Supplementary Material for
Generalizing Face Forgery Detection with High-frequency Features

1. Overview

In this supplementary material, we present more implementation details and experimental results, including

- Detailed network architecture (see Sec. 2).
- Detailed implementation settings (see Sec. 3).
- More experimental results (see Sec. 4), including additional ablation studies on FF++ and comparisons with other state-of-the-art methods.

2. The Network Architecture

2.1. Details of the proposed modules

SRM kernels. Following [17], we adopt three commonly used kernels from the original SRM bands [8], and the weights are presented in Fig. 1.

$$\frac{1}{4}\begin{bmatrix}0&0&0&0&0\\0&-1&2&-1&0\\0&2&-4&2&0\\0&-1&2&-1&0\\0&0&0&0&0\end{bmatrix} \quad \frac{1}{12}\begin{bmatrix}-1&2&-2&2&-1\\2&-6&8&-6&2\\-2&8&-12&8&-2\\2&-6&8&-6&2\\-1&2&-2&2&-1\end{bmatrix} \quad \frac{1}{2}\begin{bmatrix}0&0&0&0&0\\0&0&0&0&0\\0&1&-2&1&0\\0&0&0&0&0\\0&0&0&0&0\end{bmatrix}$$

Figure 1: The adopted three SRM kernels.

SRM convolution blocks. We design two SRM-based convolution blocks, *i.e.*, the SRM Block (see Fig. 2) and the Separable SRM Block (see Fig. 3).

The SRM Block is a conventional convolution block, except that its weight is fixed as the presented three kernels. We apply it to the raw RGB image, and the output size is the same as the input image. To capture the high-frequency patterns at different scales and compose more abundant information, we devise the Separable SRM Block to process each feature map separately. We use 1×1 convolution to recover the channel dimension. Following [15], we employ a truncated linear unit as the activation function.

Attention-related modules. Fig. 4 shows the dual cross-modality attention block in detail. The residual guided spatial attention block and the channel attention based fusion block are presented in Fig. 5 and Fig. 6, respectively.

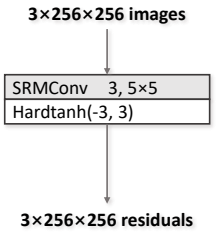

Figure 2: The SRM Block. A hardtanh layer is used for non-linear activation of the high-frequency features.

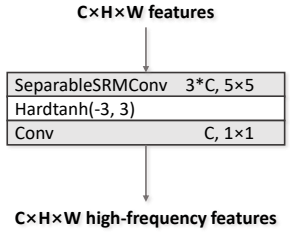

Figure 3: The Separable SRM Block. We extract high-frequency features for each channel.

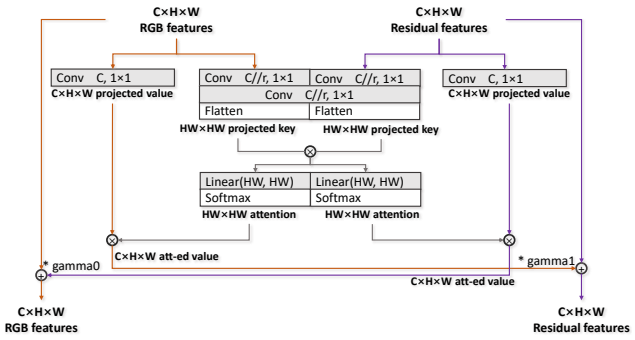

Figure 4: The Dual Cross-Modality Attention Block. The attention map is calculated based on the spatial correlation between the two modalities. Features from one modality are strengthened by those from the other modality.

2.2. Detailed architecture of the proposed model

Fig. 7 illustrates the proposed two-stream model in detail. We employ two modified Xception models as the backbone to process the RGB image and the high-frequency noises extracted by the SRM Block separately.

In the entry flow, the initial noise features are strengthened by high-frequency features from two separable SRM convolution blocks at different scales, which constitute the multi-scale high-frequency feature extraction module. Meanwhile, low-level features in the RGB stream are calibrated under the guidance of the residual guided spatial attention, which helps focus more on the forgery traces.

In the middle flow, we place two DCMA blocks to model the correlation and interaction between the regular RGB modality and the novel high-frequency modality.

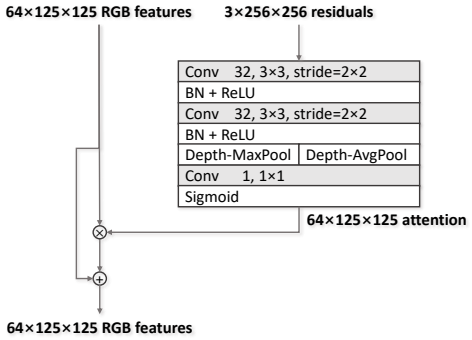

Figure 5: The Residual Guided Attention Block. Attention map calculated from high-frequency features is exploited to guide the feature extraction in the RGB modality.

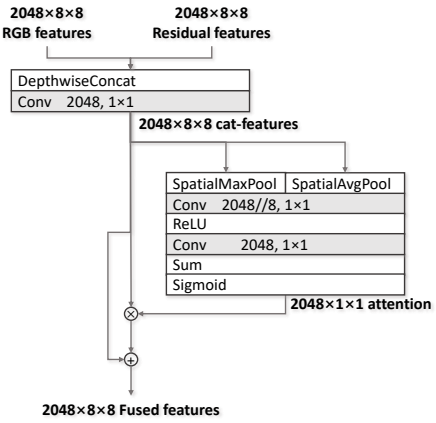

Figure 6: The Fusion Block. The concatenated high-level features are forwarded to a Channel Attention Block for further calibration.

In the exit flow, we fuse the features from the two modalities in an attention-based manner. Specifically, we first stack features in the depth direction and then apply the channel-wise attention to get the output features. Note that we place a dropout layer before the final classification layer to reduce overfitting.

3. Implementation Details

Following FF++ [13], we extract 270 frames from each training video to construct the training set. The extracted and aligned face images are resized to 256×256 and then normalized to $[-1, 1]$, with no data augmentation applied. We implement the model in PyTorch [5]. Adam optimizer is set with the learning rate of 0.0002 and the weight decay rate of 0.0005. We set the batch size to 32 in both the training and testing phases. The training loss converges after around 120k iterations. We compute the AM-Softmax loss with cosine margin under the default hyper-parameter setting ($\gamma = 0.0$, $m = 0.5$, $s = 30$, $t = 1.0$). Two Nvidia Tesla M40 GPUs are used in each experiment.

Table 1: More ablation studies on FF++. Results colored in gray indicate the within-database performance.

Training	Method	Testing AUC			
		DF	F2F	FS	NT
DF	RGB	0.993	0.736	0.485	0.736
	SRM	0.992	0.701	0.445	0.759
	Two-stream Fusion (Fusion)	0.993	0.758	0.472	0.761
	Fusion + RSA	0.992	0.753	0.454	0.778
	Fusion + RSA + DCMA	0.992	0.760	0.466	0.766
	Fusion + RSA + DCMA + Multi-scale	0.992	**0.764**	**0.497**	**0.814**
FS	RGB	0.664	0.889	0.994	0.713
	SRM	0.627	0.901	0.995	0.878
	Two-stream Fusion (Fusion)	0.670	0.961	0.995	0.905
	Fusion + RSA	0.687	0.977	0.995	0.918
	Fusion + RSA + DCMA	**0.704**	0.984	0.995	0.947
	Fusion + RSA + DCMA + Multi-scale	0.685	**0.993**	0.995	**0.980**
NT	RGB	0.800	0.813	0.731	0.991
	SRM	0.785	0.986	0.991	0.995
	Two-stream Fusion (Fusion)	0.811	0.986	0.989	0.995
	Fusion + RSA	0.825	0.988	0.991	0.994
	Fusion + RSA + DCMA	0.840	0.991	0.992	0.994
	Fusion + RSA + DCMA + Multi-scale	**0.894**	**0.995**	**0.993**	0.994

Table 2: Comparison with methods employing novel networks on CelebDF.

Model	Training Set	Testing AUC on CelebDF
Two-stream [16]	SwapMe [16]	0.557
Meso4 [7]	private DF [7]	0.526
MesoInception4 [7]	private DF [7]	0.496
Ours	FF++/DF	0.692
Ours	FF++/FS	**0.722**

Table 3: Comparison with methods employing novel networks on F2F (LQ).

Model	Training/Testing Set	Testing Acc
Two-stream [16]	F2F (LQ)	0.868
Meso4 [7]		0.832
MesoInception4 [7]		0.813
Capsule [12]		0.812
Ours		**0.897**

4. Additional Experiments

4.1. More ablation studies

For the ablation study in Sec. 5.2 of the main text, we use F2F of FF++ [13] as the training set and other databases as the testing sets. Here we present more ablation study results using the other three databases in FF++ as the training set separately. As shown in Tab. 1, the entire model assembling all the proposed modules achieves the best performance consistently under these settings.

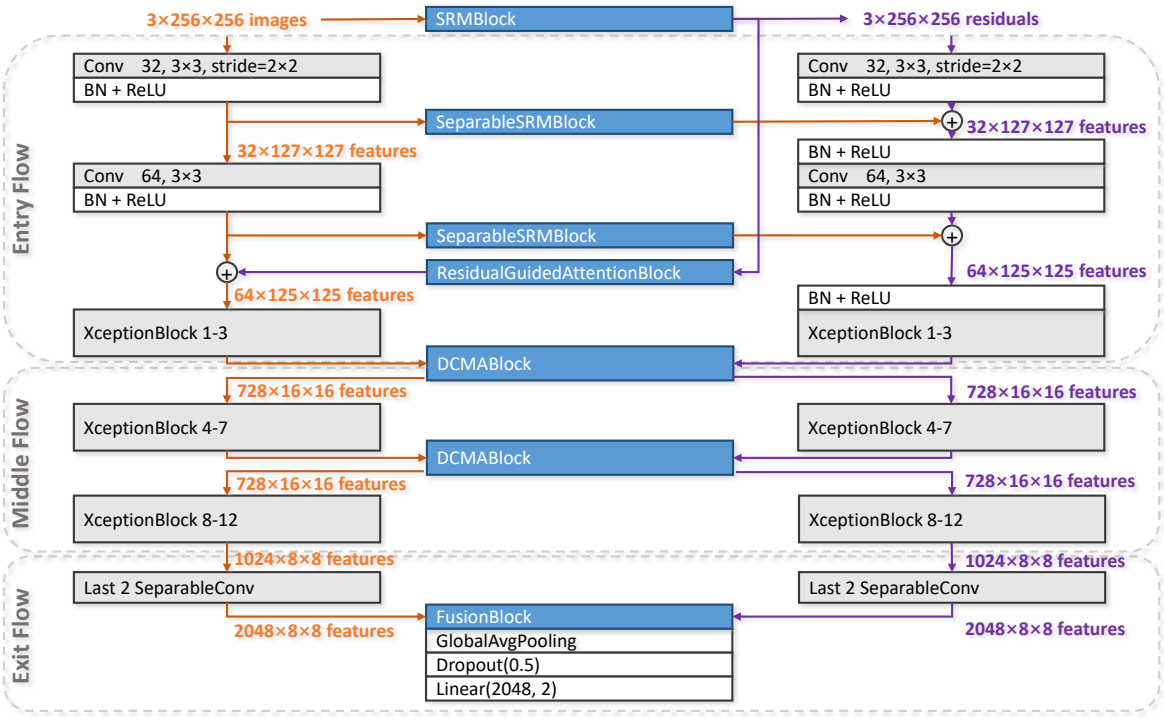

Figure 7: The architecture of the proposed model. We employ two modified Xception models as the backbone. On one hand, the residual features are enhanced by high-frequency features extracted by two separable SRM convolution blocks at different scales. On the other hand, the regular spatial features are calibrated under the guidance of the residual guided attention. We adopt two DCMA blocks to model the correlation and interaction between the two modalities in the middle flow. In the end, the class label is predicted based on the fused features.

Table 4: Cross-database evaluation from FF++ to others. The metrics are AUC, AP (Average Precision), and EER (Equal Error Rate).

Training	Model	DFD [2]			DFDC [1]			CelebDF [11]			DF1.0 [9]		
		AUC ↑	AP ↑	EER ↓	AUC ↑	AP ↑	EER ↓	AUC ↑	AP ↑	EER ↓	AUC ↑	AP ↑	EER ↓
FF++ [13]	Xception [13]	0.831	0.867	0.228	0.679	0.716	0.380	0.594	0.715	0.460	0.698	0.807	0.329
	Face X-ray [10]	0.856	0.866	0.240	0.700	0.737	0.35 0	0.742	0.823	0.336	0.723	**0.819**	0.302
	Ours	**0.919**	**0.930**	**0.175**	**0.797**	**0.819**	**0.299**	**0.794**	**0.861**	**0.276**	**0.738**	0.816	**0.300**

4.2. Comparison with methods employing novel network designs

In Sec. 5.4 of the main paper, we compare our method with multi-task learning methods, high-frequency-based methods, and data generation-based methods. Here we further compare against methods that design effective network architectures, *i.e.*, Two-stream [16], Meso4/MesoInception4 [7], and Capsule [12]. Two-stream utilizes a patch-based triplet network and leverage steganalysis features as a second stream. Meso4 and MesoInception4 are two light-weight CNN models that perform analysis at a mesoscopic level. Capsule introduces a capsule network to detect tampered faces.

Considering that Two-stream is trained on faces forged by the Swapme App [6] and two Meso-nets are trained on a collected Deepfakes [3] dataset, we train our model on the DF and FS [4] datasets in FF++, separately. We compare these models on the CelebDF database [11]. As presented in Tab. 2, our method achieves much better performance in this cross-database setting.

In addition, we evaluate these models on the low-quality F2F [14] set. As shown in Tab. 3, our model shows better robustness in detecting the heavily compressed forgeries.

4.3. More statistics of the cross-database evaluation

In Sec. 5.3 in the main script, we conduct a cross-database evaluation on four large-scale databases. Here we present more statistics of this experiment in Tab. 4

Table 5: Cross-database evaluation with HQ-BI data. (Results of Face X-ray* are from the original paper.)

Training Set	Model	Testing Set (AUC)		
		DFD	DFDC	CelebDF
raw-BI	Face X-ray*	0.935	0.712	0.748
raw-BI & raw-FF++		0.954	0.809	0.806
HQ-BI	Face X-ray	0.727	0.715	**0.824**
	Ours	**0.838**	**0.788**	0.813
HQ-BI & HQ-FF++	Face X-ray	0.908	0.783	0.833
	Ours	**0.951**	**0.822**	**0.840**

4.4. Comparison with Face X-ray on the BI dataset

Since real-world face forgeries have a limited quality, we run the released code[1] and generate a BI dataset with real faces from HQ (lightly compressed) FF++, namely *HQ-BI*. The comparison is shown in Tab. 5. Note that we also present the reported results of Face X-ray [10] (* marked) that is trained in raw (no compression) images, namely *raw-BI*. Both two models are promoted with HQ-BI data, and the proposed model achieves superior performance.

References

[1] Deepfake detection challenge. https://www.kaggle.com/c/deepfake-detection-challenge. Accessed: 2020-05-10. 3

[2] Deepfakedetection. https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html. Accessed: 2020-05-10. 3

[3] Deepfakes. https://github.com/iperov/DeepFaceLab. Accessed: 2020-05-10. 3

[4] Faceswap. https://github.com/MarekKowalski/FaceSwap. Accessed: 2020-05-10. 3

[5] Pytorch. https://pytorch.org/. An open source machine learning framework that accelerates the path from research prototyping to production deployment. 2

[6] Swapme. https://itunes.apple.com/us/app/swapme-by-faciometrics/.*. This company hasbeen acquired by Facebook and no longer available in App-Store. 3

[7] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. In *WIFS*, 2018. 2, 3

[8] Jessica Fridrich and Jan Kodovsky. Rich models for steganalysis of digital images. *TIFS*, 2012. 1

[9] Liming Jiang, Ren Li, Wayne Wu, Chen Qian, and Chen Change Loy. Deeperforensics-1.0: A large-scale dataset for real-world face forgery detection. In *CVPR*, 2020. 3

[10] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. Face x-ray for more general face forgery detection. *CVPR*, 2020. 3, 4

[11] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A new dataset for deepfake forensics. *arXiv preprint arXiv:1909.12962*, 2019. 3

[12] Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. Capsule-forensics: Using capsule networks to detect forged images and videos. In *ICASSP*, 2019. 2, 3

[13] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *ICCV*, 2019. 2, 3

[14] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *CVPR*, 2016. 3

[15] J. Ye, J. Ni, and Y. Yi. Deep learning hierarchical representations for image steganalysis. *IEEE Transactions on Information Forensics and Security*, 12(11):2545–2557, 2017. 1

[16] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Two-stream neural networks for tampered face detection. In *CVPRW*, 2017. 2, 3

[17] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Learning rich features for image manipulation detection. In *CVPR*, 2018. 1

[1] https://github.com/AlgoHunt/Face-Xray.git