Hierarchical Fine-Grained Image Forgery Detection and Localization — Supplementary material —

Xiao Guo[1], Xiaohong Liu[2], Zhiyuan Ren[1], Steven Grosz[1], Iacopo Masi[3], Xiaoming Liu[1]
[1] Michigan State University [2] Shanghai Jiao Tong University [3] Sapienza University of Rome
{guoxia11, renzhiy1, groszst}@msu.edu, xiaohongliu@sjtu.edu.cn,
masi@di.uniroma1.it, liuxm@msu.edu

In this supplementary material, we include many details of our work: 1) the details of the proposed HiFi-IFDL dataset; 2) The generalization performance against images generated from unseen forgery methods and real images in the unseen domain; 3) the HiFi-Net performance against different types of post-processing in the image editing domain; 4) the complete HiFi-Net performance on the DFFD dataset [23]; 5) we offer the forgery attribute classification results on seen and unseen forgery attributes; 6). the detailed implementation of the proposed HiFi-Net.

1. Dataset Collection Details

Table 1 reports all the forgery methods used in our dataset. In the last column, the table shows if the method used to generate the manipulated images is pre-trained, self-trained, or we used the released images. In Fig. 4 and Fig. 5, we show several examples taken from our dataset that represents a variety of objects, scenes, faces, animals. The real image dataset is the combination of LSUN [27], CelebaHQ [10], FFHQ [9], AFHQ [1], MSCOCO [13] and real face images in face forensics [21]. We either take the entire dataset or randomly select 100k images from these real datasets.

Forgery Method	Image Source	Images #	Source
DDPM [4]	LSUN	100k	pre-trained
DDIM [22]	LSUN	100k	pre-trained
GDM. [18]	LSUN	100k	pre-trained
LDM. [20]	LSUN	100k	pre-trained
StarGANv2 [1]	CelebaHQ	100k	pre-trained
HiSD [12]	CelebaHQ	100k	pre-trained
StGANv2-ada [7]	FFHQ, AFHQ	100k	pre-trained
StGAN3 [8]	FFHQ, AFHQ	100k	pre-trained
STGAN [15]	CelebaHQ	100k	self-train
Faceshifter [11]	Youtube video	100k	released

Table 1. The details of the collected dataset. Each column in order shows forgery method; the image source used for the generation; the image number; if the images are generated with a pre-trained/self-trained models or released images.

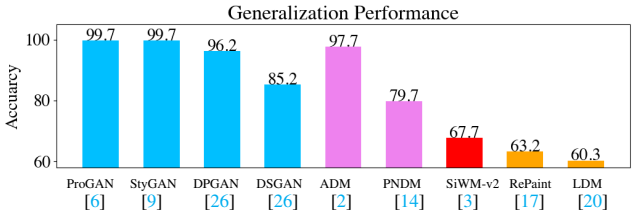

Figure 1. The image-level forgery detection accuracy on images generated by unseen GAN ( ) and diffusion ( ) methods, and unseen domain real images ( ). The pixel-level localization accuracy on images inpainted by unseen diffusion model ( ). From left to right, first 6 methods produce *bedroom* in LSUN [27]. The SiW-Mv2 contains real human face. Images generated from the last two diffusion model inpainting methods, are human face and general objects. All these images are either obtained directly from the open source github or the pre-trained weight.

2. Generalization Performance

Fig. 1 reports our method's generalization performance. Specifically, for each generative method and unseen domain real images, we have collected 1000 images and use these images to form an inference dataset. After that, we apply the pre-trained HiFi-Net on such an inference to compute the classification accuracy, given 0.5 fixed-threshold.

Our first conclusion is the same as the most recent work [19] that some generative methods such as DSGAN [26] and PNDM [14] can generate rather sophisticated images that fool the powerful forgery detector.

Secondly, we hypothesize that powerful forgery detector can largely fail when being applied on real images in different domain. For example, real images from SiW-Mv2 dataset [3], where facial image has spoof traces, such as funny glasses and wigs.

Lastly, and more importantly, we observe the well-trained model always generalizes poorly on the image that is partially manipulated by diffusion model. We think this is because of two reasons: (1) conventional image editing methods are distinct by nature to the most recently pro-

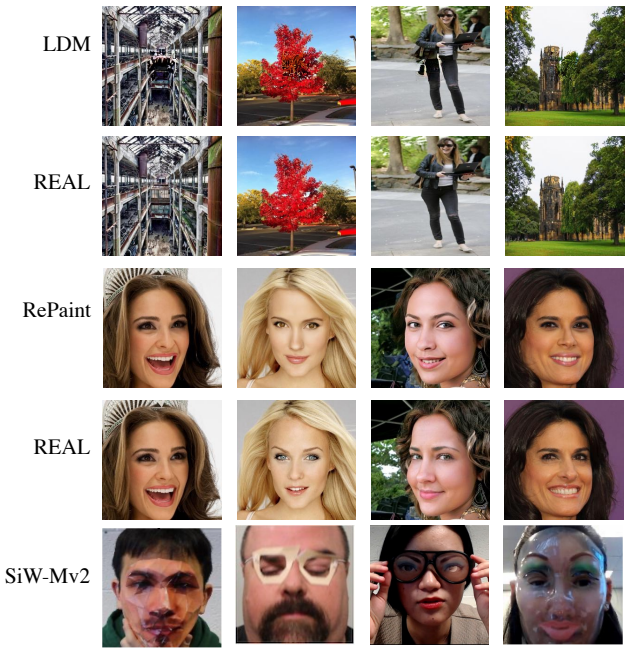

Figure 2. Images manipulated by LDM [20] and RePaint [17] and their corresponding real images. The last row is the SiW-Mv2 dataset, where real human faces have spoof traces, such as funnyeyes and masks.

posed inpainting methods based on diffusion model; (2) the forgery area edited by diffusion model can have variations, not a rigid copy-move or removal manner that is commonly used by the traditional editing methods.

We believe these three aspects are valuable for the future research, namely: (a) how to make the model generalize well on detecting forged images created by advanced methods, (b) how to maintain the precision when we have real images from the new domain, (c) diffusion model based inpanting method can raise an issue for the existing forgery localization methods, indicating that new algorithm is needed.

3. Image Editing Experiment

Following previous works [5,16,25], we evaluate the performance of our method against different post-processing steps, which is reported in Tab. 2. Our proposed method is more robust than the previous work, except for the post-processing of resizing 0.78 times the image and JPEG compression with 50% quality. Meanwhile, more qualitative results can be found in Fig. 6.

4. The DFFD Dataset Performance

We have included the complete version of our method performance on the DFFD dataset in Tab. 3. As we can see, compared to Attention Xception [23], our method still achieves more accurate localization performance on Partial Manipulated and Fully Synthesized images. For the localization performance on the real images, our performance is comparable with the Attention Xception [23].

Post.	SPAN [5]	PSCC [16]	Obj.Fo. [25]	Ours
Resize (0.78)	83.24	85.29	**87.2**	86.9
Resize (0.25)	80.32	85.01	86.3	**86.5**
Gau.Blur (3)	83.10	85.38	85.97	**86.1**
Gau.Blur (15)	79.15	79.93	80.26	**81.0**
Gau.Noi (3)	75.17	78.42	79.58	**81.9**
Gau.Noi (15)	67.28	76.65	78.15	**79.5**
JPEG Co. (100)	83.59	85.40	86.37	**86.5**
JPEG Co. (50)	80.68	85.37	**86.24**	86.0

Table 2. IFDL performance on *NIST16* with different post-processing steps. [Key: **Best**; Gau.: Gaussian; JPEG Co.: JPEG Compression.].

IoU (↑) / PBCA (↑)	Real	Fu. Syn.	Par. Man.
Att. [23]	−/**0.998**	0.847/0.847	0.401/0.786
Ours	−/0.978	**0.893**/**0.893**	**0.411**/**0.801**

(a)

IINC (↓) / C.S. (↓)	Real	Fu. Syn.	Par. Man.
Att. [23]	0.015/−	0.077/**0.095**	**0.311**/0.429
Ours	**0.010**/−	**0.060**/0.107	0.323/**0.410**

(b)

Table 3. The localization performance: (a) Metrics are IoU and PBCA, the higher the better, (b) Metrics are IINC and Cosine Similarity, the lower the better. [Keys: Fu. Syn.: Fully-synthesized; Par. Man.: Partially-manipulated]

5. Forgery Attribute Classification

We have included specific classification results for a variety of samples. Tab. (6) of the paper reports the fine-grained classification result of HiFi-IFDL. Here we show the classification probability at different levels. In examples (5) and (10) of Fig. 3, we can see the robustness of our proposed method that learns the hierarchical structure. The 3rd level fine-grained prediction probability on 5th example and 10th example is lower than the fine-grained classification prediction probability on the 4th level. This means our algorithm can recover the accuracy at the fine level classification even the classification on the coarser level does not perform excellent.

6. Implementation Details

In our HiFi-Net, the feature map resolution for different branches are 256, 128, 64, and 32 pixels. In the experiment on HiFi-IFDL, the fine-grained classification for 1st, 2nd, 3rd and 4th levels are 2-way, 4-way, 6-way and 14-way multi-class classification, respectively. The 3rd level fine-grained classification categories are: `unconditional diffusion`, `conditional diffusion`, `unconditional GAN`, `conditional GAN`, `CNN-based partial manipulation` and `Image editing`.

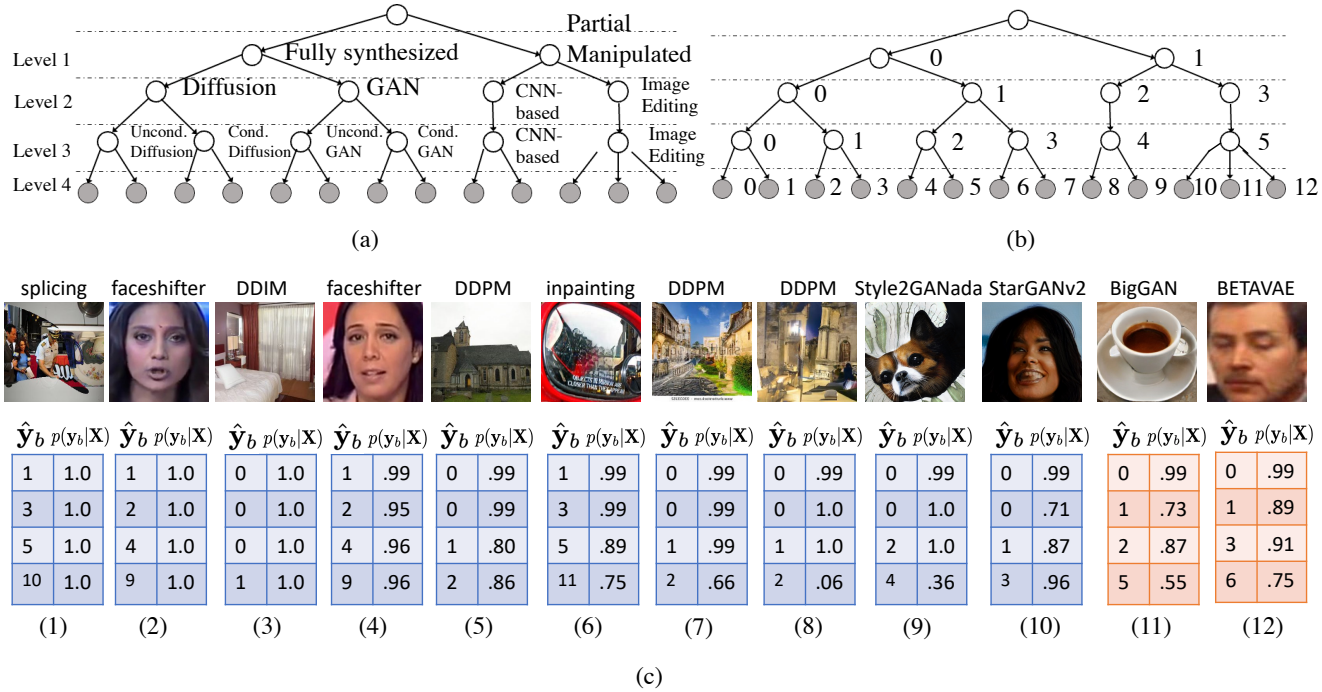

Figure 3. (a) The original hierarchical structure used in the fine-grained classification at different levels. (b). For convenience, we assign category number to each forgery attribute, at different levels. (c). The detailed prediction probability for each input forged images. The table below each image reports the categorical value $\hat{\mathbf{y}}_b$ and corresponding prediction probability $p(\mathbf{y}_b|\mathbf{X})$. From the top to down order, the results shown are for level1 to level4. For example, in the splicing image (example (1)): `partial manipulated→image editing→image editing→splicing`, which corresponds to the label index $1, 3, 5, 10$ that are shown in the first table. Lastly, the example (1) – (10) are seen forgery method in the training, and the example (11) and (12) are unseen forgery attribute.

As for the details of $\mathcal{L}_{loc}$ implementation, we first use the initialized HiFi-Net to convert each pixel in the input image to the high-dimensional feature $\mathbf{F}'_{ij} \in R^D$, where $D = 18$. Then we average the feature $\mathbf{F}'_{ij} \in R^D$ for all pixels in the real image from the HiFi-IFDL, and this average value then is used as $\mathbf{c}$. Then, we compute the ℓ_2 distance between each pixel feature $\mathbf{F}'_{ij} \in R^D$ and $\mathbf{c}$, and denote the largest distance as D_{max}. In the Eq. (1) of the paper, we set the threshold τ as $2.5 \cdot D_{max}$.

The architecture is trained end-to-end with different learning rates per layers. The detailed objective function is:

$$\mathcal{L}_{tot} = \begin{cases} 100{*}\mathcal{L}_{loc} + \mathcal{L}^1_{cls} + \mathcal{L}^2_{cls} + \mathcal{L}^3_{cls} + 100{*}\mathcal{L}^4_{cls} & \text{if } \mathbf{X} \text{ is forged} \\ \mathcal{L}_{loc} + \mathcal{L}^4_{cls} & \text{if } \mathbf{X} \text{ is real} \end{cases}$$

Where $\mathbf{X}$ is the input image. When the input image is labeled as "real", we only apply the last branch ($\boldsymbol{\theta}_4$) loss function, otherwise, if it is labeled as "manipulated", we use all the branches.

The entire architecture is trained for 13 epoches, and all training samples are seen by the model in each epoch. The feature extractor is modified based on the pre-trained HRNet [24], in which we add more layers such that each branch of our multi-branch feature extractor can have identical number of convolutional layers, and the details can be found in our source code that will be released upon the acceptance. We use `1e-4` to train the multi-branch feature extractor and classification module, and `3e-4` to train the localization modules. During the training, we use `ReduceLROnPlateau` as the learning rate scheduler to reduce the learning rate.

7. Societal Impact

Our work has the positive societal impact to the community. Because our work is dealing with various categories of forgery methods, which enable the algorithm to detect all kinds of manipulation, including seen and unseen forgeries, as indicated by Supplementary section 4. Our algorithm can enable a tool that makes general public in our society to have more trust in media contents.

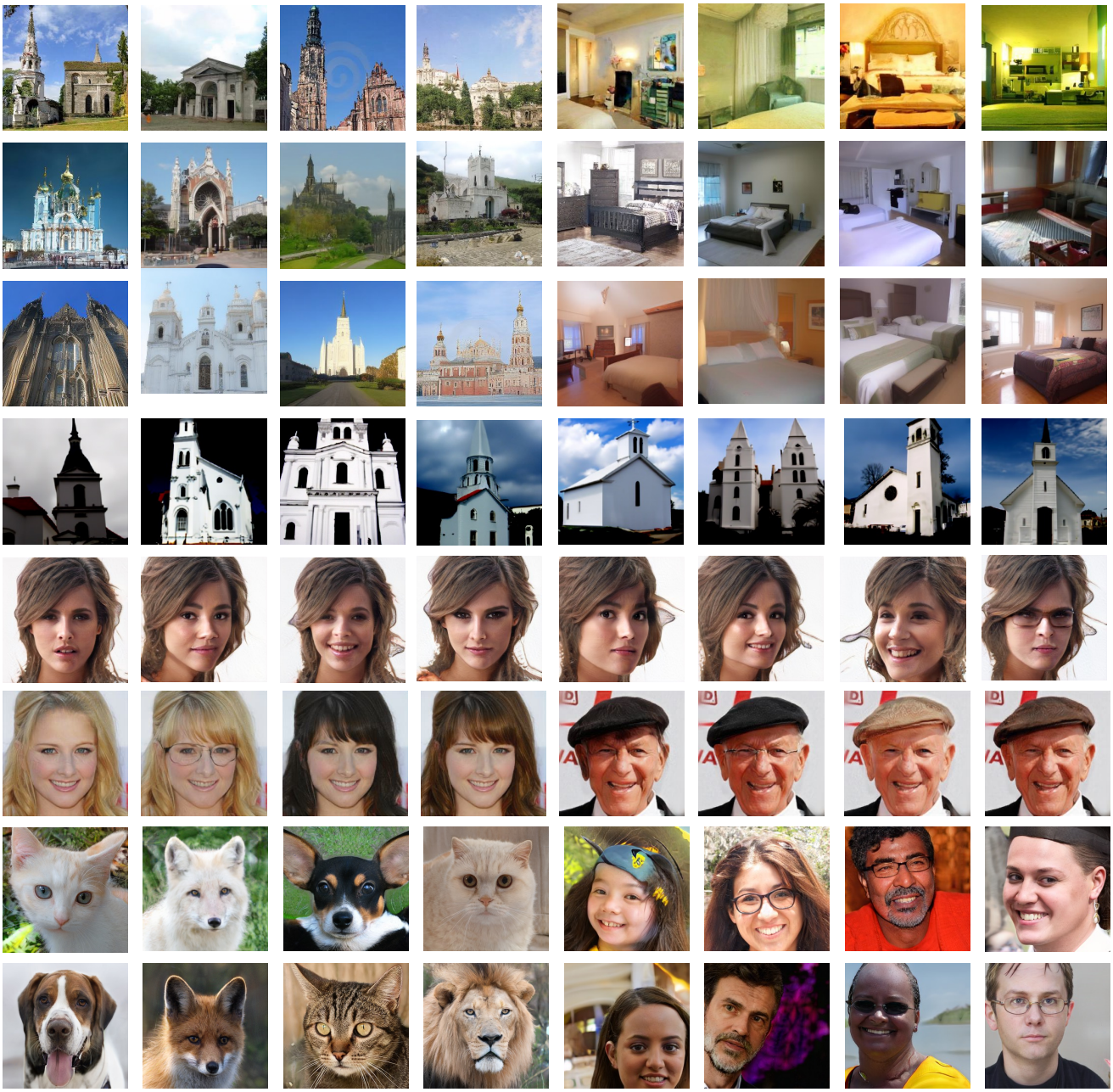

Figure 4. The samples from the proposed HiFi-IFDL dataset. From top to bottom, the images are generated by DDPM, DDIM, LDM, GDM, StarGANv2, HiSD, StyleGANv2ada, StyleGANv3.

Figure 5. Additional samples from the proposed HiFi-IFDL dataset. From top to bottom, the images are generated by STGAN, Faceshifter, and two image editing methods.

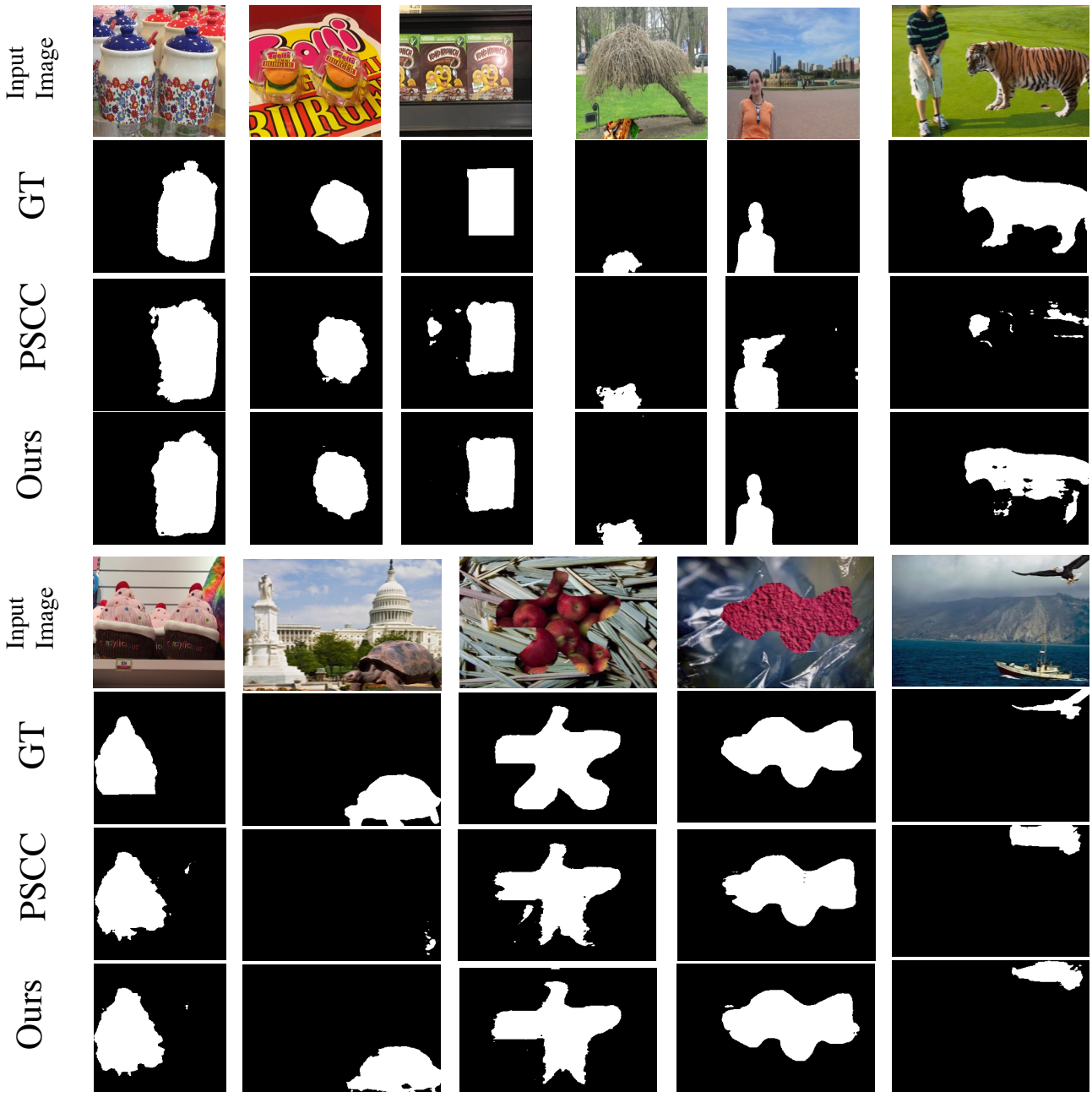

Figure 6. Additional qualitative results on *CASIA*, *NIST16* and *Coverage* dataset.

References

[1] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *CVPR*, 2020.

[2] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. 2021.

[3] Xiao Guo, Yaojie Liu, Anil Jain, and Xiaoming Liu. Multi-domain learning for updating face anti-spoofing models. In *ECCV*, 2022.

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[5] Xuefeng Hu, Zhihan Zhang, Zhenye Jiang, Syomantak Chaudhuri, Zhenheng Yang, and Ram Nevatia. Span: spatial pyramid attention network for image manipulation localization. In *ECCV*, 2020.

[6] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018.

[7] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *NeurIPS*, 2020.

[8] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *NeurIPS*, 2021.

[9] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[10] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *CVPR*, 2020.

[11] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. Faceshifter: Towards high fidelity and occlusion aware face swapping. In *CVPR*, 2020.

[12] Xinyang Li, Shengchuan Zhang, Jie Hu, Liujuan Cao, Xiaopeng Hong, Xudong Mao, Feiyue Huang, Yongjian Wu, and Rongrong Ji. Image-to-image translation via hierarchical style disentanglement. In *CVPR*, 2022.

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[14] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. 2022.

[15] Ming Liu, Yukang Ding, Min Xia, Xiao Liu, Errui Ding, Wangmeng Zuo, and Shilei Wen. Stgan: A unified selective transfer network for arbitrary image attribute editing. In *CVPR*, 2019.

[16] Xiaohong Liu, Yaojie Liu, Jun Chen, and Xiaoming Liu. Pscc-net: Progressive spatio-channel correlation network for image manipulation detection and localization. *TCSVT*, 2022.

[17] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022.

[18] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2021.

[19] Jonas Ricker, Simon Damm, Thorsten Holz, and Asja Fischer. Towards the detection of diffusion model deepfakes. *arXiv preprint arXiv:2210.14571*, 2022.

[20] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

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

[22] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. 2021.

[23] Joel Stehouwer, Hao Dang, Feng Liu, Xiaoming Liu, and Anil Jain. On the detection of digital face manipulation. In *CVPR*, 2020.

[24] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[25] Junke Wang, Zuxuan Wu, Jingjing Chen, Xintong Han, Abhinav Shrivastava, Ser-Nam Lim, and Yu-Gang Jiang. Objectformer for image manipulation detection and localization. In *CVPR*, 2022.

[26] Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Diffusion-gan: Training gans with diffusion. *arXiv preprint arXiv:2206.02262*, 2022.

[27] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.