-Supplementary Document-
TruFor: Leveraging all-round clues for trustworthy image forgery detection and localization

Fabrizio Guillaro[1] Davide Cozzolino[1] Avneesh Sud[2] Nicholas Dufour[2] Luisa Verdoliva[1]

[1]University Federico II of Naples [2]Google Research

In this supplementary document, we report the details of our approach (Sec. 1) and of the datasets used in the experiments (Sec. 2). Then, we include additional results to prove the robustness capability of our method (Sec. 3) and its ability to provide good results also for the detection task (Sec. 4). Furthermore, we show qualitative results by means of localization and confidence maps and finally we present failure cases (Sec. 5). Code is publicly available at `https://grip-unina.github.io/TruFor/`.

1. Implementation details

Architecture. The anomaly localization network is shown in Fig. 2. The feature extraction backbone in the encoder is based on a transformer-based segmentation architecture [22]. The RGB and the Noiseprint++ feature maps are combined using a Cross-Modal Feature Rectification Module (CM-FRM) [14]. Each feature extraction branch has 4 Transformer blocks, and a CM-FRM block between each transformer block. The Transformer blocks are based on the Mix Transformer encoder B2 (MiT-B2) proposed for semantic segmentation and are pretrained on ImageNet, as suggested in [22]. The Mix Transformer encoder includes self-attention mechanisms and channel-wise operations. It relies on spatial convolutions and not on positional encodings. This is important in order to work with images of any size and to obtain a localization map with the same resolution as the input image.

The CM-FRM block exploits the interactions between the image semantic (RGB) and residual (Noiseprint++) features. It performs channel-wise and spatial-wise rectifications, which consists of a weighted sum of the feature map of both branches. The weights are calculated along the channel dimension and the spatial dimension separately, combining both feature maps. The Feature Fusion Module (FFM) uses an efficient cross-attention mechanism, without positional encoding, to merge the feature maps of Noiseprint++ and RGB image and the outputs of the four FFMs represent the input of the decoder. We use the All-MLP decoder proposed in [22], which is a lightweight architecture formed by only 1×1 convolution layers and bilinear up-samplers. The decoder for the confidence map has the same All-MLP architecture. The forgery detector network takes as input the pooled features from anomaly and confidence maps, and consists of 2 fully connected layers with RELU activation: $8D \rightarrow 128D \rightarrow 1D$ output.

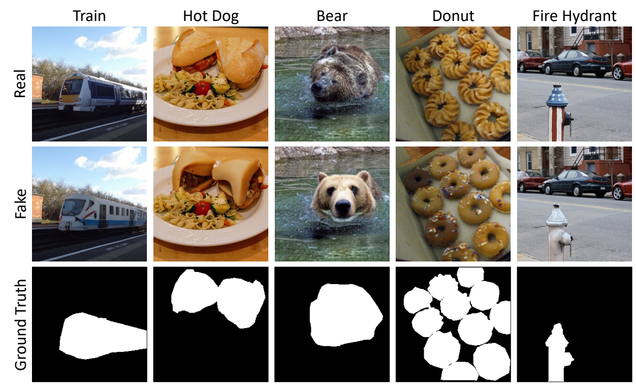

Figure 1. Some examples of real and manipulated images and related reference maps from the CocoGlide dataset. For each image we indicate the prompt that drives the synthetic generation.

Experiments have been conducted using one NVIDIA RTX A6000 GPU. Training times for each phase are 6.5 days, 6 days, 2 days, respectively. The inference time is about 1.17 sec for an image of 3.2 megapixels. As for the model size, the number of parameters for TruFor is 68.7M, that are less than those used by the top three competitors: CAT-Net v2 (114.3M), MVSS-Net (146.9M) and IF-OSN (128.8M).

Noiseprint++ training. For Noiseprint++ training, each batch includes 160 patches of 64×64 pixels. These patches are obtained from 5 camera models and 4 different images for each camera model. The resulting 20 images are subjected to 4 different editing histories, which are a combination of random resizing, compression and con-

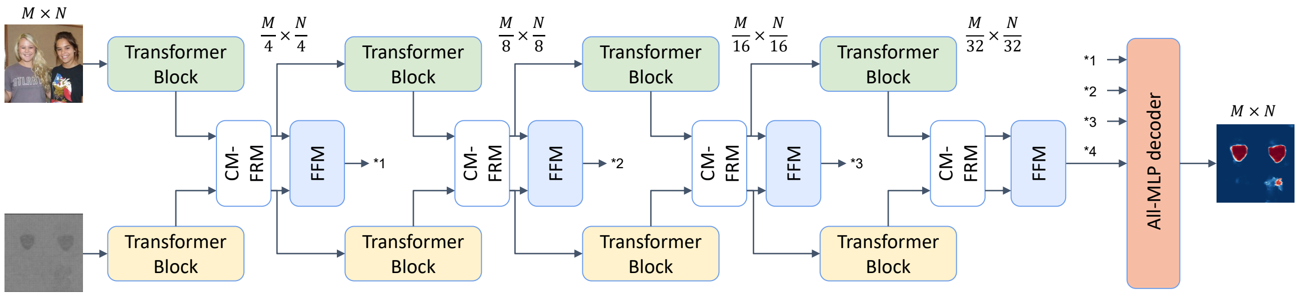

Figure 2. Anomaly localization network.

trast/brightness adjustments, for a total of 512 possible editing histories. Training is performed for a total of 50 epochs, and each epoch includes 8800 training steps. An Adam optimizer is used with an initial learning rate of 0.001, that is reduced by 10 times every 10 epochs.

Localization and detection training. For localization and detection tasks we adopted the datasets used for training and validation also used in [11], which comprises both pristine and fake images with the corresponding reference maps. The input image is cropped to 512×512 during training. Details of the dataset are reported in Tab. 1. To avoid biases due to an imbalance in training dataset size, we sample each dataset equally for each training epoch. The networks are trained for 100 epochs with a batch size equal to 18 and a learning rate that starts with 0.005 and decays to zero. An SGD optimizer is used with a momentum of 0.9. Before Noiseprint++ extraction, we apply the following augmentations on RGB inputs: resizing in the range [0.5 - 1.5] and JPEG compression with quality factor from 30 to 100.

2. Datasets

To ensure that Noiseprint++ is trained on unaltered images, we verified that for each camera model, all collected images have the same resolution, are in JPEG format with the same quantization matrix and that no photo editing software is present in the metadata (e.g. photoshop, gimp).

As for the anomaly localization and detection, the datasets used for training and testing are reported in Tab. 1. Training includes CASIA v2 [6], FantasticReality [10], IMD2020 [17] and a dataset of manipulated images created by [11] by applying splicing and copy-move using either COCO [13] training set or RAISE [3] as a source and object masks from COCO as target regions. For OpenForensics [12] and NIST16 [7], we evaluate the performance on a test subset of 2000 images (out of 19,000) and 160 images, respectively. The latter choice follows the common train/test split that most of the recent works apply [9, 18, 23, 24]. CocoGlide is a manipulated dataset generated by us using the COCO validation dataset [13]. We extract 256×256 pixel crops and then use an object mask and its corresponding label as the forgery region and the text prompt that are fed to GLIDE [16]. In this way, we generated new synthetic objects of the same category for a total of 512 manipulated images. Some examples are shown in Fig. 1. Note that we avoided overlap with [11], since CocoGlide is based on images from the validation set, while the tampered COCO dataset from the training set.

3. Additional robustness analysis

In this Section we include additional experiments to show the ability of our method to be robust to different forms of degradations and compare them with those obtained by the top performers [2, 11, 21]. We apply the following transformations on the CASIA v1 dataset: gaussian blur (varying the kernel size), gaussian noise (varying the standard deviation), gamma correction (varying the power

	Number of images		**Manipulation**	
Name [ref]	Real	Fake	Sp	CM
CASIA v2 [6]	7491	5105	✓	✓
FantasticReality [10]	16592	19423	✓	
IMD2020 [17]	414	2010	✓	✓
tampered COCO [11]	-	400K	✓	✓
tampered RAISE [11]	24462	400K		✓
CASIA v1+ [5]	800	921	✓	✓
Coverage [20]	100	100		✓
Columbia [8]	183	180	✓	
NIST16 [7]	160	160	✓	✓
DSO-1 [4]	100	100	✓	
VIPP [1]	69	69	✓	✓
OpenForensics [12]	-	2000	✓	
CocoGlide	512	512	✓	

Table 1. List of datasets used for training and testing (Sp=splicing, CM=copymove).

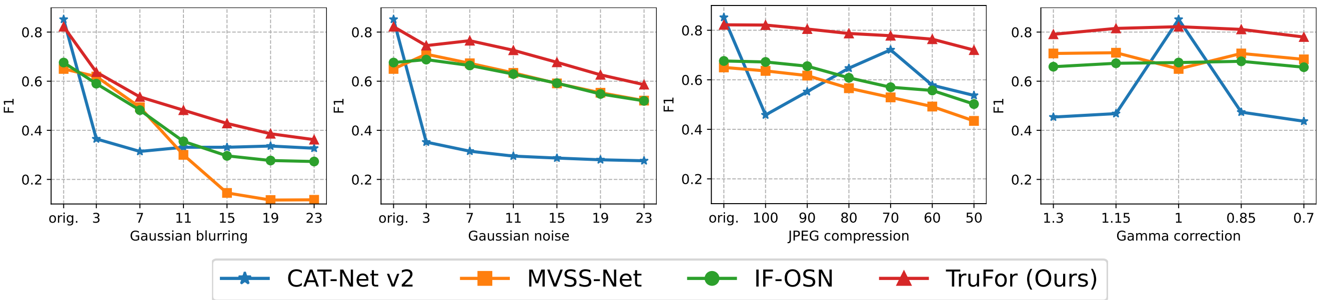

Figure 3. Robustness analysis against different processing operations on CASIA v1. Pixel-level F1 performance (best threshold) is shown.

Method	CASIA v1				Columbia				DSO-1				NIST16				AVG			
	Fb	Wa	Wb	Wc	Fb	Wa	Wb	Wc	Fb	Wa	Wb	Wc	Fb	Wa	Wb	Wc	Fb	Wa	Wb	Wc
IF-OSN [21]	.513	.524	.507	.454	.741	.752	.756	.760	.484	.395	.416	.414	.315	.302	.292	.282	.513	.493	.493	.478
CAT-Net v2 [11]	.681	.508	.469	.206	**.964**	**.952**	**.958**	**.903**	.310	.247	.240	.237	.219	.238	.243	.244	.544	.486	.478	.398
MVSS-Net [2]	.469	.444	.480	.339	.752	.747	.758	.752	.356	.308	.354	.329	.305	.252	.300	.269	.471	.438	.473	.422
TruFor (ours)	**.716**	**.713**	**.676**	**.615**	.797	.798	.835	.820	**.685**	**.465**	**.515**	**.469**	**.338**	**.384**	**.308**	**.358**	**.634**	**.590**	**.584**	**.566**

Table 2. Pixel-level F1 performance (fixed threshold) on datasets uploaded on Facebook (Fb), WhatsApp (Wa), Weibo (Wb), WeChat (Wc).

Method	Columbia	Coverage	CASIA v1	NIST16	Avg
ManTraNet	.824	.819	.817	.795	.814
SPAN	.936	.922	.797	.840	.874
PSCCNet	**.982**	.847	.829	.855	.878
ObjectFormer	.955	**.928**	.843	.872	.900
TruFor	.947	.925	**.957**	**.877**	**.927**

Table 3. Pixel-level AUC, for the comparisons the values are taken from Tab. 1 of [18]

factor) and JPEG compression (varying the quality level). The results are shown in Fig. 3. We can observe that our method is more robust than the state-of-the-art irrespective of the type of degradation.

We also check robustness to other social media networks, beyond those already considered in the main paper, i.e. Facebook and Whatsapp (Tab. 4 in main paper). More specifically, we use the whole dataset proposed in [21], where images from some standard forensic datasets, CASIA v1[1] [6], Columbia [8], DSO-1 [4] and NIST16 [7], were also uploaded on Weibo and WeChat. Results are presented in Tab. 2 and show a consistent gain over all the different datasets and social platforms except on Columbia, where CAT-Net v2 achieves better performance. On average however, we have a gain of around 16%, 19%, 18% and 18% with respect to the second best on Facebook, Whatsapp, Weibo and WeChat, respectively.

Comparison with ObjectFormer. Note that an exhaustive and equitable comparison with [18] is not feasible as they do not provide their trained model. We provide a pixel-level comparison of localization performance in Tab. 3 using values for [15, 18] from the paper. Our method is competitive or better than [18] across various test datasets, and outperforms that on average.

4. Additional detection results

In this Section, we give some more insights on the image level detection performance of our method. We first investigate the role of the confidence map in the detection strategy. In Tab. 4, we perform an ablation where we observe substantial improvements with the confidence maps both in terms of AUC and Accuracy.

Image-level metrics require calibrating the detection score for a particular dataset (or certain methods fine-tune on specific datasets [18]). In Tab. 2 of the main paper, we report the balanced accuracies evaluated on seven datasets, and the average of them considering a fixed threshold equal to 0.5. For methods that do not provide an explicit detection score, we use max pooling on the localization map. In Fig. 4, we show the accuracy (averaged over the seven datasets) as a function of the threshold. One can observe the accuracy of other methods, which rely on max pooling, increase with higher thresholds - this is indicative of many false positives in the localization maps from these methods. In contrast, our method combines various confidence weighted pooling statistics, making it more robust.

Table 5 shows the true negative rate (TNR), true positive rate (TPR), and average accuracy considering both a fixed threshold of 0.5 and the best threshold for each technique.

[1] Actually, we used the v1+ version [5], where real images of v1 (shared with v2 and present in our training set) are replaced by images from the COREL dataset [19].

	Original		Res		Res&Cmp	
	AUC	Acc	AUC	Acc	AUC	Acc
w/o conf. map	.877	.785	.847	.730	.719	.610
w conf. map	**.996**	**.905**	**.949**	**.910**	**.740**	**.675**

Table 4. Ablation image-level results in terms of AUC and accuracy considering the use (or not) of the confidence map.

We can notice that using a fixed threshold with our method we can significantly decrease the false alarms rate (around 80% lower) at the cost of increasing miss detection (around 30% higher), by achieving an average improvement in terms of accuracy of 25%. All the state-of-the-art approaches have the problem of a high number of false alarms with a best threshold that assumes values almost equal to 1. Also in this experiment where results are averaged on all seven datasets, we can appreciate the importance to include the confidence analysis during detection.

5. Qualitative results

In Fig. 5 we show some results on fake and pristine images together with the relative confidence map and the final integrity score. We can see that the confidence map can help to correct false positive predictions and provide a more reliable integrity score. Instead, in Fig. 6 different failure cases. In the first row, the manipulation was correctly localized, however, the confidence map wrongly hints that it could be a false alarm. A possible explanation is that the area is very uniform, which can lead to false positives. A similar situation is presented in the second row, since the plant has a very uniform and dark texture, which misleads the confidence extractor. Another failure case can be represented by the other way around, where we have a false positive on a pristine image, and the confidence map not correcting it.

In Fig. 7 we show some qualitative results on manipulated images (the forged area is outlined in yellow) and compare with the state-of-the-art. For these examples, the localized area appears sharper and more accurate than the other methods. We also add the confidence maps that can tell us the level of reliability of the anomaly maps and remove potential false alarms. Note the dark regions on the boundary of real forgeries - indicating lower confidence in the anomaly label assignment of intermediate regions.

Finally, in Fig. 8, we show a few examples of false alarms on pristine images. Other methods tend to focus on semantically relevant or highly saturated regions leading to false detections. TruFor's localization maps exhibit a weaker response, and most of these are discarded due to the confidence map, leading to a correct image-level decision.

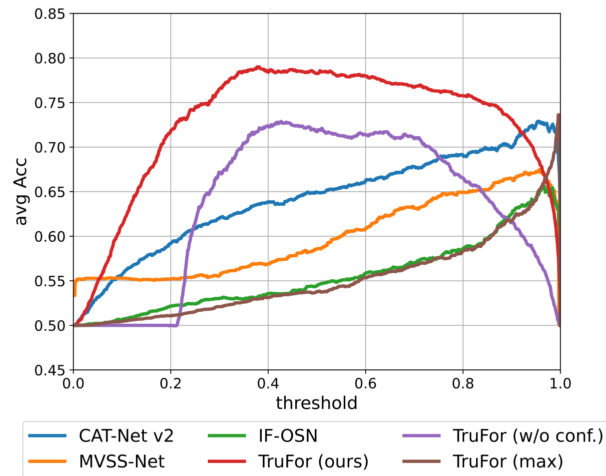

Figure 4. Image-level Detection Accuracy as a function of detection score threshold (averaged over 7 test datasets).

	fixed			best			
	TNR	TPR	Acc	th	TNR	TPR	Acc
CAT-Net v2	.416	.882	.649	.955	.840	.618	.729
IF-OSN	.182	.907	.545	.968	.763	.548	.656
MVSS-Net	.285	.886	.586	.957	.806	.544	.675
TruFor (max)	.109	**.967**	.538	.996	**.900**	.573	.736
TruFor (w/o c.)	.859	.575	.717	.427	.818	.640	.729
TruFor	**.909**	.656	**.783**	.380	.851	**.729**	**.790**

Table 5. Detection results: Image-level TNR, TPR, and Accuracy averaged on seven datasets (fixed and best threshold).

References

[1] Tiziano Bianchi and Alessandro Piva. Image Forgery Localization via Block-Grained Analysis of JPEG Artifacts. *IEEE Transactions on Information Forensics and Security*, 7(3):1003–1017, 2012. 2

[2] Xinru Chen, Chengbo Dong, Jiaqi Ji, Juan Cao, and Xirong Li. Image manipulation detection by multi-view multi-scale supervision. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021. 2, 3

[3] 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*, MMSys '15, page 219–224, 2015. 2

[4] Tiago José de Carvalho, Christian Riess, Elli Angelopoulou, Hélio 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. 2, 3

[5] Chengbo Dong, Xinru Chen, Ruohan Hu, Juan Cao, and Xirong Li. MVSS-Net: Multi-View Multi-Scale Supervised Networks for Image Manipulation Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3539–3553, 2023. 2, 3

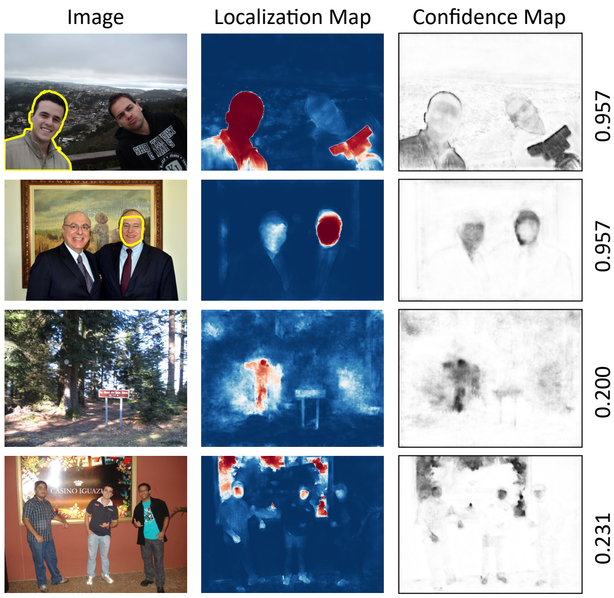

Figure 5. Examples of forged (top) and pristine (bottom) images (forgeries are highlighted in yellow). We show the localization map, the confidence map and the integrity score. For real images despite the anomaly map presenting some false alarms, the confidence analysis helps to make a correct prediction.

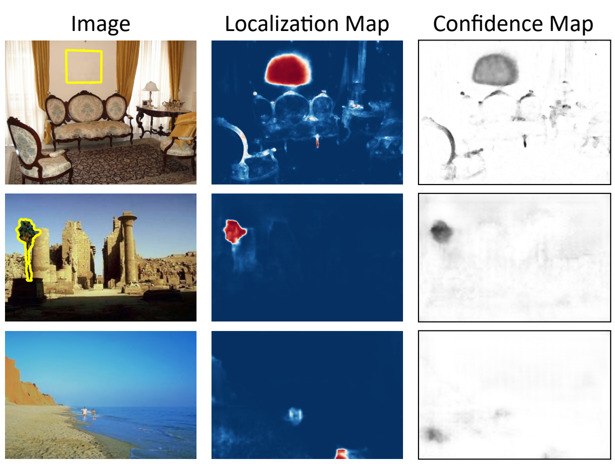

Figure 6. Examples of failure cases on fake and real images (forgeries are highlighted in yellow).

[6] Jing Dong, Wei Wang, and Tieniu Tan. CASIA image tampering detection evaluation database. In *IEEE China Summit and International Conference on Signal and Information Processing*, pages 422–426, 2013. 2, 3

[7] H. Guan, M. Kozak, E. Robertson, Y. Lee, A.N. Yates, A. Delgado, D. Zhou, T. Kheyrkhah, J. Smith, and J. Fiscus. MFC datasets: Large-scale benchmark datasets for media forensic challenge evaluation. In *IEEE WACV Workshops*, pages 63–72, 2019. 2, 3

[8] Yu-feng Hsu and Shih-Fu Chang. Detecting image splicing using geometry invariants and camera characteristics consistency. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 549–552, 2006. 2, 3

[9] Xuefeng Hu, Zhihan Zhang, Zhenye Jiang, Syomantak Chaudhuri, Zhenheng Yang, and Ram Nevatia. SPAN: Spatial pyramid attention network for image manipulation localization. In *European Conference on Computer Vision (ECCV)*, pages 312–328. Springer, 2020. 2

[10] 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 (NeurIPS)*, volume 32, 2019. 2

[11] 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*, pages 1–21, 2022. 2, 3

[12] 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 *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10117–10127, October 2021. 2

[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 *European Conference on Computer Vision (ECCV)*, pages 740–755, 2014. 2

[14] Huayao Liu, Jiaming Zhang, Kailun Yang, Xinxin Hu, and Rainer Stiefelhagen. CMX: Cross-Modal Fusion for RGB-X Semantic Segmentation with Transformers. *arXiv preprint arXiv:2203.04838*, 2022. 1

[15] 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, november 2022. 3

[16] Alexander Quinn 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 *International Conference on Machine Learning*, volume 162, pages 16784–16804, Jul 2022. 2

[17] Adam Novozamsky, Babak Mahdian, and Stanislav Saic. IMD2020: A large-scale annotated dataset tailored for detecting manipulated images. In *IEEE/CVF WACV Workshops*, 2020. 2

[18] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022. 2, 3

[19] 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. 3

Figure 7. Some qualitative results, compared with the state-of-the-art, on manipulated images (the forged area is outlined in yellow). Dark regions in the confidence map indicate regions of low confidence in the TruFor localization map.

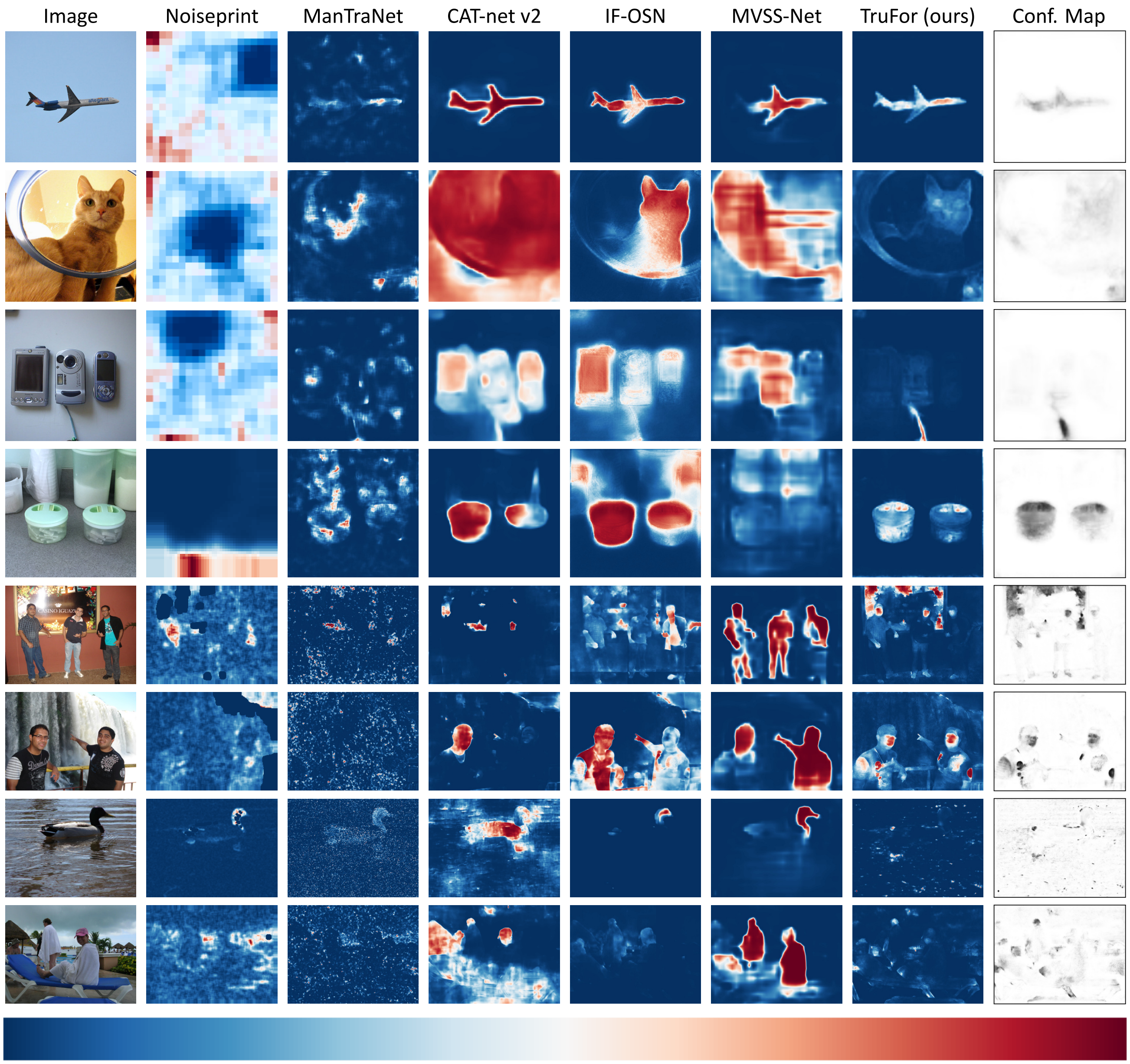

Figure 8. Some qualitative results, compared with the state-of-the-art, on pristine images. Dark regions in the confidence map indicate regions of low confidence in the TruFor localization map.

[20] Bihan Wen, Ye Zhu, Ramanathan Subramanian, Tian-Tsong Ng, Xuanjing Shen, and Stefan Winkler. COVERAGE — A novel database for copy-move forgery detection. In *IEEE International Conference on Image Processing (ICIP)*, pages 161–165, 2016. 2

[21] Haiwei Wu, Jiantao Zhou, Jinyu Tian, and Jun Liu. Robust image forgery detection over online social network shared images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022. 2, 3

[22] 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 (NeurIPS)*, 34:12077–12090, 2021. 1

[23] Chao Yang, Huizhou Li, Fangting Lin, Bin Jiang, and Hao Zhao. Constrained R-CNN: A General Image Manipulation Detection Model. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2020. 2

[24] Peng Zhou, Xintong Han, Vlad I. Morariu, and Larry S. Davis. Learning rich features for image manipulation detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2