

Deep Video Inpainting Detection

Peng Zhou¹

pengzhou@umd.edu

Ning Yu¹²

ningyu@cs.umd.edu

Zuxuan Wu¹

zxwu@cs.umd.edu

Larry S. Davis¹

lsd@umiacs.umd.edu

Abhinav Shrivastava¹

abhinav@cs.umd.edu

Ser-Nam Lim³

sernamlim@fb.com

¹ University of Maryland

College Park, MD, USA

² Max Planck Institute for Informatics

Saarbrücken, Saarland, Germany

³ Facebook AI

New York City, NY, USA

A Appendix

This supplementary document mainly contains the following:

1. Implementation details of our approach.
2. Details for our baselines.
3. Additional Ablation analysis of each component in our approach.
4. Robustness analysis of our approach under different perturbations.
5. Failure example visualization and analysis.

A.1 Implementation Details

We use PyTorch for implementation. Our model is trained on a NVIDIA TITAN P6000. The input to the network is resized to 240×427 . The length of our video clips is set to 3 frames during training. To extract ELA frames, we recompress the corresponding RGB frames by quality factor 50 and compute their difference. Our feature extraction backbone is VGG-16 [1] for both RGB and ELA features. To increase the generalization ability, we add instance normalization [2] layers to the backbone. The encoder is initialized from VGG-16 model pretrained on ImageNet [3] and the decoder is initialized by Xavier initialization [4]. We concatenate both RGB and ELA features up to the penultimate encoding layer. Afterwards, the features are passed into one convolutional and normalization layer to reduce the dimension by half to reduce training parameters. The QDLA module is added to the last encoder layer to extract directional feature information. The decoder is a 4-layer ConvLSTM. We use Adam [5] optimizer with a fixed learning rate of 1×10^{-4} for encoder and 1×10^{-3} for decoder. The optimizer of the encoder and decoder network are updated in an alternating

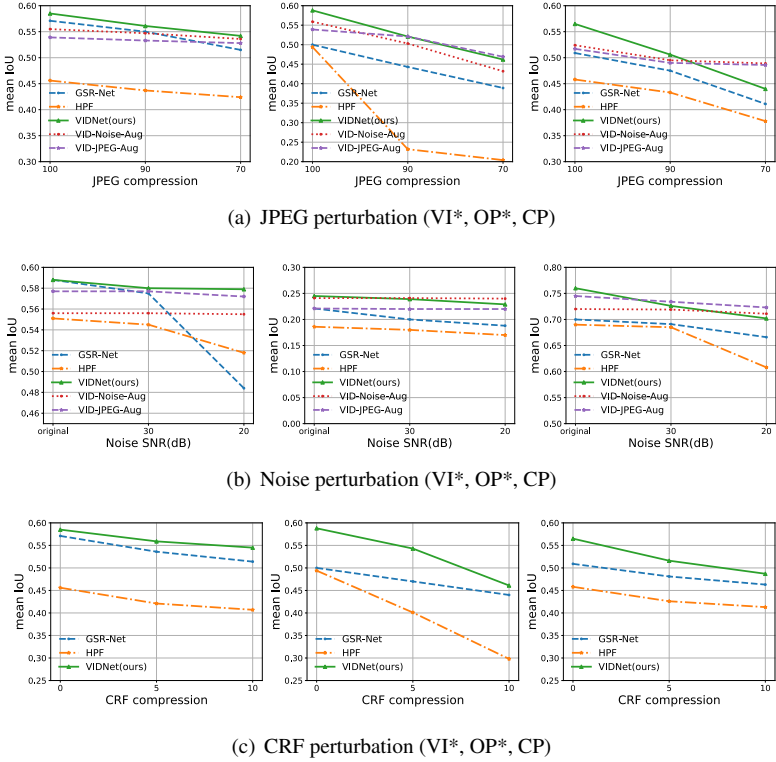


Figure 1: Mean IoU comparison under different perturbations. Perturbation in JPEG compression consists of the quality factor with 90 and 70; perturbation in noise consists of SNR 30dB and 20dB; perturbation in video CRF compression consists of the quality factor with 5 and 10. Column from left to right is the result on VI, OP and CP inpainting. ‘*’ denotes that the model is trained on these inpainting algorithms.

fashion. To avoid overfitting, weight decay with a factor of 5×10^{-5} and 50% dropout [14] are applied. Random horizontal flipping augmentation is applied during training. We train the whole network end-to-end for 40 epochs with a batch size of 4.

A.2 Baseline Details

NOI [8]: A traditional approach which aims to find inconsistent noise region as the clue of manipulation. The code for evaluation is from Zampoglou *et al.* [13]. We directly test on the VI, OP and CP test set as it is unsupervised.

CFA [8]: An approach that estimates Camera Filter Array (CFA) and regards the region with different CFA patterns as the manipulated region. We directly test on the VI, OP and CP test set as it is unsupervised.

COSNet [8]: To compare with video segmentation methods, we compare VIDNet with zero shot video segmentation method SOTA COSNet, which aligns with our setting. COSNet is based on deeplab [15], and attends to the flow difference between frames to segment out object.

HPF [8]: A learning based image inpainting detection approach that applies one high pass filter layer as an initialization to reveal high frequency inpainting artifacts. We imple-

Methods	VI*	OP*	CP
	IoU / F1	IoU / F1	IoU / F1
Ours ELA	0.460 / 0.578	0.509 / 0.631	0.417 / 0.546
Ours RGB (baseline)	0.552 / 0.671	0.456 / 0.580	0.493 / 0.625
Ours w/o QDLA	0.559 / 0.682	0.557 / 0.681	0.512 / 0.644
Ours RF edge	0.540 / 0.661	0.460 / 0.591	0.555 / 0.670
Co-Attention [14]	0.565 / 0.685	0.489 / 0.625	0.548 / 0.667
QDLA both features	0.555 / 0.680	0.580 / 0.700	0.495 / 0.635
VIDNet-IN (ours)	0.585 / 0.704	0.588 / 0.707	0.565 / 0.685

Table 1: **Ablation analysis.** The model is trained on VI and OP inpainting algorithms (denoted as ‘*’).

ment their filter kernel and train the network frame-by-frame from the ImageNet pretrained weights for comparison. For fair comparison, we also adapt it with LSTM to consider temporal information and report the LSTM version results.

GSR-Net [14]: A deeplab [10] based generic image manipulation segmentation approach that applies generative models and exploits boundary artifacts to improve the generalization ability. We use their released code and retrain on inpainted DAVIS frame-by-frame for evaluation. For fair comparison, we also adapt it with LSTM to consider temporal information and report the LSTM version results.

A.3 Additional Ablation Analysis

We analyze the importance of each key component in our framework and the details are as follows:

QDLA both features: Our full model except that the input to QDLA module is the concatenation of both RGB and ELA feature from the 5-th layer.

Ours RF edge: Following Chen *et al.* [10], we add additional edge branch and replace QDLA with recursive filter to the final prediction. The output of edge branch is used as the reference to recursive filter layer.

Co-attention [14]: We replace our QDLA module with contextual attention [14], which is also designed to learn from adjacent regions.

Tab. 1 displays the comparison results. Compared to *Ours RF edge*, our full model which contains QDLA module yields better performance because the boundary prediction degrades in video inpainting scenario and thus edge map contains false positives to guide the segmentation branch. Also, thanks to the disentanglement of four directions, our QDLA module captures better adjacent artifacts than *co-attention* [14]. The result also shows that the high level ELA features are less helpful than lower ones when comparing ours with *QDLA both features*. Eventually, with QDLA module, ELA feature and temporal information, the performance gets boosted further.

A.4 Robustness Analysis

To test the robustness of our approach under noise, JPEG and video compression perturbation, we conduct experiments listed in Fig. 1. We add Gaussian noise to the input frame with Signal-to-Noise Ratio (SNR) 30 and 20 dB and evaluate on these noisy frames, or re-compress test frame with JPEG quality 90 and 70 for perturbation, or compress video under

H264 constant rate factor (CRF) value 5 and 10. Moreover, to study the effect of specific augmentation on performance, we apply noise and JPEG augmentation to our approach and make comparison together. The details of our augmentation is as follow.

VID-Noise-Aug: Randomly apply Gaussian noise with SNR 20 dB to the input frames during training.

VID-JPEG-Aug: Randomly apply JPEG compression with quality factor 90 to the input frames during training.

The robustness of our approach stands out under different perturbations. Compared to other approaches, HPF suffers more from perturbation because more high frequency noises will be introduced. With generative models for augmentation, GSR-Net shows good robustness. However, our approach outperforms GSR-Net as more modalities of video inpainting clues have been considered. Even though adding noise augmentation yields a small degradation on the original performance, the robustness to both noise and JPEG perturbation has been improved. Similar observation is made on JPEG augmentation. Unsurprisingly, the robustness of our method under video compression perturbation is also better than other methods as more temporal features are utilized in our approach.

A.5 Failure Example Visualization and Analysis

We visualize some failure cases of our approach in Fig. 2. The failure cases could be summarized as 1) Unusual inpainting ratio cases. It is expected to be improved by advanced multi scale training methods. 2) The spatial discontinuity or multiple instances (The first and second case in Fig. 2). It might due to the limited multi-instance samples in the training set. 3) Noise in ELA features. As shown in Fig. 2 (the second case), the false positive is likely to be caused by the noisy background in ELA frames.

References

- [1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. In *TPAMI*, 2018.
- [2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.
- [3] Pasquale Ferrara, Tiziano Bianchi, Alessia De Rosa, and Alessandro Piva. Image forgery localization via fine-grained analysis of cfa artifacts. In *TIFS*, 2012.
- [4] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feed-forward neural networks. In *AISTATS*, 2010.
- [5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [6] Haodong Li and Jiwu Huang. Localization of deep inpainting using high-pass fully convolutional network. In *ICCV*, 2019.
- [7] Xiankai Lu, Wenguan Wang, Chao Ma, Jianbing Shen, Ling Shao, and Fatih Porikli. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *CVPR*, 2019.

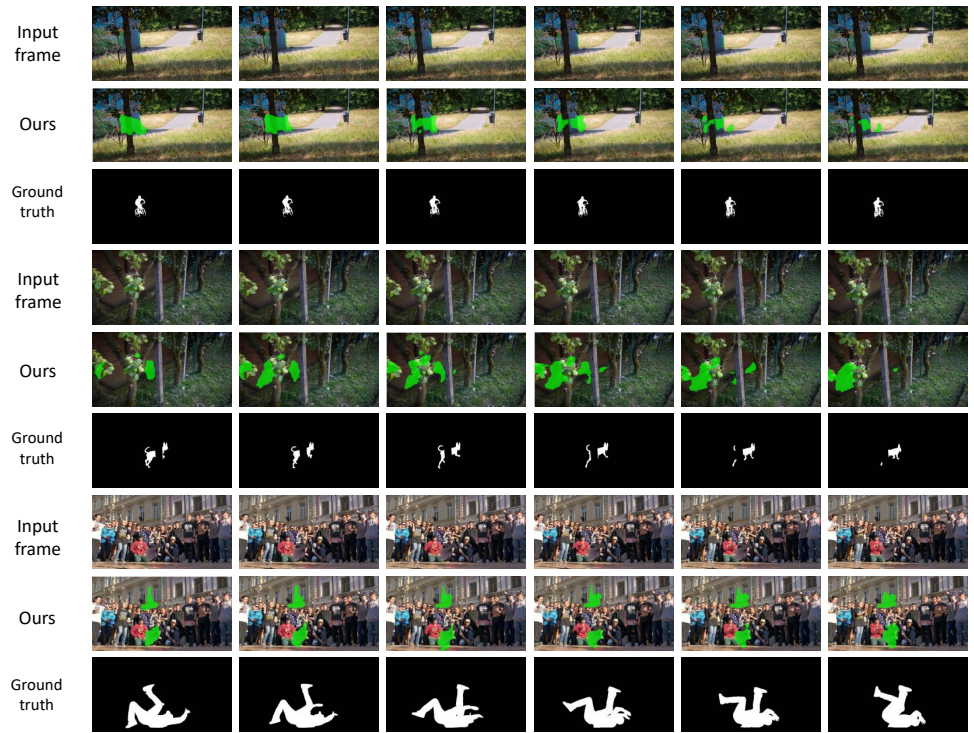


Figure 2: Failure case visualization on DAVIS. The first row shows the inpainted video frame. The second row indicates the final predictions of our approach. The third row is the ground truth.

[8] Babak Mahdian and Stanislav Saic. Using noise inconsistencies for blind image forensics. In *IMAVIS*, 2009.

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[10] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. In *JMLR*, 2014.

[11] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[12] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *CVPR*, 2018.

[13] Markos Zampoglou, Symeon Papadopoulos, and Yiannis Kompatsiaris. Detecting image splicing in the wild (web). In *ICMEW*, 2015.

- [14] Peng Zhou, Bor-Chun Chen, Xintong Han, Mahyar Najibi, Abhinav Shrivastava, Ser Nam Lim, and Larry S Davis. Generate, segment and refine: Towards generic manipulation segmentation. *AAAI*, 2020.