End-to-End Reconstruction-Classification Learning for Face Forgery Detection Supplementary Material

Junyi Cao[1] Chao Ma[1*] Taiping Yao[2] Shen Chen[2] Shouhong Ding[2] Xiaokang Yang[1]
[1] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[2] Youtu Lab, Tencent
{junyicao, chaoma, xkyang}@sjtu.edu.cn {taipingyao, kobeschen, ericshding}@tencent.com

In this supplementary material, we supplement more details for our proposed reconstruction-classification learning (RECCE) framework. Concretely, Section 1 provides the detailed implementation of our method. Section 2 presents additional experiments to better demonstrate the superiority of the proposed method, including more ablation studies, generalization, and robustness evaluation. In Section 3, we analyze the limitation of our method.

1. More Implementation Details

For the FaceForensics++ [11], Celeb-DF [7] and DFDC [3] datasets, we use RetinaFace [2] to extract the face region from video sequences and adopt a conservative crop which enlarges the facial region by a factor of 1.3 around the center of the tracked face. Then, we resize the aligned face images to 299 × 299. For WildDeepfake [17] where facial images are already cropped to 224 × 224, we use the original setting as in [17]. We implement our method based on Xception [1] structure. The decoder in the proposed reconstruction network is built upon the fourth Xception Block. We use the nearest up-sampling in the decoder to restore the spatial size. For detailed implementation, please refer to our source code. During training, we apply random horizontal flipping as data augmentation. We use Adam [5] optimizer with an initial learning rate of 2e-4 and a weight decay of 1e-5. A step learning rate scheduler is used to adjust the learning rate. Following [12], we additionally use the loss term in [4] with $b = 0.04$ to stabilize training of the proposed method. Both our method and the re-implemented approaches are based on PyTorch [9]. All the experiments are conducted on 8 Nvidia GeForce 2080 Ti GPUs.

2. Additional Experiments

2.1. Ablation Study

Effect of the proposed constraints on generalization. We focus on the common compact representations for genuine faces via the proposed constraints depicted in Section 3.1 of the manuscript, aiming to ensure the generalization of our method to unseen forgeries. In Section 4.3 of the manuscript, we validate the effectiveness of the proposed constraints under within-dataset evaluation. To investigate the effect of these constraints on generalization, we conduct an ablation study by training on WildDeepfake [17] and testing on FF++ [11]. The results are shown in Table 1. Comparing (a) and the proposed approach RECCE, we can see that reconstruction over only real faces significantly outperforms reconstruction over both the real and fake faces on the cross-testing evaluation. This affirms that exploring the common characteristics of genuine faces is more suitable than overfitting specific forgery patterns presented in the training set. Comparing (b) and our method RECCE, we observe that the proposed metric-learning loss yields a 1.24% AUC gain, which shows that it is beneficial to model the distribution of real samples while strengthening the difference between real and fake faces in the embedding space.

ID	$\mathcal{L}_\mathrm{r}$	$\mathcal{L}_\mathrm{m}$	AUC (%) ↑	EER (%) ↓
(a)	real & fake	√	56.40	46.15
(b)	real	–	58.62	43.34
RECCE	real	√	**59.86**	**42.61**

Table 1. Effectiveness of the proposed constraints in our method by training on WildDeepfake [17] and testing on FF++ [11].

ID	Operator	Multi-scale	Acc (%)	AUC (%)
(a)	summation	√	81.51	90.35
(b)	concatenation	√	82.62	90.81
(c)	non-local [14]	√	83.06	90.95
(d)	graph-based	–	82.85	90.84
RECCE	graph-based	√	**83.25**	**92.02**

Table 2. Effectiveness of the proposed graph-based operator and multi-scale structure in multi-scale graph reasoning module on WildDeepfake [17] dataset.

[*] Corresponding author.

Methods	FF++		Celeb-DF		DFDC	
	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓
Xception [11]	51.50	48.88	53.95	46.91	58.58	44.39
RFM [12]	56.70	45.27	59.53	42.94	61.54	41.70
Add-Net [17]	56.61	44.47	68.50	35.76	63.21	40.48
F^3-Net [10]	57.80	43.99	64.21	39.70	62.07	41.44
MultiAtt [15]	**60.69**	42.74	80.37	27.00	65.93	38.68
RECCE (Ours)	59.86	**42.61**	**84.79**	**23.12**	**69.87**	**35.86**

Table 3. Cross-dataset evaluation in terms of AUC (%) and EER (%) by training on WildDeepfake [17]. Our method generally outperforms other approaches on unseen forgeries.

Study on multi-scale graph reasoning. We study the effectiveness of the multi-scale graph reasoning module described in Section 3.2 of the manuscript. Specifically, we conduct ablation experiments on the aggregation operator and the multi-scale structure of multi-scale graph reasoning. The results are shown in Table 2. Note that we use the bilinear interpolation to keep summation and concatenation properly in the spatial dimension. From Table 2, we observe that the non-local [14] operator performs better than simple summation and concatenation. This means that simple operators are not sufficient to aggregate the discrepancy features embedded in the decoder blocks. Comparing with Table 2(c), our method achieves an AUC improvement of 1.07%. This is mainly because the non-local operator models dense correspondence between the encoder output and the decoder features, which may aggregate confounding factors from the irrelevant regions to hinder the reasoning process. Instead, our proposed graph reasoning keeps the spatial correspondence and is more effective in scoring the discrepancy features from the decoder to enhance the encoder output for better reasoning about forgery cues. In comparison with the variant (d) which only considers single-scale information, our method achieves a 1.18% AUC gain. The improvement validates the effectiveness of the multi-scale designing of our multi-scale graph reasoning module. These results verify the superiority of the proposed graph reasoning with the multi-scale structure.

2.2. Generalization Evaluation

In Section 4.2 of the manuscript, we conduct the cross-dataset evaluation by training on FF++ and testing on the other three datasets. In this part, we provide more experimental results on cross-dataset evaluation to better demonstrate the generalization ability of our approach. Specifically, we train our approach and existing methods on WildDeepfake [17] and then test them on FF++ [11], Celeb-DF [7], and DFDC [3]. The results are shown in Table 3. We can see that RECCE achieves better generalization performance compared with other competitors. Take the challenging DFDC [3] as an example, where the fake images are generated by various manipulations with different perturbations, our method yields a 3.94% AUC gain while decreases EER by 2.82% compared with MultiAtt [15].

Methods	Train	DF	F2F	FS	NT	Cross Avg.
Xception [11]	DF	98.44	66.21	68.67	66.79	67.22
RFM [12]		98.80	65.18	72.69	63.44	67.10
Add-Net [17]		98.04	68.67	68.61	**68.36**	68.55
Freq-SCL [6]		98.91	58.90	66.87	63.61	63.13
MultiAtt [15]		99.51	66.41	67.33	66.01	66.58
RECCE (Ours)		**99.65**	**70.66**	**74.29**	67.34	**70.76**
Xception [11]	F2F	72.93	96.21	64.26	70.48	69.22
RFM [12]		67.80	96.44	64.67	64.55	65.67
Add-Net [17]		70.24	96.35	59.54	69.74	66.51
Freq-SCL [6]		67.55	93.06	55.35	66.66	63.19
MultiAtt [15]		73.04	97.96	**65.10**	71.88	70.01
RECCE (Ours)		**75.99**	**98.06**	64.53	**72.32**	**70.95**
Xception [11]	FS	79.54	62.88	97.02	56.46	66.29
RFM [12]		81.34	61.53	98.26	55.02	65.96
Add-Net [17]		72.82	59.50	97.56	53.10	61.81
Freq-SCL [6]		75.90	54.64	98.37	49.72	60.09
MultiAtt [15]		82.33	61.65	**98.82**	54.79	66.26
RECCE (Ours)		**82.39**	**64.44**	**98.82**	**56.70**	**67.84**
Xception [11]	NT	74.50	78.23	60.19	87.67	70.97
RFM [12]		75.39	72.24	62.83	85.51	70.15
Add-Net [17]		77.55	75.42	54.30	84.96	69.09
Freq-SCL [6]		**79.09**	74.21	53.99	88.54	69.10
MultiAtt [15]		74.56	80.61	60.90	93.34	72.02
RECCE (Ours)		78.83	**80.89**	**63.70**	**93.63**	**74.47**

Table 4. Cross-testing in terms of AUC (%) on different manipulation techniques of FF++ [11]. Gray background indicates intra-testing results. Our method generalizes better on unseen forgery types compared with other approaches.

Methods	Compress	Blur	Contrast	Saturate	Pixelate	Avg.
Xception [11]	91.66	69.12	97.23	98.35	84.29	88.13
RFM [12]	91.27	58.51	96.49	98.62	84.50	85.88
Add-Net [17]	86.10	69.29	97.77	95.42	75.89	84.89
F^3-Net [10]	89.00	47.63	96.83	97.51	76.23	81.44
MultiAtt [15]	90.70	69.68	98.71	99.21	85.71	88.80
RECCE (Ours)	**91.75**	**76.92**	**98.87**	**99.22**	**90.13**	**91.38**

Table 5. Robustness evaluation in terms of AUC (%) on FF++ c23 [11] dataset. "Avg." indicates the mean score. Our framework consistently outperforms other competitors under different interference.

Furthermore, we supplement more results for Table 4 displayed in the manuscript. The complete comparison results are shown in Table 4, from which we observe that RECCE consistently outperforms the state-of-the-art approaches on unseen forgery types. These additional experiments demonstrate that it is more suitable to explore the common features of real faces rather than overfitting specific forgery patterns presented in the training samples.

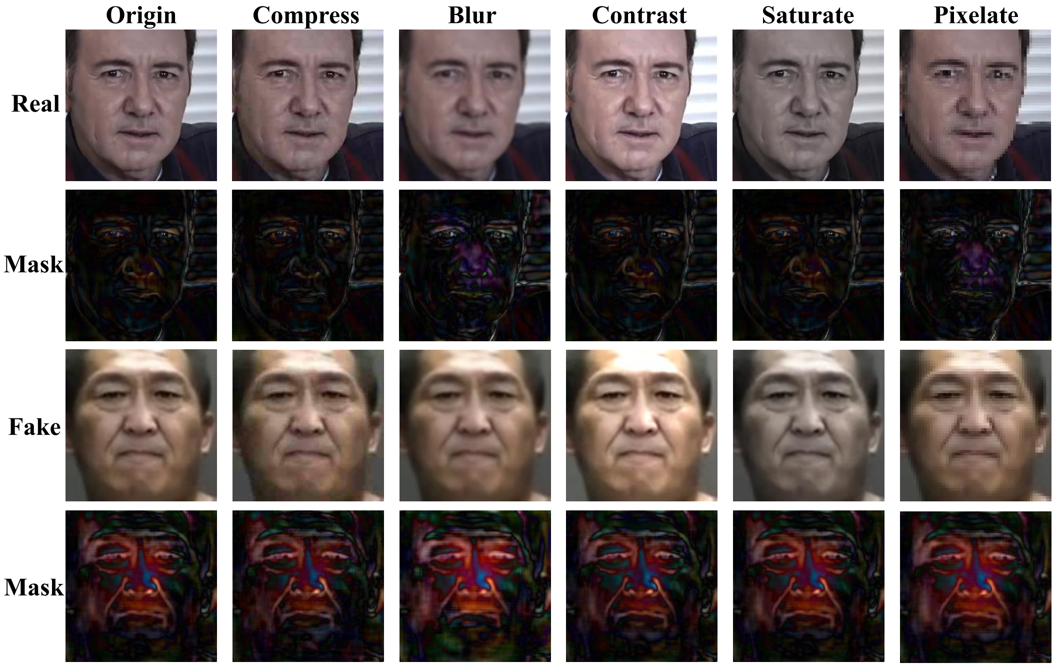

Figure 1. The visualization of the difference masks of the proposed method on WildDeepfake [17] dataset under various perturbations. We can see that our method produces consistent difference masks across the listed perturbations for real and fake faces, respectively. This implies that our method effectively captures the essential discrepancies between real and fake samples. Best viewed in color.

2.3. Robustness Evaluation

In Section 4.4 of the manuscript, we demonstrate the robustness of the proposed method under several unseen perturbations on WildDeepfake [17]. Here we additionally use FF++ c23 [11] (*i.e.*, high-quality) as training data to further justify the robustness of the proposed method. The results are shown in Table 5. We can see that our proposed framework RECCE consistently outperforms other competitors under the considered perturbations, often by a large margin. For instance, our approach achieves an AUC gain of 4.42% compared with the second best method (*i.e.*, MultiAtt [15]) under pixelation. Moreover, to understand the robustness of RECCE in an intuitive way, we visualize the difference masks obtained from the reconstruction guided attention module under different interference in Figure 1. We observe that our method generates consistent difference masks across various distortions for genuine and forged samples respectively, which implies that our method can effectively capture the essential discrepancies between real samples and fake samples even with unseen perturbations. The visualization, from another viewpoint, demonstrate the robustness of our proposed method.

3. Limitation

In this work, we focus on modeling the distribution of real faces to separate forged images from genuine ones. Thus, when the real faces in the training dataset are severely biased, *e.g.*, only includes real faces for a single gender, race, or age group, our method may yield a relatively high false rejection rate. We think that using larger databases for genuine faces like [8, 13, 16] to train the reconstruction network could relieve this problem in the future.

References

[1] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017.

[2] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. RetinaFace: Single-stage dense face localisation in the wild. *arXiv preprint arXiv:1905.00641*, 2019.

[3] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) dataset. *arXiv preprint arXiv:2006.07397*, 2020.

[4] Takashi Ishida, Ikko Yamane, Tomoya Sakai, Gang Niu, and Masashi Sugiyama. Do we need zero training loss after achieving zero training error? In *ICML*, 2020.

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[6] Jiaming Li, Hongtao Xie, Jiahong Li, Zhongyuan Wang, and Yongdong Zhang. Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection. In *CVPR*, 2021.

[7] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-DF: A large-scale challenging dataset for deepfake forensics. In *CVPR*, 2020.

[8] Aaron Nech and Ira Kemelmacher-Shlizerman. Level playing field for million scale face recognition. In *CVPR*, 2017.

[9] Adam Paszke, S. Gross, Soumith Chintala, G. Chanan, E. Yang, Zachary Devito, Zeming Lin, Alban Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017.

[10] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *ECCV*, 2020.

[11] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *ICCV*, 2019.

[12] Chengrui Wang and Weihong Deng. Representative forgery mining for fake face detection. In *CVPR*, 2021.

[13] Fei Wang, Liren Chen, Cheng Li, Shiyao Huang, Yanjie Chen, Chen Qian, and Chen Change Loy. The devil of face recognition is in the noise. In *ECCV*, 2018.

[14] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[15] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deepfake detection. In *CVPR*, 2021.

[16] Zheng Zhu, Guan Huang, Jiankang Deng, Yun Ye, Junjie Huang, Xinze Chen, Jiagang Zhu, Tian Yang, Jiwen Lu, Dalong Du, and Jie Zhou. WebFace260M: A benchmark unveiling the power of million-scale deep face recognition. In *CVPR*, 2021.

[17] Bojia Zi, Minghao Chang, Jingjing Chen, Xingjun Ma, and Yu-Gang Jiang. WildDeepfake: A challenging real-world dataset for deepfake detection. In *ACM MM*, 2020.