UnionFormer: Unified-Learning Transformer with Multi-View Representation for Image Manipulation Detection and Localization

Supplementary Material

In this supplementary document, we provide details about the training and testing datasets used in the experiments. Then, we show additional comparative results. Furthermore, we include more information about implementation and analyze our method's limitations.

1. Datasets

Training datasets. The training datasets comprise two publicly available datasets and three custom-synthesized datasets:

- CASIA v2 [3]: This dataset provides spliced and copy-moved forgery images featuring various objects, which is widely utilized for model training.
- Fantastic Reality [7]: It includes many spliced images across diverse scenes, accompanied by ground truth masks.
- Tampered COCO: The images in this dataset are constructed using the COCO 2017 datasets [9]. Inspired by [8, 19], we employ the annotations in [9] to randomly copy and paste one or more arbitrary objects within the same image or to splice objects from one image into another. Random rotations and resizing operations are then applied to these images. To facilitate the Unionformer to accurately model the continuity between objects, 60% of the tampered images in this dataset contain multiple manipulated objects.
- Tampered RAISE: This dataset is constructed based on the RAISE dataset [2]. We eliminate one or several objects from an authentic image and use a GAN-based inpainting technique [18] to restore the contents. Similarly to tampered COCO, 60% of the images have multiple objects removed.
- Pristine images: These images are selected from the COCO 2017 and RAISE datasets.

To simulate the visual quality and tampering artifacts present in real-world scenarios, we randomly add Gaussian noise and apply JPEG compression on the synthetic data.

The training process of our method executed in three distinct stages. Initially, parts three, four, and five of the training set are used to train our encoding module, BSFI-Net. Drawing on the work of [12, 17], the Transformer and Convolutional Blocks within BSFI-Net undergo pre-training on the ImageNet-1K dataset. Subsequently, the synthesized COCO datasets, including both tampered and authentic samples from COCO 2017 are utilized to train the Region Proposal Network (RPN). Finally, the entire training datasets, comprising five parts, is employed to train the complete UnionFormer. We perform equal sampling from every part in each training epoch to eliminate bias caused by the varying scales of different parts in the training dataset.

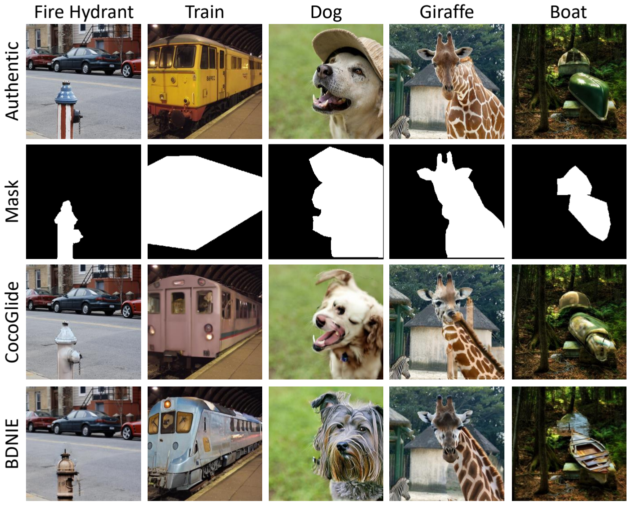

Figure 1. Some examples from the CocoGlide and BDNIE datasets with the same original images, reference masks, and guided prompts. The guided texts are annotated above the first line.

Testing datasets. To comprehensively evaluate the performance of our model, we employ five commonly traditional datasets: CASIA v1[3], Columbia[6], Coverage[15], NIST16[4], and IMD20[11], as well as two challenging diffusion-based datasets: CoCoGlide and BDNIE. CocoGlide, created by [5], consists of 512 images generated from the COCO 2017 validation set using the GLIDE model. We constructed BDNIE dataset, comprising 512 hyerrealistic fake images generated by the advanced bended Diffusion model [1] for text-driven natural image editing. In BDNIE, we adopt the same forgery regions and guided text prompts as CocoGlide.

Figure 1 displays a comparison of examples from the CoCoGlide and BDNIE datasets. Although both datasets are based on diffusion models, BDNIE undergoes a global diffusion process and spatially blends noised versions of the input image with the local text-guided diffusion latent at a progression of noise levels, seamlessly integrating the edited region with the unchanged parts [1]. Consequently,

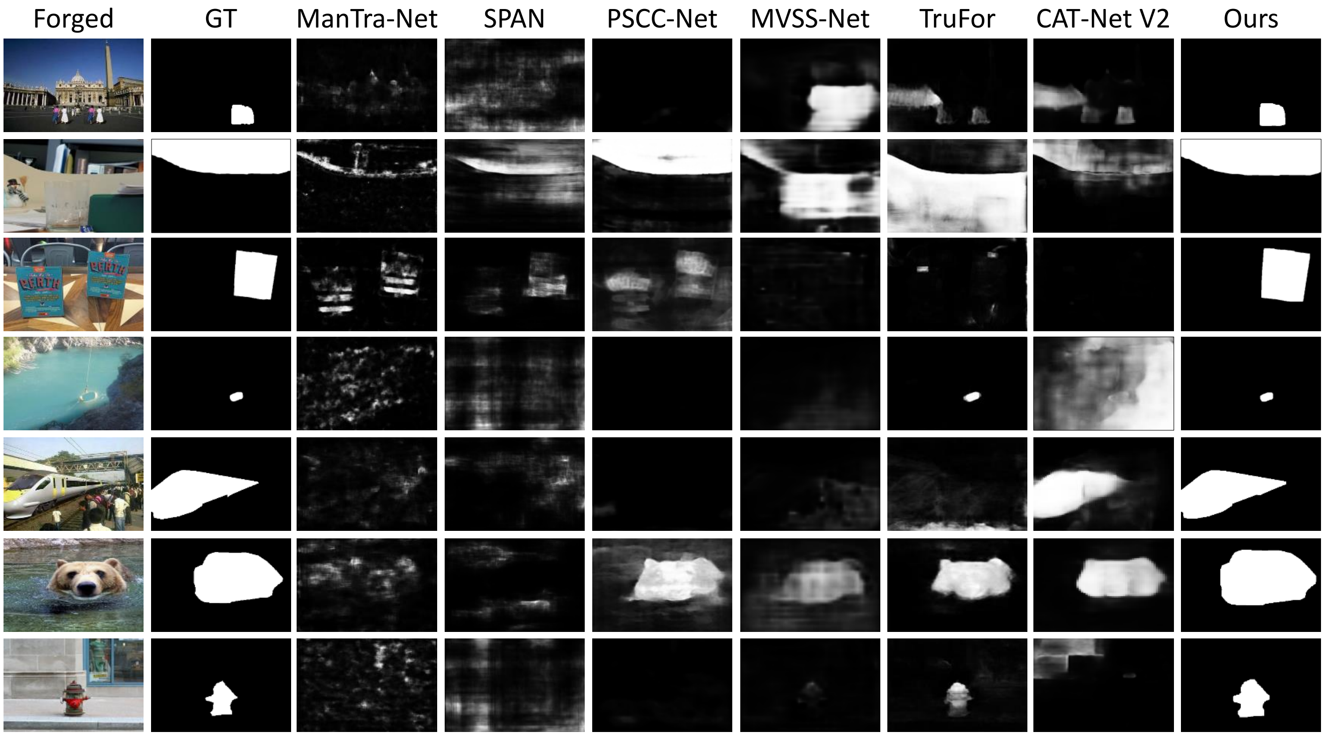

Figure 2. Some qualitative comparison results with the state-of-the-art methods. The forgery images, from top to bottom, are respectively from CASIA v1, Columbia, Coverage, NIST16, IMD20, CocoGlide and BDNIE.

	MVSS-Net	CAT-Net v2	ObjectFormer	TruFor	Ours
F1	0.515	0.547	-	0.624	0.632
AUC	-	-	0.884	0.927	0.929
Params (M)	142.79	114.26	257.97	262.05	210.63
FLOPs (G)	327.14	314.30	402.80	519.91	392.82
Training Data (K)	96.60	875.50	-	900.25	832.50

Table 1. Comparison of computing costs and dataset sizes.

	ManTra-Net	SPAN	MVSS-Net	PSCC-Net	CAT-Net v2	TruFor	Ours
optimal	0.620	0.324	0.606	0.662	0.587	0.699	**0.738**
fixed (0.5)	0.481	0.272	0.447	0.503	0.415	0.508	**0.531**

Table 2. Results of pixel-level F1 with optimal and fixed threshold on the BDNIE dataset.

the images in BDNIE appear more realistic and exhibit fewer tampering artifacts, such as traces around editing boundaries and inconsistencies between different regions.

2. Additional comparative results

Qualitative results. In Figure 2, we compare additional tampering localization results from all seven testing datasets with the state-of-the-art methods. These comparative examples illustrate that our localization results are more accurate than other methods, with more precise edges and fewer false alarms for the real regions. Moreover, our method also achieves satisfactory performance on two challenging datasets based on the Diffusion models, while most other methods tend to fail.

Computing overhead and training data. Table 1 compares the training set size and computational cost of different methods. We include pixel-level F1 and AUC scores for image manipulation localization tasks to understand model efficacy better. To compare the computational cost between models, we utilize Floating Point Operations (FLOPs) and the number of model parameters as evaluation criteria. The data are obtained based on models released by the authors. Compared to those relying solely on convolutional neural networks, transformer-based models attain higher accuracy but also demand more computational resources. As shown in Table 1, our approach has less computing overhead than other transformer-based methods. With comparable performance, we utilized less training data.

Quantitative comparison on the BDNIE dataset. To further analyze the detection capabilities of our method for identifying diffusion-based tampering, we compare its performance with other methods on the BDNIE dataset, as shown in Table 2. This evaluation focuses on the pixel-level F1 scores at optimal and fixed thresholds. Our method attained

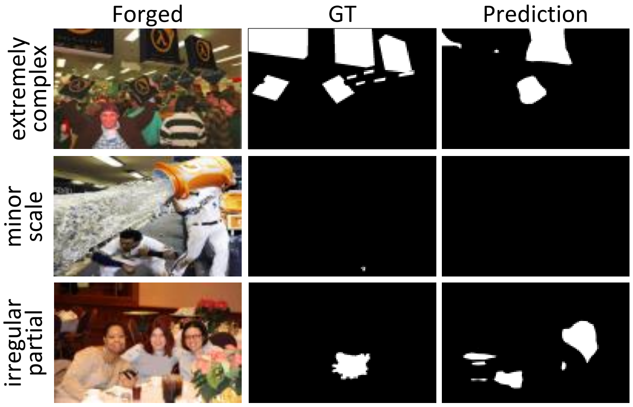

Figure 3. Some failure cases of our methods.

superior results by leveraging the continuity in modeling relationships between objects within an image.

3. Implementation Details

We sequentially employ the cross-entropy loss, the loss proposed by Faster R-CNN [13], and the unified loss $\mathcal{L}_{union}$ introduced in the main paper to train BSFI-Net, RPN, and the complete UnionFormer. We trained the BSFI-Net for 100 epochs using the AdamW optimizer [10], with a batch size of 512 and a weight decay of 0.05. The initial learning rate is set to 0.001 and decayed following a cosine schedule. The Intersection-over-Union (IoU) threshold for positive examples (potentially manipulated regions) in the PRN is set to 0.7, while for negative examples (authentic regions), it is set to 0.3. To train the RPN, we employ SGD with a momentum of 0.9 for optimization. The initial learning rate is set to 0.001 for the first 60K iterations and then reduced to 0.0001 for the subsequent 40K iterations. In the training of the complete UninonFormer, inspired by [14, 16], we adopt a 36-epoch (3×) schedule, where we train the Unionformer for 2.7×10^5 iterations with a batch size of 16. An AdamW optimizer is also used in this stage, with the learning rate initially set to 10^{-4} and then multiplied by 0.1 at 1.8×10^5 and 2.4×10^5 iterations. Following [14], the learning rate warm-up is applied in the first 1000 iterations, and the weight decay is set to 0.0001.

4. Limitation

The limitations of our method primarily lie in three scenarios: 1) extremely complex multi-object manipulation, which fails to deconstruct complex object relationships; 2) minor-scale tampering regions; and 3) irregular non-component partial modifications. Note that our method can accurately locate most regular partial modifications because the proposals generated by RPN cover these components. Figure 3. illustrates some examples of localization failures.

References

[1] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM Transactions on Graphics (TOG)*, 42 (4):1–11, 2023. 1

[2] Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato. Raise: A raw images dataset for digital image forensics. In *Proceedings of the 6th ACM Multimedia Systems Conference*, page 219–224, New York, NY, USA, 2015. Association for Computing Machinery. 1

[3] 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, 2013. 1

[4] Haiying Guan, Mark Kozak, Eric Robertson, Yooyoung Lee, Amy N. Yates, Andrew Delgado, Daniel Zhou, Timothee Kheyrkhah, Jeff Smith, and Jonathan Fiscus. Mfc datasets: Large-scale benchmark datasets for media forensic challenge evaluation. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 63–72, 2019. 1

[5] 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

[6] 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, 2006. 1

[7] Vladimir V. Kniaz, Vladimir Knyaz, and Fabio Remondino. The point where reality meets fantasy: Mixed adversarial generators for image splice detection. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 1

[8] 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 (WACV)*, pages 375–384, 2021. 1

[9] 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 *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. 1

[10] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 3

[11] 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 (WACV) Workshops*, 2020. 1

[12] Zhiliang Peng, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye. Conformer: Local features coupling global representations for visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 367–376, 2021. 1

[13] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 2015. 3

[14] Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris M Kitani. Rethinking transformer-based set prediction for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3611–3620, 2021. 3

[15] Bihan Wen, Ye Zhu, Ramanathan Subramanian, Tian-Tsong Ng, Xuanjing Shen, and Stefan Winkler. Coverage — a novel database for copy-move forgery detection. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 161–165, 2016. 1

[16] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019. 3

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

[18] Haitian Zheng, Zhe Lin, Jingwan Lu, Scott Cohen, Eli Shechtman, Connelly Barnes, Jianming Zhang, Ning Xu, Sohrab Amirghodsi, and Jiebo Luo. Image inpainting with cascaded modulation gan and object-aware training. In *Computer Vision – ECCV 2022*, pages 277–296, Cham, 2022. Springer Nature Switzerland. 1

[19] Peng Zhou, Xintong Han, Vlad I. Morariu, and Larry S. Davis. Learning rich features for image manipulation detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1