolutional layers as shown in the Fig. 5. Table 5 shows the performance of transferring the adaptive convolutional-features fusion layer to the higher layers in a CNN. From Table 5, it can be observed that the as we move the adaptive convolutional-feature fusion layer to the higher layers in a CNN, the overall performance of the proposed system degrades. One possible reason is that the real-world input face image features and its corresponding DNG face image features at a lower convolution layers have more generalized and discriminative clues than at higher convolution layers. Further, the placement of the adaptive convolutional-features fusion layer at the lower level in CNN is more economical in terms of total number of trainable parameters as compared to placing the adaptive convolutional-features fusion layer at the higher layer.

4.7. Comparison with state-of-the-art method

We further compared the performance of the proposed method with state-of-the-art face liveness detection methods in Table 6. Particularly, we compared the performance of the proposed method with the work in [17,40,44,49–51] in cross-database scenarios on CASIA and Replay-Attack database. As shown in Table 6, the proposed method provides better performance in terms of HTER (%) as compared to other state-of-the-art methods. On CASIA database, we achieved a lower HTER of 25.20%, while a lower HTER of

[1] The 80% area in cut-photo attack consist of printed photo.

Table 2
Performance comparison in % using different kernel windows for blending and disparity operations among convoltuional-feature maps.

Input	OULU (dev)			OULU (test)		
	BPCER	APCER	ACER	BPCER	APCER	ACER
RGB	7.37	1.87	4.62	9.44	3.36	6.40
Blend 3×3	5.76	1.42	3.50	9.40	2.79	6.10
Blend 5×5	5.02	1.25	3.13	9.89	2.26	6.08
Disparity 3×3	**4.04**	**1.01**	**2.52**	**9.54**	**1.90**	**5.72**
Disparity 5×5	5.35	1.35	3.35	10.46	1.83	6.15

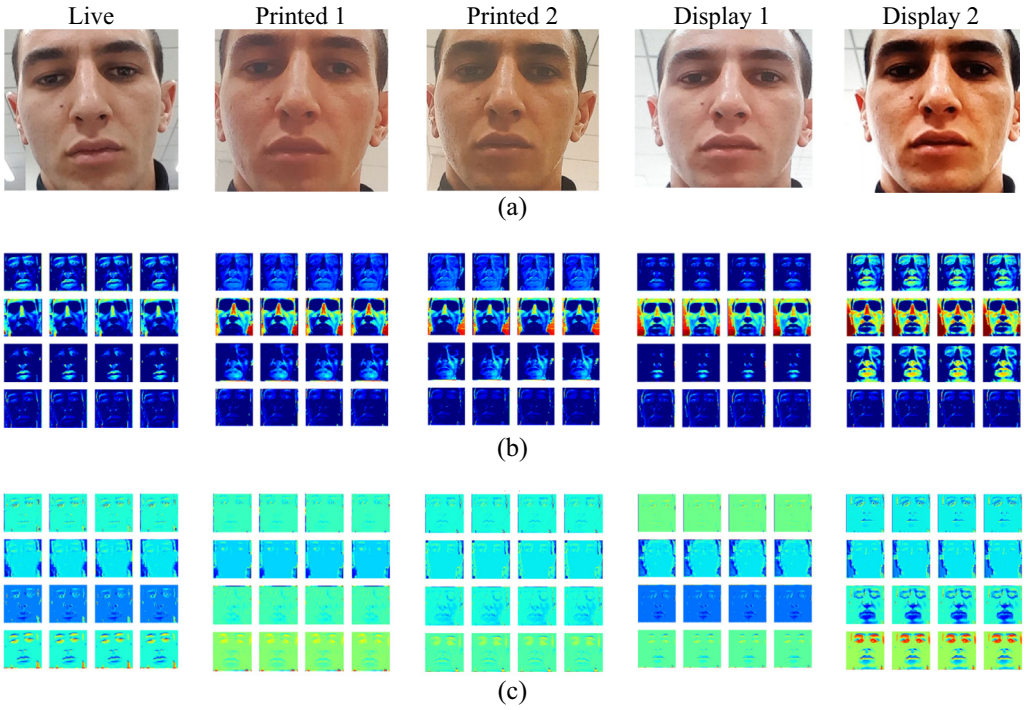

Fig. 4. (a) Sample of live face and corresponding PA in OULU database. (b) Feature maps output by first convolution layer in CNN with RGB face input. (c) Feature maps output by proposed adaptive convolutional-features fusion layer after the first convolution layer in CNN with RGB and DNG face images as inputs.

Table 3
Intra-database performance in (%) on state-of-the-art face anti-spoofing databases.

Input	OULU (development)			OULU (test)		
	BPCER	APCER	ACER	BPCER	APCER	ACER
RGB	5.85	1.45	3.66	10.78	2.88	6.83
Blend	5.76	1.42	3.50	9.40	2.79	6.10
Disparity	4.04	1.01	2.52	9.54	1.90	**5.72**
	–			CASIA (test)		
RGB	–	–	–	0.40	0.12	**0.26**
Blend	–	–	–	0.70	0.23	0.47
Disparity				0.43	0.14	0.29
	Replay-attack (development)			Replay attack (test)		
RGB	1.61	0.11	0.87	1.30	0.03	0.66
Blend	0.67	0.08	0.38	2.27	0.19	1.23
Disparity	2.33	0.37	1.35	0.20	0.40	**0.30**

Table 4
HTER in (%) for cross-database performance.

Training Set	Database	RGB	Blend	Disparity
OULU	CASIA	10.95	9.29	**9.03**
	Replay-Attack	22.70	23.31	**12.81**
CASIA	OULU	26.80	20.10	**18.73**
	Replay-Attack	22.96	11.33	**11.26**
Replay-Attack	OULU	**10.73**	30.00	21.68
	CASIA	**17.95**	28.23	25.20

11.26% was achieved with Replay-Attack database. Further, it can be noticed that the disparity between the convolutional-features of real-world face images and DNG face images using adaptive fusion layer provide better performance than blending between the convolutional-features. Additionally, the HTER on CASIA and Replay-Attack databases has been reduced to 9.03% and 12.81%, when the proposed system was trained on OULU database and tested on Replay-Attack and CASIA face anti-spoofing database.

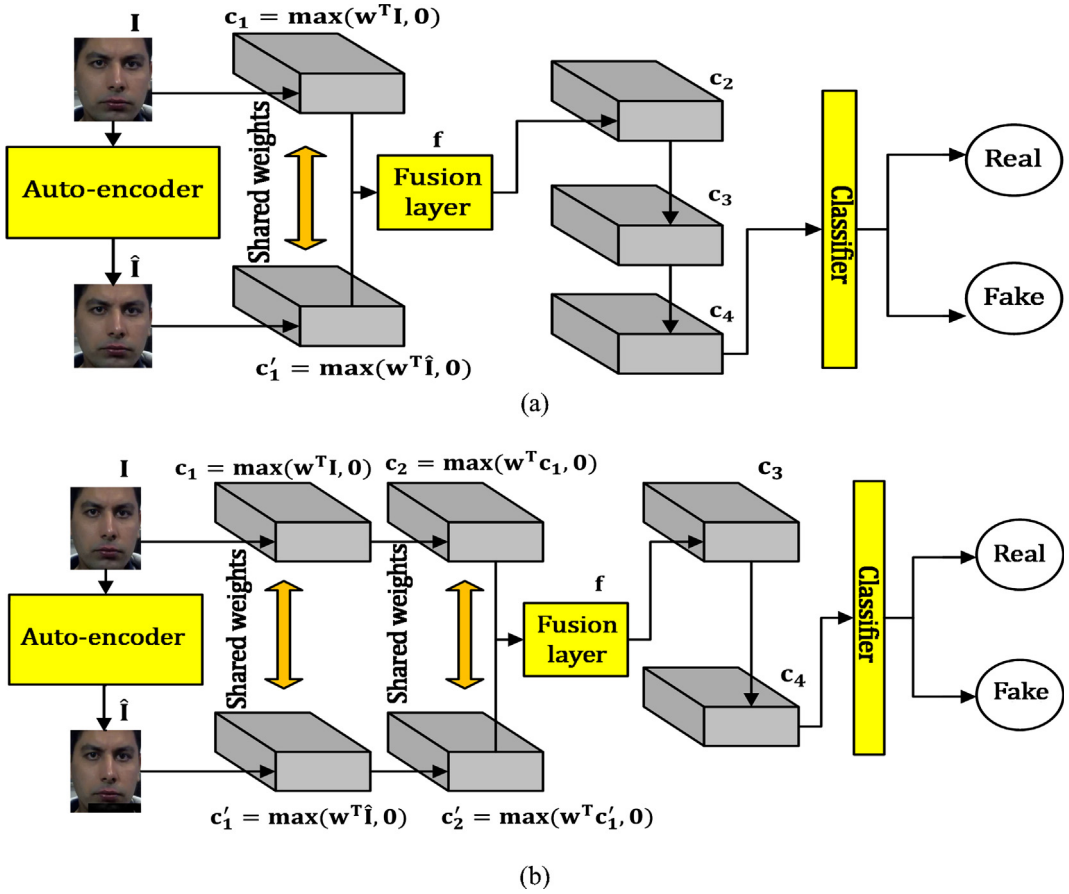

Fig. 5. The transfer of fusion layer to the deeper layers. When the fusion layer is moved to the next layer, the previous layers are duplicated with shared weights but with different inputs.

Table 5
Performance in (%) by transferring the adaptive convolutional-feature fusion layer to deeper convolutional layers.

Input	OULU (dev)			OULU (test)		
	BPCER (%)	APCER (%)	ACER (%)	BPCER (%)	APCER (%)	ACER (%)
Level 1						
Blend	5.76	1.42	3.50	9.40	2.79	**6.10**
Disparity	4.04	1.01	2.52	9.54	1.90	**5.72**
Level 2						
Blend	6.56	1.65	4.10	10.52	2.53	6.53
Disparity	6.44	1.64	4.04	9.71	2.75	6.23
Level 3						
Blend	7.68	1.94	4.78	10.89	2.90	6.89
Disparity	5.81	2.74	4.28	8.82	3.34	6.08

Table 6
Comparison of the proposed method and state-of-the-art face liveness detection methods. HTER in (%) on cross-database scenarios.

Method	*CASIA	**Replay-attack
Li et al. [40]	36.0	27.4
Manjain et al. [44]	27.4	22.8
Pinto et al. [49]	50.0	34.4
Siddiqui et al. [50]	44.6	35.4
Boulkenafet et al. [51]	37.7	30.3
Liu et al. [17]	28.4	27.6
Proposed (Blend)	28.2	11.3
Proposed (Disparity)	**25.2**	**11.3**
Proposed (Blend)	**9.3**†	**23.2**†
Proposed (Disparity)	**9.0**†	**12.8**†

* Train set: Replay Attack.
** Train set: CASIA.
† Train set: OULU.

5. Conclusion and future work

In this paper, we proposed a framework for exploiting the adaptive fusion of convolutional features fusion of real-world and DNG face images for face liveness detection. Instead of using only RGB face images, the proposed method utilized the disparities and blending among the convolutional-feature maps of real-world face images and DNG face images. Extensive experimental results with state-of-the-art face anti-spoofing databases in both intra-database and cross-database scenarios indicated that the proposed method is effective in both intra-database and cross-database scenarios. We further noticed that the placement of adaptive convolutional-feature fusion layer in the early layers of CNN boost the overall performance of the proposed method for face liveness detection task. Additionally, we found that the placement of adap-

tive convolutional-feature fusion layer at the initial level in CNN provide better performance and is more economical in terms of network parameters as compared to placing them at higher levels.

In the proposed work, we only utilize the real-world and corresponding DNG face images. However, a broader combination of real-world face images and corresponding DNG face images generated in other feature space such as LBP, HOG and Shearlet and their effectiveness for face liveness detection is a good direction for the future work.

Acknowledgment

The work in this paper is supported by City University of Hong Kong under the research project with grant number 7004430.

Declarations of Interest

None.

References

[1] Y. Duan, J. Lu, J. Feng, J. Zhou, Context-aware local binary feature learning for face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 40 (2018) 1139–1153, https://doi.org/10.1109/TPAMI.2017.2710183.
[2] J. Lu, V.E. Liong, J. Zhou, Simultaneous local binary feature learning and encoding for homogeneous and heterogeneous face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 40 (2018) 1979–1993, https://doi.org/10.1109/TPAMI.2017.2737538.
[3] Y. Duan, J. Lu, J. Feng, J. Zhou, Learning rotation-invariant local binary descriptor, IEEE Trans. Image Process. 26 (2017) 3636–3651, https://doi.org/10.1109/TIP.2017.2704661.
[4] Y. Duan, J. Lu, Z. Wang, J. Feng, J. Zhou, Learning deep binary descriptor with multi-quantization, in: Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017. 2017–Janua, 2017, pp. 4857–4866. https://doi.org/10.1109/CVPR.2017.516.
[5] H. Liu, J. Lu, J. Feng, J. Zhou, Two-stream transformer networks for video-based face alignment, IEEE Trans. Pattern Anal. Mach. Intell. 40 (2018) 2546–2554, https://doi.org/10.1109/TPAMI.2017.2734779.
[6] J. Lu, J. Hu, J. Zhou, Deep metric learning for visual understanding: an overview of recent advances, IEEE Signal Process Mag. 34 (2017) 76–84, https://doi.org/10.1109/MSP.2017.2732900.
[7] Y. Zheng, D.K. Pal, M. Savvides, Ring loss: convex feature normalization for face recognition, 2018. https://doi.org/10.1109/CVPR.2018.00534.
[8] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, W. Liu, CosFace: large margin cosine loss for deep face recognition, 2018. https://doi.org/10.1109/CVPR.2018.00552.
[9] A. Hadid, Face biometrics under spoofing attacks: Vulnerabilities, countermeasures, open issues, and research directions, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Work., 2014, pp. 113–118.
[10] Y.A.U. Rehman, L.M. Po, M. Liu, LiveNet: Improving features generalization for face liveness detection using convolution neural networks, Expert Syst. Appl. 108 (2018) 159–169, https://doi.org/10.1016/j.eswa.2018.05.004.
[11] D. Menotti, G. Chiachia, A. Pinto, W.R. Schwartz, H. Pedrini, A.X. Falcao, A. Rocha, Deep representations for iris, face, and fingerprint spoofing detection, IEEE Trans. Inf. Forensics Secur. 10 (2015) 864–879.
[12] J. Yang, Z. Lei, S.Z. Li, Learn convolutional neural network for face anti-spoofing, ArXiv Prepr, ArXiv1408.5601, 2014.
[13] L. Li, X. Feng, Z. Boulkenafet, Z. Xia, M. Li, A. Hadid, An original face anti-spoofing approach using partial convolutional neural network, in: Image Process. Theory Tools Appl. (IPTA), 2016 6th Int. Conf., 2016, pp. 1–6.
[14] J. Dong, C. Tian, Y. Xu, Face liveness detection using color gradient features, in: 2017 Int. Conf. Secur. Pattern Anal. Cybern. SPAC 2017, 2018: pp. 377–382. https://doi.org/10.1109/SPAC.2017.8304308.
[15] L.B. Zhang, F. Peng, L. Qin, M. Long, Face spoofing detection based on color texture Markov feature and support vector machine recursive feature elimination, J. Vis. Commun. Image Represent. 51 (2018) 56–69, https://doi.org/10.1016/j.jvcir.2018.01.001.
[16] Z. Boulkenafet, J. Komulainen, A. Hadid, Face anti-spoofing based on color texture analysis, 2015. https://doi.org/10.1109/ICIP.2015.7351280.
[17] Y. Liu, A. Jourabloo, X. Liu, Learning deep models for face anti-spoofing: binary or auxiliary supervision, 2018. http://arxiv.org/abs/1803.11097.
[18] H. Li, B. Li, S. Tan, J. Huang, Detection of deep network generated images using disparities in color components, 2018, pp. 1–13. doi: arXiv:1808.07276v1.
[19] L. Li, Z. Xia, A. Hadid, X. Jiang, F. Roli, X. Feng, Face presentation attack detection in learned color-liked space, n.d., pp. 1–13. doi: arXiv:1810.13170v1.
[20] A. Khodabakhsh, R. Raghavendra, K.B. Raja, P. Wasnik, Fake face detection methods: can they be generalized?, 2018. https://doi.org/10.23919/BIOSIG.2018.8553251.
[21] T. de Freitas Pereira, J. Komulainen, A. Anjos, J.M. De Martino, A. Hadid, M. Pietikäinen, S. Marcel, Face liveness detection using dynamic texture, EURASIP J. Image Video Process. 2014 (2014) 2.
[22] J. Määttä, A. Hadid, M. Pietikäinen, Face spoofing detection from single images using texture and local shape analysis, IET Biom. 1 (2012) 3–10.
[23] J. Galbally, S. Marcel, J. Fierrez, Biometric antispoofing methods: A survey in face recognition, IEEE Access 2 (2014) 1530–1552, https://doi.org/10.1109/ACCESS.2014.2381273.
[24] D. Gragnaniello, G. Poggi, C. Sansone, L. Verdoliva, An investigation of local descriptors for biometric spoofing detection, IEEE Trans. Inf. Forensics Secur. 10 (2015) 849–863.
[25] Y.A.U. Rehman, L.M. Po, M. Liu, Deep learning for face anti-spoofing: An end-to-end approach, in: Proc. Appl. 2017 Signal Process. Algorithms, Archit. Arrange., 2017, pp. 195–200. https://doi.org/10.23919/SPA.2017.8166863.
[26] A. Alotaibi, A. Mahmood, Deep face liveness detection based on nonlinear diffusion using convolution neural network, Signal, Image Video Process. 11 (2017) 713–720.
[27] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proc. - 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, CVPR 2005. I (2005) pp. 886–893. https://doi.org/10.1109/CVPR.2005.177.
[28] L. Li, X. Feng, Z. Xia, X. Jiang, A. Hadid, Face spoofing detection with local binary pattern network, J. Vis. Commun. Image Represent. 54 (2018) 182–192, https://doi.org/10.1016/j.jvcir.2018.05.009.
[29] Z. Xu, S. Li, W. Deng, Learning temporal features using LSTM-CNN architecture for face anti-spoofing, in: Pattern Recognit. (ACPR), 2015 3rd IAPR Asian Conf., 2015, pp. 141–145.
[30] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. (2012) 1097–1105.
[31] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, ArXiv Prepr, ArXiv1409.1556, 2014.
[32] X. Tu, F. Yuchun, Ultra-deep Neural Network for Face Anti-spoofing, in: Int. Conf. Neural Inf. Process., 2017, pp. 686–695. https://doi.org/10.1007/978-3-319-70096-0.
[33] K. He, Deep Residual Learning for Image Recognition, n.d.
[34] L. Feng, L.-M. Po, Y. Li, X. Xu, F. Yuan, T.C.-H. Cheung, K.-W. Cheung, Integration of image quality and motion cues for face anti-spoofing: A neural network approach, J. Vis. Commun. Image Represent. 38 (2016) 451–460.
[35] L. Feng, L.-M. Po, Y. Li, F. Yuan, Face liveness detection using shearlet-based feature descriptors, J. Electron. Imaging. 25 (2016) 43014.
[36] D.T. Nguyen, T.D. Pham, N.R. Baek, K.R. Park, Combining deep and handcrafted image features for presentation attack detection in face recognition systems using visible-light camera sensors, Sensors. 18 (2018) 699.
[37] N.N. Lakshminarayana, N. Narayan, N. Napp, S. Setlur, V. Govindaraju, A discriminative spatio-temporal mapping of face for liveness detection, in: Identity, Secur. Behav. Anal. (ISBA), 2017 IEEE Int. Conf., 2017, pp. 1–7.
[38] Y. Wang, F. Nian, T. Li, Z. Meng, K. Wang, Robust face anti-spoofing with depth information, J. Vis. Commun. Image Represent. 49 (2017) 332–337.
[39] G.B. de Souza, D.F. da Silva Santos, R.G. Pires, A.N. Marana, J.P. Papa, Deep texture features for robust face spoofing detection, IEEE Trans. Circuits Syst. II Express Briefs 64 (2017) 1397–1401, https://doi.org/10.1109/TCSII.2017.2764460.
[40] H. Li, W. Li, H. Cao, S. Wang, F. Huang, A.C. Kot, Unsupervised domain adaptation for face anti-spoofing, IEEE Trans. Inf. Forensics Secur. (2018).
[41] H. Li, P. He, S. Wang, A. Rocha, X. Jiang, A.C. Kot, Learning generalized deep feature representation for face anti-spoofing, IEEE Trans. Inf. Forensics Secur. 13 (2018) 2639–2652, https://doi.org/10.1109/TIFS.2018.2825949.
[42] J. Gan, S. Li, Y. Zhai, C. Liu, 3D convolutional neural network based on face anti-spoofing, in: 2017 2nd Int. Conf. Multimed. Image Process., 2017, pp. 1–5. https://doi.org/10.1109/ICMIP.2017.9.
[43] Y. Atoum, Y. Liu, A. Jourabloo, X. Liu, Face anti-spoofing using patch and depth-based CNNs, in: Proc. IEEE Int. Jt. Conf. Biometrics, 2017: pp. 319–328. https://doi.org/10.1109/BTAS.2017.8272713.
[44] I. Manjani, S. Tariyal, M. Vatsa, R. Singh, A. Majumdar, Detecting silicone mask based presentation attack via deep dictionary learning, IEEE Trans. Inf. Forensics Secur. (2017).
[45] A. Jourabloo, Y. Liu, X. Liu, Face de-spoofing: anti-spoofing via noise modeling, n.d.
[46] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, S.Z. Li, A face antispoofing database with diverse attacks, in: Biometrics (ICB), 2012 5th IAPR Int. Conf., 2012, pp. 26–31.
[47] I. Chingovska, A. Anjos, S. Marcel, On the effectiveness of local binary patterns in face anti-spoofing, in: Biometrics Spec. Interes. Gr. (BIOSIG), 2012 BIOSIG-Proceedings Int. Conf., 2012, pp. 1–7.
[48] Z. Boulkenafet, J. Komulainen, L. Li, X. Feng, A. Hadid, OULU-NPU: A Mobile Face Presentation Attack Database with Real-World Variations, in: Proc. - 12th IEEE Int. Conf. Autom. Face Gesture Recognition, FG 2017 - 1st Int. Work. Adapt. Shot Learn. Gesture Underst. Prod. ASL4GUP 2017, Biometrics Wild, Bwild 2017, Heteroge, 2017, pp. 612–618. https://doi.org/10.1109/FG.2017.77.
[49] A. Pinto, H. Pedrini, W.R. Schwartz, A. Rocha, Face spoofing detection through visual codebooks of spectral temporal cubes, IEEE Trans. Image Process. 24 (2015) 4726–4740.
[50] T.A. Siddiqui, S. Bharadwaj, T.I. Dhamecha, A. Agarwal, M. Vatsa, R. Singh, N. Ratha, Face anti-spoofing with multifeature videolet aggregation, in: Pattern Recognit. (ICPR), 2016 23rd Int. Conf., 2016, pp. 1035–1040.
[51] Z. Boulkenafet, J. Komulainen, A. Hadid, Face spoofing detection using colour texture analysis, IEEE Trans. Inf. Forensics Secur. 11 (2016) 1818–1830.