Leveraging Intermediate Features of Vision Transformer for Face Anti-Spoofing

Mika Feng[1], Koichi Ito[1], Takafumi Aoki[1], Tetsushi Ohki[2], and Masakatsu Nishigaki[2]

[1] Graduate School of Information Sciences, Tohoku University, Japan

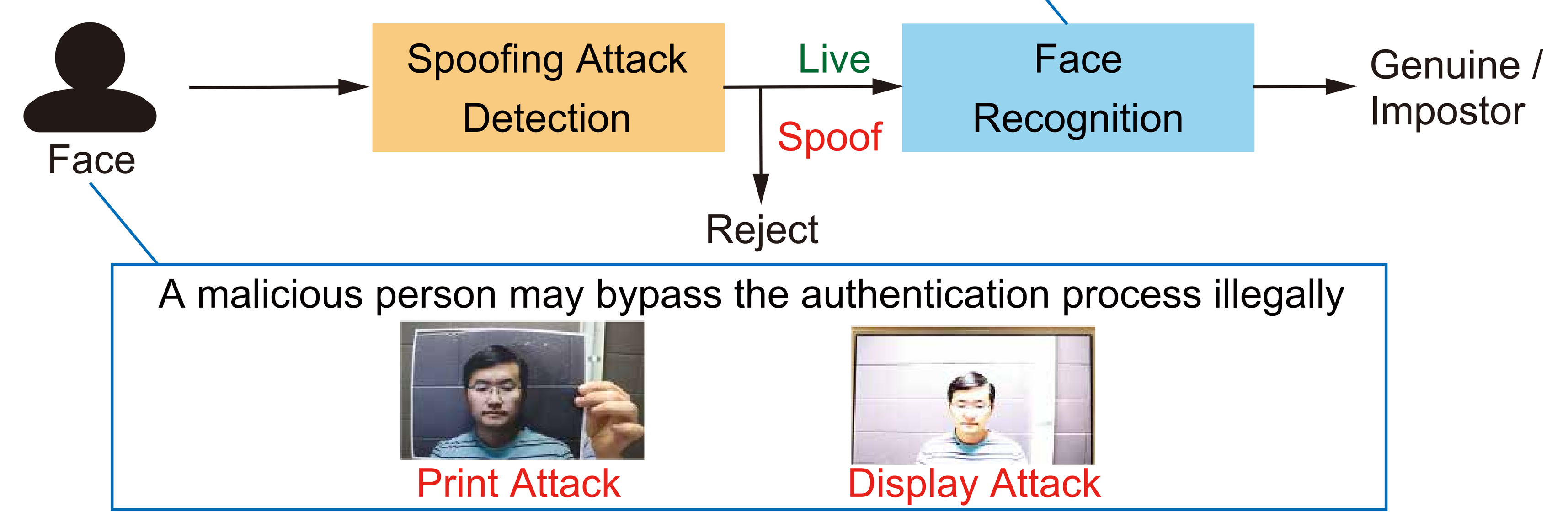
[2] Faculty of Informatics, Shizuoka University, Japan

CVPR Nashville JUNE 11-15, 2025

Background

Spoofing attacks need to be detected before face recognition

Face recognition systems are designed to be robust against changes in head pose, illumination, and blurring during image capture



To detect spoofing attacks, it is necessary to detect minute differences between the live and spoofed face image

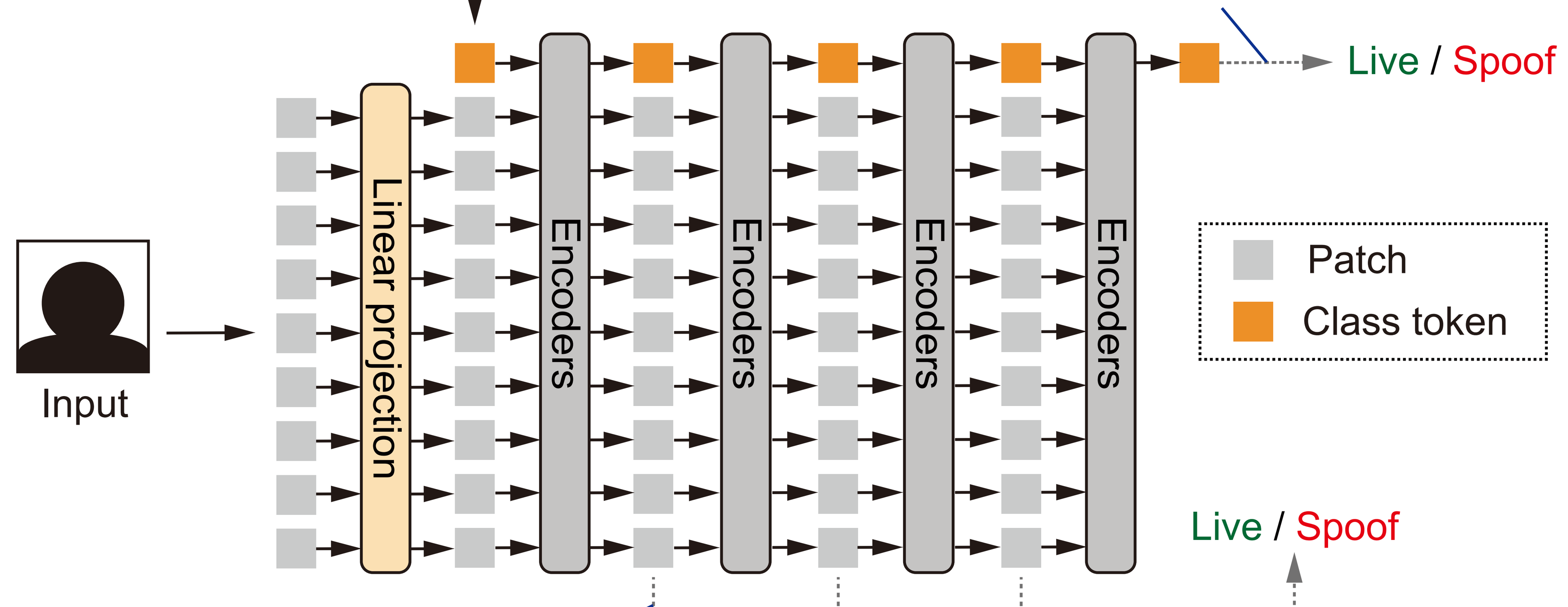
- It is necessary to extract features from the input image that can detect a global and/or local difference such as texture of paper or display device, reflection, interference fringes of display device, depth, etc

Motivation

ViT [1]-based methods are not specialized for detection of spoofing attacks

- Local features can be extracted in the shallow layer of ViT [1] and global features can be extracted in the deep layer of ViT [1]

Watanabe et al.' s method [2] and Chen et al.' s method [3] use only features extracted from the final layer of ViT [1] and do not consider local features

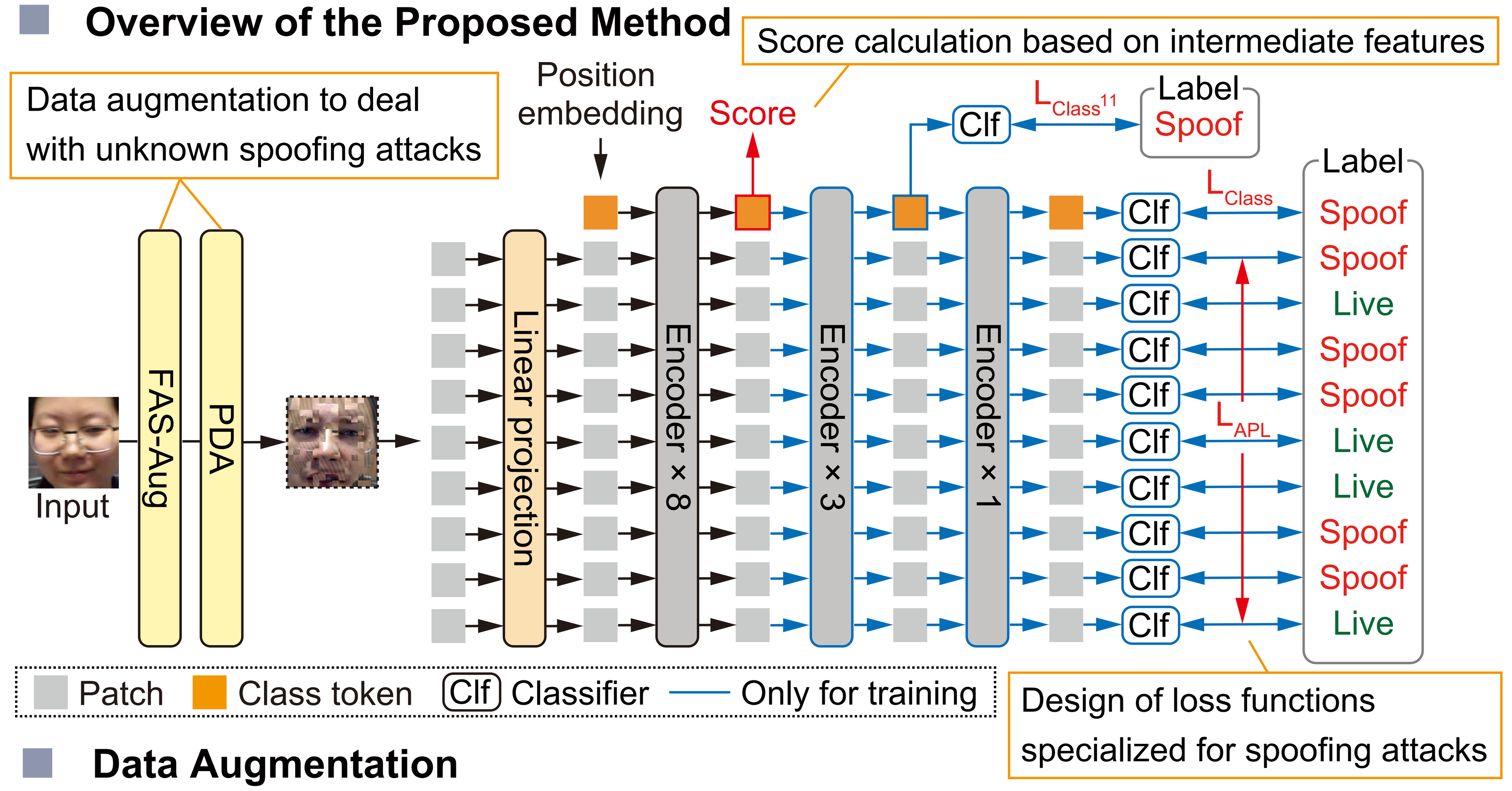


TransFAS [4] uses the intermediate features of ViT [1] , however, the location of the intermediate layer from which features are extracted has not been verified

The proposed method extracts features from the intermediate layer of ViT [1] that balance local and global features and have high generalization performance without overfitting the training data

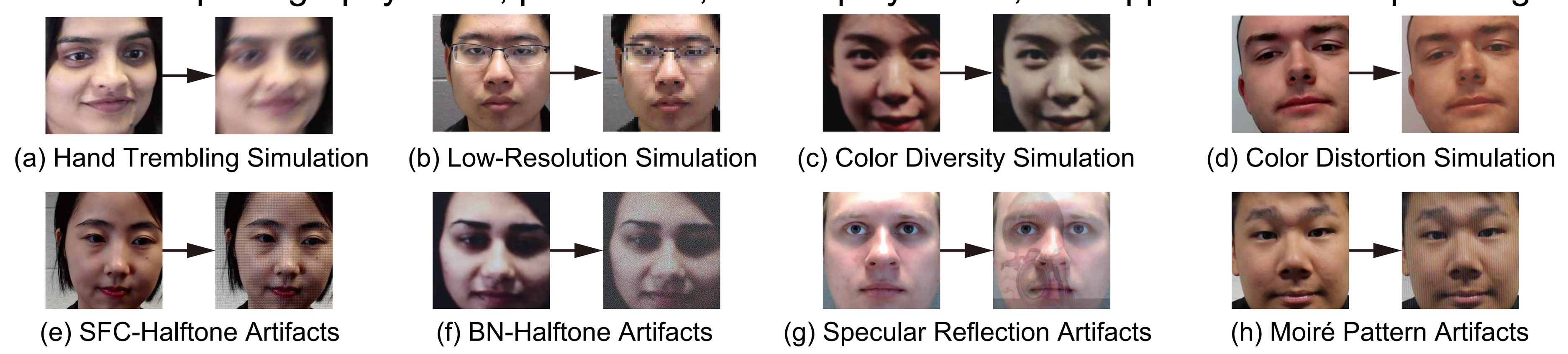
Methodology

Overview of the Proposed Method



Data Augmentation

- **FAS-Aug** [5] : The proposed method randomly selects one of the 8 data augmentations that emulate photography noise, print attack, and display attack, and applies it to the input image



(a) Hand Trembling Simulation (b) Low-Resolution Simulation (c) Color Diversity Simulation (d) Color Distortion Simulation

(e) SFC-Halftone Artifacts (f) BN-Halftone Artifacts (g) Specular Reflection Artifacts (h) Moiré Pattern Artifacts

- **PDA** [2] : Some patches of the spoofed image is replaced by patches of the live image, and the model is trained to recognize the patch of the spoofed image as “Spoof,” the replaced patch of the live image as “Live,” and the entire image as “Spoof”

Loss Function

- L_{Class} : Loss function used in standard ViT [1]-based classification
- L_{APL} [2] : Take into account patch-wise spoofing attack detection
- $L_{Class^{11}}$: Refine the features of the class token used to calculate the score

$$\mathrm{L2Softmax} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^{\top} f(x_i)+b_{y_i}}}{\sum_{j=1}^{C} e^{W_j^{\top} f(x_i)+b_j}},$$

$$\text{s.t. } \|f(x_i)\|_2 = \alpha, \quad \forall i = 1, 2, \ldots, N \quad \text{[6]}$$

Score Calculation

- After training, class tokens from the 8th encoder block of ViT are extracted from “Live” training images and used as reference vectors
- In inference, the class token from the 8th encoder block of ViT is compared to “Live” reference vectors, and the maximum cosine similarity is taken as the score
- If Score is greater than or equal to the threshold, the image is considered “Live,” otherwise, it is considered “Spoof”

Experiments

Ablation Study

Table 1. Ablation study when changing the features used for score calculation, where the values indicate ACER ↓ [%]

Features	OULU-NPU Prot. 4	SiW Prot. 3
Encoder Block #6	9.75±6.24	4.1±1.04
Encoder Block #7	10.42±6.95	3.07±0.08
Encoder Block #8	**8.83±7.51**	**2.2±0.57**
Encoder Block #9	12.46±8.02	3.45±1.7
Encoder Block #10	13.13±8.95	3.13±2.08
Encoder Block #11	13.46±8.83	3.37±2.43
Encoder Block #12	15.17±8.37	3.58±2.81
Normalization Layer	15.25±8.03	3.6±2.91

Table 2. Ablation study when changing the features used for loss calculation, where the values indicate ACER ↓ [%]

Features	OULU-NPU Prot. 4	SiW Prot. 3
Encoder Block #8	10.46±5.60	3.02±2.37
Encoder Block #9	6.54±5.09	3.07±1.74
Encoder Block #10	6.13±3.81	2.63±0.73
Encoder Block #11	**5.54±3.39**	**2.41±0.89**
Encoder Block #12	5.71±4.83	2.59±0.08

Table 3. Ablation study on loss functions, where the values indicate ACER ↓ [%]

L_{Class}	$L_{Class^{11}}$	L_{APL}	OULU-NPU Prot. 4	SiW Prot. 3
✓			9.46±8.01	1.73±0.33
✓		✓	8.83±7.51	2.2±0.57
✓	✓		9.0±6.0	**1.63±0.19**
✓	✓	✓	**5.54±3.39**	2.41±0.89

Table 4. Ablation study on data augmentation, where “ViT” indicates with and without using the intermediate features, and the values indicate ACER ↓ [%]

ViT	FAS-Aug	PDA	OULU-NPU Prot. 4	SiW Prot. 3
			15.25±8.03	3.6±2.91
		✓	10.42±6.76	3.08±2.57
	✓		7.63±5.19	4.08±3.49
	✓	✓	6.08±2.76	2.07±1.63
✓			5.54±3.39	2.41±0.89
✓		✓	5.5±4.15	2.0±0.45
✓	✓		**2.21±2.6**	1.34±0.58
✓	✓	✓	2.54±2.38	**0.83±0.13**

Comparison between Proposed and Conventional Methods

Table 5. Experimental results of each method for OULU-NPU (Unit: %).

Prot.	Method	APCER↓	BPCER↓	ACER↓
1	CDCN++	0.4	**0.0**	0.2
	NAS-FAS	0.4	**0.0**	0.2
	PatchNet	**0.0**	**0.0**	**0.0**
	Watanabe	3.7	2.4	3.0
	Chen	—	—	0.1
	TransFAS	0.8	**0.0**	0.4
	Li	0.4	**0.0**	0.2
	Proposed w/o PDA	1.21	1.08	1.15
	Proposed w/ PDA	0.92	0.5	0.71
2	CDCN++	1.8	0.8	1.3
	NAS-FAS	1.5	0.8	1.2
	PatchNet	1.1	1.2	1.2
	Watanabe	1.1	0.7	0.9
	Chen	—	—	1.1
	TransFAS	1.5	0.5	1.0
	Li	1.5	0.5	1.0
	Proposed w/o PDA	1.36	0.69	1.02
	Proposed w/ PDA	**0.69**	**0.39**	**0.54**
3	CDCN++	1.7±1.5	2.0±1.2	1.8±0.7
	NAS-FAS	2.1±1.3	1.4±1.1	1.7±0.6
	PatchNet	1.8±1.47	0.56±1.24	1.18±1.26
	Watanabe	1.2±1.0	0.7±1.0	1.0±1.0
	Chen	—	—	1.4±1.21
	TransFAS	**0.6±0.7**	1.1±2.5	0.9±1.1
	Li	0.7±0.9	1.1±2.7	0.9±1.3
	Proposed w/o PDA	1.39±0.67	0.86±0.52	1.13±0.57
	Proposed w/ PDA	0.85±0.62	**0.33±0.3**	**0.59±0.45**
4	CDCN++	4.2±3.4	5.8±4.9	5.0±2.9
	NAS-FAS	4.2±5.3	1.7±2.6	2.9±2.8
	PatchNet	2.5±3.81	3.33±3.73	2.9±3.0
	Watanabe	9.6±7.0	5.6±4.7	7.6±5.4
	Chen	—	—	2.8±2.6
	TransFAS	**2.1±2.2**	3.8±3.5	2.9±2.4
	Li	2.9±2.9	1.7±2.6	2.3±2.2
	Proposed w/o PDA	2.75±3.01	**1.67±2.25**	**2.21±2.6**
	Proposed w/ PDA	2.75±2.41	2.33±2.39	2.54±2.38

Table 6. Experimental results of each method for SiW (Unit: %).

Prot.	Method	APCER↓	BPCER↓	ACER↓
1	CDCN++	0.07	0.17	0.12
	NAS-FAS	0.07	0.17	0.12
	PatchNet	**0.00**	**0.00**	**0.00**
	Watanabe	0.11	0.08	0.10
	TransFAS	**0.00**	**0.00**	**0.00**
	MFAE	**0.00**	**0.00**	**0.00**
	Li	**0.00**	0.16	0.08
	Proposed w/o PDA	0.45	0.33	0.39
	Proposed w/ PDA	0.1	0.08	0.09
2	CDCN++	**0.00±0.00**	0.09±0.10	0.04±0.05
	NAS-FAS	**0.00±0.00**	0.09±0.10	0.04±0.05
	PatchNet	**0.00±0.00**	**0.00±0.00**	**0.00±0.00**
	Watanabe	0.01±0.01	0.01±0.01	0.01±0.01
	TransFAS	**0.00±0.00**	**0.00±0.00**	**0.00±0.00**
	MFAE	**0.00±0.00**	**0.00±0.00**	**0.00±0.00**
	Li	**0.00±0.00**	0.16±0.00	0.08±0.00
	Proposed w/o PDA	0.05±0.07	0.05±0.05	0.05±0.06
	Proposed w/ PDA	0.02±0.03	0.02±0.03	0.02±0.03
2	CDCN++	1.97±0.33	1.77±0.10	1.90±0.15
	NAS-FAS	1.58±0.23	1.46±0.08	1.52±0.13
	PatchNet	3.06±1.1	1.83±0.83	2.45±0.45
	Watanabe	3.07±2.75	3.07±2.75	3.07±2.75
	TransFAS	1.95±0.40	1.92±0.11	1.94±0.26
	MFAE	2.57±1.83	1.92±1.06	2.42±1.45
	Li	2.13±1.22	2.25±1.06	2.19±1.14
	Proposed w/o PDA	1.34±0.58	1.34±0.59	1.34±0.58
	Proposed w/ PDA	**0.83±0.13**	**0.84±0.14**	**0.83±0.13**

Conclusion

We proposed a face spoofing detection method using ViT's intermediate features and data augmentations, which may improve the accuracy of other detection approaches.

[1] A. Dosovitskiy et al., ICLR, 2021. [2] K. Watanabe et al., APSIPA, 2022.
[3] X. Chen et al., arXiv, 2023. [4] Z. Wang et al., IEEE T-BIOM, 2022.
[5] R. Cai et al., IJCV, 2024. [6] R. Ranjan et al., arXiv, 2017.