MVFNet: Multipurpose Video Forensics Network using Multiple Forms of Forensic Evidence
Supplementary Material

Tai D. Nguyen, Matthew C. Stamm
Drexel University
Philadelphia, PA, USA
tdn47@drexel.edu, mcs382@drexel.edu

A. Details on Spatial Forensic Residual Feature Extractor

Here we provide more details about the architecture of the Spatial Forensic Residual Feature Extractor. The system diagram for this subnetwork can be found in Fig. 1. As stated in Sec. 3.1, we first create a rich set of learned forensic residuals using a constrained convolutional layer proposed in [2]. Then, these residuals are analyzed using a series of fused inverted residual (FIR) blocks [13] with specifications shown in Fig. 1. In general, the number of embedding dimension increases ($24 \rightarrow 48 \rightarrow 64 \rightarrow 128 \rightarrow 256$) and the input dimension is subsequently reduced by a total factor of 8.

B. Details on Temporal Forensic Residual Feature Extractor

The temporal residual forensic features are also extracted using a residual extractor and series of FIR blocks. Here, the focus is on extracting low-level pixel-intensity-related features that when taken the difference between adjacent frames, reveals temporal inconsistencies. This process starts with the RGB input frame undergoing a shallow residually connected convolutional layer, which is succeeded by an array of convolutional, normalization, and SiLU activation layers to compress the dimensionality of the input while modestly amplify the feature set. Subsequent to this dimensionality reduction, the extracted low-level residuals are fed through a series of 4 FIR blocks, with the number of embedding dimension increases from $8 \rightarrow 16 \rightarrow 32 \rightarrow 64$, and the input dimension is reduced by a total factor of 8.

C. Spatial Forensic Residual Feature Extractor Pretraining

In order to pre-train the Spatial Forensic Residual Feature Extractor in our network, we developed a novel pre-training strategy. Here, we build upon prior approaches used in many multimedia forensic methods, in which forensic embeddings are learned by pre-training a network to perform forensic camera model identification (i.e. determining which camera model was used to capture an image) [2, 5, 8–10]. Unlike prior work [2, 5, 8–10, 12], however, our network does not produce a single embedding. Instead, it produces a high dimensional feature set which is subsequently pooled into several embeddings across multiple scales. As a result, existing forensic pre-training approaches cannot be used in our network.

Our pre-training strategy requires the high dimensional features produced by our module to produce camera model identification results that are both accurate and consistent across multiple spatial scales. To accomplish this, we create a multi-headed pre-training network in which the output of the spatial forensic residual feature extractor is passed through our multi-scale pooling module to produce localized embeddings across multiple scales. Each embedding is then passed to an classification head, which produces an output camera model identification likelihood $\theta_{i,j}^{k,c}$ for camera model c, where i and j denote the spatial index of the embedding at scale k. These multi-scale, spatially distributed decisions are then aggregated and used to produce an overall pre-training loss:

$$\mathcal{L}_{\mathrm{F}} = \sum_{k} \frac{-\lambda_k}{2^{2k}} \sum_{i,j} \sum_{c} \log \left(\frac{\exp(\theta_{i,j}^{k,c}) \mathbb{1}(c = c^*)}{\sum_c \exp(\theta_{i,j}^{k,c})} \right) \quad (1)$$

where c^* is the true source camera model, $\mathbb{1}(\cdot)$ is the indicator function and λ is the relative weight of each scale. In practice, we aggregate across three scales $k = \{3, 4, 5\}$ such that $\lambda_k = \{0.01, 0.0075, 0.005\}$. After pre-training, we discard the multi-headed classification network and retain only the initialized spatial forensic residual feature extractor trunk. As our ablation study in Sec. 6 shows, this

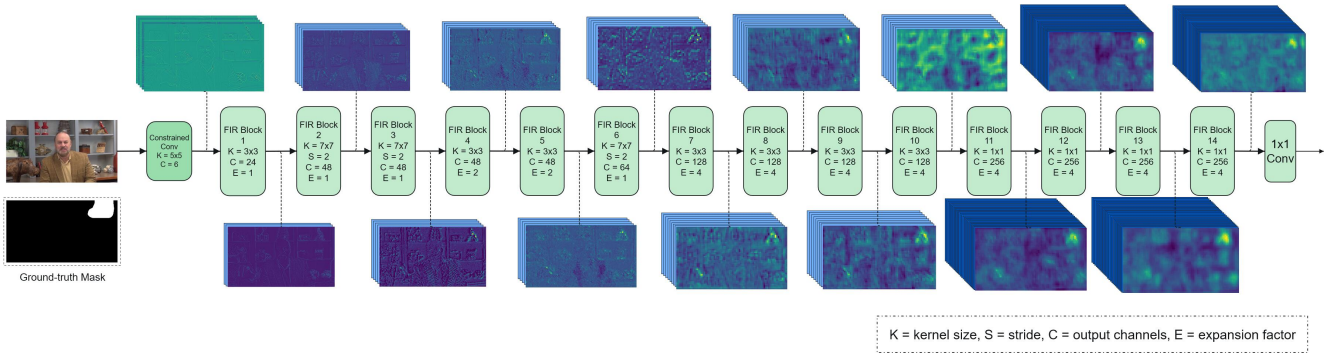

Figure 1. This figure shows the details of the Spatial Forensic Residual Feature Extractor.

pre-training procedure substantially increases the performance of our network.

D. Network Training and Hyperparameters

Our training strategy involves two steps. Initially, we pretrained the Spatial Forensic Residual Module on the camera model identification task with the Video-ACID dataset [7]. Subsequently, we trained the entire network on all types of manipulations present in the UVFA-IND dataset. During training, we started all loss parameters at 1.0 and decayed γ by 0.95, α by 0.80, while increased β by 1.18 every epoch. Additionally, our optimizer of choice for all training stages is SGD. In the pretraining stage, we initialize the learning rate at $1.00E-03$ with momentum of 0.96 and learning rate decay of 0.65 every 2 epochs. In the full-network training stage, we set the learning rate at $6.00E-04$ with momentum of 0.90 and learning rate decay of 0.85 every 2 epochs. During this final stage, we used the validation detection accuracy as the metric to choose the best checkpoint of our network.

E. Retraining Competing Forensic Networks

In our main paper, we compared against 9 state-of-the-art detection systems. These algorithms can be divided into four major groups: 1) Splicing and Editing: MVSS-Net [3], MantraNet [16], FSG [9], 2) Inpainting: VIDNet [18], DVIL [15], 3) Deepfake: Self-Blended Images (SBI) [14], Multi-Attentional Deepfake Detection (MADD) [17], Cross-Efficient-ViT (CE.ViT) [4], and 4) General: VideoFACT [12]. For a fair comparison, we retrained top-performing methods of each group using the exact same training data as ours. We note that we excluded deepfake detectors in this process because they need to isolate and operate on a face, which is impossible in the multi-manipulation setting. Nonetheless, these systems have all seen similar deepfake data as ours.

Therefore, in summary, we retrained MVSS-Net, DVIL, and VideoFACT. In addition, we have to trained VIDNet from scratch due to the fact that model weights for this network were not publicly available.

To retrain MVSS-Net, we used the authors' public code on github to initialize a new model. Next, we trained this model on the video frames in the training portion of our Unified Video Forgery Dataset (denoted in our main paper as UVFA-IND) using the following parameters: `batch_size=16`, `optimizer=SGD`, `init_lr=0.0001`, `decay_step=2`, `decay_rate=0.8`. To obtain these parameters, we performed a grid-search and we chose the set of parameters with the best training loss. Additionally, since MVSS-Net can only accept an input resolution of 512 by 512 pixels, we resized all input frames and ground-truth masks to this resolution before feeding them to the model.

To retrain DVIL, we faithfully reimplemented the authors' public code on github, written in Tensorflow 1.x, in modern Pytorch, and used this reimplementation to initialize a new model. Next, we trained this model on the video frames in the training portion of UVFA-IND using the following parameters: `batch_size=4`, `max_sequence_length=4`, `optimizer=AdamW`, `init_lr=0.006`, `weight_decay=0.0001`, `decay_step=10`, `decay_rate=0.5`. To obtain these parameters, we performed a grid-search and we chose the set of parameters with the best training loss. Additionally, since DVIL can only accept an input resolution of 240 by 432 pixels, we resized all input frames and ground-truth masks to this resolution before feeding them to the model.

To retrain VIDNet, we used the authors' public code on github to initialize a new model. Next, we trained this model on the video frames in the training portion of UVFA-IND using the following parameters: `batch_size=1`, `max_sequence_length=5`, `encoder_optimizer=AdamW`, `encoder_init_lr=0.0001`, `encoder_weight_decay=5.0e-05`, `encoder_decay_step=30`,

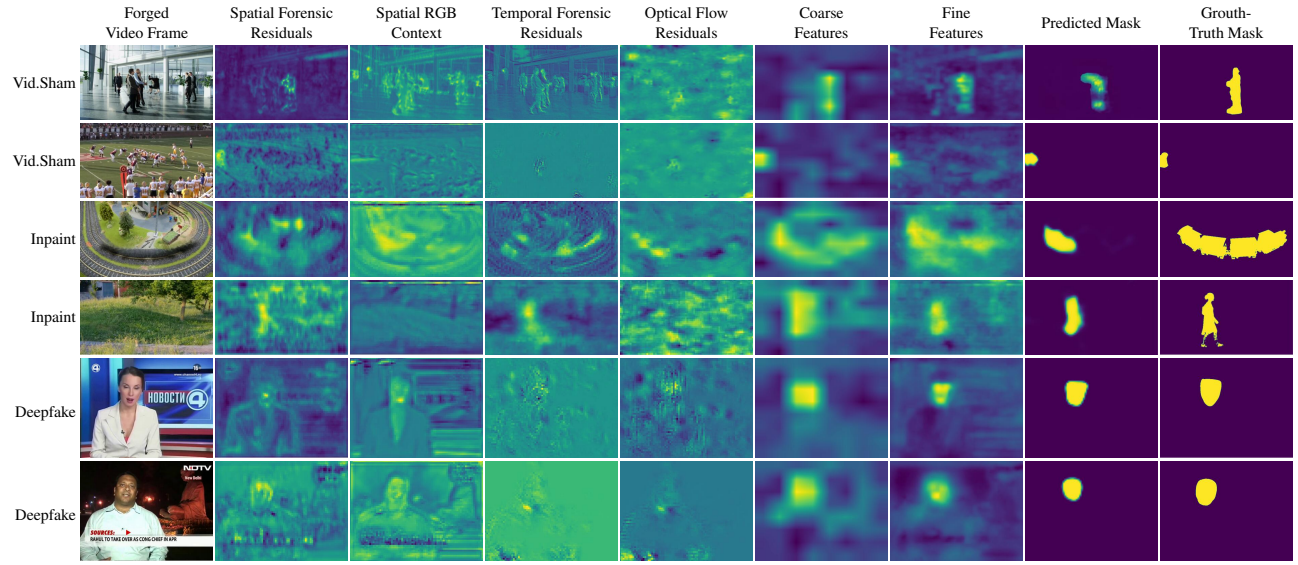

Figure 2. Intermediate layers' results are shown for different manipulation types, Splicing/Editing, Inpainting, and Deepfake. We see that for splicing/editing, the spatial features are more effective at uncovering forgery, while the temporal features are inadequate. For inpainting, temporal features are much more helpful in detecting anomalies than spatial features. And for deepfakes, both features are useful in detecting and localizing deepfakes. Further analysis is provided in Sec. F.

`encoder_decay_rate=0.1`, `decoder_optimizer=AdamW`, `decoder_init_lr=0.001`, `decoder_weight_decay=5.0e-05`, `decoder_decay_step=30`, `decoder_decay_rate=0.1`. These hyperparameters were directly provided by the authors and we found that this set of hyperparameters led to good performance. Additionally, other network hyperparameters are also chosen according to the authors' recommendations in their paper: `encoder_base_model=vgg16_bn`, `encoder_dropout=0.5`, `encoder_hidden_size=512`, `encoder_kernel_size=3`, `decoder_dropout=0.5`, `decoder_hidden_size=512`, `decoder_kernel_size=3`, `decoder_skip_mode=concat`, `max_sequence_length=5`. Additionally, since VIDNet can only accept an input resolution of 480 by 854 pixels, we resized all input frames and ground-truth masks to this resolution before feeding them to the model.

And finally, to retrain VideoFACT, we used we used the authors' public code on github to initialize a new model. Next, we trained this model on the video frames in the training portion of UVFA-IND using the authors' originally recommended parameters: `batch_size=3`, `optimizer=SGD`, `init_lr=0.001`, `momentum=0.95`, `decay_step=2`, `decay_rate=0.5`, `alpha=0.4`.

We also note that all networks, both pretrained and retrained versions, are subjected to the same benchmarking protocols to ensure fairness in our experiments.

F. Importance of Different Forensic Modalities

Fig. 2 illustrates the discerning power of our network's intermediate layers when facing diverse video manipulations that are all unseen during training. Each type of forgery imparts distinct traces that our network's specialized modules are finely tuned to detect and analyze.

For splicing and editing manipulations (see "Vid.Sham" rows), the spatial features, like the spatial forensic residuals, stand out in the intermediate layer results. This is because spliced or edited regions typically maintain temporal consistency across frames, leaving spatial discrepancies as the primary cue for detection. Temporal features, which excel at uncovering temporal inconsistencies, are less pronounced in these cases since the alterations do not significantly affect the temporal flow.

Conversely, inpainting and deepfake manipulations (see "Inpaint" and "Deepfake" rows) exhibit a higher degree of temporal inconsistency due to the nature of the forgery. In these instances, the temporal features, like the temporal forensic residuals and the optical flow residuals, become increasingly vital, as they can detect the abnormal changes over time that spatial features might miss within a single frame. For deepfakes, however, the network leverages both spatial and temporal features synergistically, capturing the frame-to-frame irregularities in facial features and expres-

sions, providing a robust detection framework.

In addition to showing traces from different forensic modalities, we also included feature maps resulting from our multi-scale hierachical analysis. Here we show the coarsest and finest resulting feature maps in the "Coarse Features" and the "Fine Features" columns. From these results, we can see that the hierarchical multi-scale approach is able to accurately identify manipulated regions. Furthermore, the prediction from larger scale is refined multiple times, such that the finest scale can accurately localize fake content with very minimal false alarms.

G. Qualitative Results on Authentic Media

In this section, we present our network's ability to correctly classify authentic, unmanipulated video. As shown in Fig. 3 and many experimental results in our main paper, our network can not only detect manipulated media, but also correctly classify unmanipulated media to be real (authentic). This behavior can be observed as our network will output a higher detection score (close to 1.0) if the input video is forged and a low detection score (close to 0.0) otherwise. Additionally, when our network is given a real video, then its localization result is often an empty, zero matrix, which is a desired outcome.

H. Space-Time Complexity Analysis & Comparisons

The table below shows the model sizes & inference speeds of ours and competing methods. We conducted this experiment using an A100 NVIDIA GPU. We measure the FPS as by dividing the total amount of time to process 1000, 512x512 frames by 1000. From the results shown in Table 1, with only 36.5M parameters, MVFNet is among the smallest models. This demonstrates that our superior performance is not due to the size of our network, but rather to the use of novel forensic modalities such as temporal forensic residuals and the innovative multi-scale hierarchical approach in which they are analyzed. Additionally, while efficiency is not part of our paper's novelties, these results show that MVFNet's inference speed, measured in frames per second (FPS), is moderate and sufficient for practical applications.

I. Additional Experimental Results

In this section, we provide additional experimental results that are specifically requested by our reviewers. Particularly, our reviewers would like to us to evaluate more single-frame based methods that were originally designed for images such as: TruFor, Noiseprint, SpliceBuster, and Adapt-CFA. Since the author of SpliceBuster did not provide official publicly accessible code or model weights, we will only provide benchmarks for the other three methods.

Table 1. This table shows the space-time complexity comparisons between ours and other competing methods. With only 36.5M parameters, MVFNet is among the smallest models. This demonstrates that our superior performance is not due to the size of our network, but rather to the use of novel forensic modalities such as temporal forensic residuals and the innovative multi-scale hierarchical approach in which they are analyzed. See more in Sec. H

Method	**Params(M)**	**FPS**
Ours	**36.5**	**11.3**
VideoFACT	135.4	21.6
MVSS-Net	146	34.3
VIDNet	337	21.8
DVIL	82.8	65.5
MantraNet	3.8	10.1
FSG	1.2	7.1
SBI	17.6	42.4
CE.ViT	101.4	32.1

Table 2. Detection and localization performance of methods specifically requested by our reviewers on the Unified Video Forgery Analysis (UVFA) and VideoSham [11] dataset.

Manip. Group	**Method**	**UVFA-IND**		**UVFA-OOD**		**VideoSham**	
		mAP	*F1*	*mAP*	*F1*	*mAP*	*F1*
Splice/Edit	TruFor [6]	0.74	0.47	0.79	0.57	**0.66**	**0.21**
	Noiseprint [5]	0.49	0.08	0.32	0.12	0.50	0.09
	Adapt-CFA [1]	0.50	0.10	0.54	0.18	0.49	0.09
	MVSS-Net [3]	0.66	0.29	0.45	0.14	0.56	0.10
	MVSS-Net (R)	0.94	0.77	0.63	0.09	0.52	0.01
General	VideoFACT [12]	0.78	0.36	0.74	0.28	0.54	0.07
	VideoFACT (R)	0.88	0.50	0.79	0.39	0.55	0.08
	Ensemble of (R)*	0.90	0.54	0.73	0.44	0.52	0.11
	Ours	**0.95**	**0.77**	**0.91**	**0.59**	0.63	0.14

From the results shown in Table 2, we see that our network still obtained state-of-the-art performance on most datasets. It is notable that TruFor slightly outperformed our network on VideoSham. However, this is due to the fact that our network has training examples of splicing and editing. By contrast, TruFor has not only seen more almost twice as many training examples of these type of forgeries, but also TruFor's training examples include more advanced manipulation techniques like Photoshopped image editing and AI-guided splicing.

J. Statistical Evaluation of Ablation Results

In this section, we provide additional experimental results showing the statistical variation of the results provided in the ablation study in Table 6 of our main paper.

To achieve this, first, we randomly sample without replacement 5 separate subsets of the UVFA-IND's test set used for our ablation study. Then, we benchmarked all variations of our network on each subsets and report the mean and variance of each metrics. We then present our results in Table 3.

Table 3. Ablation study of the components in our proposed network and their performance evaluations.

Setup	UVFA-IND			
	Reported mAP	*Mean / Var mAP*	*Reported F1*	*Mean / Var F1*
Proposed	**0.95**	**0.95 / 0.13**	**0.77**	**0.75 / 0.16**
No Spatial Foren. Resid.	0.64	0.63 / 0.07	0.57	0.55 / 0.06
No RGB Context	0.82	0.80 / 0.10	0.52	0.55 / 0.07
No Temporal Residual	0.84	0.82 / 0.09	0.59	0.53 / 0.14
No Optical Flow Residual	0.72	0.76 / 0.15	0.34	0.35 / 0.07
Standard Transformer	0.86	0.83 / 0.06	0.56	0.55 / 0.02
No M.S.H Transformer	0.75	0.70 / 0.12	0.52	0.53 / 0.05
Fine-to-Coarse	0.90	0.91 / 0.02	0.68	0.66 / 0.04

From the results in Table 3, we see that our reported numbers in Table 6 of our main paper remains statistically meaningful. This is because the standard deviations for each metric is small and the mean for each metric is close to what we previously reported.

K. Examples of Failure Modes

While our network is robust and can detect many types of video forgeries (splicing, editing, deepfake, inpainting), it can encounter challenges when encountering any of these conditions:

- Extreme video compression.
- Subtle manipulations that does not noticeable alter the forensic traces, such as remapping the color of one object to another.
- Advanced forgery techniques that combines a series of different editing operations, such as those in the VideoSham [11] dataset.

To further illustrate the difficulties of these conditions, we present examples of them with outputs from our network in Figure 4.

L. Comprehensive Qualitative Examples

In this section, we present a comprehensive set of examples in each dataset used in this paper to show our network and others' ability in detecting and localizing a wide variety of video forgery. The order of presentation is: Videosham (Fig. 5), DAVIS-E2FGVI-Inpaint (Fig. 6), DAVIS-FuseFormer-Inpaint (Fig. 7), DeepFakeDetection (Fig. 8), FaceShifter (Fig. 9), Face2Face (Fig. 10), VMMP-Inpaint (Fig. 11), VMMP-Editing (Fig. 12), and VMMP-Splicing (Fig. 13).

We observe a general trend emerges from these examples: our network is able to detect and localize a wide variety of video forgery, including those that are not seen during training. In contrast, other networks are only able to detect and localize the specific type of forgery they are trained on. For example, the splicing and editing detectors work well only for splicing and editing forgeries; the inpainting detector works well only for inpainting forgeries; and the deepfake detector works well only for deepfake forgeries.

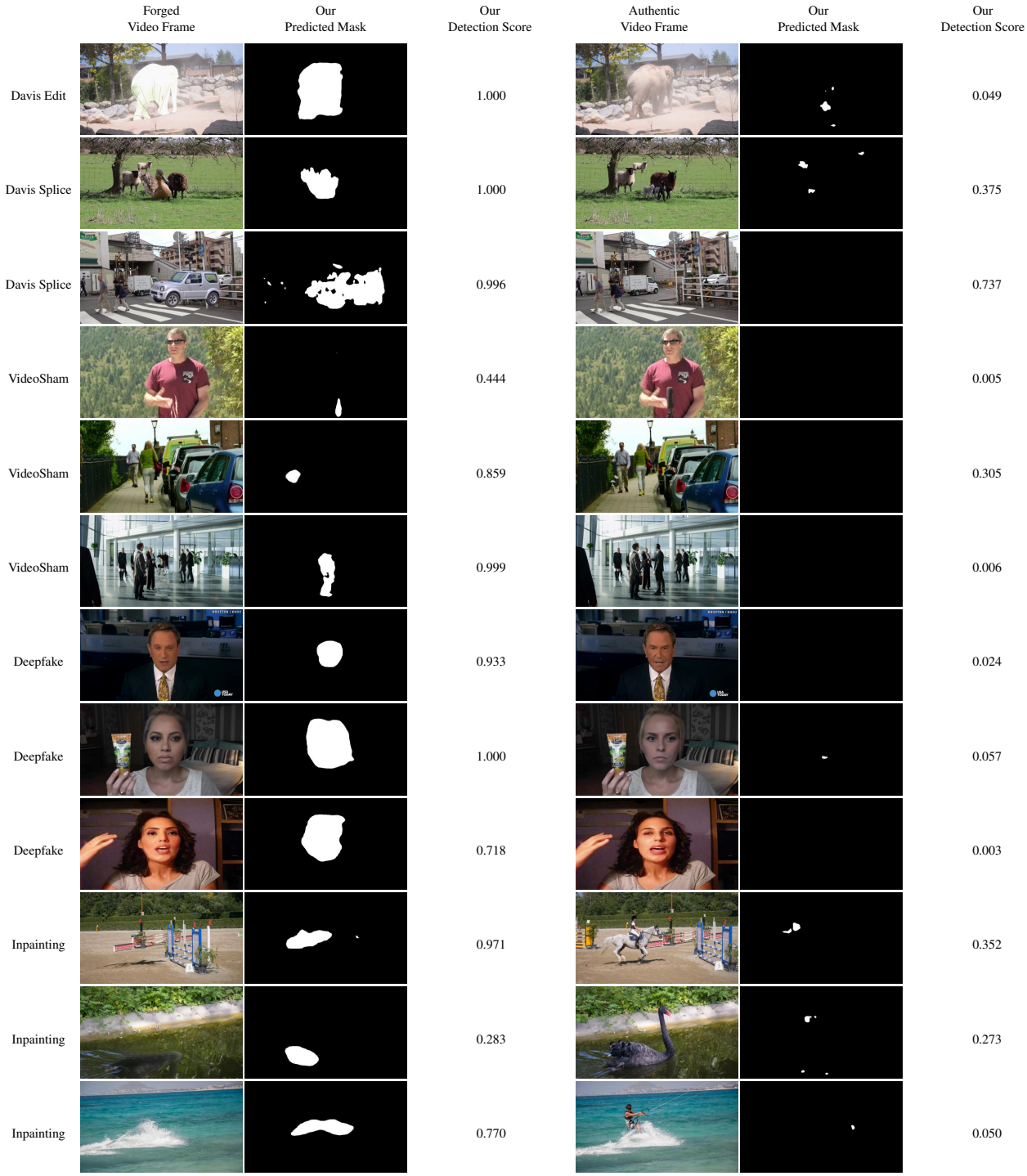

Figure 3. This figure shows our network's output on manipulated videos and their corresponding unmanipulated copies. As evident here and in the experiments presented in our main paper, our network can not only detect manipulated media, but also accurately classify authentic ones to be real. Further analysis is provided in Sec. G.

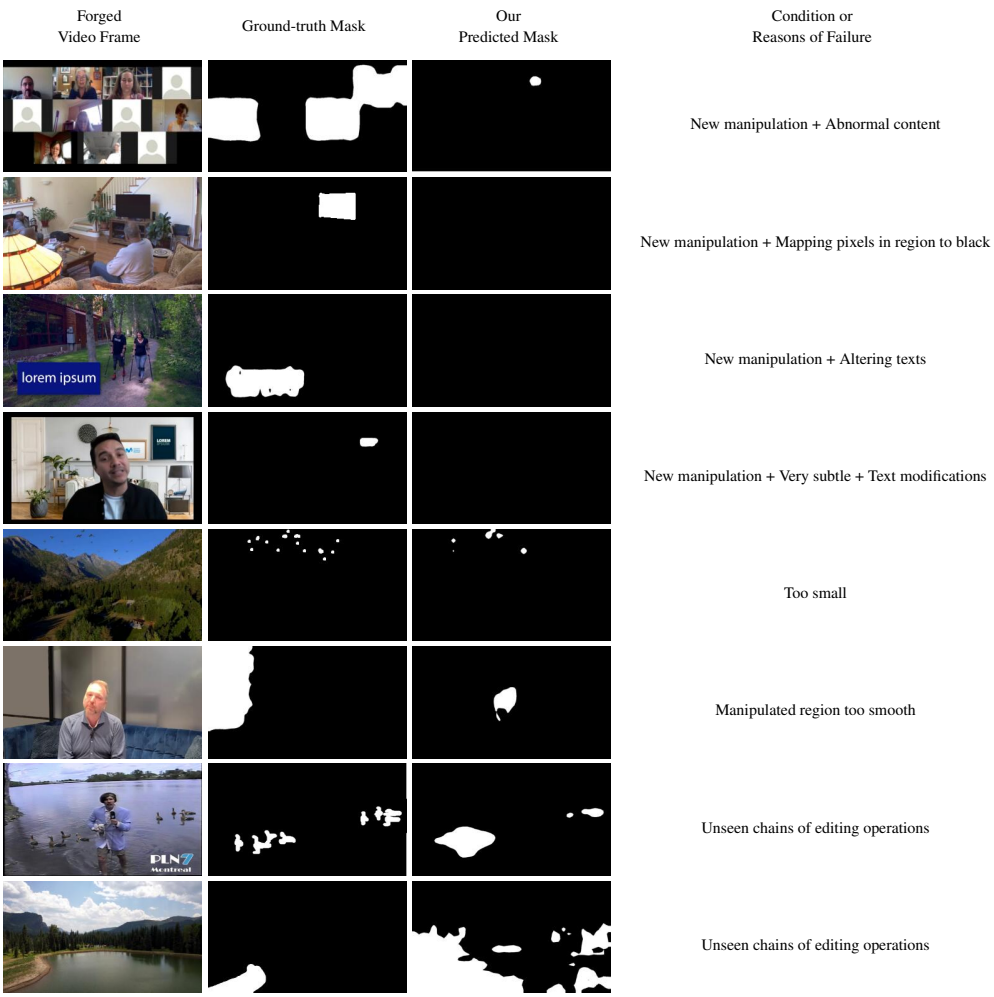

Figure 4. This figure shows cases where our network failed to identify forgeries. Further analysis is provided in Sec. K.

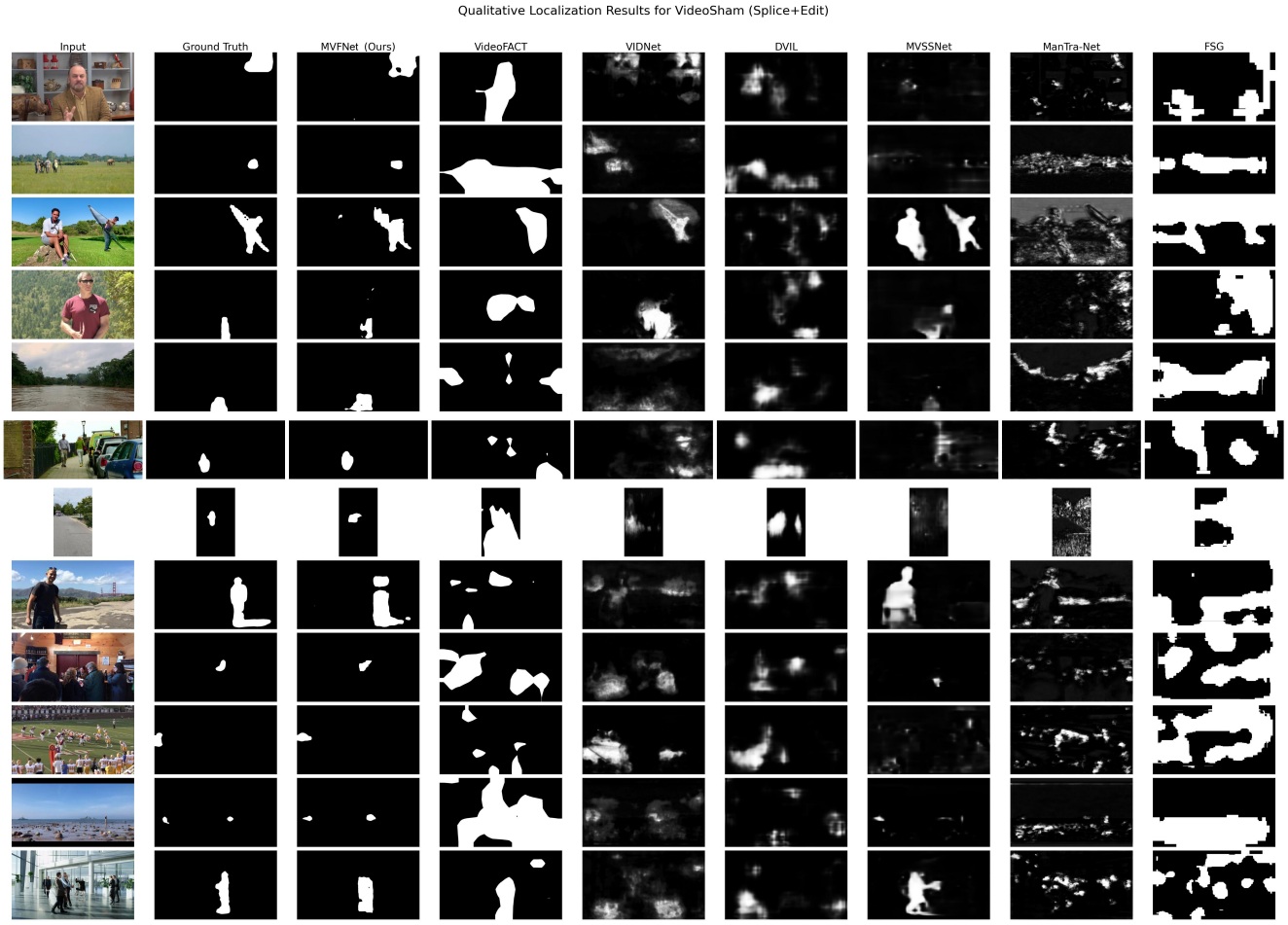

Figure 5. Comparative localization results for video manipulation detection on the Adobe VideoSham dataset (Splice+Edit). The Ground Truth column displays the actual manipulated areas, followed by the detection results from our method (MVFNet) and other leading approaches: VideoFACT, VIDNet, DVIL, MVSS-Net, ManTra-Net, and FSG. Our method demonstrates superior localization precision, closely mirroring the Ground Truth, with clear, well-defined boundaries and minimal false positives. VideoFACT and VIDNet exhibit moderate localization accuracy but with less distinct boundaries. DVIL and MVSS-Net present overgeneralized localizations with significant false positives, while ManTra-Net and FSG show mostly inaccurate detections.

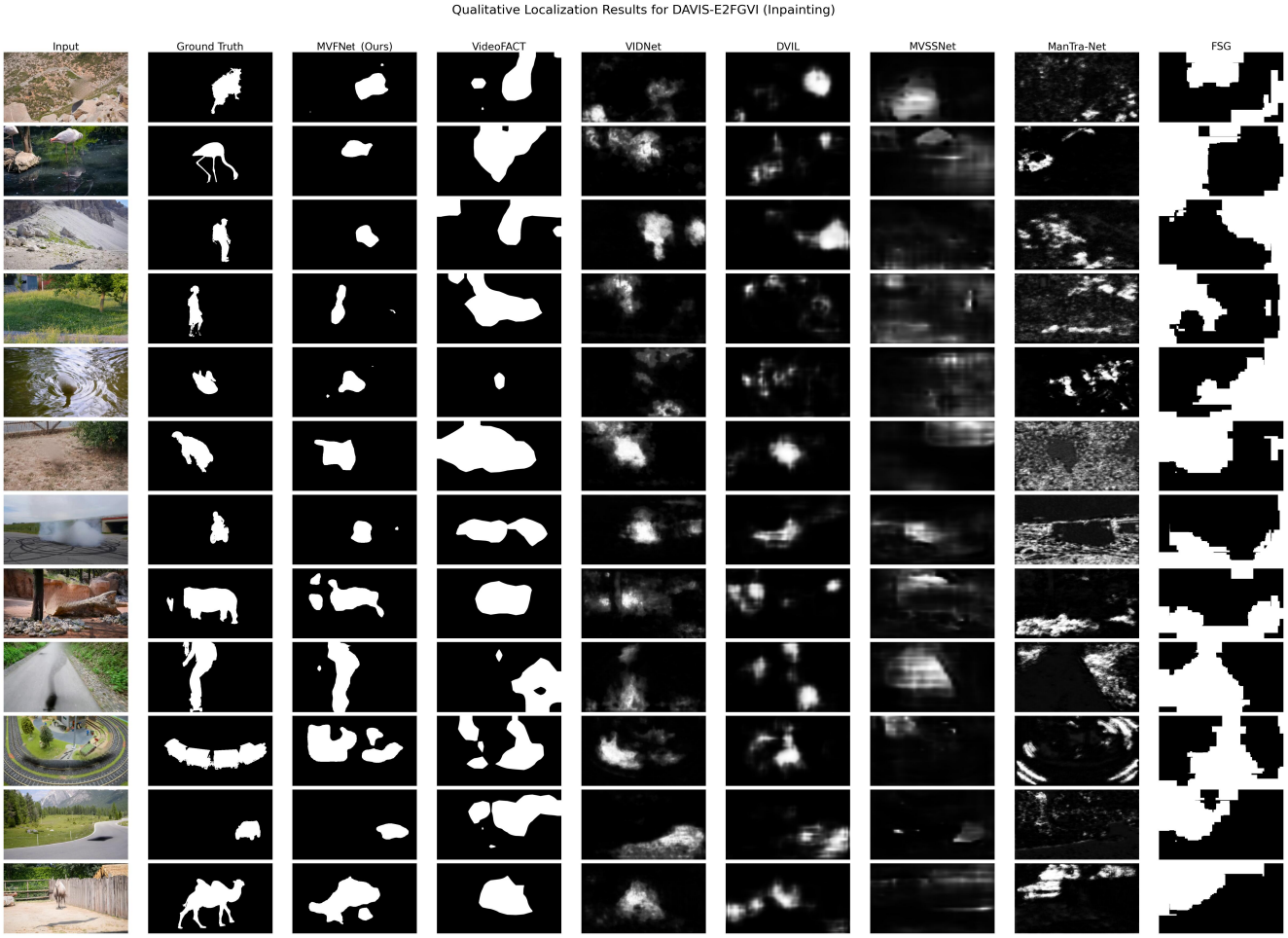

Figure 6. This figure displays localization results for inpainting detection on the DAVIS-E2FGVI dataset. Our MVFNet approach outperforms others by producing localization masks that closely match the Ground Truth, with high precision and minimal false detections. VIDNet and DVIL demonstrate reasonable accuracy, effectively capturing the manipulated regions with some minor imprecisions. VideoFACT, while detecting the general area of manipulation, tends to generate overly broad masks, leading to a significant amount of false positives. The results from MVSS-Net, ManTra-Net, and FSG show a marked lack of accuracy, failing to provide useful localization in most cases and indicating a substantial departure from the Ground Truth.

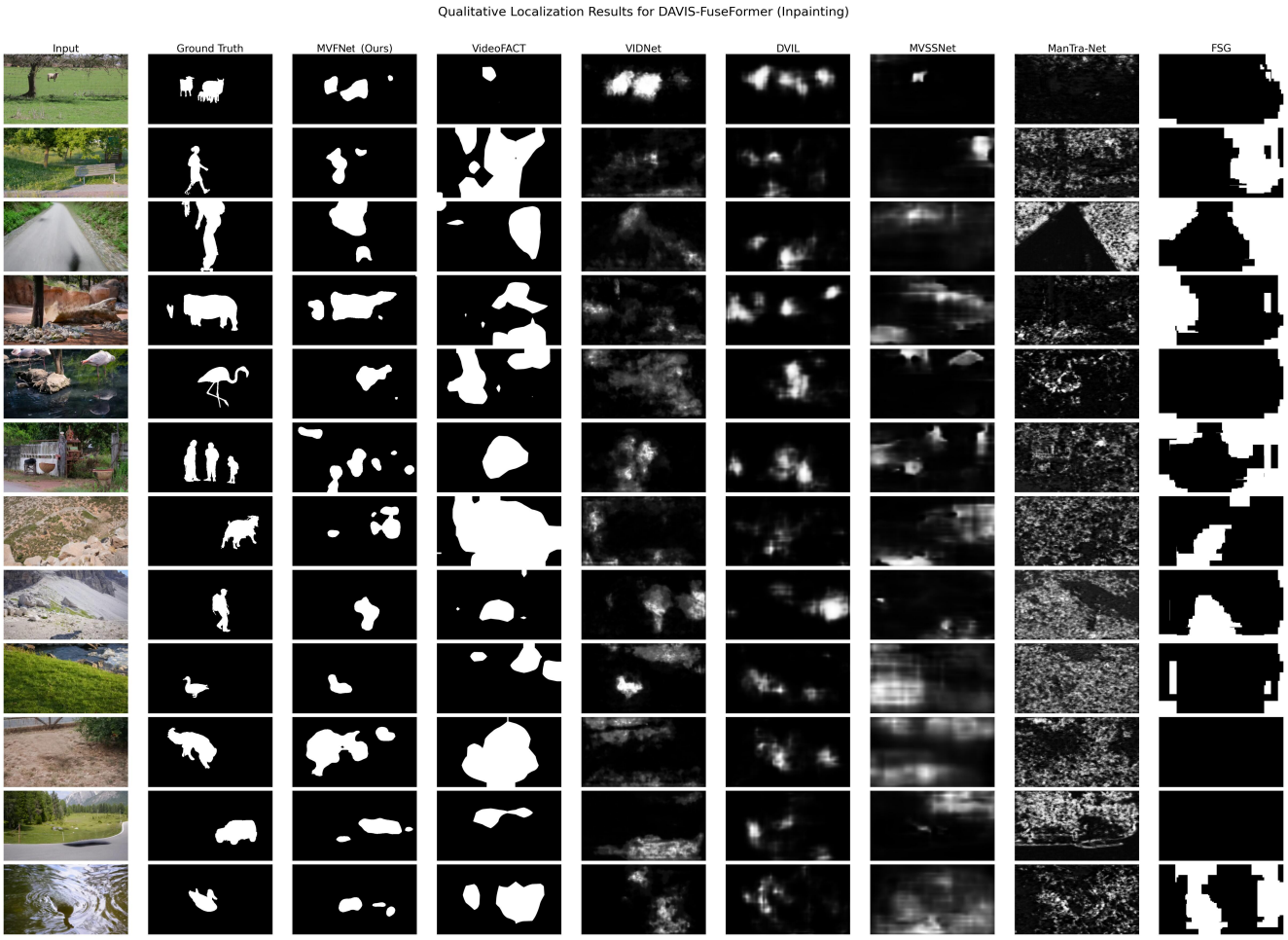

Figure 7. Localization results for inpainting detection on the DAVIS-FuseFormer dataset are compared across various methods. The MVFNet approach, our proposed method, consistently yields localization masks that are tightly aligned with the Ground Truth, demonstrating high fidelity and specificity. VideoFACT, while identifying the area of manipulation, tends to overextend the localization boundaries beyond the actual region. VIDNet shows a moderate level of accuracy, capturing significant portions of the manipulated areas but with less precision. DVIL, although somewhat effective, produces inconsistent results with varying degrees of overgeneralization. MVSS-Net, ManTra-Net, and FSG exhibit poor performance, with ManTra-Net and FSG generating particularly extensive and inaccurate localizations that deviate significantly from the Ground Truth.

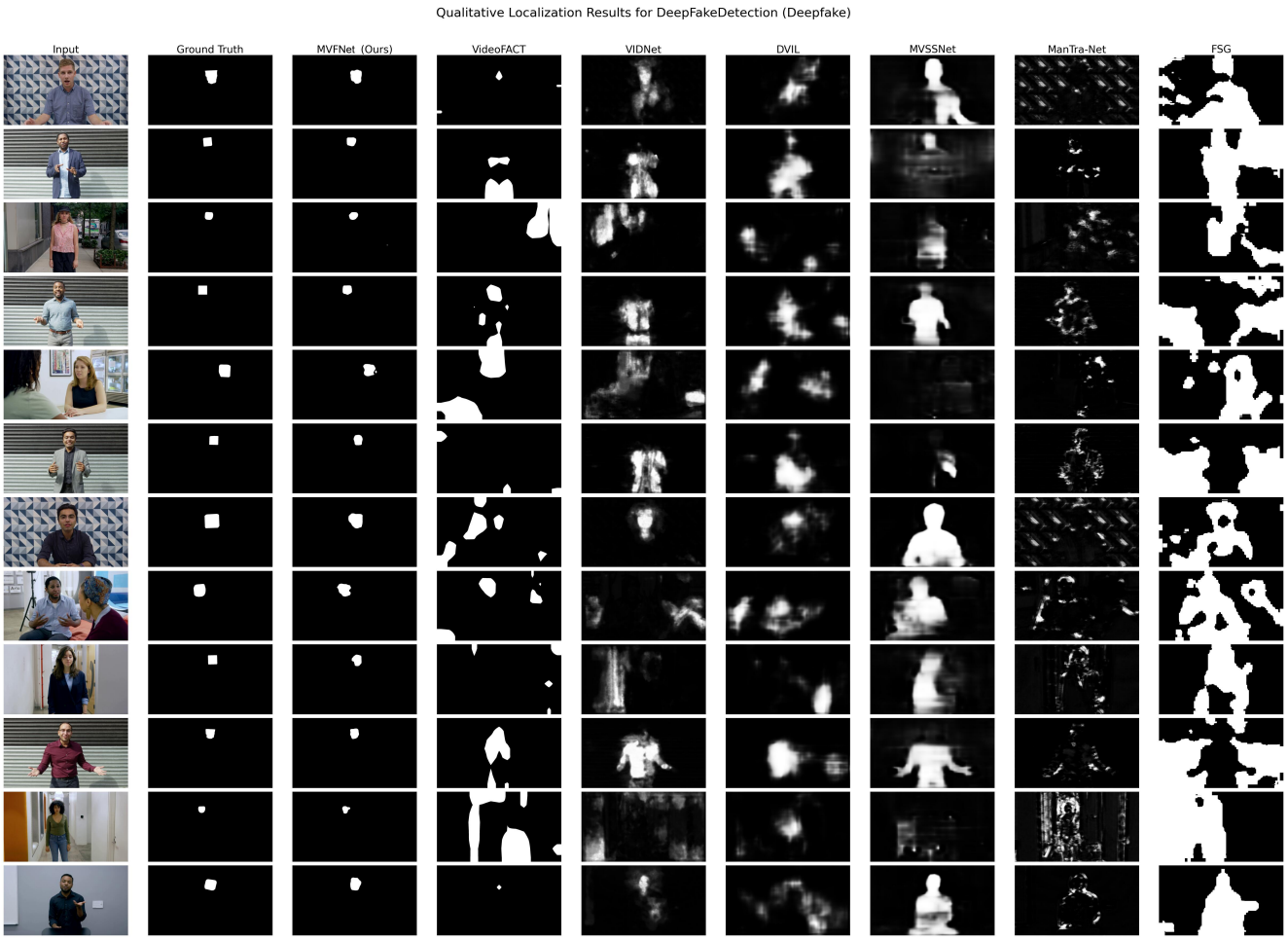

Figure 8. Localization results for deepfake detection are illustrated, highlighting the comparative effectiveness of various methods. Our MVFNet approach consistently provides precise localizations that closely match the Ground Truth. VideoFACT shows a small degree of accuracy. VIDNet achieves correct localization sporadically, with success in very few select cases. The remaining methods—DVIL, MVSS-Net, ManTra-Net, and FSG—largely fail to localize the specific regions of manipulation, often erroneously marking either the entire body or unrelated areas within the frame.

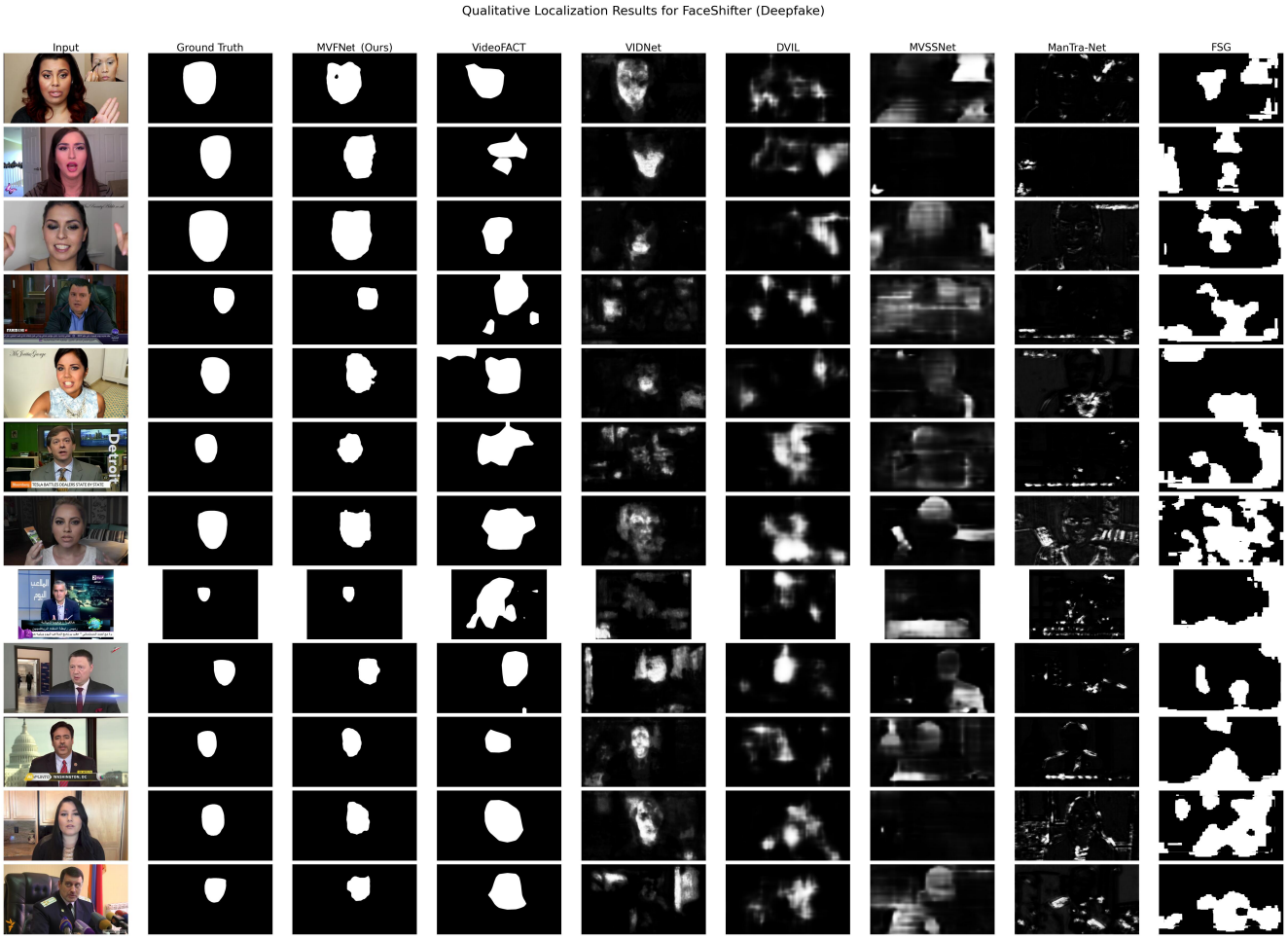

Figure 9. Comparative localization results for FaceShifter deepfake detection are depicted. Our MVFNet method demonstrates precise localization, closely aligning with the Ground Truth, while VideoFACT shows some capability to localize manipulated areas but with less specificity. VIDNet, although occasionally accurate, largely fails to consistently delineate the manipulated regions. DVIL, MVSS-Net, and ManTra-Net generally misidentify the extent of the manipulations, often erroneously attributing them to other areas or the entire body. FSG's localizations are overly broad and imprecise, significantly deviating from the targeted manipulations indicated by the Ground Truth.

Figure 10. The figure displays localization results for Face2Face deepfake detection, highlighting the superior performance of our MVFNet method, which consistently provides accurate localizations that closely match the Ground Truth. VideoFACT rarely captures the manipulated regions correctly, often missing the mark. VIDNet and DVIL achieve limited success, with correct localization in a minority of cases. The rest of the methods, including MVSS-Net, ManTra-Net, and FSG, largely fail to identify the manipulated areas accurately, with localizations that either miss or misattribute the forgeries.

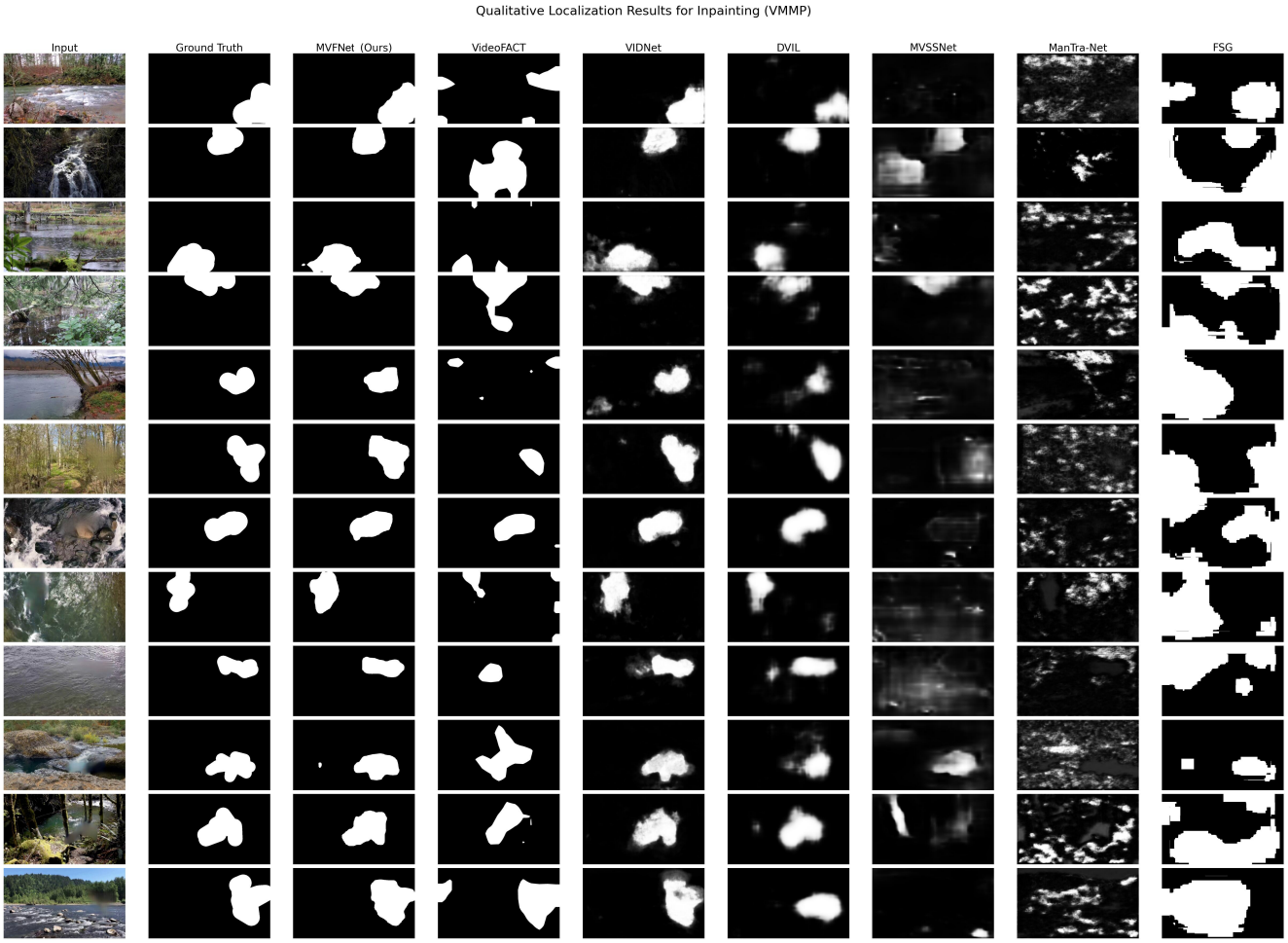

Figure 11. This figure presents localization results from various methods for an inpainting detection challenge. Our MVFNet method exhibits accurate and consistent localization closely mirroring the Ground Truth. VideoFACT achieves decent results, albeit with less precision than MVFNet. Both VIDNet and DVIL demonstrate good performance, delivering results that are on-par with our method in terms of accuracy. Conversely, MVSS-Net, ManTra-Net, and FSG struggle to identify the manipulated areas correctly, frequently misplacing the localization or entirely missing the inpainted regions.

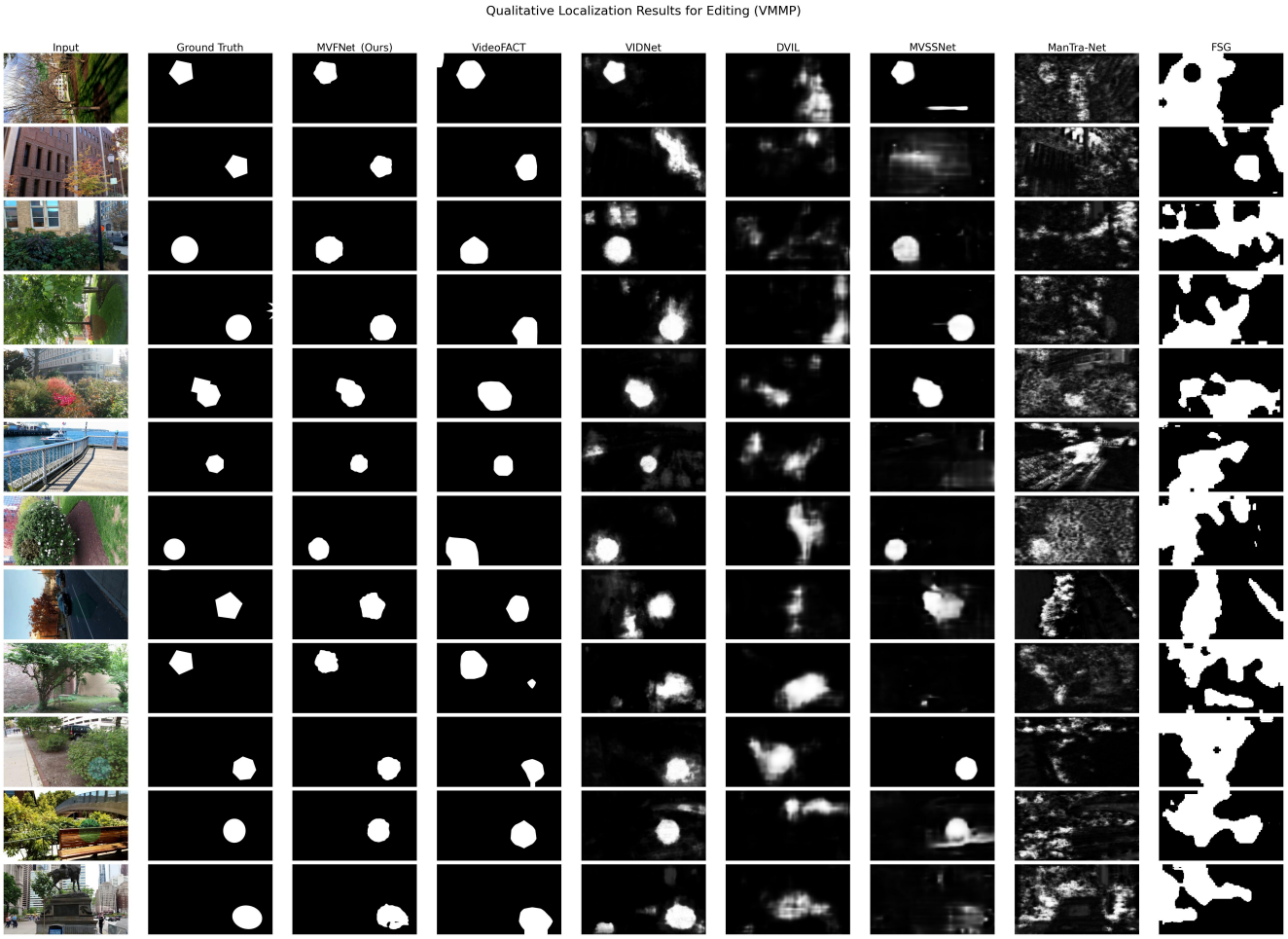

Figure 12. This figure compares the localization results for editing detection on the VMMP dataset. Our MVFNet method achieves the most accurate localization, closely reflecting the Ground Truth. VideoFACT shows competent localization capabilities, slightly less precise than MVFNet. VIDNet and DVIL, while generally effective, do not match the accuracy of VideoFACT and show a decline in performance. MVSS-Net impresses with successful localizations in the majority of cases but falls short in others. The remaining methods, ManTra-Net and FSG, largely fail to accurately localize the manipulations, with a tendency to either miss or incorrectly identify the edited regions.

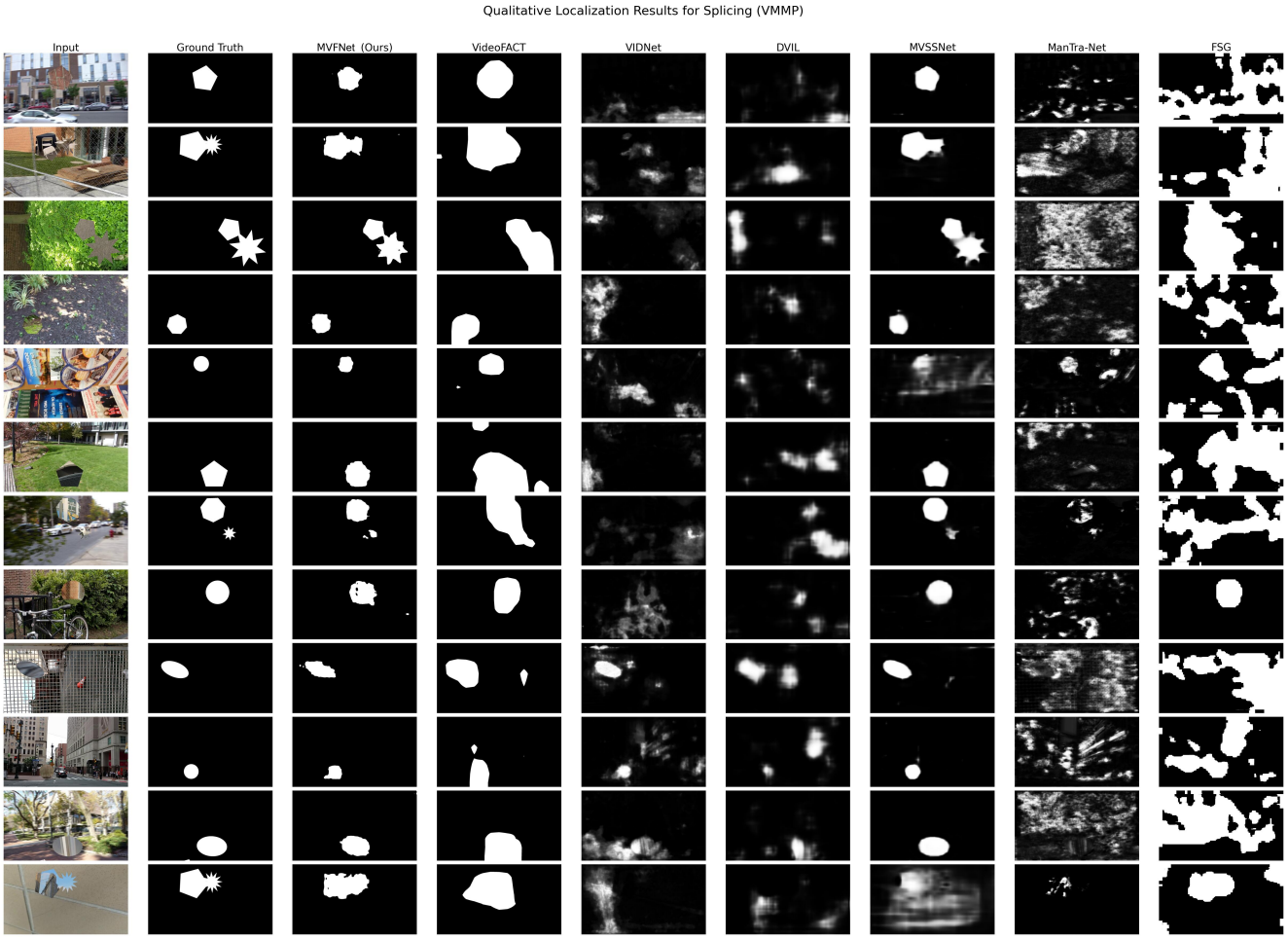

Figure 13. Localization accuracy for splicing detection is evaluated across different methods on the VMMP dataset. MVFNet showcases a high degree of precision, closely matching the Ground Truth. VideoFACT, while reasonably effective, does not attain the same level of accuracy as MVFNet. VIDNet and DVIL display less precision compared to VideoFACT, with their performance varying across different cases. MVSS-Net excels in a majority of scenarios but sometimes fails to detect the manipulation entirely. ManTra-Net and FSG generally provide inaccurate localizations, often identifying incorrect regions or missing the splicing altogether.

References

[1] Quentin Bammey, Rafael Grompone von Gioi, and Jean-Michel Morel. An adaptive neural network for unsupervised mosaic consistency analysis in image forensics. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14194–14204, 2020. 4

[2] Belhassen Bayar and Matthew C. Stamm. Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. *IEEE Trans. Inform. Forensics and Security*, 13(11):2691–2706, 2018. 1

[3] Xinru Chen, Chengbo Dong, Jiaqi Ji, Juan Cao, and Xirong Li. Image manipulation detection by multi-view multi-scale supervision. In *ICCV*, pages 14185–14193, October 2021. 2, 4

[4] Davide Alessandro Coccomini, Nicola Messina, Claudio Gennaro, and Fabrizio Falchi. Combining efficientnet and vision transformers for video deepfake detection. In *International conference on image analysis and processing*, pages 219–229. Springer, 2022. 2

[5] Davide Cozzolino and Luisa Verdoliva. Noiseprint: A cnn-based camera model fingerprint. *IEEE Trans. Inform. Forensics and Security*, 15:144–159, 2020. 1, 4

[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. 4

[7] Brian C Hosler, Xinwei Zhao, Owen Mayer, Chen Chen, James A Shackleford, and Matthew C Stamm. The video authentication and camera identification database: A new database for video forensics. *IEEE Access*, 7:76937–76948, 2019. 2

[8] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A. Efros. Fighting fake news: Image splice detection via learned self-consistency. In *ECCV*, September 2018. 1

[9] Owen Mayer and Matthew C. Stamm. Exposing fake images with forensic similarity graphs. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):1049–1064, 2020. 1, 2

[10] Owen Mayer and Matthew C. Stamm. Forensic similarity for digital images. *IEEE Trans. Inform. Forensics and Security*, 15:1331–1346, 2020. 1

[11] Trisha Mittal, Ritwik Sinha, Viswanathan Swaminathan, John Collomosse, and Dinesh Manocha. Video manipulations beyond faces: A dataset with human-machine analysis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 643–652, 2023. 4, 5

[12] Tai D. Nguyen, Shengbang Fang, and Matthew C. Stamm. Videofact: Detecting video forgeries using attention, scene context, and forensic traces. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 8563–8573, January 2024. 1, 2, 4

[13] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 1

[14] Kaede Shiohara and Toshihiko Yamasaki. Detecting deepfakes with self-blended images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18720–18729, 2022. 2

[15] Shujin Wei, Haodong Li, and Jiwu Huang. Deep video inpainting localization using spatial and temporal traces. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8957–8961. IEEE, 2022. 2

[16] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *CVPR*, pages 9535–9544, 2019. 2

[17] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deepfake detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2185–2194, 2021. 2

[18] Peng Zhou, Ning Yu, Zuxuan Wu, Larry S Davis, Abhinav Shrivastava, and Ser Nam Lim. Deep video inpainting detection. In *BMVC*, 2021. 2