ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-6654-0540-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICASSP43922.2022.9747840

DEPTH-BASED ENSEMBLE LEARNING NETWORK FOR FACE ANTI-SPOOFING

Jie Jiang and Yunlian Sun[*]

School of Computer Science and Engineering, Nanjing University of Science and Technology

ABSTRACT

Although significant progress has been made in face anti-spoofing, current methods can only achieve satisfactory results under intra-dataset settings. In other words, they tend to perform poorly when suffering from unseen attacks. Previous methods try to extract a common feature space from multiple domains, but this idea is inefficient due to the enormous distribution difference among training domains. Unlike previous methods, we assume that the data distribution of the target domain will be similar to one of the training domains. Based on this hypothesis, we draw on the idea of ensemble learning and propose a generalized framework with multiple domain-specific modules. Given a test sample, the proposed framework allows it to dynamically choose which module to use based on its similarity to the training domains. In addition, we employ GCBlock to better mine face depth information for auxiliary supervision. Since fake information is spread throughout the image, we further introduce DropBlock to avoid overfitting. Extensive experiments on four public datasets show that our approach is practical.

Index Terms— Face Anti-spoofing, Ensemble Learning, Domain Generalization

1. INTRODUCTION

With the progress of science and technology, face recognition technology has been widely applied in our daily lives, such as access control systems, smartphone unlock, and account login. However, the indiscriminate use of face recognition technology has also brought about many security risks. A variety of presentation attacks (e.g. print attack, video replay, 3D mask attack) make face recognition systems extremely vulnerable. Researchers have proposed various approaches to tackle the above issue, roughly divided into texture-based and temporal-based methods. Texture-based methods aim to exploit differences in appearance between real and fake faces such as color [1], distortion cues [2], etc. Temporal-based methods are proposed to discover distinguishing features from consecutive frames, such as rPPG [3, 4].

Although current methods can achieve satisfactory performance in intra-dataset experiments, they cannot generalize well in cross-database experiments. This limitation is because the intrinsic distribution discrepancy of the training and test domains is large, and existing methods tend to discover dataset-biased cues. As a result, previous models do not have good generalization performance in application.

Quite a few methods attempt to exploit domain adaptation (DA) to make the models generalize well to unseen scenarios such as [5]. However, DA assumes that the training process can access the target domain data, which is unrealistic because we do not know what kind of attacks the security system will encounter. Based on the above discussion, we regard face anti-spoofing as a domain generalization (DG) problem. Compared with DA methods, DG aims to extract domain invariant features from multiple source domains without accessing target domain data. For example, Shao et al. [6] proposed a multi-adversarial discriminative deep domain generalization framework to learn a generalized feature space. Jia et al. [7] developed a single-side domain generalization framework to make the features of real faces indistinguishable.

Inspired by Mancini et al. [8], our work aims to address the problem of domain generalization for face anti-spoofing in an ensemble-learning framework. For face anti-spoofing, those features unique for each domain are as important as those domain invariant features. However, traditional methods focus more on extracting domain invariant features, making some private features important for classification abandoned. To fully leverage the information from multiple training domains, we assume that the data distribution of the test domain will be more similar to that of one of the training domains. In such a case, it will be unnecessary to balance multiple training domains for extracting domain invariant features, which is extremely difficult. Specifically, we incorporate several domain-specific modules in our proposed network. Each module focuses on learning knowledge from one specific training domain without considering the other domains. Given a test sample, we first estimate its similarity to each training domain and then fuse features from all modules by different similarity weights. In addition, we utilize face depth information as auxiliary supervision to explore more discriminative information following [4]. Furthermore, we find that it is not enough to establish depth maps by only convolution. Global context is also needed. Therefore, we inte-

[*] Yunlian Sun is the corresponding author (yunlian.sun@njust.edu.cn).

Thanks to the National Natural Science Foundation of China under Grant 62076131 for funding.

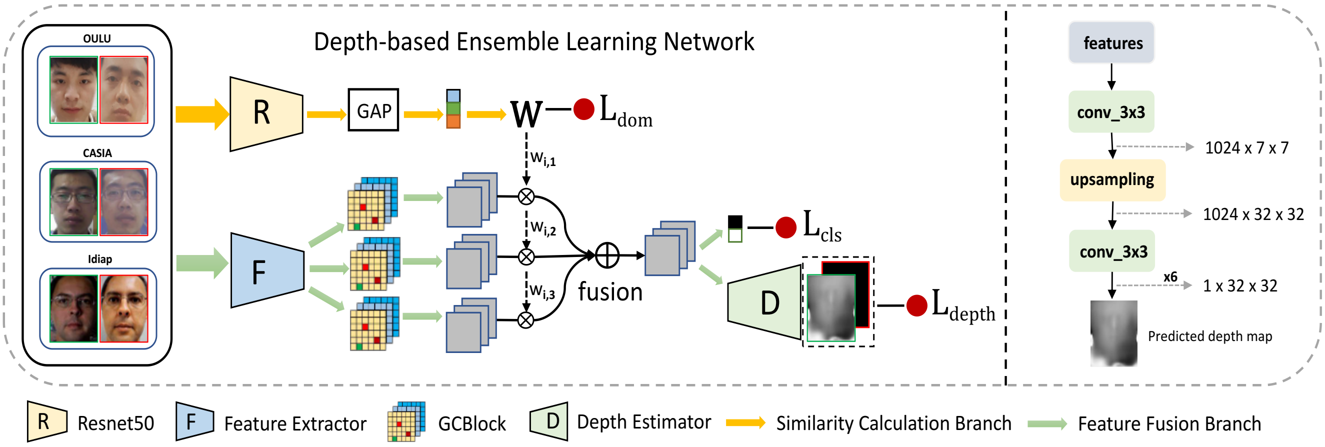

Fig. 1. An overview of our framework. The right half shows the network structure of our depth estimator, where we utilize bilinear interpolation for upsampling and then reduce the number of channels to 1 by multiple convolutions.

grate GCBlock in our framework, hoping to better mine depth information. Considering that fake information may exist in the entire image, DropBlock is further introduced to avoid overfitting.

2. METHOD

2.1. Problem Statement

This paper aims to develop a generalized face anti-spoofing framework to cope with various unseen attacks. Suppose we have access to N source domains in the face anti-spoofing task, denoted as $D = [D_1, D_2, ..., D_N]$ from two categories corresponding to attack and real, resp. Our work aims to extend the knowledge gained from multiple source domains to any unknown target domain without accessing the target domain. To this end, we randomly divide N source domains into $N-1$ training domains and one test domain. And we denote with $\{(x_i, y_i, d_i)\}_{i=1}^{M}$ all training images. Here, x_i is an image, y_i = 0/1 is the label of attack/real, $d_i \in \{1, ..., N-1\}$ indicates the domain that this image belongs to, M denotes the total number of training samples.

2.2. Ensemble Learning Network

Because of the distribution discrepancy among training domains, it is tough to align features from different domains. According to the assumption in the introduction, we conjecture that the similarities of the test domain to different training domains are different. In other words, we should give higher weights to training domains with higher similarities. In consideration of this, we design a neural network with multiple domain-specific modules corresponding to $N-1$ training domains, and leverage a parallel branch to estimate their corresponding similarities. The outputs of all domain-specific modules can then be fused with different similarity weights.

2.2.1. *Similarity Calculation Module*

Therefore, calculating the similarity weight between the test sample and each training domain is of the utmost importance. Fortunately, we have access to domain labels of images during the training phase, so we incorporate a parallel domain prediction branch into the network to identify which domain each sample belongs to. The domain prediction branch can map each input image to a weight vector $\mathbf{w_i} = [w_{i,1}, ..., w_{i,N-1}]^T$ where $0 < w_{i,j} < 1$ and $\sum_{j=1}^{N-1} w_{i,j} = 1$. Simply, we treat the optimization of this branch as a multi-classification problem with $N-1$ classes. Following [9], we adopt the label smoothing trick for the optimization. To be effective, we choose the complete Resnet50 [10] as the backbone and learn $\mathbf{w_i}$ by the following cross-entropy loss:

$$\mathcal{L}_{dom} = -\sum_{i=1}^{M}((1-\epsilon)\sum_{j=1}^{N-1}\mathbb{I}_{j=d_i}\log w_{i,j} + \frac{\epsilon}{N-1}\sum_{j=1}^{N-1}\mathbb{I}_{j\neq d_i}\log w_{i,j}) \qquad (1)$$

where ϵ controls the strength of the confidence penalty, $\mathbb{I}$ is the indicator function.

2.2.2. *Feature Fusion Module*

Specifically, we firstly feed samples into a feature extractor (denoted as F). For simplicity and efficiency, we choose the first three layers of Resnet50 as the feature extractor shared by all training domains. Afterwards, $N-1$ different domain-specific modules (denoted as $\{E_j\}_{j=1}^{N-1}$) are designed to extract corresponding domain-specific features from the output of F. Since features from different modules have different similarity weights, we adopt the weighted summation strategy for feature fusion:

$$z_i = \sum_{j=1}^{N-1} E_j(F(x_i))w_{i,j} \tag{2}$$

With the fused feature z_i, we further use 3×3 convolution and Global Average Pooling (GAP) to get corresponding logits from which we compute the predicted score vector $\hat{\mathbf{y}}$ by softmax. In the same way, we smooth the labels and calculate the cross-entropy loss $\mathcal{L}_{cls}$ based on the binary class labels y_i:

$$\begin{aligned}\mathcal{L}_{cls} = -\sum_{i=1}^{M}((1-\epsilon)\sum_{j=1}^{2}\mathbb{I}_{j=y_i}\log\hat{\mathbf{y}}_{i,j} \\ +\frac{\epsilon}{2}\sum_{j=1}^{2}\mathbb{I}_{j\neq y_i}\log\hat{\mathbf{y}}_{i,j})\end{aligned} \tag{3}$$

2.3. Depth Estimator and GCBlock

Suppose the vanilla ensemble-learning framework is applied to face anti-spoofing with only the supervision of binary class labels. In that case, we might get unsatisfactory performance since there is no guarantee that every domain-specific branch will be optimized well. An imbalance in the domain size may lead to instability. For example, the branch with less training data may be poorly optimized. In addition, if similarities of the test sample to all the training domains are actually not high, the vanilla ensemble-learning network will still have a risk of being biased and arbitrary. With these issues in mind, we further exploit face depth maps for auxiliary supervision. As shown in Fig. 1, a depth estimator (denoted as D) is designed to estimate face depth information. In order to obtain ground truth depth labels of real faces, we use PRNet [11] to estimate the depth map. For fake faces, we set all values of the depth map to zero.

Note that unlike binary classification supervision, depth supervision pays more attention to global information than local texture information. Therefore, we add GCBlock [12] to each domain-specific module to well model the global context. It should be noted that GCBlock can not only model the global context but also stay lightweight. Finally, we use $\mathcal{L}_2$ loss for constraint and optimization:

$$\mathcal{L}_{dep} = \sum_{i=1}^{M}||D(z_i) - I_i||^2 \tag{4}$$

where z_i indicates the outputs of the feature fusion module, I_i is the pre-calculated face depth maps of input face images.

2.4. Regularization Method and Loss Function

Given the particularity of face anti-spoofing, we notice spoof cues typically exist in all image positions. Instead, neural networks may focus on obvious features and ignore subtle clues. Moreover, we want our network to focus on neither face identity information nor leftover background information. To this end, we incorporate DropBlock [13] as a regularization method in our framework. Different from Dropout [14], DropBlock masks out local block regions by a certain probability such as eyes.

Integrating all things mentioned above together, the objective of our ensemble learning network for face anti-spoofing is:

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{dep} + \lambda\mathcal{L}_{dom} \tag{5}$$

where λ is the balanced parameter.

3. EXPERIMENTAL SETUP

3.1. Databases

Following [7], we utilize four public databases to evaluate our model: OULU-NPU [15] (denoted as ”O”), CASIA-FASD [16] (”C”), Idiap Replay-Attack [17] (”I”), and MSU-MFSD [2] (”M”). Then, we randomly select three databases as training domains and the remaining as the target domain to evaluate the generalization ability of the proposed model. Finally, we have a total of four test tasks: O&C&I to M, O&M&I to C, O&C&M to I, and I&C&M to O. In this case, we can simulate complex domain shift scenarios in the real world to the greatest extent possible.

3.2. Implementation Details

We choose Centerface [18] for face detection and face alignment, and first resize all detected faces to 224×224. We use only one frame randomly selected from each video for training while two frames for test. The SGD optimizer with momentum of 0.9 and weight decay of 5e-4 is utilized for training. The balanced coefficient λ,learning rate and ϵ are set to 0.5, 0.01 and 0.1, resp. Following [6], we choose the Half Total Error Rate (HTER) and the Area Under Curve (AUC) as the evaluation metrics. To be specific, HTER is half of the summation of false acceptance rate and false rejection rate. It is important to note that while DropBlock can make the network discover as many clues as possible, it may bring about some instability due to the random erasing operation (with a probability of 0.35). To this end, we repeat all experiments five times and take their average as the final result.

4. RESULTS

4.1. Comparison with State-of-the-art Methods

We compare the proposed network with several state-of-the-art methods including Image Distortion Analysis (IDA) [2], Color Texture [1], Auxiliary [4], SSDG-M [7] and MADDG [6]. In addition, two meta-learning approaches are also selected for comparison: RFM [19], SDA [20]. For Auxiliary

Table 1. Comparison to existing methods on four test settings. Bold indicates the best performance. Under-line denotes the second best performance.

Methods	O&C&I to M		O&M&I to C		O&C&M to I		I&C&M to O	
	HTER(%)	AUC(%)	HTER(%)	AUC(%)	HTER(%)	AUC(%)	HTER(%)	AUC(%)
IDA (TIFS2015)	66.27	27.86	55.17	39.05	28.35	78.25	54.20	44.59
Color Texture (TIFS2017)	28.09	78.47	30.58	76.89	40.40	62.78	63.59	32.71
Auxiliary (CVPR2018)	22.7	85.8	33.5	73.1	29.1	71.6	30.1	77.6
MADDG (CVPR2019)	17.6	88.0	24.5	84.5	22.1	84.9	27.9	80.0
SSDG- M (CVPR2020)	16.67	90.47	23.11	85.45	18.21	**94.61**	25.17	81.83
RFM (AAAI2020)	13.89	93.98	20.27	**88.16**	17.30	90.48	**16.45**	**91.16**
SDA (AAAI2021)	15.4	91.8	24.5	84.4	15.6	90.1	23.1	84.3
Ours	**8.57**	**95.01**	**20.26**	85.80	**13.52**	93.22	20.22	88.48

Table 2. Ablation study results of our proposed network on four test settings.

Methods	O&C&I to M		O&M&I to C		O&C&M to I		I&C&M to O	
	HTER(%)	AUC(%)	HTER(%)	AUC(%)	HTER(%)	AUC(%)	HTER(%)	AUC(%)
Ours_wo/depth	9.76	94.30	20.72	84.06	20.90	83.26	25.86	82.02
Ours_wo/GCBlock	10.71	93.28	20.72	83.65	20.50	80.37	20.78	87.48
Ours_wo/DropBlock	11.54	94.06	20.63	83.36	20.95	78.27	23.11	85.41
Ours	**8.57**	**95.01**	**20.26**	**85.80**	**13.52**	**93.22**	**20.22**	**88.48**

[4], we choose the version with only depth estimation component available (without rPPG), for a fair comparison.

From Table 1, our method outperforms most previous methods. Unlike meta-learning, our learning strategy does not need to sample many training and test tasks from source domains, which reduces the computation cost to some extent. Moreover, our framework is more interpretable and less dependent on network structures. In particular, our method achieves the best performance on O&C&I to M. We believe the reason is that the data distribution of the target domain is highly similar to that of a training domain. This significantly proves that our hypothesis is reasonable.

4.2. Ablation Study

In order to verify the contribution of each component, we further perform several ablation studies. Ours_wo/depth, Ours_wo/GCBlock, and Ours_wo/DropBlock denote the proposed framework without the depth supervision, GCBlock, and DropBlock, resp. As shown in Table 2, all performance degrades in different degrees with any of the three components excluded, which validates the effectiveness of each part of our proposed framework.

4.3. Visualization

We adopt Grad-CAM [21] to compute the class activation map of our method. As shown in Fig. 2, each module of our network focuses on different regions of the face to seek discriminative cues, while the fusion module selects features based on similarity weights. In fact, most test samples have high prediction scores for one specific domain. That is, they are highly similar to one of the training domains. For example, in Fig. 2, the upper sample is highly similar to the first domain, while the lower sample is highly similar to the third.

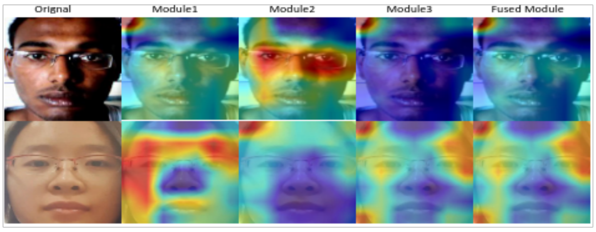

Fig. 2. Visualization of the proposed method. The first column shows the original image. The three middle columns present CAMs obtained from the three domain-specific modules. The last column illustrates the CAM from the fusion module.

5. CONCLUSION

To improve the generalization of face anti-spoofing, we developed a depth-based ensemble learning network to take full advantage of both domain private and agnostic features. The network is comprised of multiple domain-specific modules, and each test sample selects desired features based on its similarity to each training domain. Besides, a depth estimator and DropBlock were incorporated to explore more useful clues. Experiments on four public databases demonstrate that our network is effective and achieves promising results.

6. REFERENCES

[1] 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, 2016.

[2] 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.

[3] Siqi Liu, Pong C Yuen, Shengping Zhang, and Guoying Zhao, "3d mask face anti-spoofing with remote photoplethysmography," in *European Conference on Computer Vision*. Springer, 2016, pp. 85–100.

[4] Yaojie Liu, Amin Jourabloo, and Xiaoming Liu, "Learning deep models for face anti-spoofing: Binary or auxiliary supervision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 389–398.

[5] Haoliang Li, Wen Li, Hong Cao, Shiqi Wang, Feiyue Huang, and Alex C Kot, "Unsupervised domain adaptation for face anti-spoofing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1794–1809, 2018.

[6] Rui Shao, Xiangyuan Lan, Jiawei Li, and Pong C Yuen, "Multi-adversarial discriminative deep domain generalization for face presentation attack detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10023–10031.

[7] Yunpei Jia, Jie Zhang, Shiguang Shan, and Xilin Chen, "Single-side domain generalization for face anti-spoofing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8484–8493.

[8] Massimiliano Mancini, Samuel Rota Bulo, Barbara Caputo, and Elisa Ricci, "Best sources forward: domain generalization through source-specific nets," in *2018 25th IEEE international conference on image processing (ICIP)*. IEEE, 2018, pp. 1353–1357.

[9] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[11] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou, "Joint 3d face reconstruction and dense alignment with position map regression network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 534–551.

[12] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu, "Gcnet: Non-local networks meet squeeze-excitation networks and beyond," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[13] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le, "Dropblock: A regularization method for convolutional networks," *arXiv preprint arXiv:1810.12890*, 2018.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[15] Zinelabinde Boulkenafet, Jukka Komulainen, Lei Li, Xiaoyi Feng, and Abdenour Hadid, "Oulu-npu: A mobile face presentation attack database with real-world variations," in *2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017)*. IEEE, 2017, pp. 612–618.

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

[17] Ivana Chingovska, André Anjos, and Sébastien Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in *BIOSIG*. IEEE, 2012, pp. 1–7.

[18] Yuanyuan Xu, Wan Yan, Genke Yang, Jiliang Luo, Tao Li, and Jianan He, "Centerface: joint face detection and alignment using face as point," *Scientific Programming*, vol. 2020, 2020.

[19] Rui Shao, Xiangyuan Lan, and Pong C Yuen, "Regularized fine-grained meta face anti-spoofing," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 11974–11981.

[20] Jingjing Wang, Jingyi Zhang, Ying Bian, Youyi Cai, Chunmao Wang, and Shiliang Pu, "Self-domain adaptation for face anti-spoofing," *arXiv preprint arXiv:2102.12129*, 2021.

[21] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.