DiffForensics: Leveraging Diffusion Prior to Image Forgery Detection and Localization

Supplementary Material

In this supplementary material, we include many details of our work: 1) the detailed implementation of the proposed DiffForensics and the used datasets. 2) more results for image forgery localization, including qualitative and quantitative experiments. 3) the performance of DiffForensics against other two post-processing attacks, as well as the results on social media transmissions [24]. 4) more ablation experiments on self-supervised denoising diffusion pre-training.

1. Implementation Details

Architecture. The architecture of our encoder-decoder framework is shown in Fig. 1. As for the encoder, we utilize the encoder part of Segformer-B5 [26], which is a transformer-based segmentation architecture. As for the decoder, each block is mainly composed of a feature decoding structure (driven from Unet [22]) and a time embedding module (driven from [8]). During training, t is uniformly sampled from 1 to T. By inputting the 256-D embedding (the output of the encoder) and the time embedding t, we have the output of the decoder.

Datasets. Details of the datasets used for pre-training, fine-tuning and evaluation are reported in Table 2. We evaluate our model on six datasets, which are tampered by traditional image editing tools, including CASIAv1+ [2], Columbia [9], NIST16 [19], IMD2020 [20], DSO-1 [4] and Korus [13]. And two datasets are tampered by deep generative models (DGMs), including AutoSplicing [11] and OpenForensics [16].

- **CASIAv1+** [5] contains 920 images depicting different objects that have undergone tampering via copy-move and splicing. The tampered regions are carefully selected and complemented by various post-processing techniques, including filtering and blurring. To avoid data overlap between real images and tampered images, we follow the composition of [2] and use the data in the COREL [23] as real images.
- **Columbia** [9] contains 180 uncompressed spliced tampered images and 183 original images.
- **NIST16** [19] presents a challenging collection that encompasses all three tampering techniques. The manipulations included in this selection are post-processed to mask visible indicators.
- **IMD2020** [20] consists of 2010 real-life manipulated images collected from the Internet and corresponding 414 original images.
- **DSO-1** [4] contains 100 images of people undergoing splicing tampering operations.

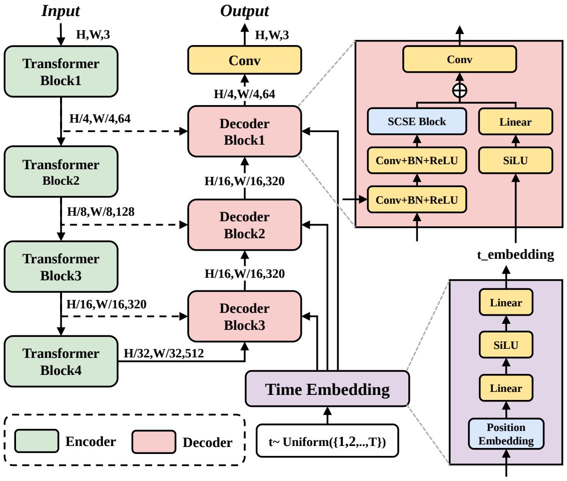

Figure 1. Architecture of our encoder-decoder framework.

- **Korus** [4] contains 220 images of daily scenes taken by four digital cameras and tampered with all three manipulations.
- **AutoSplicing** [11] uses the language-image model based on the diffusion model, DALL-E2 [21], to modify the image either locally or globally guided by text prompts. The dataset comprises 3621 manipulated images and 2273 authentic images, with varying dimensions from 256 × 256 to 4232 × 4232 pixels.
- **OpenForensics** [16] contains 18895 tampered images of several facial images, generated via GAN and incorporating both genuine and tampered facial images in the latter category.

2. More Experimental Results

Localization Evaluation. In this part, we illustrate more comparison results with the SOTA methods in terms of localization evaluation.

Quantitative results. Following [2, 6, 14], we further report the F1 metrics of the forgery localization results under the best threshold and IOU under the fixed threshold in Table 1, our method achieves the best F1 and IOU on most of the test datasets and finally obtains the best average F1 and IOU of all test datasets. It further shows the effectiveness of our method.

Qualitative results. We illustrate qualitative results of forgery localization on six forgery datasets shown in Fig. 3.

Methods	Editing												DGM				Average	
	CASIA1.0+		Columbia		NIST16		IMD2020		DSO		Korus		AutoSplice		OpenForensics			
	F1	IOU	F1	IOU	F1	IOU	F1	IOU	F1	IOU	F1	IOU	F1	IOU	F1	IOU	F1	IOU
H-LSTM [1]	.204	.072	.464	.158	.184	.064	.184	.070	.300	.109	.153	.051	.578	.188	.172	.072	.280	.098
ManTra-Net* [25]	.216	.078	.614	.248	.316	.107	.354	.120	.366	.057	.246	.069	.591	.113	.180	.024	.360	.102
HP-FCN [17]	.532	.079	.389	.027	.418	.040	.382	.015	.280	.007	**.395**	.041	.587	.015	**.377**	.014	.420	.030
GSR-Net [28]	.523	.194	.740	.266	.384	.172	.377	.070	.402	.037	.256	.041	.637	.029	.214	.016	.442	.103
SPAN [10]	.213	.053	.541	.144	.250	.076	.264	.068	.311	.034	.179	.040	.588	.027	.184	.008	.316	.056
MVSS-Net* [2]	**.674**	.397	.749	.573	.456	.239	.446	.201	.447	.188	.268	.067	.736	.241	.219	.037	.499	.243
CAT-Net [14]	.501	.363	.925	.826	.453	.275	.467	.235	.367	.111	.279	.115	.597	.136	.069	.002	.457	.258
SATL-Net [30]	.178	.041	.787	.595	.284	.138	.282	.104	.224	.060	.133	.026	.377	.067	.110	.012	.297	.130
PSCC-Net [18]	.455	.283	.798	.554	.347	.170	.423	.217	.408	.202	.260	.104	.631	.106	.172	.038	.437	.209
HiFi-Net [7]	-	.063	-	.264	-	.121	-	.114	-	.193	-	.054	-	**.486**	-	**.088**	-	.173
Ours	.636	**.480**	**.938**	**.893**	**.526**	**.359**	**.629**	**.443**	**.662**	**.413**	.383	**.204**	**.831**	.418	.277	.083	**.610**	**.412**

Table 1. Pixel-level F1 (best threshold) and IOU performance of image forgery localization. The best result is highlighted and bold. Except for the method with $*$ uses the pre-training model of the original paper, other methods keep the same training data as our method. (Hifi-Net uses a special threshold range when calculating the binary classification index, so we do not report F1 under the best threshold.)

Dataset	Neg.	Pos.	Com.	Spl.	Inp.
#Pre-Training&Fine-tuning					
Fantasitic - Reality [12]	16,592	19,423	-	19,423	-
CASIAv2 [5]	7,491	5,123	3,295	1,828	-
#Evaluation					
CASIAv1+ [5]	800	920	459	461	-
Columbia [9]	183	180	-	180	-
NIST16 [19]	-	564	68	288	208
IMD2020 [20]	414	2,010	-	-	-
DSO-1 [4]	-	100	-	100	-
Korus [13]	-	220	-	-	-
AutoSplicing [11]	2,273	3,621	-	3,621	-
OpenForensics [16]	-	18,895	-	-	18,895

Table 2. The pre-training, training, and testing data used in our experiments, the upper part of the test dataset is artificially editing data, and the lower part is DGM data.

Our method exhibits higher accuracy and lower false alarm rate not only in image editing forgery datasets but also in DGM datasets, which demonstrates its good generalization ability. Especially in subtle forgery regions, (*e.g.*, row1, row9-row11) the contours of forgery and authentic regions can still be accurately localized.

3. More Robustness Analysis

To show the robustness performance of the proposed method in a more comprehensive way, in addition to the JPEG compression and Gaussian noise mentioned in the main body, we also show the results against other two widely used attacks, *i.e.*, Gaussian blurring and Median filtering, as shown in Fig. 2. It is observed that the proposed method achieves leading performance, especially for image forgery localization.

We also validate the robustness of our method to social media network transmissions on the four datasets, *i.e.*, CASIAv1 [5], Columbia [9], DSO-1 [4] and NIST16 [19], which have undergone social media transmission [24], and the results are summarized in Table 3. In the upper part of Table 3, we show the comparison results of our method with other involved methods trained on the same experimental setting in the main body, it is ready to see that our method achieves the best forensic results on the four OSN datasets. Since IF-OSN [24] was designed for the image forgery localization task, for a fair comparison, we also report the results by training the models with tampered data only, as shown in the lower part of Table 3. To compare with other IFDL methods, *i.e.*, MVSS-Net[1] [2], CAT-Net[2] [14], PSCC-Net[3] [18], IF-OSN[4] [24], TruFor[5] [6], ReLoc[6] [29], their pre-trained models are used for testing although most of them were trained on a larger dataset than ours. Our method shows superior performance to other methods on Columbia [9], DSO-1 [4] and NIST16 [19], and achieves the best average performance. Although TruFor [6] exhibits better performance on CASIAv1 [5], it benefits from a training dataset 36 times larger than ours.

4. More Ablation Studies

Self-supervised denoising diffusion pre-training. In this part, we conduct additional ablation studies of several components for the self-supervised denoising diffusion pre-training.

[1] https://github.com/dong03/MVSS-Net

[2] https://github.com/mjkwon2021/CAT-Net

[3] https://github.com/proteus1991/PSCC-Net

[4] https://github.com/HighwayWu/ImageForensicsOSN

[5] https://github.com/grip-unina/TruFor Note: TruFor [6] uses special F1 calculations in the original paper, details of which can be seen on (https://github.com/grip-unina/TruFor/issues/3), and here we adopt the most commonly used calculations in [2, 7, 14, 18, 24, 29].

[6] https://github.com/ZhuangPeiyu/ReLoc

Methods	#Data	CASIA v1				Columbia				DSO-1				NIST16				Average			
		Fb	Wa	Wb	Wc	Fb	Wa	Wb	Wc	Fb	Wa	Wb	Wc	Fb	Wa	Wb	Wc	Fb	Wa	Wb	Wc
CAT-Net [14]	32k	.306	.300	.365	**.204**	.869	.859	.854	.866	.123	.095	.098	.106	.336	.300	.312	.313	.409	.389	.407	.372
SATL-Net [30]		.069	.069	.063	.062	.673	.685	.674	.674	.081	.064	.077	.060	.183	.128	.182	.189	.252	.237	.249	.246
PSCC-Net [18]		.108	.003	.219	.002	.628	.646	.645	.655	.312	.284	.304	.298	.229	.245	.230	.217	.319	.295	.350	.293
HiFi-Net [7]		.067	.063	.083	.029	.407	.412	.384	.398	.307	.307	.294	.315	.171	.175	.172	.180	.238	.239	.233	.231
Ours		**.348**	**.350**	**.375**	.150	**.900**	**.899**	**.910**	**.900**	**.512**	**.526**	**.493**	**.495**	**.418**	**.415**	**.400**	**.385**	**.545**	**.548**	**.545**	**.483**
MVSS-Net [2]	13k	.387	.359	.404	.248	.691	.685	.689	.690	.277	.181	.258	.214	.264	.165	.251	.211	.405	.348	.401	.341
CAT-Net v1 [14]	876k	.083	.066	.120	.042	.777	.769	.770	.765	.095	.033	.051	.050	.173	.071	.112	.089	.282	.235	.263	.237
CAT-Net v2 [15]	876k	.011	.011	.012	.015	.887	.874	.870	.862	.077	.027	.064	.040	.251	.152	.195	.179	.307	.266	.285	.274
PSCC-Net [18]	100k	.094	.079	.111	.086	.578	.563	.621	.601	.209	.214	.144	.204	.211	.154	.173	.171	.273	.253	.262	.266
IF-OSN [24]	-	.462	.405	.466	.478	.714	.727	.724	.727	.462	.405	.466	.478	.329	.286	.294	.313	.492	.456	.488	.499
TruFor [6]	876k	**.683**	**.674**	**.647**	**.578**	.750	.748	.802	.774	**.674**	.389	.478	.446	.333	.384	.322	.343	.610	.549	.562	.535
ReLoc [29]	12k	.578	.586	.560	.496	.707	.690	.716	.726	.330	.309	.303	.302	.312	.314	.279	.292	.482	.475	.465	.454
Ours	25k	.553	.570	.560	.454	**.919**	**.915**	**.916**	**.917**	.541	**.531**	**.521**	**.509**	**.430**	**.426**	**.426**	**.413**	**.611**	**.611**	**.606**	**.573**

Table 3. Pixel-level F1 performance (fixed threshold) of image forgery localization on datasets uploaded on Facebook (Fb), WhatsApp (Wa), Weibo (Wb), WeChat (Wc).

Noise		Model weights		Localization				Detection				Average	
				IMD2020		AutoSplicing		IMD2020		AutoSplicing			
Guass	Simplex	Encoder	Decoder	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC
-	-	Cityscapes	-	.438	.887	.275	.861	.831	.715	.635	.882	.545	.836
✓	-	DDPM	DDPM	.425	.885	**.498**	**.935**	.725	.682	.597	.901	.561	.851
✓	-	Cityscapes	DDPM	.488	**.910**	.289	.880	.831	.737	.522	.899	.533	.857
-	✓	DDPM	DDPM	.438	.877	.425	.851	.763	.666	.432	.843	.515	.809
-	✓	Cityscapes	DDPM	**.519**	.907	.399	.917	**.819**	**.757**	.571	**.924**	**.577**	**.876**

Table 4. For IFDL tasks, the performance of different weight settings for denoising diffusion pre-training using Simplex noise and Gaussian noise for encoder and decoder structures.

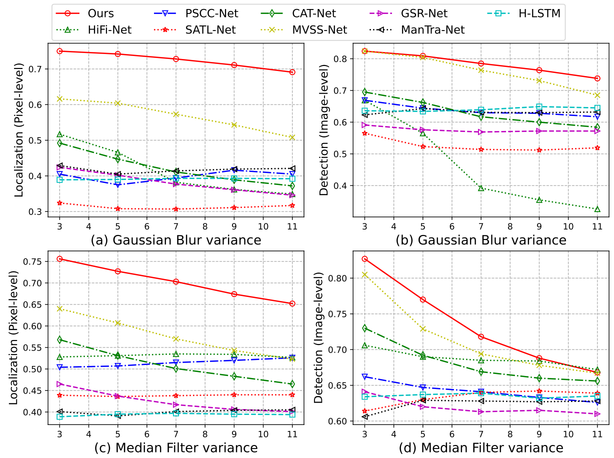

Figure 2. Robustness against Gaussian Blurring and Median Filtering effects. Tested on CASIA1.0+, Columbia, IMD2020 and AutoSplicing. Our method achieves a substantial lead in tamper localization performance.

Pre-Training		Localization		Detection		Average	
Neg.	Pos.	F1	AUC	F1	AUC	F1	AUC
1	0	.342	.860	.639	.848	.491	.854
0	1	.422	.896	.741	.834	.582	.865
1/4	1/4	.385	.896	.672	.832	.529	.864
1/2	1/2	.453	.912	**.781**	**.847**	.617	.880
3/4	3/4	.395	.913	.749	.821	.572	.867
1	1	**.509**	**.925**	.760	.846	**.635**	**.886**

Table 5. Performance of different types and amounts of the pre-training data. Tested on IMD2020 and AutoSplicing.

Firstly, to comprehensively explore the impact of our proposed scheme that combines the macro-features and micro-features for the IFDL task, we additionally adopted pre-trained weights of the encoder on another semantic segmentation dataset (*i.e.*, Cityscapes [3]), and showed the results in Table 4. Similar to the case of using pre-trained weights on ADE20K [27], our proposed training scheme of combining macroscopic features with supervised weights and using simplex noise for denoising diffusion pre-training to obtain microscopic features can effectively boost the performance for IFDL task. Furthermore, we can observe that the macroscopic feature weights pre-trained by ADE20K can better improve the performance of the IFDL task because the image contents of the ADE20K dataset are closer to the ones in widely used tampered image datasets.

Secondly, we investigate the impact of the types of pre-training data, including real-image-only data, tampered-image-only data, and the combination of real and tampered image data. As shown in the 1st, 2nd, and last row of Table 5, the performance of the model by training on real-image-only dataset is the worst, while the performance of the model increases considerably with tampered-image-only dataset. And the model performance can be further boosted with the combination of real and tampered image data. Furthermore, we investigate the impact of the amount of pre-training data. As shown in the last 4 rows of Table 5, with the increase of the amount of pre-training data, the overall performance of the model in detection and localization is getting better. It indicates that more suitable data during self-supervised pre-training could encourage DiffForensics to better learn tampering micro-features.

References

[1] Jawadul H Bappy, Cody Simons, Lakshmanan Nataraj, BS Manjunath, and Amit K Roy-Chowdhury. Hybrid lstm and encoder–decoder architecture for detection of image forgeries. *IEEE Transactions on Image Processing*, 28(7):3286–3300, 2019. 2

[2] Xinru Chen, Chengbo Dong, Jiaqi Ji, Juan Cao, and Xirong Li. Image manipulation detection by multi-view multi-scale supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14185–14193, 2021. 1, 2, 3

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 3

[4] Tiago José De Carvalho, Christian Riess, Elli Angelopoulou, Helio Pedrini, and Anderson de Rezende Rocha. Exposing digital image forgeries by illumination color classification. *IEEE Transactions on Information Forensics and Security*, 8 (7):1182–1194, 2013. 1, 2, 6

[5] Jing Dong, Wei Wang, and Tieniu Tan. Casia image tampering detection evaluation database. In *2013 IEEE China summit and international conference on signal and information processing*, pages 422–426. IEEE, 2013. 1, 2, 6

[6] Fabrizio Guillaro, Davide Cozzolino, Avneesh Sud, Nicholas Dufour, and Luisa Verdoliva. Trufor: Leveraging all-round clues for trustworthy image forgery detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20606–20615, 2023. 1, 2, 3

[7] Xiao Guo, Xiaohong Liu, Zhiyuan Ren, Steven Grosz, Iacopo Masi, and Xiaoming Liu. Hierarchical fine-grained image forgery detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3155–3165, 2023. 2, 3

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1

[9] Yu-Feng Hsu and Shih-Fu Chang. Detecting image splicing using geometry invariants and camera characteristics consistency. In *2006 IEEE International Conference on Multimedia and Expo*, pages 549–552. IEEE, 2006. 1, 2, 6

[10] Xuefeng Hu, Zhihan Zhang, Zhenye Jiang, Syomantak Chaudhuri, Zhenheng Yang, and Ram Nevatia. Span: Spatial pyramid attention network for image manipulation localization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pages 312–328. Springer, 2020. 2

[11] Shan Jia, Mingzhen Huang, Zhou Zhou, Yan Ju, Jialing Cai, and Siwei Lyu. Autosplice: A text-prompt manipulated image dataset for media forensics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 893–903, 2023. 1, 2, 6

[12] Vladimir V Kniaz, Vladimir Knyaz, and Fabio Remondino. The point where reality meets fantasy: Mixed adversarial generators for image splice detection. *Advances in neural information processing systems*, 32, 2019. 2

[13] Paweł Korus and Jiwu Huang. Multi-scale analysis strategies in prnu-based tampering localization. *IEEE Transactions on Information Forensics and Security*, 12(4):809–824, 2016. 1, 2, 6

[14] Myung-Joon Kwon, In-Jae Yu, Seung-Hun Nam, and Heung-Kyu Lee. Cat-net: Compression artifact tracing network for detection and localization of image splicing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 375–384, 2021. 1, 2, 3

[15] Myung-Joon Kwon, Seung-Hun Nam, In-Jae Yu, Heung-Kyu Lee, and Changick Kim. Learning jpeg compression artifacts for image manipulation detection and localization. *International Journal of Computer Vision*, 130(8):1875–1895, 2022. 3

[16] Trung-Nghia Le, Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. Openforensics: Large-scale challenging dataset for multi-face forgery detection and segmentation in-the-wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10117–10127, 2021. 1, 2, 6

[17] Haodong Li and Jiwu Huang. Localization of deep inpainting using high-pass fully convolutional network. In *proceedings of the IEEE/CVF international conference on computer vision*, pages 8301–8310, 2019. 2

[18] Xiaohong Liu, Yaojie Liu, Jun Chen, and Xiaoming Liu. Pscc-net: Progressive spatio-channel correlation network for image manipulation detection and localization. *IEEE Transactions on Circuits and Systems for Video Technology*, 32 (11):7505–7517, 2022. 2, 3

[19] NIST. Nimble. Datasets, 2016. `https://www.nist.gov/itl/iad/mig/nimble-challenge-2017-evaluation`. 1, 2, 6

[20] Adam Novozamsky, Babak Mahdian, and Stanislav Saic. Imd2020: A large-scale annotated dataset tailored for detecting manipulated images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, pages 71–80, 2020. 1, 2, 6

[21] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image gen-

eration with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 1

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 1

[23] James Ze Wang, Jia Li, and Gio Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on pattern analysis and machine intelligence*, 23(9):947–963, 2001. 1

[24] Haiwei Wu, Jiantao Zhou, Jinyu Tian, and Jun Liu. Robust image forgery detection over online social network shared images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13440–13449, 2022. 1, 2, 3

[25] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9543–9552, 2019. 2

[26] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 1

[27] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 3

[28] Peng Zhou, Bor-Chun Chen, Xintong Han, Mahyar Najibi, Abhinav Shrivastava, Ser-Nam Lim, and Larry Davis. Generate, segment, and refine: Towards generic manipulation segmentation. In *Proceedings of the AAAI conference on artificial intelligence*, pages 13058–13065, 2020. 2

[29] Peiyu Zhuang, Haodong Li, Rui Yang, and Jiwu Huang. Reloc: A restoration-assisted framework for robust image tampering localization. *IEEE Transactions on Information Forensics and Security*, 2023. 2, 3

[30] Long Zhuo, Shunquan Tan, Bin Li, and Jiwu Huang. Self-adversarial training incorporating forgery attention for image forgery localization. *IEEE Transactions on Information Forensics and Security*, 17:819–834, 2022. 2, 3

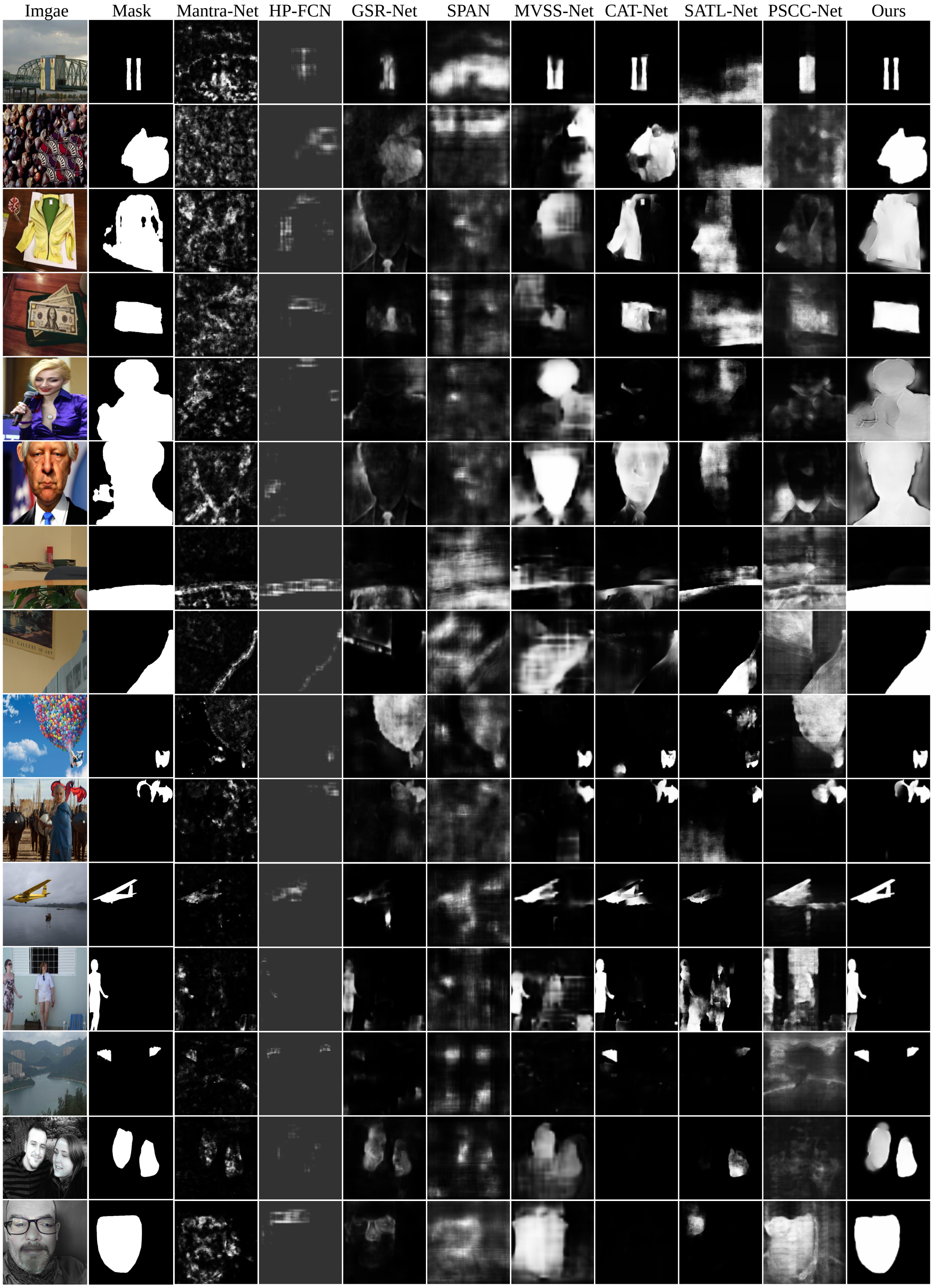

Figure 3. Qualitative localization evaluations on six standard test datasets. From top to bottom, row1-row2: CASIA1.0+ [5], row3-row6: AutoSplicing [11], row7-row8: Columbia [9], row9-row10: IMD2020 [20], row11: NIST16 [19], row12: DSO-1 [4], row13: Korus [13], row14-row15: Openforensics [16].