Enhancing Deep Discriminative Feature Maps via Perturbation for Face Presentation Attack Detection

Yasar Abbas Ur Rehman[a,c,*], Lai-Man Po[a], Jukka Komulainen[b]

[a] *Department of Electrical Engineering, City University of Hong Kong, Kowloon, Hong Kong, SAR China*

[b] *Visidon Ltd., Oulu, Finland*

[c] *TCL Corporate Research (Hong Kong) Co. Limited, Hong Kong Science Park, Hong Kong, SAR China*

Abstract

Face presentation attack detection (PAD) in unconstrained conditions is one of the key issues in face biometric-based authentication and security applications. In this paper, we propose a perturbation layer — a learnable pre-processing layer for low-level deep features — to enhance the discriminative ability of deep features in face PAD. The perturbation layer takes the deep features of a candidate layer in Convolutional Neural Network (CNN), the corresponding hand-crafted features of an input image, and produces adaptive convolutional weights for the deep features of the candidate layer. These adaptive convolutional weights determine the importance of the pixels in the deep features of the candidate layer for face PAD. The proposed perturbation layer adds very little overhead to the total trainable parameters in the model.We evaluated the proposed perturbation layer with Local Binary Patterns (LBP), with and without color information, on three publicly available face PAD databases, i.e., CASIA, Idiap Replay-Attack, and OULU-NPU databases. Our experimental results show that the introduction of the proposed perturbation layer in the CNN improved the face PAD performance, in both intra-database and cross-database scenarios. Our results also highlight the attention created by the proposed perturbation layer in the deep features and its effectiveness for face PAD in general.

Keywords: Spoofing, Presentation attack detection, CNN, Attention, Face-biometrics

---

1. Introduction

The recent surge in the deployment of face recognition based access control in electronic devices has raised serious concerns regarding potential security breaches in these electronic

---

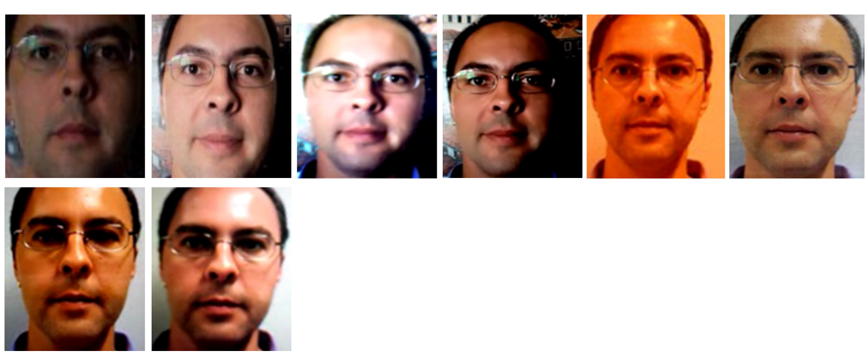
* Corresponding author

Email addresses: yaurehman2-c@my.cityu.edu.hk (Yasar Abbas Ur Rehman), eelmpo@my.cityu.edu.hk (Lai-Man Po), yty@iki.fi (Jukka Komulainen)

devices. While the accuracy of face recognition based approaches [1], [2], [3], in classifying different individuals based on their facial attributes, have been remarkably improved; face recognition based systems adopted for access control are highly vulnerable to face-spoofing attacks, also known as face Presentation Attacks (PA) [4]. Without face Presentation Attack Detection (PAD) (also known as face liveness detection and face antispoofing) support, an intruder can easily outwit a face recognition based access control system by merely using a printed photograph of a genuine user's face [5]. To make matters worse, the availability of, and easy access to, social media platforms like Facebook, WeChat, Instagram, as well as development in high-end digital cameras and printers, have made it easy to obtain realistic face portraits of any individual. As a result, it gets easier to gain illegal access to the individual's tangible or intangible assets via face-spoofing [6]. Therefore, the inclusion of face PAD support, in conjunction with face recognition systems, for access control in electronic devices, has become indispensable.

Face PA can be broadly classified into three main types: printed photo-based face PA, video display based face PA, and 3D mask-based face PA [7]. While the former two face PA can be easily produced using off the shelf printers and portable display devices; the production of 3D mask-based face PA is expensive and require more expertise in producing the realistic 3D facial face of a genuine user. Fig.1 shows examples of real faces and corresponding face PA from CASIA

(a)



(b)

Fig. 1. (a) Examples of real faces and fake faces in CASIA database. The first 2 face images in the first row and 3rd face image in the second row are real face images, while the rest are fake face images. (b) Examples of real faces and fake faces in Reply-Attack database. The first and the 5th face images in the first row are real, while the rest are fake face images.

[8] and Replay-Attack [9] face the anti-spoofing database. As shown in Fig.1, given no further information, it is difficult to classify whether these face images are the representations of a genuine face or face PA. However, after extracting some facial features from each face image, we can distinguish between a real face and a particular face PA. These facial features vary from hand-crafted features to learnable or deep features. Common examples of hand-crafted features utilized for face anti-spoofing are Histogram of Oriented Gradients (HOG) [10], Local Binary Patterns (LBP) [11], [12], Shearlet [13], [14], optical flow (OF) [15], Discrete Cosine Transform (DCT) [16], and Redundant Wavelet transforms [17]. On the other hand, deep features utilized for face anti-spoofing have been obtained from deep neural networks, such as deep convolution neural networks (CNN) [18].

In the recent decade, a multitude of state-of-the-art face PAD methods and algorithms have been proposed to detect face PA. While most of these methods have either utilized hand-crafted features-based or deep features-based classifiers, a parallel line of research has been drawn by combining hand-crafted features and deep features for solving the face PAD problem. In these approaches, the hand-crafted features are utilized as auxiliary supervision of a classification model, or the hand-crafted features and deep features are fused at the last layer of the CNN for face PAD. In general, the supervision of classification models, like CNN, by utilizing the combination of hand-crafted features and deep features have demonstrated remarkable improvement in the performance of face PAD [19]. To this end, the hand-crafted features can be utilized as auxiliary supervision, or the hand-crafted features and deep features can be utilized in combination to generate new features or attention maps for supervising a classification model, like CNN, for face PAD [12].

Our work is different from the early feature fusion [20] and late feature fusion frameworks proposed for face PAD [19]. In general, the early feature fusion frameworks first concatenate the input image and its different feature representations (HOG, LBP, OF) as input to the CNN or deep models for face PAD [15,20]. These frameworks assume that the different representations of the input image contain enough discriminative information that can be learned by the deep models for face PAD. However, the different representations of the input image are often designed to cover a specific range of scenarios in the face PAD. Additionally, the early layer in the CNN model performs weighting on the pixels of the input image and its different representations, respectively, before feeding its output to the next layer. As a result, the early layer in the CNN or deep models may give lower priority to the pixels in the original image and high priority to the other discriminative representations. While these methods may perform remarkably well in intra-database scenarios (because the hand-crafted features are designed for such scenarios), their performance drops significantly in adverse scenarios. The late feature fusion frameworks concatenate different features obtained from the CNN or deep models before performing final face PAD classification. While these methods have performed remarkably well in different face PAD scenarios, they may increase the computational cost of the final PAD classifiers as different models need to be trained to obtained various deep representations before the final classification stage.

Different from the previous works, we propose to induce the information of the hand-crafted features into the deep features of a candidate layer in CNN using a perturbation layer. This work is inspired by the Perturbative Neural Networks (PNN) [21]. However, our perturbation layer is different from the perturbation layer proposed in [21]. Our perturbation layer takes the deep features of the candidate layer in CNN, the corresponding LBP features (or other hand-crafted features in general) of the input image, and produces adaptive convolutional weights based on the joint information of the hand-crafted and the deep features of the candidate layer. These adaptive convolutional weights are then multiplied with the deep features of the candidate layer to amplify or attenuate the intensity of each pixel in the deep features of the candidate layer. The modified deep features are then served as an input to the remaining CNN layers for face PAD. In our preliminary work [22], we only evaluated the effectiveness of HOG features in perturbing convolutional features for face PAD. In this paper, we investigate the subject thoroughly by introducing LBP features, extracted from both grayscale and color images, for perturbing deep feature maps. We further provide comprehensive experimental analysis of the proposed method for face PAD, analyzing the effectiveness of the proposed method in various challenging face anti-spoofing scenarios. We further show that the introduction of hand-crafted features in CNN further strengthens the discriminative regions and introduce attention in convolutional-feature maps.

Extensive experimental results on three public face anti-spoofing databases, CASIA [8], Idiap Replay-Attack [9], and OULU-NPU [23], show excellent generalization ability of the proposed method in face PAD. In general, we show that the proposed method can be as effective as the state-of-the-art frame-level face PAD approaches, or it can further improve the performance of the state-of-the-art frame-level face PAD methods. One more advantage of our proposed method is the utilization of a small number of parameters (approximately 0.1M), which makes our proposed method lightweight and suitable for the resource-constrained application. Further, the proposed method combines hand-crafted features and deep features into one architecture using the perturbation layer that accounts for only 50-75 parameters approximately. This further reduces the requirement of Siamese or Triplet architectures for fusing the information from hand-crafted features and deep features. The main contributions of this paper are as follows:

1) We propose a novel approach to combine hand-crafted features and deep feature maps in a CNN in an end-to-end learning fashion. We find that the proposed method enhances the discriminative ability of deep features in face PAD.

2) We utilize LBP features, with and without color information, with deep features to learn adaptive convolutional weights, also called perturbative weights, for perturbing candidate layer features for face PAD.

3) We provide comprehensive experimental analysis on three publicly available face anti-spoofing databases and discuss the pros and cons of the proposed method in various face PAD scenarios. The proposed method performs comparatively better in the category of models that utilizes combined hand-crafted and deep features in face PAD.

4) Compared with other deep learning-based methods, the proposed CNN based method embeds the RGB face data and corresponding hand-crafted features into one architecture while being lightweight and computationally efficient.

The rest of this paper is organized as follows: In Section 2, we review state-of-the-art methods proposed in the face PAD domain using fixed feature-based classifiers and CNN classifiers. The details of the proposed approach are presented in Section 3. The experimental setup and the description of the face anti-spoofing databases are presented in Section 4, and evaluation and discussion of the proposed approach are presented in Section 5. An additional discussion section summarizing the experimental results and the pros and cons of the proposed method is provided in Section 6. Finally, the paper concludes with a conclusion and future work in Section 7.

2. Literature Review

In recent years, a wide range of state-of-the-art techniques has been developed for face PAD. Taking the context of this paper into consideration, we make a division of these state-of-the-art face PAD techniques into three categories, i.e., hand-crafted features-based face PAD approaches, deep CNN based face PAD approaches and combined hand-crafted and deep CNN based face PAD approaches.

2.1. Hand-crafted features-based face PAD

Face PAD utilizing hand-crafted features have been extensively studied in the literature. The core idea behind the utilization of hand-crafted features for face PAD is to explore the liveness cues in the face image by either utilizing the texture [24], [25], motion, or spectral reflectance properties of the face image. If the liveness cues existed in the face image, the face image was considered as live otherwise PA. Commonly used hand-crafted features for detecting liveness cues in the face image were HOG [10], LBP [26], [27] LPQ [28] , Shearlet [15], and their variants. Additionally, the transformation of RGB color space to other color domains, such as HSV, YCbCr, and features describing image quality, such as specular reflections and blurriness was also explored in literature for face PAD [29], [30], [31], [32] [33]. Other cues-based face PAD methods exploited motion cues, such as eye blinking, lips movement, Moiré patterns, and optical flow for detecting whether the given face sequence was live or face PA [34], [35], [36] [37], [16].

2.2. CNN-based face PAD

Because the features learned by CNN are more dynamic in contrast to hand-crafted features, recent works utilized CNN classifiers for face PAD [38]. For example, in [39], a 3 layer CNN network was utilized for fingerprint, iris, and face PAD that achieved remarkable accuracy in intra-database face PAD scenarios on Replay-Attack and 3D-MAD database. Rather than training a CNN from scratch, the authors in [40], selected the discriminative feature maps from different layers of well-known VGG-Net [41] and then utilized PCA for dimensionality reduction and an

SVM classifier for classifying whether the input face image was real or face PA. In [42], the authors utilized AlexNet [18] with different data-augmentation techniques for face PAD. LSTM (Long Short Term Memory)-CNN architecture was proposed by [43], to learn the spatial-temporal structure from face sequences fed to the CNN for face PAD. Similarly, in [44], the authors utilized 3D-CNN architecture with spatial and gamma correction based augmentations and Maximum Mean Discrepancy (MMD) loss for face PAD. A dictionary-based approach was utilized by [45] to face PAD. In [46], the face-depth and face-patches like eyes, nose, mouth, and eyebrows were utilized along with CNN for face PAD. Two CNN were utilized: one for classification of face-patch and second for obtaining the face depth map from the input face image. The output face depth map was then fed to an SVM classifier, and a score-level fusion strategy was used to improve the face PAD rate. The authors, in [47], proposed to detect face PA by supervising a CNN based architecture by exploiting various noise patterns in the input face imaage. In [48], the authors proposed to map the existing RGB, YCbCr, and HSV into a learned color like space model for face PAD.

2.3. Combined Hand-crafted and CNN-based face PAD

In [49], a Spatio-temporal representation was first obtained from the input RGB face images by computing the energy representation in each color channel. These face images in Spatio-temporal representation were then fed to CNN to detect the liveness of face. Similarly, in [15], Shearlet based feature descriptors, face optical-flow map, and scene optical-flow map were utilized for training a deep auto-encoder for face PAD. Recently, a combination of hand-crafted features such as LBP and the features produced by CNN were utilized by [50] for face PAD task. Extensive experiments were performed using feature-level and score-level fusion to analyze the performance of the proposed method for face PAD. In [19], the authors proposed to train a CNN using a depth map and rPPG signals supervision for face PAD. In [20], the authors proposed LBP-net for classification of the live face and face PA. Their method utilized CNN with LPB feature maps computed from a grayscale image. Similarly, [51], [12] proposed to extract the LBP features from convolutional feature maps for face PAD.

3. Methodology

The proposed pipeline for face PAD is shown in Fig.2. As depicted in Fig.2, the convolutional feature maps of the candidate convolutional layer, "$Conv_1$", with input image I_{RGB} and the corresponding I_{LBP} features, are first concatenated using the concatenation layer "[.]", and subsequently fed to the custom-designed perturbation layer "$Conv_p$." The perturbation layer first computes the adaptive perturbative weights "C_2^p", followed by perturbing "$Conv_1$" layer's feature maps "C_1" using "C_2^p" to generate P^h. Afterward, the output P^h of the perturbation layer is passed through the rest of CNN for face PAD.

3.1. Design of Perturbation Layer

Suppose that the input face image to CNN is represented by I_{RGB}, and the corresponding LBP features are represented as I_{LBP}. Additionally, let the feature maps of candidate convolution layer is represented as C_1. In this work, we select the first convolutional layer in the proposed CNN. To learn adaptive perturbative weights, we first concatenate each j^{th}convolutional feature map $c_{1,j}$ of candidate convolutional layer C_1 with $I_{LBP,}$ as shown in Fig.3. The combination of LBP images and

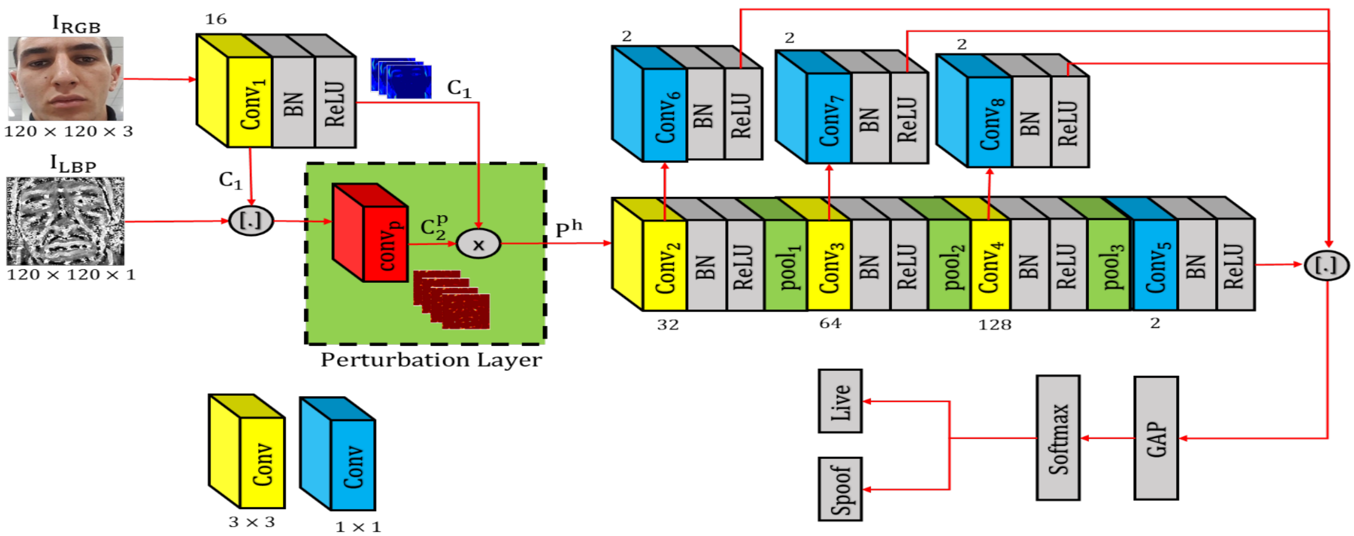


Fig. 2. Generalized pipeline of the proposed method for face liveness detection

convolutional feature maps resulted in a hybrid tensor F^h :

$$F^h = \{f_1^h, f_2^h, \dots, f_j^h, \dots, f_n^h\} \tag{1}$$

$$f_j^h = [c_{1,j}, I_{LBP}] \qquad j = 1,2, \dots, n \tag{2}$$

Each j^{th} element of the hybrid tensor F^h is then convolved with the shared weight matrix $W^T$and passed through the sigmoid activation σ. We represent the output of this convolution layer as C_2^p:

$$C_2^p = \{c_{2,1}^p, c_{2,2}^p, \dots, c_{2,j}^p, \dots, c_{2,n}^p\} \tag{4}$$

$$c_{2,j}^p = \sigma\left(W^T * f_j^h\right) \tag{5}$$

$$c_{2,j}^p(x,y) = \sigma\left(\sum_n^N \sum_m^M w_0^T(x-n, y-m)c_{1,j}(x,y) + \ w_0^T(x-n, y-m)\ I_{LBP}(x,y)\right) \tag{6}$$

As depicted in Fig.3, each of the j^{th} element of convolutional feature maps C_2^p is represented by a weighted combination of corresponding elements of j^{th} convolutional feature map $c_{1,j}$ and I_{LBP}. The convolutional weights $W^T = \{w_0^T, w_0^T\}$ are learnable weights that are optimized while training the proposed CNN using backpropagation. After obtaining the feature maps C_2^p, we calculated the Hadamard product between the feature maps of C_1 and C_2^p to obtain the perturbed feature maps P^h:

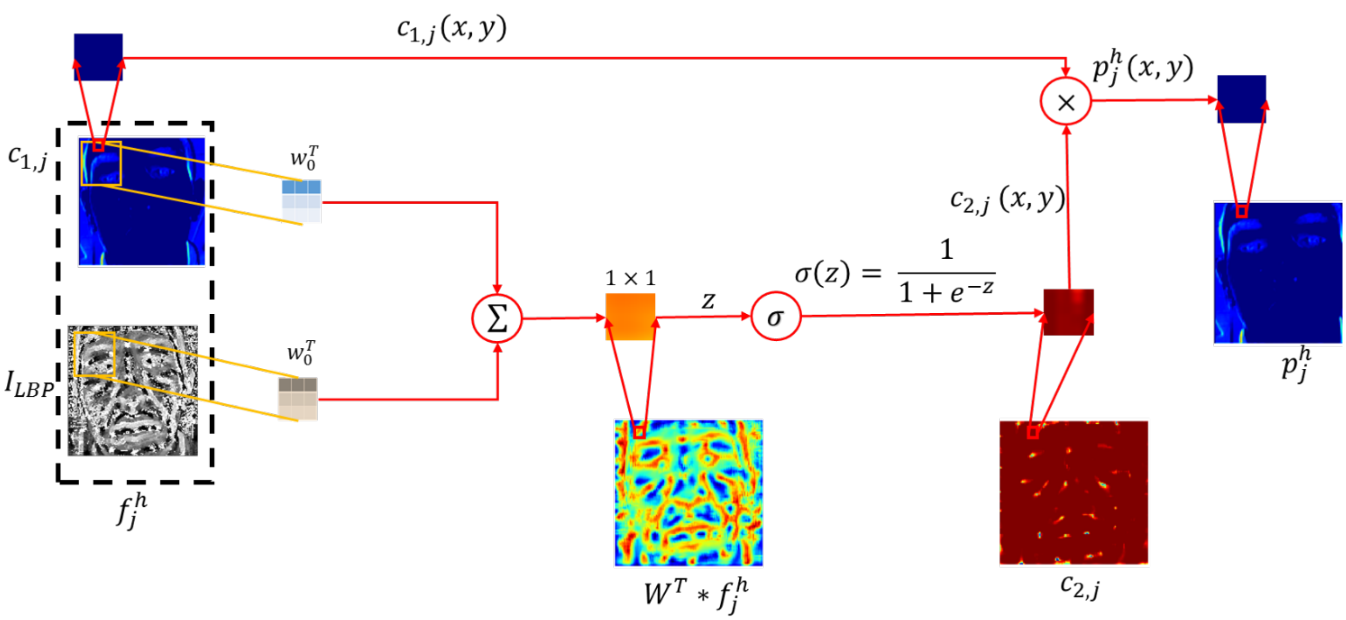


Fig. 3. Pipeline for generating the j^{th} adaptive perturbative weights $c_{2,j}$, and corresponding j^{th} perturbed convolutional-feature maps p_j^h

$$P^h = \{p_1^h, p_2^h, \ldots, p_j^h, \ldots, p_n^h\} \tag{7}$$

$$p_j^h = c_{1,j} \times c_{2,j}^p \tag{8}$$

Rewriting equation (7) reveals some salient information about the perturbation layer:

$$p_j^h = c_{1,j} \times \sigma\left(\textstyle\sum_n^N \sum_m^M w_0^T(x-n, y-m)c_{1,j}(x,y) + w_0^T(x-n, y-m)\, I_{LBP}(x,y)\right) \tag{9}$$

It can be concluded from equation (9) that each element in the j^{th} convolutional feature map $c_{1,j}$ is perturbed by the information extracted from the local region, say 5×5 patch, of the j^{th} feature map $c_{i,j}$, and the LBP features I_{LBP}. This provides certain advantages. For example, it enables the integration of RGB face image features and corresponding LBP image features into one architecture, which would otherwise require Siamese or triplets CNN for each input image. Further, each pixel in the j^{th} convolutional feature map $c_{1,j}$ of the candidate convolution layer is scaled by taking into the weighted neighborhood information around that pixel in $c_{1,j}$ and I_{LBP}. This determines which discriminative pixels in the j^{th} convolutional feature map $c_{1,j}$ of candidate convolutional layer should retain its original value and which discriminative pixels should be down-weighted, according to the information obtained from adaptive perturbative weights $c_{2,j}^p$. This is analogous to introducing attention in the convolution feature maps.

3.2. CNN architecture and training

The architecture of the proposed method has been shown in Fig. 2. The proposed CNN consists of 8 convolutional layers except for the perturbation layer. All convolutional layers, except the perturbation layer, are followed by Batch Normalization (BN) and Rectified Linear Unit (ReLU). Additionally, the convolutional layer 6, 7, and 8 takes the outputs of convolutional layer 2, 3, and 4, respectively. Subsequently, the outputs of convolutional layer 2, 3, and 4 are concatenated with convolutional layer 5, followed by Global Average Pooling (GAP) that averages all the features maps of the convolutional layer 5, 6, 7, and 8 and produces an 8 element feature vector. This 8 element feature vector is then fed to a fully-connected layer with a two-way softmax classifier for face PAD. Since GAP has no parameter to learn, a direct relationship can be established between the convolutional layers and output of softmax. We further used a dropout of 0.2 after each max-pooling layer and *l2* regularization factor of 0.0005 in each convolution layer except for the perturbation layer. The total number of trainable parameters in the proposed CNN is 99,000, of which the perturbation layer has only 50-75 trainable parameters.

The proposed system was trained for a total of 30 epochs. The initial learning rate was set to 0.01, which was reduced by a factor of 0.5 after every 2 epochs. The batch-size was set to 32. Before feeding the training data to the proposed CNN, samples in the training data were randomly shuffled and normalized. The proposed network took approximately 3 to 4 hours to train on GTX 1080 GPU. Each epoch took approximately 11 minutes to 15 minutes, depending upon the data size and input image resolution. Also, most of the time was taken by online batch-wise computation of LBP from the input batch of RGB images.

3.3. Visualizing the class activation maps of C_1 and perturbation layer

The perturbed feature maps P^h represent scaled versions of the candidate layer feature maps C_1. To further visualize the information induced by perturbing the convolutional features maps C_1 with perturbative weights C_2^p; we visualize the discriminative regions selected by the candidate convolution layer and the perturbation layer for classifying an input face image being live or face PA. For this purpose, we took a sample of real face images and samples of face PA from OULU-NPU database, and generated the class activation maps (CAM) of the candidate convolution layer and perturbation layer. This serves two purposes. First, it helps to determine the discriminative facial regions selected by the candidate layer in the input face image. Second, it helps to determine scaling performed by the perturbation layer of the discriminative facial regions selected by the candidate layer. To generate the CAM from the candidate layer and the perturbation layer, we followed the procedure defined in [52]. Fig.4 shows the sample of real face image and corresponding samples of face PA, the corresponding CAM obtained from the candidate layer, and the corresponding CAM obtained from the perturbation layer. Comparing the CAM of the live face with the face PA, we can see that, for a particular class (real or face PA), the perturbation layer has further enhanced (in case of live face) or down-weighted (in case of face PA) the discriminative regions selected by candidate convolution layer, and utilized by the proposed CNN for classifying an input face image as being live or fake. For example, in the case of the live face image, as shown in Fig.4 (first column), the perturbation layer focuses more on the eyes, nose, and mouth region of the input face image, whereas for the case of face PA, it down-weights those

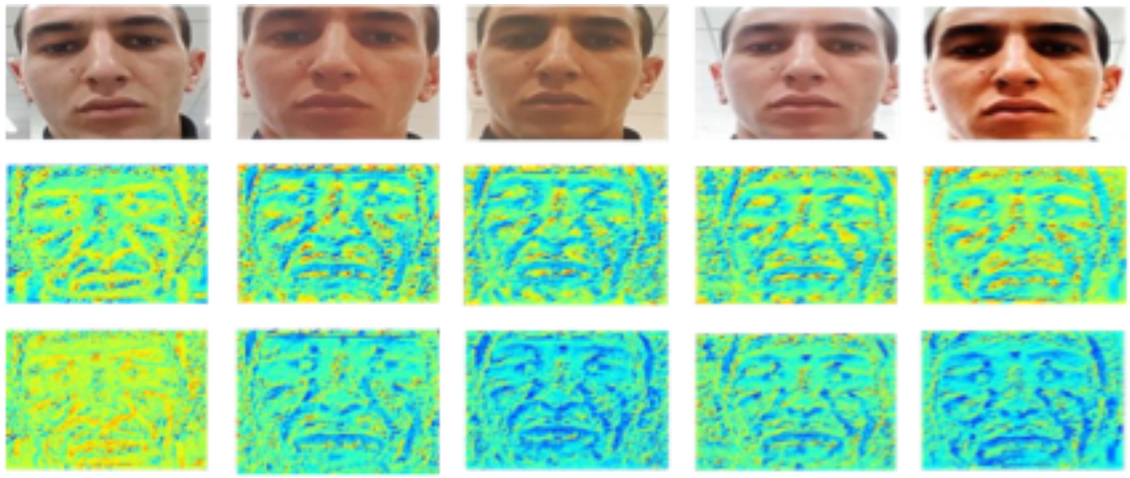


Fig.4. Samples of face images from OULU-NPU face anti-spoofing database and corresponding CAM of candidate convolution layer (2nd Row) and perturbation layer (3rd Row). The first column represents the live face while the rest of the columns represent face PA.

regions. Thus, we can clearly see that the perturbation layer generate attention in the feature maps, by retaining or down-weighting the elements of feature maps, for supervising the rest of the CNN layers for classifying an input face image being live or face PA.

4. Experimental Setup:

For our experimental analysis, we considered three public face PAD databases: CASIA-FASD [8], Replay-Attack [9], and OULU-NPU [23]. A brief introduction of these databases and the metrics used for evaluation of the proposed method have been given in the following sub-sections.

4.1. CASIA-FASD

This video face PAD database contains 50 subjects with 3 face PA types, i.e., display medium attack, cut photo-attack, and printed photo-attack. The training set consists of 20 subjects, while the testing set consist of 30 subjects. Additionally, each category of face PA and real access was produced in 3 different imaging qualities, i.e., low quality, normal quality, and high quality.

4.2. Idiap Replay-Attack

This video face PAD database also contains 50 subjects with 3 face PA types, i.e., printed photo attack, iPad display attack, and mobile display attack. Additionally, two different illumination conditions were provided, i.e., controlled and adverse. The training set and development set contain 60 real access and 300 face PA videos, and the test set contains 80 real access and 400 attack videos.

4.3. OULU-NPU

This video face PAD database contains 55 subjects with 2 PA types, i.e., printed and display that were captured under 3 different illumination conditions and background scenes. The overall training set and overall test set contain 20 subjects, while the overall development set contains 15 subjects. In addition, there are 4 protocols to test the generality of face PAD method under varying scenarios, like illumination, face PA types, various camera types, and their combination [23].

4.4. Evaluation Metrics

We evaluated the performance of the proposed method using Equal Error Rate (EER), Half Total Error Rate (HTER), and the Attack Presentation Classification Error Rate (APCER), Bona Fide Presentation Classification Error Rate (BPCER), and Average Classification Error Rate (ACER) [4]. In general, APCER = FAR, BPCER=FRR, and ACER=HTER. The only difference between these metrics is that: in APCER, BPCER, and ACER, the worst-case scenario for each face PA is considered. For intra-database evaluation, we employed the APCER, BPCER, 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, or training set, such as in the case of CASIA database, is used to calculate HTER on the database under consideration. We utilized the evaluation protocol defined in [19] for OULU-NPU, Idiap Replay-Attack, and CASIA database. Since we also provided cross-database performance among the databases utilized in this work, we provide the intra-database and cross-database results on the overall development and test databases.

5. Experimental Results and Discussion

5.1. Effect of kernel size in the perturbation layer

Table 1 Face liveness detection performance in % of the proposed method by using different kernel sizes in generating perturbed feature maps

Kernel size	**OULU-NPU (development)**			**OULU-NPU (test)**		
	BPCER	**APCER**	**ACER**	**BPCER**	**APCER**	**ACER**
1 × 1	6.32	1.61	3.96	7.04	3.65	5.35
3 × 3	4.70	1.13	2.92	6.31	2.95	4.63
5 × 5	5.06	1.27	3.16	**5.81**	**1.97**	**3.89**
7 × 7	5.56	1.37	3.46	7.97	3.02	5.50

The kernel size is an essential hyper-parameter in the design of the proposed perturbation layer. Before performing any evaluation using the proposed CNN configuration, we first analyze the utilization of different kernel window sizes in the perturbation layer and its effect on face PAD in general. For this purpose, we utilized the overall training, development, and test set of OULU-NPU database. We evaluated the kernel sizes of 1 × 1, 3 × 3, 5 × 5, and 7 × 7 in the perturbation layer and reported the results in Table 1. It can be seen in Table 1 that using a 5 × 5 kernel size in the perturbation layer results in the lowest ACER of 3.89% on the overall OULU-NPU test set. Further increasing the kernel size from 5 × 5 deteriorates the performance of the proposed system in face PAD task. Therefore, for the rest of our analyses, we present the performance of the proposed method using 5 × 5 kernel size in the perturbation layer.

5.2. Effectiveness of perturbation layer for face PAD

To show the effectiveness of utilizing the perturbation layer in CNN for face PAD, we trained the CNN with and without perturbation layer on the first 2 protocols of OULU-NPU database. The first protocol of the OULU-NPU database evaluates the performance of face PAD method under unseen environmental conditions, while the second protocol evaluates the performance of face PAD methods against face PA created with different PA mediums, like printers and display. These two protocols are sufficient to select the best configuration for the rest of the two challenging protocols in the OULU-NPU database.

We performed analysis by incorporating a different combination of color spaces and their corresponding LBP features in the perturbation layer. For example, the $I_{RGB} + I_{LBP\_G}$ denotes the utilization of the LBP image (in the perturbation layer) extracted from the grayscale version of the I_{RGB}. On the other hand, the $I_{RGB} + I_{LBP\_C}$ denotes the utilization of the LBP image extracted from each color channel of the I_{RGB}. Further, we also performed analysis by perturbing the deep feature maps with the LBP image extracted from each channel of HSV and YCRCB color spaces, while feeding the face image in these colorspaces as an input to the proposed CNN.

Table 2 shows the face PAD performance of the proposed CNN on Protocol 1 and Protocol 2 of the OULU-NPU database [23]. As it can be seen in Table 2, without using the perturbation layer, we obtained an ACER of 18.96% and 16.81% on Protocol 1 and Protocol 2, respectively. The use of perturbation layer with $I_{RGB} + I_{LBP\_C}$ significantly reduced the ACER to 7.81% and 13.13% on Protocol 1 and Protocol 2. In comparison, the use of perturbation layer with $I_{RGB} + I_{LBP\_G}$ obtained the ACER of 22.92% and 14.17% on Protocol 1 and Protocol 2. From these results, it can be

Table 2 Face liveness detection performance in % of the proposed CNN with and without perturbation layer on Protocol 1 and Protocol 2 of OULU- NPU database

	Input	Dev	Test				
			Print	Display	Overall		
		EER	APCER	APCER	APCER	BPCER	ACER
	Protocol 1						
w/o perturbation layer	I_{RGB}	1.13	1.88	4.17	4.17	33.75	18.96
w/ perturbation layer	$I_{RGB} + I_{LBP\_G}$	1.46	1.67	1.04	1.67	42.92	22.92
	$I_{RGB} + I_{LBP\_C}$	1.62	2.71	2.71	2.71	12.92	**7.81**
	$I_{HSV} + I_{LBP\_C}$	1.08	10.42	9.79	10.42	11.67	11.04
	$I_{YCRCB} + I_{LBP\_C}$	1.49	1.46	0.21	1.46	20.83	11.15
	Protocol 2						
w/o perturbation layer	I_{RGB}	1.55	9.17	26.25	26.25	7.36	16.81
w/ perturbation layer	$I_{RGB} + I_{LBP\_G}$	1.19	18.19	23.61	23.61	4.72	14.17
	$I_{RGB} + I_{LBP\_C}$	1.72	22.5	23.75	23.75	2.50	**13.13**
	$I_{HSV} + I_{LBP\_C}$	1.44	20.83	32.92	32.92	2.08	17.50
	$I_{YCRCB} + I_{LBP\_C}$	1.84	28.61	17.78	28.61	6.53	17.57

inferred that on average, the utilization of $I_{RGB} + I_{LBP\_C}$ provides better performance compared to $I_{RGB} + I_{LBP\_G}$ and other color spaces.

5.3. Performance on all 4 protocols of OULU-NPU database

We further compared the performance of the proposed method with the baseline method proposed in the IJCB competition [53], [54] in Table 3. We found significant improvement of the proposed method over the baseline method. It should be noted that our proposed method performs the face PAD at frame level as opposed to other state-of-the-art methods that also incorporate video sequence-based methods. Particularly in protocol 4, the proposed method obtained an ACER of 20.42 ± 11.00 % compared to the baseline that obtained an ACER of 26.3±16.9%. It can be further observed in Table 3, that $I_{RGB} + I_{LBP\_G}$ did not perform well compared $I_{RGB} + I_{LBP\_C}$, particularly in challenging scenarios, such as protocol 4. This suggested that the utilization of each color channel information is necessary for face PAD under varying scenarios.

Table 3 Face liveness detection performance in % of the proposed method with the baseline on OULU- NPU database

Protocol	**Dev**	**Test**				
		Print	Display	Overall		
	EER	APCER	APCER	APCER	BPCER	ACER
	Baseline					
1	4.4	**1.3**	5.0	5.0	20.8	12.9
2	4.1	**22.5**	**15.6**	22.5	6.7	14.6
3	3.9±0.7	11.8±10.8	9.3±4.3	14.2±9.2	8.6±5.9	11.4±4.6
4	4.7±0.6	22.5±38.3	19.2±17.4	29.2±37.5	23.3±13.3	26.3±16.9
	Proposed ($I_{RGB} + I_{LBP\_G}$)					
1	1.46	1.67	1.04	1.67	42.92	22.92
2	1.19	18.19	23.61	23.61	4.72	14.17
3	0.86±0.27	6.25±8.59	10.56±10.72	15.00±9.53	20.28±19.36	17.64±8.69
4	1.61±0.45	5.00±12.25	12.5±16.96	15.83±18.28	57.5±40.71	36.67±14.80
	Proposed ($I_{RGB} + I_{LBP\_C}$)					
1	**1.62**	2.71	**2.71**	**2.71**	**12.92**	**7.81**
2	**1.72**	22.5	23.75	23.75	**2.50**	**13.13**
3	**1.8±0.13**	**9.17±6.71**	**9.17±8.05**	**13.47±6.57**	**8.33±9.19**	**10.90±2.13**
4	**2.02±0.27**	**17.5±14.75**	**13.33±11.69**	**23.33±13.66**	**17.5±15.73**	**20.42±11.00**

5.4. Intra-database performance on CASIA and Idiap Replay-Attack database

We also evaluated the performance of the proposed method on the two commonly used face PAD benchmarks, namely CASIA, and Idiap Replay-Attack database. Table 4 shows the results of the proposed method for each database. As can be seen in Table 4, the introduction of the perturbation layer in CNN significantly improved the results in intra-database test scenarios. In the case of CASIA database, the introduction of the perturbation layer improved the performance in the test set by reducing the ACER from 1.73% to 0.23%. In the case of Replay-Attack database, the proposed method improved the performance in the test set by reducing the ACER from 2.07% to 1.06%. Comparing results, of without using the perturbation layer and using perturbation layer, in Table 4, we further observe the $I_{RGB} + I_{LBP\_G}$ provide better performance on both CASIA and Idiap Replay-Attack database compared to $I_{RGB} + I_{LBP\_C}$. Nevertheless, the proposed architecture

Table 4 Face liveness detection performance in % of the proposed method with and without perturbation layer on CASIA and Idiap Replay-Attack database in intra-database scenarios

	BPCER	APCER	ACER	BPCER	APCER	ACER
	CASIA					
	Development			Test		
$\mathbf{I_{RGB}}$	-	-	-	2.13	1.33	1.73
$\mathbf{I_{RGB}+I_{LBP\_G}}$	-	-	-	**0.33**	**0.12**	**0.23**
$\mathbf{I_{RGB}+I_{LBP\_C}}$	-	-	-	11.67	3.87	7.77
	Replay Attack					
	Development			Test		
$\mathbf{I_{RGB}}$	4.72	0.95	2.84	1.26	2.89	2.07
$\mathbf{I_{RGB}+I_{LBP\_G}}$	3.02	0.60	1.81	**1.73**	**0.38**	**1.06**
$\mathbf{I_{RGB}+I_{LBP\_C}}$	7.32	1.45	4.38	6.94	3.34	5.14

still achieved relatively better performance compared to the result obtained without using perturbation layer. It must be noted that in the CASIA and Idiap Replay-Attack database, the training data and testing data differ only by the number of subjects, while the environmental conditions such as illumination, and the various face PA types are nearly same. Therefore the results obtained in Table 4 only represents an upper bound on the performance of the proposed PAD method in intra-database scenarios.

5.5. Performance in cross-database face PAD scenarios

We evaluated the generalization of the proposed method across the three face PAD databases, namely OULU-NPU, CASIA, and Idiap Replay-Attack database, in cross-database setup. Table 5 shows the results of the proposed method for each cross-database test scenarios. It can be seen in Table 5 that the proposed method, using $\mathbf{I_{RGB}+I_{LBP\_C}}$ in the perturbation layer and trained with the CASIA database has obtained better performance on Idiap Replay-Attack and OULU-NPU database by lowering the HTER to 8.95% and 30.31%. Similarly, the proposed method trained on Idiap Replay-Attack database, using $\mathbf{I_{RGB}+I_{LBP\_C}}$ as an input, obtained the ACER of 22.82% and 9.24% on CASIA and OULU-NPU database. In the case of OULU-NPU database, the $I_{RGB} + I_{LBP\_G}$ and $I_{RGB} + I_{LBP\_C}$ obtain comparative results. Nevertheless, we still found that the utilization of the perturbation layer with LBP features improved the performance of face liveness detection in cross-database scenarios as well.

5.6. Comparison with state-of-the-art face PAD databases

We further compared the performance of the proposed method with state-of-the-art face PAD approaches both in intra-database and cross-database scenarios on CASIA and Idiap Replay-Attack databases. Table 6 shows the intra-database performance of the proposed method compared with state-of-the-art face PAD approaches, based on EER and HTER metric. For intra-database performance comparison, we compared the performance of our proposed method with the

Table 5 Face liveness detection performance in % of the proposed method on state-of-the-art face anti-spoofing databases in cross-database scenarios

Train set	**Test set**	**Input**	**HTER**	
CASIA	Idiap Replay Attack	I_{RGB}	19.22	w/o perturbation
		$I_{RGB} + I_{LBP\_G}$	20.38	w/ perturbation
		$\mathbf{I_{RGB} + I_{LBP\_C}}$	**8.95**	
	OULU-NPU	I_{RGB}	32.58	w/o perturbation
		$I_{RGB} + I_{LBP\_G}$	31.70	w/ perturbation
		$\mathbf{I_{RGB} + I_{LBP\_C}}$	**30.31**	
Idiap Replay Attack	CASIA	I_{RGB}	23.75	w/o perturbation
		$I_{RGB} + I_{LBP\_G}$	24.37	w/ perturbation
		$\mathbf{I_{RGB} + I_{LBP\_C}}$	**22.82**	
	OULU-NPU	I_{RGB}	14.82	w/o perturbation
		$I_{RGB} + I_{LBP\_G}$	12.60	w/ perturbation
		$\mathbf{I_{RGB} + I_{LBP\_C}}$	**9.24**	
OULU-NPU	CASIA	I_{RGB}	37.5	w/o perturbation
		$\mathbf{I_{RGB} + I_{LBP\_G}}$	**9.76**	w/ perturbation
		$I_{RGB} + I_{LBP\_C}$	10.45	
	Idiap Replay Attack	I_{RGB}	41.67	w/o perturbation
		$\mathbf{I_{RGB} + I_{LBP\_G}}$	**9.19**	w/ perturbation
		$I_{RGB} + I_{LBP\_C}$	10.06	

following state-of-the-art face PAD approaches: LBP-TOP [25], LBP+LDA[27], multi-cue integration (MCI) [15], Image Distortion Analysis (IDA) [30], Spoof-Net [39], DP-CNN [40], 3D-CNN+MMD [44], DDGL [45], Patch-CNN [46], Learned color space[48], LBP-Net [12], Ultra-deep CNN [55], SPMT + SSD [56], color texture [57]. As can be seen in Table 6, the proposed method significantly lowered down the HTER on Idiap Replay-Attack database, and the EER on the CASIA database, in intra-database scenarios. From Table VI, it can be noticed that the approach proposed in [56] performs much better than our proposed method on the Idiap-Replay Attack database. However, the complexity of their method is comparatively high and required several stages of feature extraction without utilizing end-to-end learning when using CNN with hand-crafted features. In contrast, our proposed method utilized only a single CNN with a perturbation layer with end-to-end learning, which has obtained comparative results with the work in [56], while being computationally efficient.

In cross-database scenarios, we perform comparison with the following state-of-the-art approaches: LBP-TOP[25] , color texture [57], visual-codebook [58], Videolet aggregation [59], domain adaption [60], De-spoof [47], Auxiliary [19], deep dynamic textures [61], DDGL [45], GFA-CNN [62], STASN [63]. We summarized these results in Table 7. It can be seen in Table 7 that the proposed method obtained a significantly lower HTER of 24.37% on CASIA and 20.38% on Idiap Replay-Attack database using $I_{RGB}+I_{LBP\_G}$. However, using color LBP in the perturbation layer achieved much better results on Idiap Replay-Attack database by achieving significantly lower HTER of 8.95%, whereas on CASIA database it achieved comparative performance by obtaining

Table 6 Performance comparison in % HTER of the proposed method with state-of-the-art face anti-spoofing methods in intra-database tests

Type	Method	CASIA	Replay Attack
Hand-crafted	LBP-TOP [25]	10.0	7.60
	LBP+LDA [27]	21.01	19.62
	IDA [30]	12.97	7.41
	Color texture [57]	2.1	2.8
CNN	Ultra-deep CNN [55]	1.00	1.03
	Patch-CNN [46] †	4.44	0.72
	DP-CNN [40]	4.55	5.78
	Spoof-Net [39]	-	0.75
	DDGL [45]	1.3	0
	3D CNN+ MMD [44] †	1.2	1.40
	Learned color space [48] †	-	0.70
	Proposed (I_{RGB})	0.77	1.31
Hand-crafted +CNN	MCI [15] †	5.83	0
	LBP-Net [12]	2.5	1.3
	SPMT + SSD [56]	0.04	0.06
	Proposed ($I_{RGB} + I_{LBP\_G}$)	**0.09**	**0.30**
	Proposed ($I_{RGB} + I_{LBP\_C}$)	2.91	1.97
† Utilized video-sequence as opposed to frame-level			

Table 7 Performance comparison in % HTER of the proposed method with state-of-the-art face anti-spoofing methods in cross-database tests

Type	Method	CASIA *	Idiap Replay Attack**
Hand-crafted	LBP-TOP [25]	60.6	49.7
	Color texture [57]	37.70	30.30
	Visual codebook [58]	50.0	34.38
	Videolet aggregation[59] †	44.6	35.4
CNN	FaceDs [47]	41.1	28.5
	Deep dynamic texture [61] †	35.0	22.2
	GFA-CNN [62]	34.3	21.4
	STASN [63] †	25.0	18.7
	DDGL [45]	27.4	22.8
Hand-crafted + CNN	Auxiliary [19] †	28.4	27.6
	Domain adaption [60] †	36.0	27.4
	Proposed ($I_{RGB}+I_{LBP\_G}$)	**24.37**	**20.38**
	Proposed($I_{RGB}+I_{LBP\_C}$)	**22.82**	**8.95**
*** Train set: Replay Attack** **** Train set: CASIA** **† Utilized video-sequence as opposed to frame-level**			

HTER of 22.82%. Nevertheless, the proposed method significantly improved the state of the art in cross-database scenarios.

The current state-of-the-art hand-crafted and CNN based face PAD techniques have shown great success in on various protocols of the OULU-NPU database. We further compared the performance of the proposed method with these state-of-the-art methods in IJCB [54] competition such as Baseline, GRADIANT, CPqD and NWPU, and recent CNN based techniques such as STASN [63], GFA-CNN[62], FaceDs [47], DeepPixBis [64] and Auxilary [19]. Compared to these methods, the proposed method is light-weight and performs liveness detection from a single image (frame-level). It should be further noted that the results reported for the proposed method are obtained from a single CNN architecture, i.e. without any ensemble of deep models. Table 8 summarizes the performance of the proposed method on protocol 1, 2, 3, and 4 of the OULU-NPU database against state-of-the-art hand-crafted, CNN, and hand-crafted + CNN based techniques. It can be observed that the proposed method performed better than the frame-based Baseline approach and obtained comparative performance to CPqD based technique on protocol 1. Further, the state-of-the-art deep CNN based methods utilized very deeper architectures compared to the proposed method. However, the proposed method still obtained comparative performance to these state-of-the-art techniques at the cost of reducing the computational complexity. Compared to the state-of-the-art method, our proposed method provides comparative results in the category of hand-crafted + CNN based approaches. Notably, on protocol 4, our proposed algorithm performed second best after the method Auxiliary [19], while being computationally efficient.

6. Discussion

It is worth highlighting that we considered a relatively shallow CNN network consisting of only 10 layers, including the perturbation layer, with approximately 0.1M parameters in this work, unlike many other recent state-of-the-art approaches utilizing deep networks[53], [47], [19], [64] . Our aim was to investigate the importance of the perturbation layer, with deep features and LBP features (with and without color information) as input, and its effectiveness in CNN-based face PAD in general. We believe that the performance of the proposed approach could be further improved by learning more high-level features, e.g. by incorporating the proposed deep feature in the early layer of the state-of-the-art (face PAD) frameworks.

As stated in the introduction section of this work, the early feature fusion frameworks feeding the input image along with its various representations may fail to perform reliably in diverse scenarios. As an example, this is evident from the results obtained with HKBU method [54] in Table 8 (on the 4th protocol of OULU-NPU), where the authors fused hand-crafted IDA and multi-scale LBP features with deep features to learn a classifier for face PAD. The late feature fusion performed remarkably well both on frame-level and video sequence-level across all the protocols of the OULU-NPU database. However, it should be noted that results obtained from the frame-level face PAD approaches shown in Table 8 have incorporated very deep models (either single or multiple) to obtain state-of-the-art performance across all the protocols of OULU-NPU database. For example, CPqD [54] method provided the average results obtained from Inception-v3 model and the baseline (color LBP) method scores, respectively. Similarly, the MixedFASNet [54] stacked various deep CNN models, each of over 30 layers, to obtain state-of-the-art performance. On the other hand, the video sequence-level based face PAD approaches utilized one or more CNN models for face PAD detection. The state-of-the-art result has been obtained by late fusion of the features obtained from various deep models, with each model output estimating certain features of the input video sequence. For example, the Auxilary [19] models utilized the deep models to estimate the depth and rPPG signals from the input sequences to achieve state-of-the-art performance on OULU-NPU database.

Compared to the aforementioned approaches, our proposed approach is unique in the sense that we utilized only single CNN architecture with the original image and its LBP features as input. The LBP features only serve as an input to the perturbation layer to learn the adaptive convolutional weights. Further, we did not construct an ensemble of models, although in practice it may improve the robustness of the proposed face PAD method in general. Our work may serve as a starting point for further exploration of adaptively engineering the deep features of the CNN models for face PAD. Although the proposed method is simple, yet we show that it can achieve significantly improved performance gain in face PAD. One drawback of the proposed method is the uncertainty in the selection of appropriate hand-crafted features to be fed to the perturbation layer. Although we showed that the feeding LBP features (extracted from the color images) to the perturbation layer could improve the performance in general face PAD, this is only a single possible solution in the pool of existing hand-crafted features.

7. Conclusion and Future Work

In this paper, we proposed a novel approach for face PAD by inducing the information of hand-crafted features such as LBP into deep CNN models. We aimed to learn adaptive perturbative weights from a weighted combination of deep convolutional feature maps, and LBP features with and without color information, obtained from the input face image, to perturb the convolutional features maps of the candidate convolutional layer for face PAD. Our extensive experimental results showed that the proposed method strengthens the discriminative regions by introducing attention in the convolutional feature maps of the candidate convolutional layer for face PAD. Furthermore, the proposed approach obtained comparative results with the state-of-the-art in both intra-database and cross-database scenarios. In the future, we will study other hand-crafted features and their influence on various CNN configurations for face PAD. Further, we will explore novel approaches for perturbing deep features with hand-crafted features.

TABLE 8 Comparison of the proposed method with state-of-the-art on protocol 2, 3, and 4 of OULU NPU database

Protocol 1				
Input	Method	APCER	BPCER	ACER
Hand-crafted	Baseline [54]	5.0	20.8	12.9
	GRADIANT [54] †	1.3	12.5	6.9
CNN	DeepPixBiS [64]	0.83	0	0.42
	STASN [63] †	1.2	0.8	1.0
	FaceDs[47]	1.2	1.7	1.5
	GFA-CNN [62] †	2.5	8.9	5.7
Hand-crafted + CNN	Auxilary [19] †	1.6	1.6	1.6
	CPqD [54]	2.9	10.8	6.9
	Proposed	**2.71**	**12.92**	**7.81**
	HKBU [54]	13.9	5.6	9.7
	NWPU [54]	8.8	21.7	15.2
Protocol 2				
Input	Method	APCER	BPCER	ACER
Hand-crafted	GRADIANT [54] †	3.1	1.9	2.5
	Baseline [54]	22.5	6.7	14.6
CNN	STASN [63] †	1.4	0.8	1.1
	GFA-CNN [62]	2.5	1.3	1.9
	FaceDs [47]	4.2	4.4	4.3
	MixedFASNet [54]	9.7	2.5	6.1
	DeepPixBiS [64]	11.39	0.56	5.97
Hand-crafted + CNN	Auxilary [19] †	2.7	2.7	2.7
	CPqD [54]	14.7	3.6	9.2
	HKBU [54]	13.9	5.6	9.7
	Proposed	**23.75**	**2.5**	**13.13**
	NWPU [54]	12.5	26.7	19.6
Protocol 3				
Hand-crafted	GRADIANT [54] †	2.6±3.9	5.0±5.3	3.8±2.4
	Baseline [54]	14.2±9.2	8.6±5.9	11.4±4.6
CNN	STASN [63] †	1.4±1.4	3.6±4.6	2.5±2.2
	FaceDs [47]	4.0±1.8	3.8±1.2	3.6±1.6
	GFA-CNN [62]	4.3	7.1	5.7
	MixedFASNet [54]	5.3±6.7	7.8±5.5	6.5±4.6
	DeepPixBiS [64]	11.67±19.6	10.56±14.1	11.11±9.4
Hand-crafted + CNN	Auxilary [19] †	2.7±1.3	3.1±1.7	2.9±1.5
	CPqD [54]	6.8±5.6	8.1±6.4	7.4±3.3
	Proposed	**13.47±6.6**	**8.33±9.2**	**10.90±2.1**
	HKBU	12.8±11.0	11.4±9.0	12.1±6.5
	NWPU [54]	3.2±2.6	33.9±10.3	18.5±4.4
Protocol 4				
Hand-crafted	GRADIANT [54] †	5.0±4.5	15.0±7.1	10.0 ±5.0
	Baseline [54]	29.2±37.5	23.3±13.3	26.3±16.9
CNN	MassyHNU	35.8±35.3	8.3±4.1	22.1±17.6
	STASN [63] †	0.9±1.8	4.2±5.3	2.6±2.8
	FaceDs [47]	1.2±6.3	6.1±5.1	5.6±5.7
	GFA-CNN [62]	7.4	10.4	8.9
	DeepPixBiS [64]	36.67±29.7	13.33±16.8	25.0±12.7
CNN + Hand-crafted	Auxilary [19] †	9.3±5.6	10.4±6.0	9.5±6.0
	Proposed	**23.3±13.7**	**17.5±15.7**	**20.4±11.0**
	CPqD [54]	32.5±37.5	11.7±12.1	22.1±20.8
	HKBU [54]	33.3±37.9	27.5±20.4	30.4±20.8
	NWPU [54]	30.8±7.4	44.2±23.3	37.5±9.4
† Utilized video-sequence as opposed to frame-level				

8. Acknowledgment

Internal SRG Funding from the City University of Hong Kong substantially supported the work described in this paper under Project No. 7004430.

References

[1] 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.

[2] 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.

[3] Y. Wen, K. Zhang, Z. Li, Y. Qiao, A Discriminative Feature Learning Approach for Deep Face Recognition, in: Comput. Vis. – ECCV 2016. ECCV 2016. Lect. Notes Comput. Sci., 2016: pp. 499–515. https://doi.org/10.1007/978-3-319-46478-7_31.

[4] ISO/IEC JTC 1/SC 37 Biometrics. information technology biometric presentation attack detection part 1: Framework. international organization for standardization, 2016., n.d. https://www.iso.org/obp/ui/iso.

[5] W. Kim, S. Suh, J.-J. Han, Face liveness detection from a single image via diffusion speed model, IEEE Trans. Image Process. 24 (2015) 2456–2465.

[6] Y. Xu, T. Price, J. Frahm, F. Monrose, Virtual U: Defeating Face Liveness Detection by Building Virtual Models From Your Public Photos, in: Proc. 25th USENIX Secur. Symp., 2016: pp. 497–512. https://doi.org/https://www.usenix.org/system/files/conference/usenixsecurity16/sec16_paper_xu.pdf.

[7] S. Marcel, M.S. Nixon, J. Fierrez, N. Evans, eds., Handbook of Biometric Anti-Spoofing, 2nd ed., 2019. https://doi.org/https://doi.org/10.1007/978-3-319-92627-8.

[8] 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.

[9] 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.

[10] J. Määttä, A. Hadid, M. Pietikäinen, Face spoofing detection from single images using texture and local shape analysis, IET Biometrics. 1 (2012) 3–10.

[11] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 971–987. https://doi.org/10.1109/TPAMI.2002.1017623.

[12] 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.

[13] Gitta Kutyniok, Demetrio Labate, Introduction to Shearlets, in: Appl. Numer. Harmon. Anal., Springer, 2012: pp. 1–38. https://doi.org/10.1007/978-0-8176-8316-0.

[14] L. Feng, L.-M. Po, Y. Li, F. Yuan, Face liveness detection using shearlet-based feature descriptors, J. Electron. Imaging. 25 (2016) 43014.

[15] 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.

[16] Y. Tian, S. Xiang, Detection of Video-Based Face Spoofing Using LBP and Multiscale DCT, in: Digit. Forensics Watermarking, 2016: pp. 16–28. https://doi.org/10.1007/978-3-319-64185-0.

[17] A. Agarwal, R. Singh, M. Vatsa, Face anti-spoofing using Haralick features, 2016 IEEE 8th Int. Conf. Biometrics Theory, Appl. Syst. BTAS 2016. (2016). https://doi.org/10.1109/BTAS.2016.7791171.

[18] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Adv. Neural Inf. Process. Syst., 2012: pp. 1097–1105.

[19] Y. Liu, A. Jourabloo, X. Liu, Learning Deep Models for Face Anti-Spoofing: Binary or Auxiliary Supervision, in: 2018 IEEE/CVF Conf. Comput. Vis. Pattern Recognit., IEEE, 2018: pp. 389–398. https://doi.org/10.1109/CVPR.2018.00048.

[20] 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.

[21] F. Juefei-Xu, V.N. Boddeti, M. Savvides, Perturbative Neural Networks, in: IEEE Conf. Comput. Vis. Pattern Recognit., 2018: pp. 3310–3318.

[22] Y.A.U. Rehman, L. Po, M. Liu, Z. Zou, W. Ou, Perturbing Convolutional Feature Maps with Histogram of Oriented Gradients for Face Liveness Detection, in: F. Martínez Álvarez, A. Troncoso Lora, J.A. Sáez Muñoz, H. Quintián, E. Corchado (Eds.), Int. Jt. Conf. 12th Int. Conf. Comput. Intell. Secur. Inf. Syst. (CISIS 2019) 10th Int. Conf. Eur. Transnatl. Educ. (ICEUTE 2019). CISIS 2019, ICEUTE 2019. Adva, Springer International Publishing, Cham, 2020: pp. 3–13. https://doi.org/10.1007/978-3-030-20005-3_1.

[23] 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), IEEE, 2017: pp. 612–618. https://doi.org/10.1109/FG.2017.77.

[24] J. Määttä, A. Hadid, M. Pietikäinen, Face spoofing detection from single images using micro-texture analysis, in: Biometrics (IJCB), 2011 Int. Jt. Conf., 2011: pp. 1–7.

[25] T. De, F. 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) 1–15. http://jivp.eurasipjournals.com/content/2014/1/2.

[26] Z. Boulkenafet, J. Komulainen, A. Hadid, Face anti-spoofing based on color texture analysis, in: 2015 IEEE Int. Conf. Image Process., IEEE, 2015: pp. 2636–2640. https://doi.org/10.1109/ICIP.2015.7351280.

[27] T. de Freitas Pereira, A. Anjos, J.M. De Martino, S. Marcel, Can face anti-spoofing countermeasures work in a real world scenario?, in: 2013 Int. Conf. Biometrics, IEEE, 2013: pp. 1–8. https://doi.org/10.1109/ICB.2013.6612981.

[28] 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.

[29] J. Galbally, S. Marcel, J. Fierrez, Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition, IEEE Trans. Image Process. 23 (2014) 710–724.

[30] Di Wen, Hu Han, A.K. Jain, Face Spoof Detection With Image Distortion Analysis, IEEE Trans. Inf. Forensics Secur. 10 (2015) 746–761. https://doi.org/10.1109/TIFS.2015.2400395.

[31] P.P.K. Chan, W. Liu, D. Chen, D.S. Yeung, F. Zhang, X. Wang, C.C. Hsu, Face Liveness Detection Using a Flash Against 2D Spoofing Attack, IEEE Trans. Inf. Forensics Secur. 13 (2018) 521–534. https://doi.org/10.1109/TIFS.2017.2758748.

[32] A.P.S. Bhogal, D. Sollinger, P. Trung, A. Uhl, Non-reference image quality assessment for biometric presentation attack detection, in: 2017 5th Int. Work. Biometrics Forensics, IEEE, 2017: pp. 1–6. https://doi.org/10.1109/IWBF.2017.7935080.

[33] A. Sepas-Moghaddam, F. Pereira, P.L. Correia, Light Field-Based Face Presentation Attack Detection: Reviewing, Benchmarking and One Step Further, IEEE Trans. Inf. Forensics Secur. 13 (2018) 1696–1709. https://doi.org/10.1109/TIFS.2018.2799427.

[34] Y. Kim, J.-H. Yoo, K. Choi, A motion and similarity-based fake detection method for biometric face recognition systems, IEEE Trans. Consum. Electron. 57 (2011).

[35] K. Kollreider, H. Fronthaler, M.I. Faraj, J. Bigun, Real-time face detection and motion analysis with application in “liveness” assessment, IEEE Trans. Inf. Forensics Secur. 2 (2007) 548–558.

[36] A. Anjos, M.M. Chakka, S. Marcel, Motion-based counter-measures to photo attacks in face recognition, IET Biometrics. 3 (2013) 147–158.

[37] K. Patel, H. Han, A.K. Jain, G. Ott, Live face video vs. spoof face video: Use of moiré patterns to detect replay video attacks, Proc. 2015 Int. Conf. Biometrics, ICB 2015. (2015) 98–105. https://doi.org/10.1109/ICB.2015.7139082.

[38] L. Li, Z. Xia, L. Li, X. Jiang, X. Feng, F. Roli, Face anti-spoofing via hybrid convolutional neural network, Conf. Proc. - 2017 Int. Conf. Front. Adv. Data Sci. FADS 2017. 2018-Janua (2018) 120–124. https://doi.org/10.1109/FADS.2017.8253209.

[39] 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.

[40] 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.

[41] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Int. Conf. Learn. Represent. (ICLR), 2015, 2015: p. 14. https://arxiv.org/pdf/1409.1556.pdf http://arxiv.org/abs/1409.1556.pdf.

[42] J. Yang, Z. Lei, S.Z. Li, Learn convolutional neural network for face anti-spoofing, in: ArXiv Prepr. ArXiv1408.5601, 2014. https://arxiv.org/pdf/1408.5601.pdf.

[43] 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.

[44] 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.

[45] 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. 12 (2017) 1713–1723. https://doi.org/10.1109/TIFS.2017.2676720.

[46] 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.

[47] A. Jourabloo, Y. Liu, X. Liu, Face de-spoofing: Anti-spoofing via noise modeling, in: Ferrari V., Hebert M., Sminchisescu C., Weiss Y. Comput. Vis. – ECCV 2018. ECCV 2018. Lect. Notes Comput. Sci., 2018: pp. 297–315. https://doi.org/10.1007/978-3-030-01261-8_18.

[48] L. Li, Z. Xia, A. Hadid, X. Jiang, F. Roli, X. Feng, Face Presentation Attack Detection in Learned Color-liked Space, (n.d.) 1–13. https://doi.org/arXiv:1810.13170v1.

[49] 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.

[50] 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 (Switzerland). 18 (2018). https://doi.org/10.3390/s18030699.

[51] L. Li, X. Feng, X. Jiang, Z. Xia, A. Hadid, Face anti-spoofing via deep local binary patterns, Proc. - Int. Conf. Image Process. ICIP. 2017-Septe (2018) 101–105. https://doi.org/10.1109/ICIP.2017.8296251.

[52] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning Deep Features for

Discriminative Localization, in: 2016 IEEE Conf. Comput. Vis. Pattern Recognit., IEEE, 2016: pp. 2921–2929. https://doi.org/10.1109/CVPR.2016.319.

[53] J. Komulainen, Z. Boulkenafet, Z. Akhtar, Review of Face Presentation Attack Detection Competitions, in: Handb. Biometric Anti-Spoofing, Adv. Comput. Vis. Pattern Recognit., Springer International Publishing, 2019: pp. 291–317. https://doi.org/10.1007/978-3-319-92627-8_14.

[54] Z. Boulkenafet, J. Komulainen, Z. Akhtar, A. Benlamoudi, D. Samai, S.E. Bekhouche, A. Ouafi, F. Dornaika, A. Taleb-Ahmed, L. Qin, F. Peng, L.B. Zhang, M. Long, S. Bhilare, V. Kanhangad, A. Costa-Pazo, E. Vazquez-Fernandez, D. Pérez-Cabo, J.J. Moreira-Pérez, D. González-Jiménez, A. Mohammadi, S. Bhattacharjee, S. Marcel, S. Volkova, Y. Tang, N. Abe, L. Li, X. Feng, Z. Xia, X. Jiang, S. Liu, R. Shao, P.C. Yuen, W.R. Almeida, F. Andaló, R. Padilha, G. Bertocco, W. Dias, J. Wainer, R. Torres, A. Rocha, M.A. Angeloni, G. Folego, A. Godoy, A. Hadid, A competition on generalized software-based face presentation attack detection in mobile scenarios, IEEE Int. Jt. Conf. Biometrics, IJCB 2017. 2018-Janua (2018) 688–696. https://doi.org/10.1109/BTAS.2017.8272758.

[55] 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.

[56] X. Song, X. Zhao, L. Fang, T. Lin, Discriminative representation combinations for accurate face spoofing detection, Pattern Recognit. 85 (2019) 220–231. https://doi.org/10.1016/j.patcog.2018.08.019.

[57] Z. Boulkenafet, J. Komulainen, A. Hadid, Face spoofing detection using colour texture analysis, IEEE Trans. Inf. Forensics Secur. 11 (2016) 1818–1830.

[58] 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.

[59] 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.

[60] 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).

[61] R. Shao, X. Lan, P.C. Yuen, Joint Discriminative Learning of Deep Dynamic Textures for 3D Mask Face Anti-spoofing, IEEE Trans. Inf. Forensics Secur. PP (2018) 1–1. https://doi.org/10.1109/TIFS.2018.2868230.

[62] X. Tu, J. Zhao, M. Xie, G. Du, H. Zhang, J. Li, Z. Ma, J. Feng, Learning Generalizable and Identity-Discriminative Representations for Face Anti-Spoofing, (2019). http://arxiv.org/abs/1901.05602.

[63] X. Yang, W. Luo, L. Bao, Y. Gao, D. Gong, S. Zheng, Z. Li, W. Liu, Face Anti-Spoofing: Model Matters, So Does Data, IEEE Int. Conf. Comput. Vis. Pattern Recognit. (2019) 3507–3516.

[64] A. George, S. Marcel, Deep Pixel-wise Binary Supervision for Face Presentation Attack Detection, in: Int. Conf. Biometrics (ICB), 12th IAPR Int. Conf., 2019: p. 8.