INVESTIGATION IN SPATIAL-TEMPORAL DOMAIN FOR FACE SPOOF DETECTION

Zhonglin Sun, Li Sun, and Qingli Li

Shanghai Key Laboratory of Multidimensional Information Processing,
East China Normal University, 200241 Shanghai, China

ABSTRACT

This paper focuses on face spoofing detection using video. The purpose is to find out the best scheme for this task in the end-to-end learning manner. We investigate 4 different types of structure to fully exploit the raw data in its spatial-temporal domain, which are the pure CNN, CNN with 3D convolution, CNN+LSTM and CNN+Conv-LSTM. Moreover, another stream built on optical flow is also used, and with a proper fusion method, it can improve the accuracy. In experiments, we compare schemes on the raw data in single stream and fusion methods with optical flow in two streams. The performance are not only given within each dataset, but also measured across different datsets, which is crucial to avoid the overfitting.

Index Terms— Spoofing, CNN, LSTM, Conv-LSTM

1. INTRODUCTION

Computer vision application based on facial image has already been widely used. Particularly, face recognition emerges in varieties of vision systems such as security control, surveillance monitoring or human-computer interaction. On the other hand, it is known that most of existing recognition systems are vulnerable to spoofing attacks. The spoofing attack means someone tries to bypass a face biometric system by presenting a fake face in front of the camera. In [1], researches demonstrate the vulnerability of current commercial face authentication systems by using only the photo attack from social network. Therefore, to distinguish a real or fake face image has important applications.

Fake attacks are divided into three types: photo attack, video attack and mask attack. In photo attack, the intruder presents a real photo to the system. In such an attack, motion or depth in the areas of attackers image tends to be consistent. Moreover, it dose not have local facial movements. Video attacker provides a real video playback to the system. The face in the video is with natural expressions, but since the playback is on the screen, there are still consistent motion or depth pattern. In mask attack, the intruder wears a mask on his/her face. Some silicone mask gives a very misleading facial appearance, only texture cues help to distinguish it.

Traditional methods deal with spoof detection by going through two individual stages, which are feature extraction and classifier design. Most of them choose SVM for the binary classification, but they give different features to the classifier. In [2], the optical flow, which specifies the horizontal and vertical displacement between adjacent frames, is calculated, then magnitude of the flow vector is quantized into histogram. In [3], a method based on texture feature is proposed. Several feature operators are adopted to express the texture in image. Wen *et al.* [4] analyze the different types of image distortions caused by spoofing face, and incorporate more features for classification. Features, reflecting the specular reflection, blurriness, chroma moment, and color diversity, are extracted.

With the great success of the Convolution Neural Network (CNN) on image classification task, researchers take effort to investigate its application in face spoof detection. Many algorithms based on CNN have been proposed. In [5], Li *et al.* propose a deep model based on VGG-16, which is pretrained on VGG-face dataset [6]. They finetune the model with spoof detection dataset and extract features from different layers. They give the features to SVM to make final classification. Valle *et al.* [7] also use VGG-16 model, but they finetune the model in an end-to-end manner without employing SVM. Amin *et al.* [8] propose a two stream CNN-based structure. One stream, trained with the local patch in face region, treat spoof detection as regression task and produces a spoof score for each patch. The other stream are fed with whole face region with its purpose of estimating the depth map. They make the final fusion of two streams. All the above works [5, 7, 8] treat the spoof detection within a single image without exploit the temporal information. Xu *et al.* [9] consider both spatial and temporal domain representation. They use CNN for feature learning in image spatial domain and Long Short-Term Memory (LSTM) for temporal domain. But the structure of their network has only two convolution layers which is rather shallow.

This paper investigates the face spoof detection based on video within the spatial temporal domain. Our goal is to find the best way to perform the end-to-end feature learning. Since CNN achieves the best performance in all image related

This work was supported in part by the National Natural Science Foundation of China under Project 61302125, 61671376 and in part by Natural Science Foundation of Shanghai under Project 17ZR1408500. Corresponding to sunli@ee.ecnu.edu.cn

classification task, the feature learning way in 2D image spatial domain becomes normal. The key issue is how to exploit the temporal domain and how to combine the temporal domain scheme with CNN for this particular task. Therefore, we explore 4 different types of structures which are pure CNN with directly image stacking (CNN-Stacking), CNN with 3D convolution (CNN-3DConv), CNN combined with LSTM (CNN-LSTM) and CNN combined with convolutional LSTM (CNN-ConvLSTM). Moreover, we find spoof detection task is easy to lead to overfitting. Thus another stream using optical flow is also built based on CNN. This second stream is first trained individually. Then fusion scheme of the two stream is also provided. Intensive experiments on CASIA and replay datasets are performed to give both intra and cross datasets evaluation. There are also some insightful observations in this paper. First, regularization technique such as batch normalization is crucial since it is an easily overfitting task. Second, CNN achieves better results even in the spatial temporal domain than other complex structure.

We organize the remainder of the paper as follows. Details about our investigation on different structures are given in Section 2. Results of experiment and their comparisons are given in Section 3. Brief conclusions are finally given in Section 4.

2. INVESTIGATION ON DIFFERENT STRUCTURES IN SPATIAL-TEMPORAL DOMAIN

Different structures in spatial temporal domain are compared in this paper. The overview flowchart is given in Fig.1. As is shown, 3 different types of inputs, constrained in the face region, are investigated in this paper. They are single or stacked region(s) of raw images, and the optical flow in face region calculated between the two adjacent frames. The face detection algorithm in [10] help to locate the face region in each image. The raw images given to CNN form the first stream, and the optical flow also given to CNN makes the second stream. We try and compare different structures for the first stream and uses the simplest structure which is CNN defined on 2 channel optical flow for the second stream. Note that the number of channels for filters in the first convolution layer (conv1) are different since it depends on the number of input channels, so it separates from CNN in Fig. 1. From conv2 to conv5 layer, the same structure is used for different types of input. After conv5, a global average pooling layer (GAP) is used to reduce the feature dimension in its spatial coverage. To fully exploit the spatial temporal domain, we try both 2D conv and 3D conv in CNN for stacked regions input. With the same purpose, the output from 2D conv for the single region is further given to LSTM or Conv-LSTM. The final binary decision is made based on the feature given by CNN-Stacking, CNN-3DConv, CNN-LSTM or CNN-ConvLSTM. The best scheme among 4 candidates are chosen, and we also investigate different fusion schemes with the second stream.

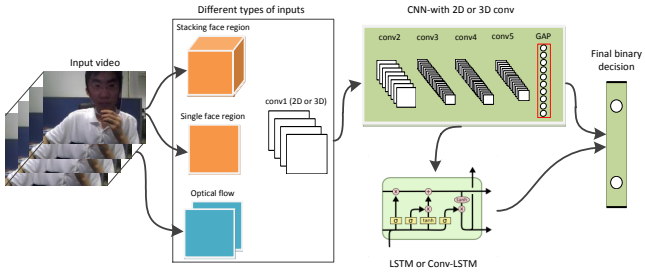

Fig. 1. Overview of different structures to perform feature learning for face spoof detection.

In summary, we extensively compare different ways to conduct feature learning on the first stream with raw images as its input. In 4 schemes, CNN-Stacking, CNN-3Dconv, CNN-LSTM and CNN-ConvLSTM, the end-to-end training are conducted independently without sharing model parameter, but they all have the same network structure except conv1 layer. In the following subsection, we illustrate the structure in following aspects. The comparison is between 2D and 3D convolution for stacked regions input, and then between LSTM or Conv-LSTM for single region input. Finally we describe the implementation on the second optical flow stream, and the fusion strategies for two steams.

2.1. 2D or 3D convolution

We have two ways to reformulate images so that it can be given to conv1 of CNN. Each image can be directly given to conv1 layer, or 15 images, at different time stamp, are stacked and given conv1 together. Here we only focus on stacked regions input and leave single region input to LSTM or Conv-LSTM.

2D conv is normal in CNN, in which the filter slides over the input tensor only in 2D spatial domain. Since there are multiple regions captured at different time stamp in the input, we are curious about whether 3D conv can improve it. 3D conv, which has already been successful in the video application [11], extends 2D conv by sliding not only in 2D spatial domain, but also along the channel direction. Fig. 2 depicts the differences between 2D and 3D conv. It is obvious that 3D conv has fewer parameters than 2D conv because it does not take all channels of input tensor. But in 3D conv, 1 conv filter generates several feature maps while 1 conv filter only generate 1 map in 2D conv. 3D conv consumes more memory and time than 2D conv because of another extra convolution along the channel direction

2.2. LSTM or Conv-LSTM

As is described in previous subsection, the single face region is first given to CNN to perform feature learning in only spatial domain. In order to incorporate the temporal domain

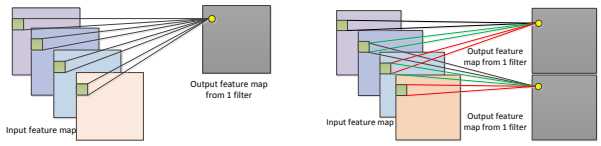

Fig. 2. Comparison between 2D and 3D conv. The normal 2D conv is shown on the left, and 3D conv is shown on the right.

into it, we use two types of Recurrent Neural Networks (RNN), which are LSTM and Conv-LSTM. Here we only give brief descriptions about them and more details can be found in [12, 13].

LSTM takes feature vector x_t from the GAP layer in CNN at time t. x_t corresponds only to the image at time t because CNN uses the single face region as its input in this case. The output of LSTM is h_t which will be used as a feature vector for the final decision. The cell state is represented by vector c_t. There are also vectors specified by the *input*, *forget* and *output* gate, represented by i_t, f_t and o_t. At each time stamp t, LSTM update i_t, f_t, c_t, o_t and h_t in sequence. The update is carried out as follows.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci} \circ c_{t-1} + b_i) \quad (1a)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} \circ c_{t-1} + b_f) \quad (1b)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (1c)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \circ c_t + b_o) \quad (1d)$$

$$h_t = o_t \circ tanh(c_t) \quad (1e)$$

In the above equations, $\sigma(\cdot)$ is the sigmoid function used for *input*, *forget* and *output* gate. $\circ$ denotes the Hadamard product. In each gate, W is the model parameter that is learned from the data.

LSTM handles the spatial temporal data through the inner product in input-to-state and state-to-state transitions. Conv-LSTM is a variant which intends to change the inner product into convolution. In Conv-LSTM, x_t and h_{t-1} become 2D matrixes and make their contribution through convolution in each gate, so they only. The state update equation in Conv-LSTM is similar with (1) except minor but crucial change from inner product to convolution, and it can be found in [12].

2.3. Optical flow stream and fusion strategies for two streams

Optical flow is the pattern of apparent motion which is calculated based on two adjacent images. It defines both horizontal and vertical displacement for each pixel, and reflects motion about object and scene. Optical flow has demonstrated its effectiveness for action recognition in video [14]. In face spoof detection, motion is considered as a useful cue by traditional algorithms [2], because the attacking like photo or video can be observed by human eyes from its motion pattern.

In this paper, we propose to form another stream based on CNN which takes advantage of optical flow as its input feature. The algorithm in [15] is adopted to calculated the dense optical field for each pixel between two images. Fig. 3 shows optical flow features for attack and real faces respectively. The 2-channel input is given to a CNN to perform further feature learning based on it, hoping to find out the useful cues from it.

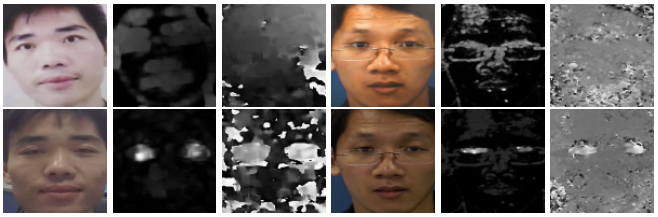

Fig. 3. Optical flow feature demonstration. The first row is the results for attack faces and the second row is for the real one. We show both horizontal and vertical components in the flow. For references, one original image is also provided for each identity. Note that the two person identities in the first row are the same with the second row.

We now consider making fusion for two streams and give the inference result on the video. In this second stream, CNN gives the binary inference on each optical flow calculated from two frames. In the first stream, the candidate model can be either CNN-Stacking, CNN-3DConv, CNN+LSTM or CNN+Conv-LSTM. In this paper, we choose a simple fusion scheme of 2 streams. For each stream, we take feature vector before softmax layer from each stream, and concatenate them into a longer vector. Finally, the long vector is given to a FC layer with two output nodes. Note that we do not change the parameters in low-level conv layers, in other words, only parameters in FC layer get trained in fusion step.

3. EXPERIMENTS AND DISCUSSIONS

3.1. Implementation details

Our experiments are conducted on 3 datasets, which are replay-attack [16], CASIA [17] and 3DMAD [18]. Replay-attack and CASIA are given by a series of large video clips. So we need some preprocessing methods for efficiency. First the video is decoded into images by ffmpeg and down-sampled temporally, leaving only one frame in every three, and each video is divided into subclips with $N = 15$ frames in each of them. Face detection [10] is performed on each frame and optical flow [15] is calculated on two adjacent frames in subclips. Note that each frame is resized to 96×96 before given to CNN. During testing, we accumulate the binary inference results on each subclip. When 50% or above of the subclips are considered to be attacking, then the test video is regarded as the attacker.

Tensorflow 1.1.0 is used as our deep learning platform for training and testing. As is shown in Fig.1, the structure of our network is the variant of CaffeNet [19], which is rather simple, with only 5 conv layers. The accuracy can be easily improved by using more complex architecture. Our hyper-parameters are listed as follows: weight decay is 0.01, initial learning rate is 0.001, batchsize is 50, and the learning rate is divided by 10 for each iteration, and the optimization method is Adam. To accelerate the convergence, we use CaffeNet pretrained model in conv1 to conv5 layers. For CNN with single face region, the parameters in pretrained model can be directly applied, but for CNN-Stacking, parameters need to extend in channel depth direction. Batch normalization is also applied in each conv layer and fc layer in CNN. We find BN is important in face spoof detection task to avoid overfitting. In LSTM, BN is adopted in each gate in (1) and is independent on input x_t and historical output h_{t-1}. Details about BN in LSTM can be found in [20]. During training, each stream is first trained independently, then they join together to get the fusion results.

3.2. Quantitative evaluations and discussions

Half Total Error Rate (HTER) and Equal Error Rate (EER) are two numerics which are often used in biometric recognition system. HTER depends on both False Accept Rate (FAR) and False Reject Rate (FRR), and is the arithmetic mean. EER is also determined by FAR and FRR, and it is the value when FAR equals to FRR. HTER and EER scores are given as %, and we ignore % in this paper.

We compared HTER and EER of 5 separate models, which are CNN-Stacking, CNN-3DConv, CNN+LSTM, CNN+Conv-LSTM, and CNN-Optical. Moreover, the first 4 models, forming the first stream, are fused with the second stream CNN-Optical to improve the performance.

	model	3DMAD	Replay-attack	CASIA
	Spoofnet [21]	0/-	0.70/-	-
	FASNet [7]	0/-	1.20/-	-
	Pluse [22]	7.94/4.71	-	-
	LSTM-CNN [9]	-	-	5.93/5.17
	Multi-cues Integration [23]	0/-	0/-	-/5.83
	Diffusion-based Kernel Matrix [24]	-	4.30/-	-
	Dynamic Texture [25]	-	7.60/-	-/10.00
	Motion Mag [26]	-	1.25/-	-
	Moire pattern [27]	-	3.30/-	0/-
	Colour Texture [3]	-	2.80/0	-/2.10
	Patch and Depth CNN [28]	-	0.72/0.79	2.27/2.67
	CNN-Stacking	0/0	0.64/3.84	3.72/6.74
	CNN-3Dconv	0/3.30	1.80/3.84	6.51/11.23
single	CNN+LSTM	0/0	1.80/2.50	6.51/16.85
	CNN+Conv-LSTM	1.16/3.30	5.13/5.12	14.60/22.40
	CNN-Optical	1.60/0	3.60/11.26	13.84/13.48
	CNN-Stacking	0/0	0.38/2.66	3.49/6.70
fusion	CNN-3Dconv	0/0	2.56/3.77	9.12/13.40
	CNN+LSTM	0/0	1.68/1.28	5.22/14.60
	CNN+Conv-LSTM	0.81/1.66	1.92/6.40	11.44/23.50

Table 1. Comparison of HTER/EER performance on 3 datasets.

On 3DMAD, CNN-Stacking, CNN-3DConv, and CNN+LSTM have all reached 0 by within only 1 stream structure. CNN+Conv-LSTM and CNN-Opitical have slightly worse values but they can be improved after fusion. On replay-attack, CNN-Stacking reaches 0.64 within single stream, and 0.38 for bi-stream, which is the slightly worse than [23]. In addition, compared with the single stream model, we find that fusion actually improves the performance on replay-attack. Moreover, the structure of CNN used in this paper is rather simple, and the performance will be significantly improved if VGG or RetNet is applied. It is also worth mentioning that the simple CNN-Stacking scheme performs the best in both single or two streams. This demonstrates that even for task of spoof detection in video, CNN is still easy to train and powerful.

For practical considerations, cross dataset evaluation is also necessary. Since we find face spoof detection often leads to overfitting, it is doubtful that the above good performances in Table 1 are actually caused by overfiting. The cross dataset evaluation are carried out on replay-attack and CASIA. In other words, the model get trained on replay-attack dataset and is tested on CASIA or vice versa.

	model	replay-attack	CASIA
	Motion [29]	48.28	50.25
	LBP [29]	57.90	47.05
	LBF-TOP [29]	61.33	50.64
	Motion Mag [26]	47.00	50.10
	Spectral cubes [30]	50.00	34.38
	Colour Texture [3]	37.70	30.30
	CNN-Stacking	41.60	22.72
	CNN-3DConv	49.60	37.74
single	CNN+LSTM	42.73	41.10
	CNN+Conv-LSTM	48.70	33.20
	optical flow	30.14	36.80
fusion	stacking	40.40	20.59

Table 2. HTER performance for cross-dataset evaluation.

4. CONCLUSION

In this paper, we conduct an extensive investigation on a series of spatial temporal models based on deep learning for face anti-spoofing task. We compare the performance of the single stream model, named as CNN-Stacking, CNN-3DConv, CNN+LSTM, CNN+ConvLSTM, which take different types of raw images as the input. A two-stream structure, with a second stream constructed on optical flow, is also proposed. With proper fusion scheme, the two stream structure, with its first stream using CNN-Stacking, gives the state-of-the art performance.

5. REFERENCES

[1] Yan Li, Ke Xu, Qiang Yan, Yingjiu Li, and Robert H Deng, "Understanding osn-based facial disclosure against face authentication systems," in *Proceedings of the 9th ACM symposium on Information, computer and communications security*. ACM, 2014, pp. 413–424.

[2] André Anjos, Murali Mohan Chakka, and Sébastien Marcel, "Motion-based counter-measures to photo attacks in face recognition," *IET biometrics*, vol. 3, no. 3, pp. 147–158, 2013.

[3] Zinelabidine Boulkenafet, Jukka Komulainen, and Abdenour Hadid, "Face spoofing detection using colour texture analysis," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1818–1830, 2017.

[4] Di Wen, Hu Han, and Anil K Jain, "Face spoof detection with image distortion analysis," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 746–761, 2015.

[5] Lei Li, Xiaoyi Feng, Zinelabidine Boulkenafet, Zhaoqiang Xia, Mingming Li, and Abdenour Hadid, "An original face anti-spoofing approach using partial convolutional neural network," in *Image Processing Theory Tools and Applications (IPTA), 2016 6th International Conference on*. IEEE, 2016, pp. 1–6.

[6] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al., "Deep face recognition.," in *BMVC*, 2015, vol. 1, p. 6.

[7] Eduardo Valle and Roberto Lotufo, "Transfer learning using convolutional neural networks for face anti-spoofing," in *Image Analysis and Recognition: 14th International Conference, ICIAR 2017, Montreal, QC, Canada, July 5–7, 2017, Proceedings*. Springer, 2017, vol. 10317, p. 27.

[8] Yousef Atoum, Yaojie Liu, Amin Jourabloo, and Xiaoming Liu, "Face anti-spoofing using patch and depth-based cnns," in *Biometrics (IJCB), 2017 IEEE International Joint Conference on*. IEEE, 2017, pp. 1–6.

[9] Zhenqi Xu, Shan Li, and Weihong Deng, "Learning temporal features using lstm-cnn architecture for face anti-spoofing," in *Pattern Recognition (ACPR), 2015 3rd IAPR Asian Conference on*. IEEE, 2015, pp. 141–145.

[10] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[11] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu, "3d convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[12] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015, pp. 802–810.

[14] Karen Simonyan and Andrew Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, 2014, pp. 568–576.

[15] Gunnar Farnebäck, "Two-frame motion estimation based on polynomial expansion," *Image analysis*, pp. 363–370, 2003.

[16] Ivana Chingovska, André Anjos, and Sébastien Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," 2012.

[17] Zhiwei Zhang, Junjie Yan, Sifei Liu, Zhen Lei, Dong Yi, and Stan Z Li, "A face antispoofing database with diverse attacks," in *Biometrics (ICB), 2012 5th IAPR international conference on*. IEEE, 2012, pp. 26–31.

[18] Nesli Erdogmus and Sébastien Marcel, "Spoofing in 2d face recognition with 3d masks and anti-spoofing with kinect," in *Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on*. IEEE, 2013, pp. 1–6.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[20] Tim Cooijmans, Nicolas Ballas, César Laurent, Çağlar Gülçehre, and Aaron Courville, "Recurrent batch normalization," *arXiv preprint arXiv:1603.09025*, 2016.

[21] David Menotti, Giovani Chiachia, Allan Pinto, William Robson Schwartz, Helio Pedrini, Alexandre Xavier Falcao, and Anderson Rocha, "Deep representations for iris, face, and fingerprint spoofing detection," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 864–879, 2015.

[22] Xiaobai Li, Jukka Komulainen, Guoying Zhao, Pong-Chi Yuen, and Matti Pietikäinen, "Generalized face anti-spoofing by detecting pulse from face videos," in *Pattern Recognition (ICPR), 2016 23rd International Conference on*. IEEE, 2016, pp. 4244–4249.

[23] Litong Feng, Lai-Man Po, Yuming Li, Xuyuan Xu, Fang Yuan, Terence Chun-Ho Cheung, and Kwok-Wai Cheung, "Integration of image quality and motion cues for face anti-spoofing: A neural network approach," *Journal of Visual Communication and Image Representation*, vol. 38, pp. 451–460, 2016.

[24] Changyong Yu and Yunde Jia, "Anisotropic diffusion-based kernel matrix model for face liveness detection," *arXiv preprint arXiv:1707.02692*, 2017.

[25] Tiago de Freitas Pereira, Jukka Komulainen, André Anjos, José Mario De Martino, Abdenour Hadid, Matti Pietikäinen, and Sébastien Marcel, "Face liveness detection using dynamic texture," *EURASIP Journal on Image and Video Processing*, vol. 2014, no. 1, pp. 2, 2014.

[26] Samarth Bharadwaj, Tejas I Dhamecha, Mayank Vatsa, and Richa Singh, "Computationally efficient face spoofing detection with motion magnification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 105–110.

[27] Keyurkumar Patel, Hu Han, Anil. K. Jain, and Greg Ott, "Live face video vs. spoof face video: Use of moir patterns to detect replay video attacks," in *International Conference on Biometrics*, 2015, pp. 98–105.

[28] Yousef Atoum, Yaojie Liu, Amin Jourabloo, and Xiaoming Liu, "Face anti-spoofing using patch and depth-based cnns," in *The International Joint Conference on Biometrics*, 2017.

[29] Tiago De Freitas Pereira, Andr Anjos, Jos Mario De Martino, and Sbastien Marcel, "Can face anti-spoofing countermeasures work in a real world scenario?," in *International Conference on Biometrics*, 2013, pp. 1–8.

[30] Allan Pinto, Helio Pedrini, William Robson Schwartz, and Anderson Rocha, "Face spoofing detection through visual codebooks of spectral temporal cubes," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4726–4740, 2015.