Performance Evaluation of Face Anti-Spoofing Method Using Deep Metric Learning from a Few Frames of Face Video

Koichi Ito*, Asateru Kimura * and Takafumi Aoki*
* Graduate School of Information Sciecnes, Tohoku University, Japan.
E-mail: ito@aoki.ecei.tohoku.ac.jp

***Abstract*—Recent advances in face recognition and deep learning technologies are enabling us to identify individuals from images captured by a camera from a distance. On the other hand, there is a problem that a malicious person can impersonate the registered user by presenting a photo or video of the registered user's face. Spoofing detection using video input, from which more features can be extracted than images, has not been studied very much. In this paper, we propose a method for detecting spoofing from video images of a small number of frames. The proposed method uses features extracted from video images using 3D Convolutional Neural Network (3D CNN). We also use deep metric learning to improve the accuracy of detection. We demonstrate the effectiveness of the proposed method through performance evaluation experiments using a large-scale spoofing attack dataset.**

I. Introduction

Biometrics, which uses physical and behavioral characteristics of an individual, has been attracting much attention as an emerging personal authentication method that is an alternative to keys and passwords [1]. Biometric recognition employs physical characteristics such as faces, fingerprints, DNA, and palm prints, and behavioral characteristics such as handwriting and keystrokes. Among them, face recognition is highly acceptable since it is the same as natural personal authentication, and is highly convenient since it is possible to authenticate from a face image taken by a common RGB camera [2]. On the other hand, there is a risk that a malicious person can impersonate a registered user using a duplicate of the registered user's biometric information.

There are two types of major face spoofing attacks against face recognition systems: "Print-Attack," which presents a printed picture of the registered user's face, and "Display-Attack," which presents a video image of the registered user's face. We have to determine whether the input image or video is real or fake to prevent spoofing attacks. Spoofing detection methods such as Local Binary Pattern (LBP) and Histogram of oriented Gradients (HoG) are based on manually designed features [3], [4], [5], [6], [7]. There is a problem of decreasing accuracy due to differences in the acquisition environment since general-purpose image features are used. Recently, many methods based on Convolutional Neural Network (CNN) have been proposed with the development of deep learning [8], [9], [10], [11], [12], [13]. This approach is more accurate than conventional general-purpose feature-based methods since it can extract features based on training data.

Although face recognition systems often use video images, most of the spoofing detection methods using CNN are based on images, and there are few methods using video images. We expect to improve the accuracy of spoofing detection by using video images as input since video images can extract not only spatial features but also temporal features. In this paper, we propose a spoofing detection method using a 3D CNN with video input. The proposed method improves the accuracy by using (2+1)D CNN [14] instead of standard 3D CNN. We also use deep metric learning to improve the accuracy against spoofing attacks that are not included in the training data. Deep metric learning is one of the learning methods that takes into account the relationships between feature vectors in feature space (e.g., distance and similarity). In particular, our method learns to reduce the intra-class variance and increase the inter-class variance for real and fake by deep metric learning. We demonstrate the effectiveness of the proposed method in detecting spoofing through performance evaluation experiments using the Spoof in the Wild (SiW) dataset[1] [15], a large-scale dataset of spoofing attacks against face recognition.

II. Related Work

Spoofing detection in face recognition can be divided into two types of approaches, one based on texture features and the other based on temporal features. We summarize their major methods in the following.

A. Approach based on Texture Features

This approach is divided into two categories: general-purpose image features [3], [4], [5], [6], [7] and CNN [8], [9], [10], [16]. Manually designed features such as LBP [3], [4], HoG [5], Scale-Invariant Feature Transform (SIFT) [6], and Speeded-Up Robust Features (SURF) [7] are used as general-purpose features. Support Vector Machine (SVM) and Linear Discriminant Analysis (LDA) are used to discriminate between real and fake features. They are vulnerable to changes in the environment and have a low detection accuracy since these features are designed manually. Recently, many methods using CNNs have been studied with the development of deep learning [8], [9], [10], [16]. Most of methods use CNN models

[1] http://cvlab.cse.msu.edu/siw-spoof-in-the-wild-database.html

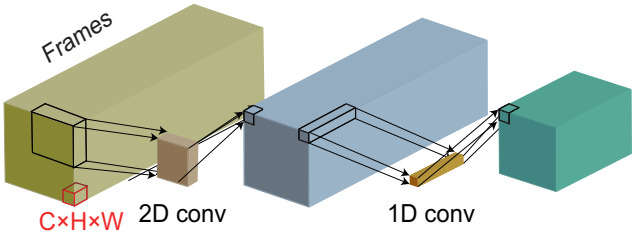

Fig. 1. Overview of (2+1)D convolution used in the proposed method.

trained by ImageNet to extract features and detect spoofing attacks by these features [8], [9], [10]. Ref. [16] employs two CNNs: one is to extract texture features of each patch and another is to extract features based on the depth information of the face. CNN-based methods achieve higher detection accuracy than those using general-purpose image features. On the other hand, there is a problem that the detection accuracy of spoofing attacks is not good enough to detect unknown spoofing attacks since the binary classification of real and fake is based on training data.

B. Approach based on Temporal Features

For print attacks, methods using temporal features have been proposed since the real thing is in motion [10], [11], [12], [13], [16]. Spoofing detection methods based on the detection of eye blinks between video frames were proposed [10], [13] It is effective against print attacks, but not necessarily effective against spoofing attacks such as display attacks, which present moving images since they contain eye blinks. In [11], the detection accuracy was improved by extracting dynamic features from multiple frames using a 3D CNN. In [12], the detection accuracy was improved by adding optical flow to the CNN input and extracting features that take motion into account. In [15], the detection accuracy is improved by using the depth information of the face estimated by CNN and remote photoplethysmography (rPPG) estimated from the dynamic change of the face texture. There is a problem that some methods require many frames to extract temporal features.

III. Proposed Method

The proposed method detects spoofing based on features extracted from the input video image using 3D CNN. In the following, we describe the 3D CNN architecture and deep metric learning for the proposed method.

A. Network Architecture

In the case of spoofing attacks on face recognition systems, it is expected that the detection accuracy of spoofing attacks on face recognition systems can be improved by extracting temporal features from video images instead of still images [11]. In this paper, we use a network using (2+1)D convolution [14] with a reduced number of 3D CNN parameters to extract temporal features. (2+1)D convolution is a decomposition

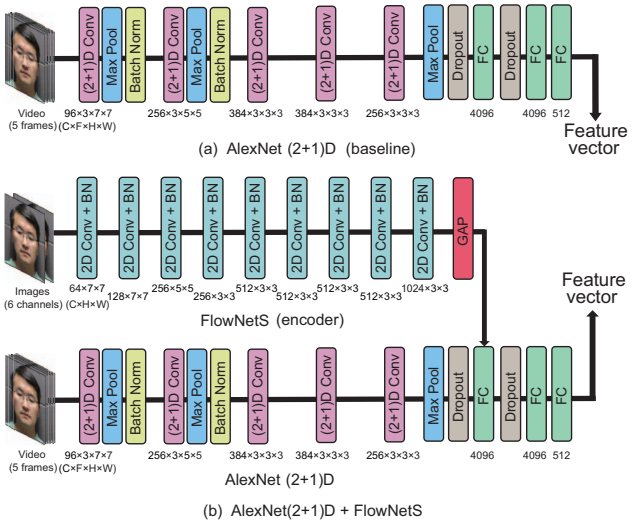

Fig. 2. Pipeline of the network architecture used in the proposed method: (a) baseline and (b) baseline with FlowNetS.

of a 3D filter into spatial and temporal filters as shown in Fig. 1. (2+1)D convolution allows us to reduce the number of parameters while extracting temporal features, and thus suppress overfitting. In this paper, we designate AlexNet [17] with (2+1)D convolution as the baseline, which is called ALexNet(2+1)D, as shown in Fig. 2 (a).

To improve the accuracy of the baseline, we add optical flow-based features as shown in [12]. We use the features extracted by FlowNetS [18], which is a network for estimating optical flows. FlowNetS is a method for estimating optical flows between images consisting of encoder decoders. In this paper, we use only the encoder part to utilize optical flow-based features. The features obtained by FlowNetS are aggregated by Global Average Pooling (GAP) and given as the input to the first fully-connected layer as shown in Fig. 2 (b). FlowNetS uses a model that has been trained on the Flying Chairs dataset. AlexNet(2+1)D inputs a 5 frame movie image, and FlowNetS inputs the 1st and 5th frame images.

B. Deep Metric Learning

Detection of spoofing attacks is a binary classification problem to distinguish between real and fake, and therefore it is not always possible to accurately distinguish spoofing attacks that do not exist in the training data. In order to deal with such unknown spoofing attacks, we employ deep metric learning. Deep metric learning is a method of learning in which the intra-class variance is small and the inter-class variance is large. Deep metric learning may allow us to detect unknown spoofing attacks since the score between feature vectors of data not included in the training data and the feature vectors of the real class may be reduced. We consider the following methods to confirm the effectiveness of deep metric learning in detecting spoofing attacks.

Contrastive Loss

Let $\boldsymbol{x}_i$ and $\boldsymbol{x}'_j$ be the feature vectors of class i and j, respectively. The contrastive loss is a loss function that is trained to reduce the distance between feature vectors when the classes i and j are of the same class and to increase the distance when the classes are different, and is defined by

$$L_{\text{cont}} = \begin{cases} D(\boldsymbol{x}, \boldsymbol{x}') & i = j \\ \max(m - D(\boldsymbol{x}, \boldsymbol{x}'), 0) & i \neq j \end{cases}, \quad (1)$$

where $D(\boldsymbol{x}, \boldsymbol{x}')$ indicates a function calculating the distance between two feature vectors and m indicates a margin.

Triplet Loss

Let consider 3 feature vectors: $\boldsymbol{x}_a$, $\boldsymbol{x}_p$, and $\boldsymbol{x}_n$. The triplet loss is a loss function that makes the distance between a feature vector $\boldsymbol{x}_a$ and a closely related feature vector $\boldsymbol{x}_p$ closer and farther away from a distant feature vector $\boldsymbol{x}_n$ when focusing on a feature vector $\boldsymbol{x}_a$, and is defined by

$$L_{\text{Triplet}} = \max(D(\boldsymbol{x}_a, \boldsymbol{x}_p) - D(\boldsymbol{x}_a, \boldsymbol{x}_n) + \alpha, 0), \quad (2)$$

where α indicates a margin.

Cosine Similarity Loss

Major cosine similarity-based deep metric learning includes ArcFace [19], SphereFace [20], and CosFace [21]. All the methods penalize the angle between the feature vectors of the correct class and the input feature vectors while training the class classification. The cosine similarity loss is a loss function that is trained to increase the inter-class variance and to decrease the intra-class variance, and is defined by

$$L_{\cos} = \log \frac{e^{s\{\cos(m_1\theta_i + m_2) - m_3\}}}{e^{s\{\cos(m_1\theta_i + m_2) - m_3\}} + \sum_{j=1, j\neq i}^{n} e^{s\cos\theta_j}}, \quad (3)$$

where θ_i indicates the angle between the input feature vector and the feature vector of class i, i is a correct class label, and s is a scaling parameter. m_1, m_2, and m_3 indicates the penalty parameter for SphereFace, ArcFace, and CosFace, respectively.

IV. Experiments and Discussion

We describe experiments to evaluate the accuracy of spoofing attacks on face recognition systems using the SiW dataset [15].

A. SiW Dataset

The SiW dataset [15] consists of real, print attack, and display attack moving images taken from 165 subjects. The distance between the camera and the face, head pose, facial expression, and lighting were varied to evaluate the robustness to environmental changes. Each frame of the video image has 1,920×1,080 pixels and was captured at 30 fps for about 15 seconds. In the print attack, two types of paper with the subject's face printed on them are held up to the camera. In the display attack, 4 display devices are used per subject to play a video of their face. Fig. 3 shows some examples of images in the SiW dataset.

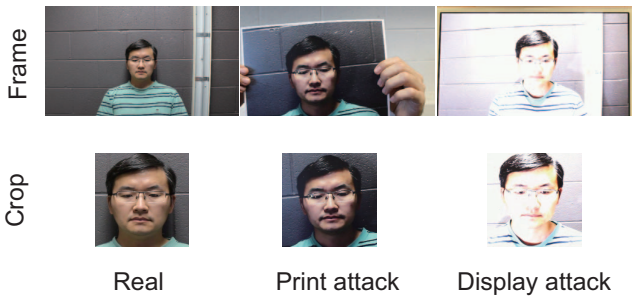

Fig. 3. Example of images in the SiW dataset.

TABLE I
Evaluation protocol provided by the SiW dataset.

Protocol	Subset	Subject	Attack
1	Train	90	First 60 frames
	Test	75	All
2	Train	90	3 Displays
	Test	75	1 Display
3	Train	90	Print (Display)
	Test	75	Display (Print)

B. Evaluation Protocol in SiW

The SiW dataset was created to evaluate the generalization performance of spoofing detection methods against spoofing attacks, and three evaluation protocols are provided as shown in Table I. Protocol 1 is designed to evaluate the generalization performance for various facial angles, expressions, and poses. At 60 frames from the start of the video image, the subject did not move much. Only the first 60 frames are used for training, and all frames are tested. Protocol 2 is designed to evaluate the generalization performance for the display attacks. Train on three display devices and test on the remaining one. The purpose of protocol 3 is to evaluate the performance against unknown spoofing attacks. We train using only one of the print attack or display attack and test against the untrained spoofing data. Note that cross-validation is performed for protocol 2 and protocol 3.

C. Evaluation Metrics

In this paper, we evaluate the Attack Presentation Classification Error Rate (APCER), Bona fide Presentation Classification Error Rate (BPCER), and Average Classification Error Rate (ACER). APCER represents the maximum False Acceptance Rate (FAR) for spoofing attacks. BPCER is equivalent to the False Rejection Rate (FRR). ACER is calculated by the average of APCER and BPCER. A smaller value indicates a higher accuracy for both metrics.

D. Experimental Condition

In this experiment, 60, 30, and 75 subjects were used for training, threshold determination, and testing, respectively. Some video images show a hand holding a printed paper or the edge of the display, which can be detected easily as a spoofing attack. Therefore, the 244×244-pixel face area extracted by the face detector is taken as input. The FrontalFace Detector

of Dlib[2] is used as a face detector in the experiments. For each frame of the video image, the average of the pixel values is set to 0 and the variance is set to 1. The optimization method Nesterov Accelerated Gradient (NAG) [22] is used. The maximum epoch is 20, and if the value of the loss function does not improve with respect to the data for determining the threshold for 4 consecutive epochs, training is terminated. The initial value of the learning rate is 0.005, and the learning rate is multiplied by 0.9 for each epoch. The cosine similarity between the representative vectors of the real class and the extracted feature vectors is used as a score. The threshold is the score at which the FAR and FRR are equal. Data augmentation methods of left-right flipping and random erasing [23] are applied to images during training.

E. Results

We compare the performance of the proposed method for (i) the number of input frames, (ii) deep metric learning, and (iii) conventional method, following the SiW evaluation protocol.

1) The number of input frames: In the baseline and the method combining the baseline and FlowNetS, the detection accuracy is compared by changing the number of input frames. Table II shows the experimental results. For all the protocols, the method combined with FlowNetS was more accurate than the baseline. Features based on the optical flows estimated by FlowNetS were distributed at different locations in the feature space for real, print attack, and display attack, which may have contributed to reducing the intra-class variance and increasing the inter-class variance. The accuracy is lower when the number of input frames is low and higher when the number of input frames is high, however, the greater the number of input frames, the longer it takes to process, therefore an input of about 5 frames is considered to be suitable.

2) Deep metric learning: We compare the accuracy of the proposed method trained by deep metric learning. In this experiment, we use cosine similarity as the distance function D between the feature vectors. Table III shows the results of the experiments with each deep metric learning method. From the experimental results, the proposed method trained by ArcFace showed the best accuracy. The result is expected to be due to the better discrimination of spoofing attacks in terms of penalties compared to SphereFace and CosFace, which are methods based on the same cosine similarity. In particular, in protocol 3, the method using deep metric learning is more accurate than the method using the cross entropy Loss. In this paper, we demonstrate that deep metric learning can improve detection accuracy against unknown spoofing attacks.

3) Comparison with conventional method: The accuracy of the proposed method is compared with that of the conventional method, FAS-BAS[15]. Table IV shows the results of the experiment. In protocol 2, the accuracy of the proposed method is higher than that of the conventional method. On the other hand, the accuracy of the conventional method is high, especially in protocol 3. While the proposed method requires 5 frames of video input, FAS-BAS requires about 100 frames of input to obtain rPPG. Therefore, it is possible to detect the display attack with higher accuracy compared to the proposed method.

[2]http://dlib.net/

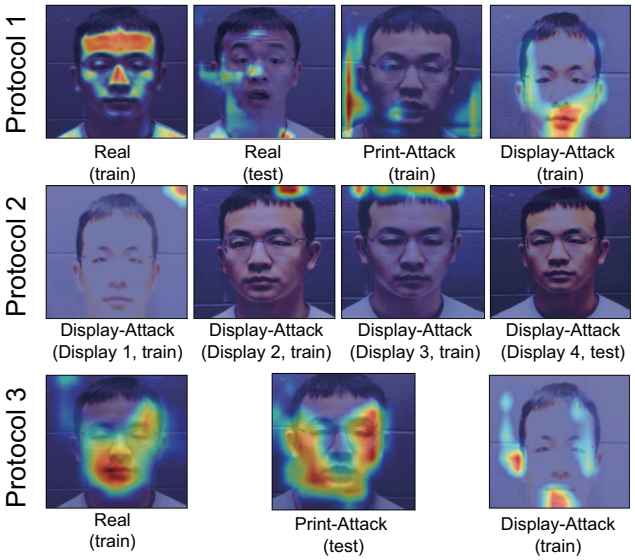

Fig. 4. Examples of activation maps for each protocol obtained using Grad-CAM

F. Discussion

We discuss what kind of evidence the proposed method uses to discriminate between real and spoofing attacks by visualizing regions of interest in CNN using Gradient-weighted Class Activation Mapping (Grad-CAM) [24]. An example of an activation map generated by testing the SiW dataset according to the evaluation protocol is shown in Fig. 4. This is a kind of the heat map, where the more important the area is with the closer to red. In protocol 1, we focus on the entire face area for the real, the reflection of printed matter for print attack, and the reflection and noise of the display for display attack. Protocol 2 focuses on reflections and noise in the display display display as well as on the trained data for untrained display displays. Protocols 1 and 3 determined that the real was the spoofing attack and print attack was the real, respectively. In conclusion, in order to correctly identify the print attack in the proposed method, it is necessary to combine it with a method using depth features and a method that is robust to expression change.

V. Conclusion

In this paper, we proposed a spoofing detection method based on 3D CNN and deep metric learning. Through performance evaluation experiments using the SiW dataset, we demonstrated the effectiveness of a combination of 3D CNN, optical flow, and deep metric learning techniques. We also demonstrated that the proposed method, which requires 5 frames as input, is more accurate in display attack than the conventional method, which requires about 100 frames as

TABLE II
COMPARISON FOR THE NUMBER OF INPUT FRAMES.

Prot.	Method	# of frames	APCER [%]	BPCER [%]	ACER [%]
1	Baseline	2	1.83	37.1	19.5
		3	0.82	28.3	14.6
		5	0.81	28.3	14.6
	Baseline w/ FlowNetS	2	0.74	27.2	14.0
		3	0.76	27.3	14.0
		5	**0.73**	**26.2**	**13.5**
2	Baseline	2	1.12±0.40	1.06±0.41	1.09±0.40
		3	0.45±0.29	0.50±0.22	0.48±0.23
		5	0.45±0.28	0.46±0.22	0.46±0.23
	Baseline w/ FlowNetS	2	0.65±0.22	0.66±0.45	0.66±0.23
		3	0.39±0.30	0.58±0.19	0.48±0.24
		5	**0.37±0.30**	**0.43±0.10**	**0.40±0.19**
3	Baseline	2	46.35±8.61	0.90±0.32	23.63±4.44
		3	33.35±7.72	0.78±0.20	17.06±3.93
		5	24.29±6.89	**0.53±0.20**	12.96±3.53
	Baseline w/ FlowNetS	2	29.35±5.89	0.78±0.20	15.06±3.94
		3	22.82±5.45	0.62±0.16	11.72±3.04
		5	**21.91±5.27**	**0.61±0.16**	**11.26±2.90**

TABLE III
COMPARISON FOR DEEP METRIC LEARNING METHODS.

Protocol	Loss	APCER [%]	BPCER [%]	ACER [%]
1	Cross Entropy	0.83	45.1	26.7
	Cont	4.30	37.1	23.7
	Triplet	1.67	31.8	16.7
	ArcFace	**0.73**	**26.2**	**13.5**
	CosFace	0.88	26.3	13.6
	SphereFace	1.33	30.1	15.7
	ArcFace+Triplet	0.77	26.8	13.8
	ArcFace+Cont	2.44	46.2	24.3
2	Cross Enptropy	**0.22±0.33**	**0.35±0.13**	**0.33±0.18**
	Cont	1.22±0.37	1.11±0.20	1.17±0.29
	Triplet	1.01±0.39	1.00±0.40	1.01±0.40
	ArcFace	0.37±0.30	0.43±0.10	0.40±0.19
	CosFace	0.37±0.30	0.45±0.11	0.41±0.20
	SphereFace	0.77±0.28	0.99±0.20	0.88±0.25
	ArcFace+Triplet	0.44±0.33	0.62±0.20	0.53±0.28
	ArcFace+Cont	1.19±0.44	1.43±0.51	1.31±0.48
3	Cross Entropy	73.13±9.45	**0.39±0.24**	56.76±4.60
	Cont	26.22±6.27	1.00±0.21	13.61±3.22
	Triplet	28.97±8.27	1.01±0.33	15.00±4.28
	ArcFace	**21.91±5.27**	0.61±0.16	**11.26±2.90**
	CosFace	22.37±5.40	0.90±0.20	11.64±2.76
	SphereFace	22.50±6.27	0.61±0.16	11.26±3.20
	ArcFace+Triplet	21.91±6.02	0.61±0.16	11.26±3.08
	ArcFace+Cont	21.93±7.27	1.33±0.21	11.63±3.73

input. In the future, we will investigate the network architecture in combination with depth estimation and evaluate the performance of the data set with other spoofing attacks.

REFERENCES

[1] A.K. Jain, P. Flynn, and A.A. Ross, *Handbook of Biometrics*, Springer, 2008.

[2] S.Z. Li and A.K. Jain, *Handbook of Face Recognition*, Springer, 2011.

[3] T.F. Pereira, A. Anjos, J.M. De Martino, and S. Marcel, "Can face anti-spoofing countermeasures work in a real world scenario?," *Proc. Int'l Conf. Biometrics*, pp. 1–8, June 2013.

[4] I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," *Proc. IEEE Conf. Biometrics Special Interest Group*, pp. 1–7, Sept. 2012.

[5] J. Komulainen, A. Hadid, and M. Pietikäinen, "Context based face anti-spoofing," *Proc. Int'l Conf. Biometrics: Theory, Applications and Systems*, pp. 1–8, Sept. 2013.

[6] K. Patel, H. Han, and A.K. Jain, "Secure face unlock: Spoof detection on smartphones," *IEEE Trans. Information Forensics and Security*, vol. 11, no. 10, pp. 2268–2283, Oct. 2016.

[7] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face anti-spoofing using speeded-up robust features and fisher vector encoding," *IEEE Signal Processing Letters*, vol. 24, no. 2, pp. 141–145, Feb. 2016.

[8] L. Li, X. Feng, Z. Boulkenafet, Z. Xia, M. Li, and A. Hadid, "An original face anti-spoofing approach using partial convolutional neural network," *Int'l Conf. Image Processing Theory, Tools and Applications*, pp. 1–6, Dec. 2016.

[9] J. Yang, Z. Lei, and S.Z. Li, "Learn convolutional neural network for face anti-spoofing," *CoRR*, pp. 1–8, Aug. 2014.

[10] H. Patel, K.and Han and A.K. Jain, "Cross-database face antispoofing with robust feature representation," July 2016.

[11] J. Gan, S. Li, Y. Zhai, and C. Liu, "3D convolutional neural network based on face anti-spoofing," *Int'l Conf. Multimedia and Image Processing*, pp. 1–5, Mar. 2017.

[12] L. Feng, L.-M. Po, Y. Li, X. Xu, F. Yuan, T.C.-H. Cheung, and K.-W. Cheung, "Integration of image quality and motion cues for face anti-spoofing: A neural network approach," *J. Visual Communication and*

TABLE IV
COMPARISON WITH THE PROPOSED METHOD AND THE CONVENTIONAL METHOD.

Protocol	Method	# of frames	APCER [%]	BPCER [%]	ACER [%]
1	FAS-BAS[15]	about 100	3.58	**3.58**	**3.58**
	Baseline	5	**0.65**	32.6	16.6
	Baseline w/ FlowNetS	5	0.73	26.2	13.5
2	FAS-BAS[15]	about 100	0.57±0.69	0.57±0.69	0.57±0.69
	Baseline	5	**0.37±0.30**	0.43±0.12	0.40±0.20
	Baseline w/ FlowNetS	5	**0.37±0.30**	**0.43±0.10**	**0.40±0.19**
3	FAS-BAS[15]	about 100	**8.31±3.81**	8.31±3.80	**8.31±3.81**
	Baseline	5	24.29±6.89	**0.53±0.20**	12.96±3.41
	Baseline w/ FlowNetS	5	21.91±5.27	0.61±0.16	11.26±2.90

Image Representation, vol. 38, pp. 451–460, June 2016.

[13] G. Pan, L. Sun, Z. Wu, and S. Lao, "Eyeblink-based anti-spoofing in face recognition from a generic webcamera," Oct. 2007.

[14] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 6450–6459, June 2018.

[15] Y. Liu, A. Jourabloo, and X. Liu, "Learning deep models for face anti-spoofing: Binary or auxiliary supervision," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 389–398, June 2018.

[16] Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu, "Face anti-spoofing using patch and depth-based CNNs," Oct. 2017.

[17] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "ImageNet: Classification with deep convolutional neural networks," *Proc. Advances in Neural Information Processing Systems*, pp. 1097–1105, Dec. 2012.

[18] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P.V.D Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," *Proc. Int'l Conf. Computer Vision*, pp. 2758–2766, Dec. 2015.

[19] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 4690–4699, June 2019.

[20] L. Weiyang, W. Yandong, Y. Zhiding, L. Ming, R. Bhiksha, and S. Le, "SphereFace: Deep hypersphere embedding for face recognition," July 2017.

[21] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "CosFace: Large margin cosine loss for deep face recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 5265–5274, June 2018.

[22] Y. Nesterov, "A method of solving a convex programming problem with convergence rate O(1/k2)," *Soviet Mathematics Doklady*, vol. 27, no. 2, pp. 372–376, 1983.

[23] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," *CoRR*, vol. abs/1708.04896, pp. 1–10, 2017.

[24] R.R. Selvaraju, A. Cogswell, M.and Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," Oct. 2017.