DEEP VIDEO INPAINTING LOCALIZATION USING SPATIAL AND TEMPORAL TRACES

Shujin Wei[†‡], *Haodong Li*[†‡*], *Jiwu Huang*[†‡]

[†]Guangdong Key Laboratory of Intelligent Information Processing and Shenzhen Key Laboratory of Media Security, Shenzhen University, Shenzhen 518060, China
[‡]Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen 518172, China

ABSTRACT

Advanced deep-learning-based video inpainting can fill a specified video region with visually plausible contents, usually leaving imperceptible traces. As inpainting can be used for malicious video manipulations, it has led to potential privacy and security issues. Therefore, it is necessary to detect and locate the video regions subjected to deep inpainting. This paper addresses this problem by exploiting the spatial and temporal traces left by inpainting. Firstly, the inpainting traces are enhanced by intra-frame and inter-frame residuals. In particular, we guide the extraction of inter-frame residual with optical-flow based frame alignment, which can better reveal the inpainting traces. Then, a dual-stream network, acting as the encoder, is designed to learn discriminative features from frame residuals. Finally, bidirectional convolutional LSTMs are embedded in the decoder network to produce pixel-wise predictions of inpainted regions for each frame. The proposed method is evaluated with tampered videos created by two state-of-the-art deep video inpainting algorithms. Extensive experimental results show that the proposed method can effectively localize the inpainted regions, outperforming existing methods.

Index Terms— Video forgery detection, video inpainting, optical flow, frame residual

1. INTRODUCTION

As a kind of video editing technique, video inpainting is originally developed to repair the damaged or unwanted regions in a video via generating spatiotemporal consistent contents. Although video inpainting is helpful in some productive applications, such as video restoration and film post-production, it can also be maliciously used for video tampering. For example, erasing visible copyright watermarks or removing certain objects to synthesize fake information. With the rapid development of video inpainting techniques, it becomes more and more difficult to recognize the inpainted videos with naked eyes. Hence, it is necessary to identify the inpainted regions within a given video from a security point of view.

The conventional video inpainting approaches [1, 2, 3] usually complete missing regions in a temporal-extended form of the patch-based image inpainting [4]. Although acceptable results can be achieved, they cannot well deal with complex motion and suffer from high complexity. Nowadays, deep-learning-based approaches have substantially improved video inpainting [5, 6]. By using 3D temporal convolution [7, 8], optical flow [9, 10], and attention mechanism [11, 12], existing state-of-the-art video inpainting methods can extract useful temporal information from neighboring frames and create contents with high spatiotemporal consistency. With such advanced deep video inpainting techniques, one can manipulate a video with photo-realistic contents, posing potential threats to video authentication.

To authenticate digital videos, many forensic methods have been developed [13]. Among them, some methods aim to detect conventional video inpainting [14, 15, 16], but their performance for deep video inpainting would be degraded due to the fact that the introduced traces are different. Currently there are only a few methods try to handle the forensics of deep video inpainting [17, 18], while the traces of deep inpainting have not been fully exploited. On the other hand, several image forensic methods could be used to locate the inpainted video regions in a frame-by-frame manner [19, 20], but they fail to utilize the temporal correlation among video frames and thus would obtain unfavorable performance.

In this paper, we propose an end-to-end framework to locate the video regions manipulated by deep inpainting. Firstly, to dig out the spatial and temporal traces left by deep inpainting, the intra-frame and inter-frame residuals are extracted, where the extraction of inter-frame residual is particularly guided by optical flow for better revealing the inpainting traces. Then, discriminative features are learned from the frame residuals with a dual-stream network. Finally, a decoder network equipped with bidirectional convolutional LSTMs is designed to produce pixel-wise localization results. We have evaluated the proposed method with tampered videos created by two state-of-the-art deep video inpainting algorithms, and the experimental results show that our method outperforms four existing related methods.

2. PROPOSED METHOD

The proposed end-to-end framework for locating the inpainted video regions is shown in Fig. 1. Firstly, intra-frame and inter-frame residuals are extracted from each video frame for enhancing the inpainting traces. To fuse the two types of residuals, a dual-stream network is then used to encode

*Corresponding author.

This work was supported in part by the Key-Area Research and Development Program of Guangdong Province (2019B010139003) and in part by the NSFC (61802262, U19B2022).

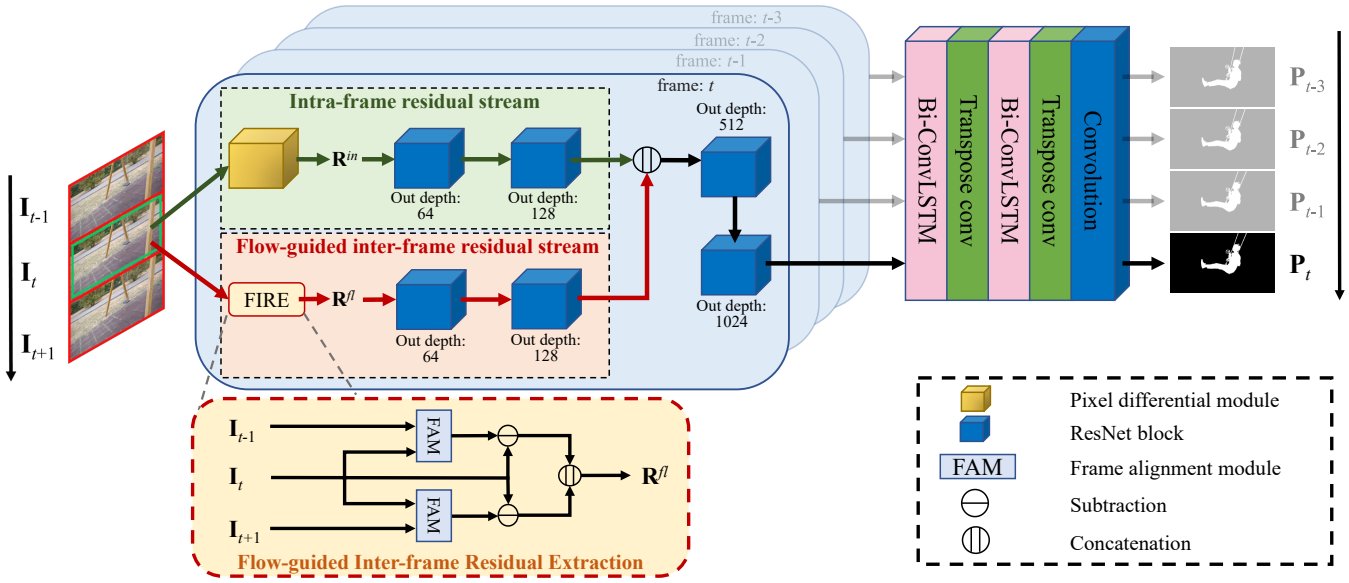


Fig. 1. The overall framework of the proposed method.

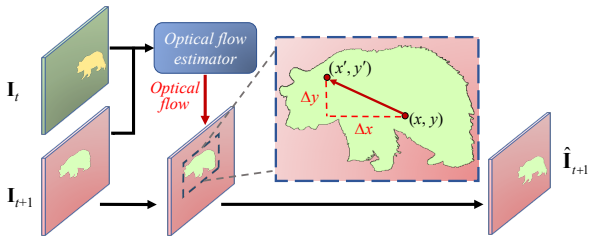


Fig. 2. The frame alignment module.

the residuals into deep features. Finally, a decoder network composed of bidirectional ConvLSTMs [21] and transpose convolutions is designed to model the temporal correlation among sequential frames and output pixel-wise predictions. The details of our method are elaborated as follows.

2.1. Extraction of frame residuals

The key to exposing the inpainted regions is to capture the inpainting traces. As the contents within the inpainted regions are inferred from the known pixels in the same frame or propagated from neighboring frames, certain spatial and temporal traces would be inevitably left. However, such traces are much weaker compared to the video contents. Hence, we try to enhance the spatial and temporal inpainting traces with intra-frame and inter-frame residuals.

To obtain the intra-frame residual, the first-order derivative high-pass filters used in [19] are adopted. Namely, each video frame is fed to a pixel differential module that takes adjacent pixel differences in horizontal, vertical, and main-diagonal directions, respectively. In this way, the video contents is largely suppressed and the obtained residuals can reflect the inpainting inconsistencies left in an individual frame.

In addition to intra-frame spatial traces, video inpainting would introduce abnormal traces w.r.t. temporal information due to the imperfect completion of moving objects, implying that inter-frame relationship is also essential for video inpainting localization. In [18], temporal convolution is applied to model inter-frame relationship via computing the residual between a target frame and its neighbors. However, the obtained results would be disturbed by the motions of video contents. To address this issue, we first align adjacent video frames based on optical flow and then construct the flow-guided inter-frame residual. The frame alignment is executed as shown in Fig. 2. Given two adjacent video frames $\mathbf{I}_t$ and $\mathbf{I}_{t+1}$, the optical flow is firstly computed by the pretrained RAFT [22] network $\mathcal{F}$:

$$\mathbf{F}_{t\to t+1} = \mathcal{F}(\mathbf{I}_t, \mathbf{I}_{t+1}). \tag{1}$$

Supposing that the flow vector for a pixel $\mathbf{I}_t(x, y)$ in the frame $\mathbf{I}_t$ is $\mathbf{F}_{t\to t+1}(x, y) = (\Delta x, \Delta y)$, we can associate $\mathbf{I}_t(x, y)$ with a pixel $\mathbf{I}_{t+1}(x', y')$ in the frame $\mathbf{I}_{t+1}$, satisfying that

$$(x', y') = ([x + \Delta x], [y + \Delta y]), \tag{2}$$

where $[\cdot]$ is the rounding operation. Then, the pixels in the frame $\mathbf{I}_{t+1}$ can be projected to new positions to form a virtual frame $\hat{\mathbf{I}}_{t+1}$ by using the following equation

$$\begin{cases} \hat{\mathbf{I}}_{t+1}(x, y) = \mathbf{I}_{t+1}(x', y'), & 0 \le x' < w, 0 \le y' < h, \\ \hat{\mathbf{I}}_{t+1}(x, y) = 0, & else, \end{cases} \tag{3}$$

where w and h are the width and height of the frame. In this way, the contents of $\hat{\mathbf{I}}_{t+1}$ are aligned with $\mathbf{I}_t$. Similarly, another virtual frame $\hat{\mathbf{I}}_{t-1}$ can be generated from $\mathbf{I}_{t-1}$ based on $\mathbf{F}_{t\to t-1}$, and the inter-frame residual is computed as

$$\mathbf{R}^{fl} = \{\mathbf{I}_t - \hat{\mathbf{I}}_{t-1}, \mathbf{I}_t - \hat{\mathbf{I}}_{t+1}\}, \tag{4}$$

where $\{\cdot, \cdot\}$ denotes the concatenation operation.

Fig. 3 visualizes an example of intra-frame and inter-frame residuals. It is observed that the inpainted region looks blurry to some extent in the intra-frame residual, and it presents the ghost of object removal in the inter-frame residual, implying that the two types of residuals are beneficial for the localization of inpainted regions.

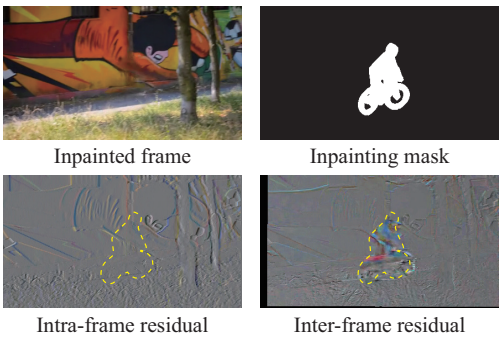


Fig. 3. Visualization of intra-frame and inter-frame residuals.

2.2. Dual-stream encoder network

Since the two types of residuals have different properties, they are not directly combined together before being fed to the subsequent deep network. Instead, a dual-stream encoder network is designed, which first separately processes the intra-frame and inter-frame residuals and then fuses the learned features. As shown in Fig. 1, the dual-stream network consists of a series of ResNet [23] blocks. Each ResNet block contains two bottleneck units, each of which is composed of three successive convolutional layers and an identity skip connection. The kernel sizes of the three convolutional layers are 1×1, 3×3, and 1×1, respectively; the convolution stride is 1, except that the last layer of each block has a stride of 2 for spatial pooling; instance normalization and ReLu activation are performed before each convolution. For each data stream, two ResNet blocks are equipped and a 128-channel feature map is learned. The feature maps from two streams are then concatenated and fed to another two ResNet blocks, and finally a 1024-channel feature map is output, whose spatial resolution is $1/16$ of the input video frame.

2.3. Decoder network

In order to produce pixel-wise predictions, it is straightforward to decode the learned features with transpose convolutions. However, this would fail to exploit the temporal correlation among sequential frames. In order to incorporate the temporal correlation for improving the localization performance, bidirectional convolutional LSTMs (Bi-ConvLSTMs) are embedded into a transpose convolution based decoder network. As shown in Fig. 1, the decoder network contains two transpose convolutions, each of which performs a $4\times$ upscaling so that the final output has the same size as the input frame. Before each transpose convolution, a Bi-ConvLSTM is used to take the learned features of four sequential frames as input, so as to model the temporal correlation in both forward and backward directions. The features output by LSTM are concatenated along the channel dimension. Lastly, to weaken the checkerboard artifacts introduced by transpose convolution, a 5×5 convolution is performed and it output 2-channel logits. By applying SoftMax to the logits we can classify the pixels within a video frame and obtain the localization results.

2.4. Loss function

Considering that the inpainted regions are usually smaller than the pristine ones, we choose to optimize the dice loss [24] for dealing with class imbalance. The dice loss can be formulated as

$$L_{dice} = 1 - \frac{2\sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2}, \tag{5}$$

where N is the total number of pixels, and p_i and g_i are the prediction and ground truth label, respectively. In fact, optimizing the dice loss is equivalent to optimizing the F1-score. Based on our experiment, the dice loss yields better results compared to the combined loss [18] (including focal loss, IoU loss and SSIM loss; refer to Table 1 for details).

Table 1. F1-scores / IoUs of different variants.

Methods	FGVC [10]	STTN [12]
Inter-frame only (Dice loss)	0.43 / 0.30	0.60 / 0.48
Inter+Intra (Dice loss)	0.50 / 0.36	0.64 / 0.52
Intra+ConvLSTM (k=4, Dice loss)	0.54 / 0.41	0.59 / 0.48
Inter+ConvLSTM (k=4, Dice loss)	0.51 / 0.39	0.65 / 0.54
Inter+Intra+ConvLSTM (k=3, Dice loss)	0.59 / 0.46	0.68 / 0.57
Inter+Intra+ConvLSTM (k=5, Dice loss)	0.58 / 0.45	0.67 / 0.56
Inter+Intra+ConvLSTM (k=4, Combined loss)	0.59 / 0.46	**0.69 / 0.58**
Inter+Intra+ConvLSTM (k=4, Dice loss)	**0.61 / 0.48**	0.68 / 0.57

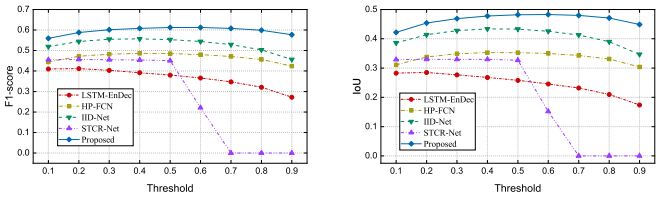
3. EXPERIMENTAL RESULTS

3.1. Experimental setup

In the experiments, we used different datasets to construct the training and testing data. The training data was generated from three datasets, *i.e.*, UAV123 [25], GOT-10k [26] and VisDrone2018 [27]. As only box-level labels were provided in these datasets, we used the video object tracking algorithm SiamMask [28] to obtain finer masks. The testing data come from DAVIS2016 [29] and DAVIS2017 [30], in which pixel-level masks were available. Based on the masks, each video was inpainted by the FGVC [10] and STTN [12] algorithms, producing two inpainted counterparts. All the inpainted videos were saved by H.264 with the constant QP 23 and with the size of 240×432. In order to conduct validation, some videos with good inpainting quality were selected from the training data. In total, the training and validation sets consisted of 1639 and 150 videos, respectively, and the testing set consisted of 100 videos.

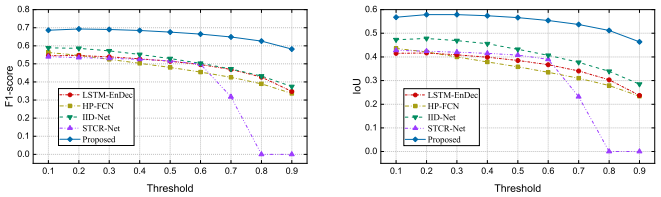
The proposed network was implemented with Tensorflow. During the training, Adam [31] optimizer was adopted, and the initial learning rate was set to 1×10^{-4} and decreased 50% after each epoch. We initialized the kernel weights with Xavier and the biases with zeros, and used L2 regularization with a weight decay of 1×10^{-4}. The final model was chosen as the best one in an 8-epoch training process. The batch size used in training and testing was 4, and all the experiments was conducted with a Nvidia Tesla P100 GPU. The implementation is available at: `https://github.com/ShujinW/Deep-Video-Inpainting-Localization`.

To evaluate the localization performance, the pixel-level F1-score and Intersection over Union (IoU) were used as the performance metrics. The metrics were independently calculated for each video frame and the mean values over all testing video frames were reported.



(a) F1-scores for FGVC

(b) IoUs for FGVC



(c) F1-scores for STTN

(d) IoUs for STTN

Fig. 4. F1-scores and IoUs obtained with different thresholds.

Table 2. F1-scores / IoUs of different forensic methods. The inpainting algorithms included in training are mark with *.

Methods	VI [5]*	OP [11]*	CP [6]
HP-FCN [19]	0.57/0.46	0.62/0.49	0.58/0.46
GSR-Net [32]	0.69/0.57	0.63/0.50	0.63/0.51
VIDNet [17]	0.70/0.59	0.71/0.59	0.69/0.57
FAST [33]	**0.73/0.61**	0.78/0.65	0.76/0.63
Proposed	**0.73**/0.60	**0.80/0.69**	**0.77/0.65**

3.2. Ablation study

In this experiment, we compared the proposed method with some of its variants, including using only one data stream, adding the ConvLSTMs with the features of k sequential frames as input, and using different loss functions. The obtained results are shown in Table 1, where the last row corresponds to the proposed method. From this table, we can observe that combining both residual streams and using ConvLSTMs with 4 sequential frames achieves more favorable performance, and the use of dice loss slightly outperforms the combined loss used in [18].

3.3. Quantitative performance evaluation

In this experiment, we compared the performance of the proposed method and four existing methods, including a general image tampering localization method (LSTM-EnDec [34]), two deep image inpainting localization methods (HP-FCN [19] and IID-Net [20]), and a deep video inpainting localization method (STCR-Net [18]). The models of all the methods were trained and tested with the data as described above. We converted the predictions obtained by different methods to binary maps by using a series of thresholds and computed the F1-scores and IoUs as shown in Fig. 4. From this figure, it is observed that the proposed method significantly outperforms all the four competitors in a wide range of thresholds.

To further study the performance for different inpainting algorithms, we also evaluated the proposed method on the DAVIS dataset and compared with GSR-Net [32], VIDNet [17] and FAST [33]. As did in [17] and [33], the videos in DAVIS2016 [29] was inpainted by VI [5], OP [11], and CP [6]. The obtained result are shown in Table 2. It is observed that our method outperforms the existing ones.

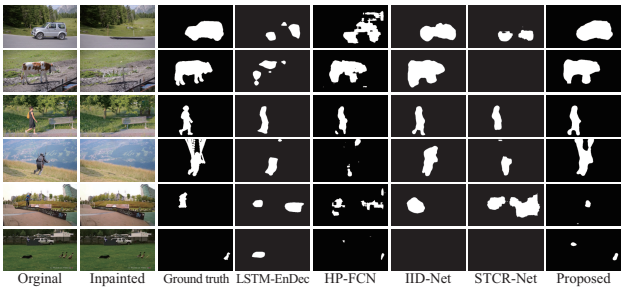
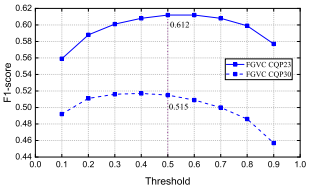
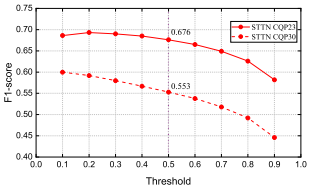


Fig. 5. Examples of localization results.



(a) F1-scores for FGVC



(b) F1-scores for STTN

Fig. 6. F1-scores for different QPs.

3.4. Qualitative performance evaluation

Some example localization results are illustrated in Fig. 5. The upper four cases clearly show that the proposed method can locate the inpainted regions in a much more accurate manner. Although some miss detections and false alarms are presented in the last two cases, the proposed method still gets better results than other methods. We also note that our method achieves a low false alarm rate (lower than 5%).

3.5. Robustness against compression quality

To evaluate the robustness of our method against different QPs, we trained the network by using videos compressed with the constant QP (CQP) 23 and then tested the model by using videos compressed with CQP23 and CQP30, respectively. As shown in Fig. 6, the F1-scores for inpainting methods FGVC and STTN are degraded about 0.1 when the compression changing from CQP23 to CQP30.

4. CONCLUSION

In this paper, we propose a method to locate the video regions manipulated by deep inpainting. The proposed method relies on the spatiotemporal traces left by deep inpainting, which is enhanced by intra-frame and inter-frame residuals. It should be highlighted that the inter-frame residual is extracted with the guiding of optical flow so that the spatiotemporal inconsistencies of inpainting is better represented. The two types of residuals are sent to a dual-stream network for learning discriminative features. To further utilize the temporal correlation among sequential frames, Bi-ConvLSTMs are embedded into the decoder network. The proposed end-to-end framework can locate the inpainted regions with pixel-wise predictions, achieving significantly better performance compared to existing methods. In the future, we will further improve the robustness of our method and enhance its generalization capability for different inpainting algorithms.

5. REFERENCES

[1] Yonatan Wexler, Eli Shechtman, and Michal Irani, "Space-time completion of video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 463–476, 2007.

[2] Timothy K Shih, Nick C Tang, and Jenq-Neng Hwang, "Exemplar-based video inpainting without ghost shadow artifacts by maintaining temporal continuity," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 3, pp. 347–360, 2009.

[3] Alasdair Newson, Andrés Almansa, Matthieu Fradet, Yann Gousseau, and Patrick Pérez, "Video inpainting of complex scenes," *SIAM Journal on Imaging Sciences*, vol. 7, no. 4, pp. 1993–2019, 2014.

[4] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 24, 2009.

[5] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon, "Deep video inpainting," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5792–5801.

[6] Sungho Lee, Seoung Wug Oh, DaeYeun Won, and Seon Joo Kim, "Copy-and-paste networks for deep video inpainting," in *IEEE International Conference on Computer Vision*, 2019, pp. 4413–4421.

[7] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu, "Free-form video inpainting with 3D gated convolution and temporal PatchGAN," in *IEEE International Conference on Computer Vision*, 2019, pp. 9066–9075.

[8] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu, "Learnable gated temporal shift module for deep video inpainting," *arXiv preprint arXiv:1907.01131*, 2019.

[9] Rui Xu, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy, "Deep flow-guided video inpainting," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3723–3732.

[10] Chen Gao, Ayush Saraf, Jia-Bin Huang, and Johannes Kopf, "Flow-edge guided video completion," in *European Conference on Computer Vision*, 2020, pp. 713–729.

[11] Seoung Wug Oh, Sungho Lee, Joon-Young Lee, and Seon Joo Kim, "Onion-peel networks for deep video completion," in *IEEE International Conference on Computer Vision*, 2019, pp. 4403–4412.

[12] Yanhong Zeng, Jianlong Fu, and Hongyang Chao, "Learning joint spatial-temporal transformations for video inpainting," in *European Conference on Computer Vision*, 2020, pp. 528–543.

[13] Simone Milani, Marco Fontani, Paolo Bestagini, Mauro Barni, Alessandro Piva, Marco Tagliasacchi, and Stefano Tubaro, "An overview on video forensics," *APSIPA Transactions on Signal and Information Processing*, vol. 1, pp. e2, 2012.

[14] Mustapha Aminu Bagiwa, Ainuddin Wahid Abdul Wahab, Mohd Yamani Idna Idris, and Suleman Khan, "Digital video inpainting detection using correlation of Hessian matrix," *Malaysian Journal of Computer Science*, vol. 29, no. 3, pp. 179–195, 2016.

[15] Shanshan Bai, Haichao Yao, Rongrong Ni, and Yao Zhao, "Detection and localization of video object removal by spatio-temporal LBP coherence analysis," in *International Conference on Image and Graphics*, 2019, pp. 244–254.

[16] Cheng-Shian Lin and Jyh-Jong Tsay, "A passive approach for effective detection and localization of region-level video forgery with spatio-temporal coherence analysis," *Digital Investigation*, vol. 11, no. 2, pp. 120–140, 2014.

[17] Peng Zhou, Ning Yu, Zuxuan Wu, Larry S Davis, Abhinav Shrivastava, and Ser-Nam Lim, "Deep video inpainting detection," *arXiv preprint arXiv:2101.11080*, 2021.

[18] Xiangling Ding, Yifeng Pan, Kui Luo, Yanming Huang, Junlin Ouyang, and Gaobo Yang, "Localization of deep video inpainting based on spatiotemporal convolution and refinement network," in *IEEE International Symposium on Circuits and Systems*, 2021, pp. 1–5.

[19] Haodong Li and Jiwu Huang, "Localization of deep inpainting using high-pass fully convolutional network," in *IEEE International Conference on Computer Vision*, 2019, pp. 8301–8310.

[20] 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*, 2021.

[21] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015, pp. 802–810.

[22] Zachary Teed and Jia Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *European Conference on Computer Vision*, 2020, pp. 402–419.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Identity mappings in deep residual networks," in *European Conference on Computer Vision*, 2016, pp. 630–645.

[24] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Fourth International conference on 3D vision*, 2016, pp. 565–571.

[25] Matthias Mueller, Neil Smith, and Bernard Ghanem, "A benchmark and simulator for UAV tracking," in *European Conference on Computer Vision*, 2016, pp. 445–461.

[26] Lianghua Huang, Xin Zhao, and Kaiqi Huang, "Got-10k: A large high-diversity benchmark for generic object tracking in the wild," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 5, pp. 1562–1577, 2021.

[27] Pengfei Zhu, Longyin Wen, Xiao Bian, Haibin Ling, and Qinghua Hu, "Vision meets drones: A challenge," *arXiv preprint arXiv:1804.07437*, 2018.

[28] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr, "Fast online object tracking and segmentation: A unifying approach," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1328–1338.

[29] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 724–732.

[30] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool, "The 2017 DAVIS challenge on video object segmentation," *arXiv preprint arXiv:1704.00675*, 2017.

[31] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[32] Peng Zhou, Bor-Chun Chen, Xintong Han, Mahyar Najibi, Abhinav Shrivastava, Ser-Nam Lim, and Larry Davis, "Generate, segment, and refine: Towards generic manipulation segmentation," in *Proceedings of the AAAI conference on artificial intelligence*, 2020, pp. 13058–13065.

[33] Bingyao Yu, Wanhua Li, Xiu Li, Jiwen Lu, and Jie Zhou, "Frequency-aware spatiotemporal transformers for video inpainting detection," in *IEEE International Conference on Computer Vision*, 2021, pp. 8188–8197.

[34] Jawadul H Bappy, Cody Simons, Lakshmanan Nataraj, BS Manjunath, and Amit K Roy-Chowdhury, "Hybrid LSTM and encoder–decoder architecture for detection of image forgeries," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3286–3300, 2019.