Supplementary material: Fusion Transformer with Object Mask Guidance for Image Forgery Analysis

Dimitrios Karageorgiou[1] Giorgos Kordopatis-Zilos[2] Symeon Papadopoulos[1]
[1]Information Technologies Institute, CERTH
[2]VRG, FEE, Czech Technical University in Prague
{dkarageo,papadop}@iti.gr kordogeo@fel.cvut.cz

1. Experimental Setup

1.1. Image Forensics Algorithms

In our evaluation, we consider only methods with publicly available code in order to fairly evaluate them under the same setup. We employ as inputs to our score-level fusion network, OMG-Fuser$_S$, the SPAN [11], MVSS-Net++ [2], CATNetv2 [17], TruFor [8] and IFOSN [33] algorithms, due to their competitive performance and their complementarity with respect to the type of artifacts they detect, i.e., artifacts in the RGB domain, edge artifacts, compression artifacts, noise artifacts, and robustness against online sharing operations, respectively. Moreover, we utilize the Noiseprint++ [8] and the DCT-domain stream [17] as inputs to our feature-level fusion network, OMG-Fuser$_F$, two learnable forensic signals that captures noise-related and compression-related artifacts, respectively. For all the aforementioned methods, we utilize the official code implementations provided by the original authors.

1.2. Datasets

To train our OMG-Fuser models, we utilize data from three publicly available datasets. We randomly sample 25k forged and 25k authentic images from the synthetic dataset used by CAT-Net [16] due to the big variability of its samples regarding compression qualities and depicted topics. We enrich this set with another 10k inpainted images sampled from the DEFACTO [24] dataset. Also, we include all images from the CASIAv2 [3] dataset into our training set to compensate for low-resolution and low-quality images. We utilize 90% of these data for the actual training of the model and the rest 10% for validation purposes. Despite some datasets consisting of more samples, we observed in our experiments that a further increase in the amount of training data did not yield any significant performance increase. Thus, our architecture requires significantly less training data than the previous state-of-the-art ones [8, 17]. Instead, as highlighted by [8], introducing more variability into the training data was more beneficial. For evaluation,

	Dataset	Forged	Authentic	Types
Train	**Tampered-50k** [16]	25k	25k	SP, CM
	DEFACTO-INP [24]	10k	-	INP
	CASIAv2 [3]	5k	7k	SP, CM
Test	**CASIAv1+** [2]	920	800	SP, CM
	Columbia [10]	180	183	SP
	COVERAGE [30]	100	100	CM
	NIST16 [7]	564	560	SP, CM, INP
	OpenForensics [18]	19k	-	SP
	CocoGlide [8]	512	512	INP
	DID [32]	6k	-	INP

Table 1. Number of samples and types of forgery included in the train and test datasets. SP stands for splicing, CM for copy-move and INP for inpainting.

following [8, 11, 21], we have selected five popular datasets, namely the CASIAv1+ [2], COLUMBIA [10], COVERAGE [30], NIST16 [7], OpenForensics [18] datasets, including common cases of image forgery. In contrast to previous works [11, 21], we benchmark our methods on the entire evaluation dataset without making any further assumptions about the type of forgery or utilizing subsets of them. Furthermore, to take into account deep-learning based inpainting operations, we further employed the recently introduced CocoGlide [8] dataset and the deep-learning based inpaintings of the DiverseInpaintingDataset (DID) [32]. A summary of all the datasets used in our research is presented in Tab. 1.

1.3. Implementation Details

We implement and train all of our models using PyTorch [27]. Following [25, 31], we train our models for 100 epochs using the SGD optimizer with momentum [22] set to 0.9. We initialize the learning rate to 10^{-3}, with 5 epochs of linear warm-up and a cosine decay until 10^{-6}. We empirically tune the weights of the optimization criterion and set them to $a = 0.3$, $b = 0.45$, $c = 0.25$. To acquire the instance segmentation masks, we utilize the Segment Any-

	Approach	CASIAv1+	Columbia	Coverage	NIST16	OpenFor.	CocoGl.	DID	Overall
Feature Fusion	**PSCC-Net** [21]	83.4	86.7	69.3	50.2	51.3	84.1	58.5	70.7
	SPAN [11]	64.2	88.6	78.6	56.6	39.0	82.7	45.9	71.3
	IFOSN [33]	87.4	86.9	63.4	71.1	49.3	79.9	37.8	73.4
	MVSSNet++ [2]	80.0	81.6	80.0	73.1	48.1	83.3	41.5	75.1
	CATNetv2 [17]	87.6	<u>91.7</u>	79.0	68.9	66.3	80.3	**80.0**	77.2
	TruFor [8]	<u>89.6</u>	90.5	<u>83.9</u>	<u>74.5</u>	<u>71.2</u>	<u>85.6</u>	66.5	<u>77.8</u>
	OMG-Fuser$_F$ (Ours)	**92.0**	**94.6**	**88.3**	**82.1**	**82.0**	**87.7**	<u>78.9</u>	**80.1**
Score Fus.	**DST-Fusion** [6]	89.3	91.8	76.6	56.0	33.3	86.5	64.8	77.4
	OW-Fusion [1]	89.1	<u>94.7</u>	82.0	72.5	66.9	84.2	<u>78.2</u>	81.9
	AVG-Fusion	<u>90.9</u>	94.4	<u>87.8</u>	<u>80.4</u>	<u>70.9</u>	**87.8**	77.5	<u>84.3</u>
	OMG-Fuser$_S$ (Ours)	**92.2**	**96.6**	**89.3**	**83.7**	**85.2**	<u>86.2</u>	**86.1**	**86.8**

Table 2. Comparison on image forgery localization. Pixel-level F1 scores, calibrated with the best threshold per image, are presented for each algorithm and dataset. The best value per column is highlighted in bold, and the second best is underlined.

thing Model (SAM) [13], a zero-shot model that is not limited to a fixed set of object classes. Furthermore, for the RGB stream, we employ the DINOv2 model [26], trained in an unsupervised manner on a large curated dataset and capable of extracting rich features suitable for a large number of downstream tasks. We utilize its ViT-S/14 variant. During the training of score-level fusion models, DINOv2 remains frozen and the resolution of inputs to all streams is 224×224. For feature-level fusion models, we increase the input resolution of all streams to 448×448 and fine-tune the DINOv2 backbone in order to capture low-level cues in finer detail. Following [15, 19], we freeze the patch-embedding layer during fine-tuning. For the computation of the input signals, the image is provided in its original resolution to all the respective algorithms.

The number of layers of each stage is set to $B_1 = B_2 = B_3 = 6$. Regarding the localization head, it consists of five upsampling layers, each including a transposed convolution [23], a ReLU [14] activation, and a Batch Normalization [12] layer, with a sigmoid activation at the end of the network. For the detection head, we employ a four-block transformer [4] with a classification token $z^{cls} \in \mathbb{R}^D$ that is used for forgery detection. After propagating through the network, the refined token passes through a single fully-connected layer with a sigmoid activation to generate the final image-level forgery detection score.

The training data are augmented using resizing, cropping, flipping, and rotation operations. Training is performed on a single HPC cluster node equipped with four Nvidia A100 40GB GPUs, with an effective batch size of 160 images for score-level fusion models and 40 for feature-level fusion models. The training requires about 30 hours for the score-level and 60 hours for the feature-level fusion models. Moreover, the stream expansion experiments are performed on a single A100 to better represent a constrained environment. Finally, all the evaluation experiments are being conducted on a single Nvidia RTX3090 GPU.

For comparison with other score-level fusion approaches, we employ the OW-Fusion [1], a deep learning-based fusion approach, and implement it with the same input signals used on OMG-Fuser$_S$. Furthermore, in order to take into account the previous statistical fusion frameworks, we reimplement a DST-based fusion framework [6, 28] in Python, using again the same inputs with our score-level fusion implementation. Finally, we use the average of all input signals as a baseline approach.

2. Additional Experiments

Image Forgery Localization on best threshold: Following [8], we conducted additional experiments on image forgery localization, computing the F1 metric for the best threshold per image as an indicator of the performance of the method when properly calibrated. The results are presented in Tab. 2. Similar to the results in the main paper, both our methods outperform the competing approaches from the state-of-the-art in both feature- and score-level fusion with a clear margin.

Instance Segmentation Models: To better evaluate the modularity of our architecture, starting from the trained models of our score- and feature-level fusion implementations, we replaced the instance segmentation masks generated by the SAM [13] with the ones computed by the EVA [5]. In particular, we considered three different types of instance segmentation masks: i) from an EVA model trained on COCO [20], ii) from an EVA model trained on LVIS [9] and iii) from aggregating the segmentation masks of both of the aforementioned models. The results of these experiments are presented in Tab. 3. They highlight that replacing the instance segmentation model used during training has only a minimal impact on performance. This allows the combination of our model with class-specific

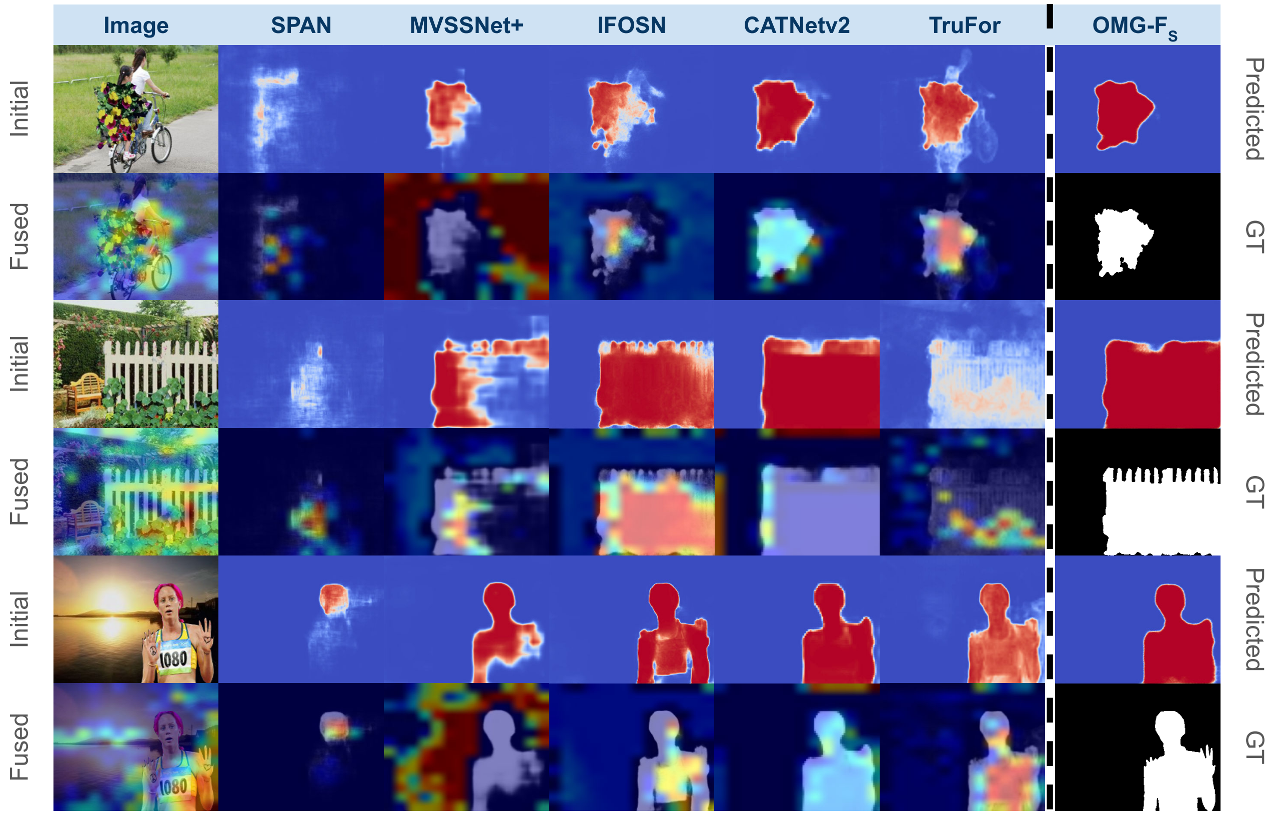

Figure 1. Explainability analysis. The top "initial" row of each sample presents the signals fused by OMG-F$_S$, while the bottom "fused" row presents the Grad-CAM overlay on top of them. Red regions in the overlay maps denote the regions of the signals with the greatest impact on the fusion process. The most-right column depicts the predicted output of our network on top and the ground-truth mask on the bottom row.

	Seg. Model	**Loc.**		**Det.**	
		F1	AUC	F1	AUC
Feat. Fus.	EVA (LVIS)	66.1	90.5	80.7	87.5
	EVA (COCO)	66.5	90.7	80.3	87.9
	EVA (COCO+LVIS)	67.0	91.3	80.8	88.0
	SAM (SA-1B)	67.3	91.5	82.4	88.0
Score Fus.	EVA (LVIS)	68.4	92.6	81.9	87.9
	EVA (COCO+LVIS)	68.8	93.0	82.4	88.6
	EVA (COCO)	69.1	93.0	81.8	88.4
	SAM (SA-1B)	70.4	93.5	83.2	89.5

Table 3. Comparison with different instance segmentation models. The average pixel-level F1 and AUC scores are reported across all evaluation datasets. The training dataset of each instance segmentation model is reported in parentheses.

	Signal	**Time**
Feat. Fus.	**DCT** [8]	75 ms
	Noiseprint++ [8]	115 ms
	SegmentAnything [13]	1.4 s
	OMG-Fuser$_F$ (Ours)	32.3 ms
Score Fusion	**SPAN** [11]	1.34s
	IFOSN [33]	6.85 s
	MVSSNet++ [2]	221 ms
	CATNetv2 [17]	1.04 s
	TruFor [8]	1.18 s
	SegmentAnything [13]	1.4 s
	OMG-Fuser$_S$ (Ours)	40.4 ms

Table 4. Computation time for the fused signals and our proposed network. The input signals utilized in the proposed feature- and score-level fusion implementations are considered.

segmentation models that better fit the needs of the downstream application.

Computation Time: In Tab. 4, we present the computation time required for extracting the input signals for both our score- and feature-level fusion implementations. The experiments have been conducted on the NIST16 dataset due to its great variability in the sizes of the included images. Our proposed fusion networks impose a minimal overhead on the overall computation time compared to the computation time required for generating the fused signals.

Additional qualitative evaluation: Finally, we present an extensive qualitative evaluation of our score- and feature-level fusion implementations with several forged and authentic samples in Fig. 2 and Fig. 3, respectively. In forged samples, our models greatly improve the localization mask, while in authentic samples, they considerably decrease the false positives.

3. Explainability

To better understand which parts of the input signals contribute the most to the fused tokens $\bar{z}^{ft}$ (eq. 5), we employ the Grad-CAM [29] method. In particular, we compute the gradients of the fused tokens with respect to the $N+1$ different inputs to the TFT in z (eq. 4) in order to isolate the token fusion process and determine from which tokens the information propagates to the next stages. We compute the gradients based on the output of the TFT, using the squared ℓ^2-norm of the $\bar{z}^{ft}$. In these experiments, we employ the OMG-Fuser$_S$ variant of our architecture due to the easier interpretation of the input signals. The explainability maps for three samples are presented in Fig. 1. Our architecture has learned to attend to the correctly estimated regions in the input signals based on the ground truth while ignoring the regions of the input signals containing erroneous predictions. Also, our network focuses on the signals that better capture the forged and the authentic regions separately, *e.g*. on MVSSNet++ for the detection of authentic regions, while exploiting information from the image to resolve ambiguity in the input signals.

4. Discussion on Research Ethics

Our primary ethical consideration while carrying out this research has been the potential for misuse of the proposed method. In particular, as with any image forensics method, the outputs of forgery localization and detection may be misinterpreted by non-experts or misused by malicious actors in an effort to discredit online digital media as being “manipulated”. This is especially true for methods that result in high false positive rates. Given that OMG-Fuser exhibits consistent improvements in detection accuracy with the integration of additional input forensic signals and demonstrates very low false positive rates, we expect the risk of misuse to be negligible, while at the same time, it enhances the current capabilities of detecting forged online content aimed at spreading disinformation.

References

[1] Polychronis Charitidis, Giorgos Kordopatis-Zilos, Symeon Papadopoulos, and Ioannis Kompatsiaris. Operation-wise attention network for tampering localization fusion. In *2021 International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–6. IEEE, 2021. 2

[2] 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*, 2022. 1, 2, 3

[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. IEEE, 2013. 1

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[5] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19358–19369, 2023. 2

[6] Marco Fontani, Tiziano Bianchi, Alessia De Rosa, Alessandro Piva, and Mauro Barni. A framework for decision fusion in image forensics based on dempster–shafer theory of evidence. *IEEE transactions on Information Forensics and Security*, 8(4):593–607, 2013. 2

[7] 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. IEEE, 2019. 1

[8] 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

[9] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019. 2

[10] 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

[11] 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. 1, 2, 3

[12] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 2

[13] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 2, 3

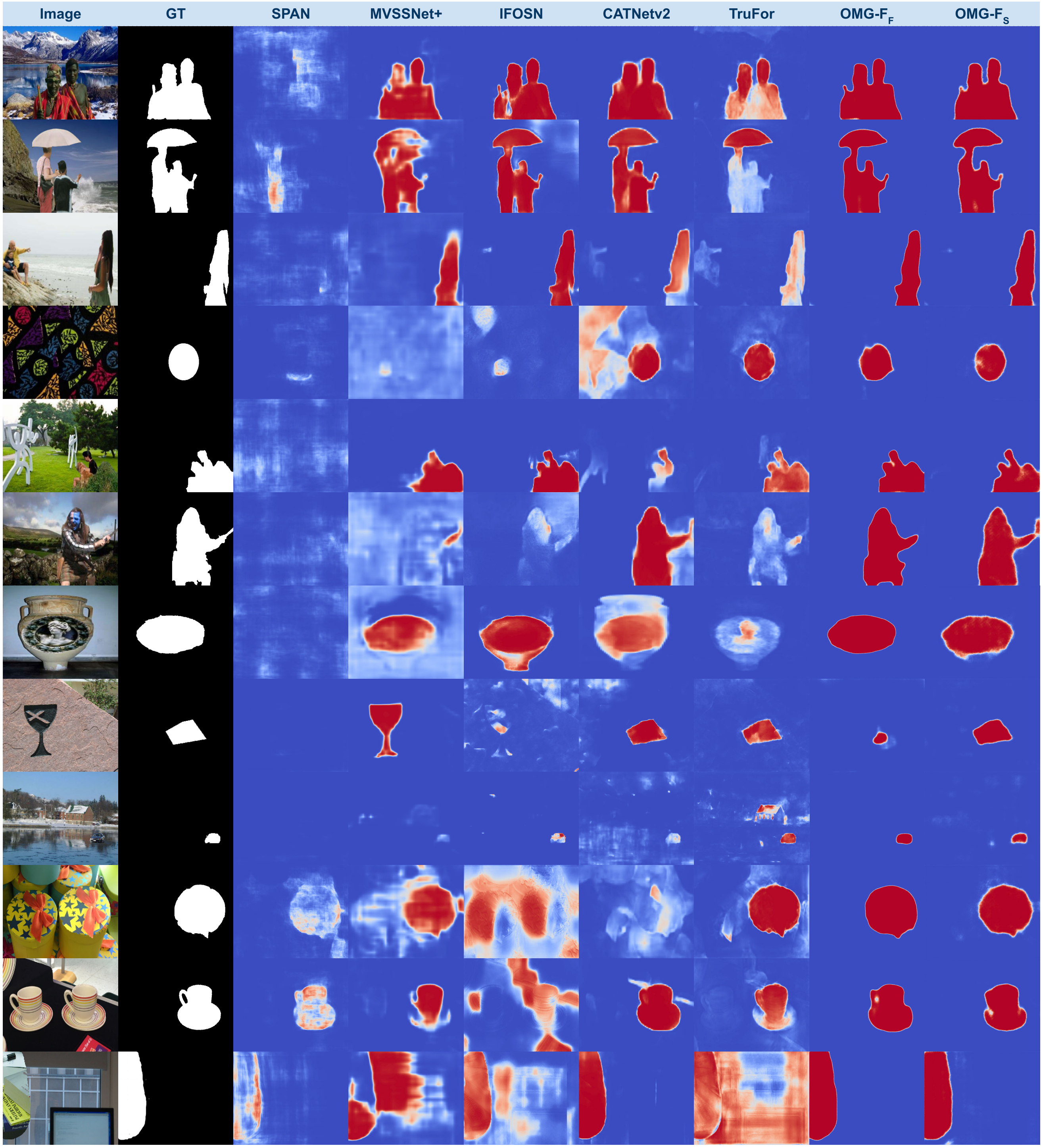

Figure 2. Additional qualitative evaluation results on forged images. From left to right, the image in question, the ground truth mask, the outputs of five recent methods, and the outputs of the two variants of our architecture are displayed.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 2

[15] Ananya Kumar, Ruoqi Shen, Sébastien Bubeck, and Suriya Gunasekar. How to fine-tune vision models with sgd. *arXiv preprint arXiv:2211.09359*, 2022. 2

[16] Myung-Joon Kwon, In-Jae Yu, Seung-Hun Nam, and

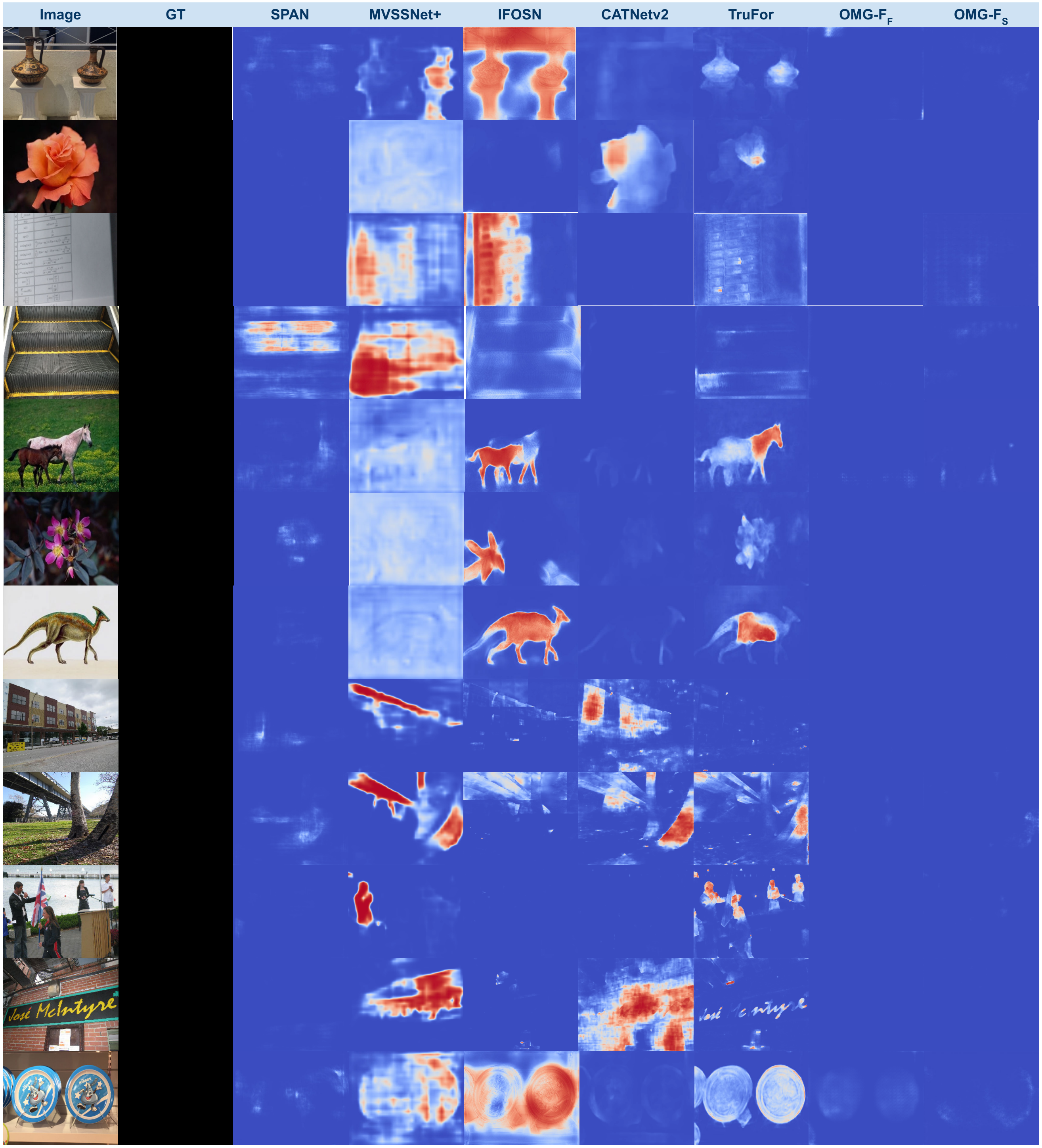

Figure 3. Additional qualitative evaluation results on authentic images. From left to right, the image in question, the ground truth mask, the outputs of five recent methods, and the outputs of the two variants of our architecture are displayed.

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

[17] 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. 1, 2, 3

[18] 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

[19] Jaejun Lee, Raphael Tang, and Jimmy Lin. What would elsa do? freezing layers during transformer fine-tuning. *arXiv preprint arXiv:1911.03090*, 2019. 2

[20] 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: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2

[21] 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. 1, 2

[22] Yanli Liu, Yuan Gao, and Wotao Yin. An improved analysis of stochastic gradient descent with momentum. *Advances in Neural Information Processing Systems*, 33:18261–18271, 2020. 1

[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 2

[24] Gaël Mahfoudi, Badr Tajini, Florent Retraint, Frederic Morain-Nicolier, Jean Luc Dugelay, and PIC Marc. Defacto: image and face manipulation dataset. In *2019 27Th european signal processing conference (EUSIPCO)*, pages 1–5. IEEE, 2019. 1

[25] Hiroki Naganuma, Kartik Ahuja, Ioannis Mitliagkas, Shiro Takagi, Tetsuya Motokawa, Rio Yokota, Kohta Ishikawa, and Ikuro Sato. Empirical study on optimizer selection for out-of-distribution generalization. *arXiv preprint arXiv:2211.08583*, 2022. 1

[26] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 1

[28] Anh-Thu Phan-Ho and Florent Retraint. A comparative study of bayesian and dempster-shafer fusion on image forgery detection. *IEEE Access*, 10:99268–99281, 2022. 2

[29] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 4

[30] 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. IEEE, 2016. 1

[31] Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. *Advances in neural information processing systems*, 30, 2017. 1

[32] Haiwei Wu and Jiantao Zhou. Iid-net: Image inpainting detection network via neural architecture search and attention. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(3):1172–1185, 2021. 1

[33] 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