Multi-View Inconsistency Analysis for Video Object-Level Splicing Localization

Pengfei Peng [1], Guoqing Liang [2] and Tao Luan [3]
[1]School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100085, China
[2]Taiyuan Coal Gasification (Group) Co., Ltd. No. 29 Heping South Road, Wanbailin District, Taiyuan, Shanxi, China
[3] The Institute of Software, Chinese Academy of Sciences, Beijing 100085, China
luantao@iscas.ac.cn

***Abstract*—In the digital era, the widespread use of video content has led to the rapid development of video editing technologies. However, it has also raised concerns about the authenticity and integrity of multimedia content. Video splicing forgery has emerged as a challenging and deceptive technique used to create fake video objects, potentially for malicious purposes such as deception, defamation, and fraud. Therefore, the detection of video splicing forgery has become critically important. Nevertheless, due to the complexity of video data and a lack of relevant datasets, research on video splicing forgery detection remains relatively limited. This paper introduces a novel method for detecting video object splicing forgery, which enhances detection performance by deeply exploring inconsistent features between different source videos. We incorporate various feature types, including edge luminance, texture, and video quality information, and utilize a joint learning approach with Convolutional Neural Network (CNN) and Vision Transformer (ViT) models. Experimental results demonstrate that our method excels in detecting video object splicing forgery, offering promising prospects for further advancements in this field.**

***Index Terms*—Video splicing forgery, Multi-view feature learning, Object-level forgery detection**

I. Introduction

In the digital era, the rapid dissemination of video content and the widespread use of video editing techniques have raised significant concerns about the authenticity and integrity of multimedia content [1–4]. Among various forms of digital content manipulation, video splicing forgery has emerged as a challenging and deceptive technology. Video splicing is commonly employed to introduce fabricated elements, involving the synthesis of objects or scenes from different sources to create fraudulent videos. Such manipulated videos can be used for various malicious purposes, including misinformation, defamation, and fraud. The widespread propagation of video object splicing poses a substantial threat to the credibility of visual information in an increasingly digital and interconnected world. Therefore, the need for robust and effective methods to detect video object splicing forgery has become paramount, even in the face of advanced adversarial attempts.

While techniques for detecting image manipulation have made some progress and various methods exist for image splicing detection, the research on video tampering localization is still relatively limited. Some image splicing detection methods utilize cues such as edge artifacts [1], pixel traces [1, 5], incongruities in physical lighting [6], and compression artifacts [7] to distinguish spliced regions originating from different sources. However, due to the inherent complexity of video data and the lack of publicly available video datasets, research on video tampering localization techniques remains underexplored. The challenge in detecting video splicing forgery lies in the fact that the spliced objects may come from entirely different real-world scenes, and common deep learning-based forgery artifacts, such as unnatural pixel or boundary artifacts, may not be present [1, 8]. Existing methods are often based on these forgery artifacts, which may explain their suboptimal performance in extracting inconsistency information.

To enhance the performance of splicing content detection, the key lies in delving deeper into the incongruity traces between two dissimilar source videos. It is observed that two non-homogeneous images/videos typically originate from different scenes, resulting in disparities due to distinct shooting angles, lighting conditions, camera settings, or other factors. These differences encompass visual effects (e.g., color/brightness variations, lighting inconsistencies, camera noise), disparities in resolution and frame rates, variations in splicing effects (e.g., transition effects, brightness changes at splicing edges caused by frame translation and scaling), as well as motion continuity and perspective changes. Another challenge is the continuous advancement of video editing and processing techniques. Spliced videos may undergo a series of alterations, such as re-rendering, re-recording, and re-compression, potentially compromising or erasing the disparities and artifacts that may have originally existed. This further complicates the detection task, necessitating the design of an effective feature extraction network and methodology.

Semantic segmentation is a pivotal computer vision technique that plays a crucial role in object splicing detection. Semantic segmentation, by assigning each pixel in an image to different semantic categories, aids in precise object boundary localization, identifying distinct objects, providing rich image descriptions, and improving the performance of segmentation networks [9–11]. By incorporating the knowledge of semantic segmentation, a better understanding and segmentation of spliced objects can be achieved. We introduce an effective network that combines the local feature learning of Convolutional Neural Networks (CNNs) with the global feature learning of Visual Transformers (ViTs) to fuse and learn incongruous features within splicing of two non-homogeneous videos.

ISSN:3006-2985(Print)
ISSN:3006-9297(Online)

In the process of addressing these challenges, this paper draws inspiration from the progress made in image splicing detection and extends it to the field of video object splicing forgery detection. We initially extract information such as video texture, quality, and splicing edges and proceed to fuse these features. Additionally, we design a temporal feature encoding module to effectively capture motion information in videos.

In summary, the primary contributions of this paper include:

- Proposing a novel method for detecting video object splicing forgery by effectively identifying distinct features in lighting, texture, and quality information between two different source videos.
- Enhancing robustness through multi-stage learning, exhibiting high performance in the face of video editing and processing techniques that significantly impact splicing detection.
- Demonstrating the effectiveness of our method through extensive experimental results on a dataset of video object splicing, outperforming existing state-of-the-art methods. This will contribute to advancing the detection of non-homogeneous video object splicing forgery.

II. Related Works

The creation of video splicing involves the use of tools similar to Photoshop to generate objects manually. Each object to be added to the video requires careful consideration, taking into account whether it fits semantically and is consistent with the background. Precise determination of the splicing location, as well as the scale and size of the spliced object in the background, is essential. Factors like brightness, color coordination between the foreground and background, and various other elements need to be considered. Currently, deep learning-based object splicing detection techniques have not yet reached the expected level of performance, as research in this field has mainly focused on images, lacking relevant video datasets[8, 12–14].

PQMECNet [7] utilizes local estimation of JPEG quantization matrices to distinguish spliced regions originating from different sources. MVSS-Net [1] learns semantically independent and more general features using noise distribution and boundary artifacts around the tampered area. ComNet [15] customizes the approximation of JPEG compression operations to improve performance on JPEG-compressed images. DCU-Net [12] effectively extracts tampered regions by leveraging double-channel encoding of deep features, utilizing dilated convolutions, and merging in the decoder. TransU2-Net [2] is a novel transformer-based architecture designed to enhance object-level forgery detection in images. The challenge in splicing localization lies in enhancing the robustness of the adopted methods against various post-processing operations, such as compression and blurring [1].

III. Method

As shown in Figure 1, our approach consists of two key stages: multi-view feature extraction and local feature and global feature learning. Our method effectively addresses the task of detecting video splicing forgery by employing the joint use of multi-view feature extraction and local and global feature learning, thereby enhancing the accuracy and robustness of detection.

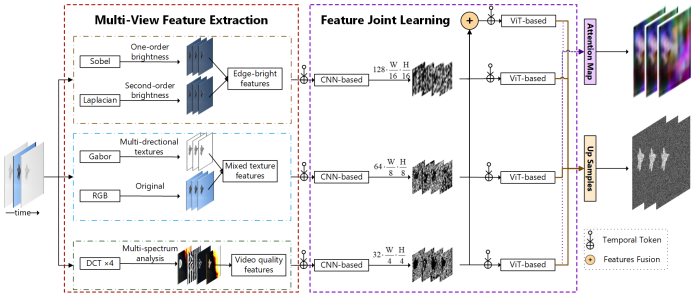

Fig. 1. An overview of the multi-view inconsistency analysis network designed for video splicing localization. We extract features from various perspectives, including edge-bright, texture, and video quality, to analyze the inconsistency between the spliced object video and the source video. Additionally, we employ modules based on CNN and ViT to enable correlation learning from local to global scales, facilitating the handling of these inconsistency features.

A. Edge and Brightness Feature Extraction

In object-level video splicing detection, extracting brightness variation features plays an important role, as lighting is one of the important factors affecting video authenticity. The lighting conditions of different objects or scenes may be different, and the forged object may have inconsistent shadows and highlights under different lighting conditions. Therefore, the extraction of brightness features can be used to detect the lighting consistency between the forged object and the background.

Firstly, the Sobel filter finds brightness variations in an image by calculating the first-order discrete derivative of the image grayscale function, especially in edge regions. This helps to highlight parts of the image with brightness and lighting variations. Subsequently, the Laplacian filter more accurately locates brightness variations and edges by calculating the second-order gradient of the image, while also enhancing detailed information in the image, including texture and subtle brightness variations. Specifically, the mathematical formulas of the Sobel and Laplacian filters are as follows:

$$\begin{aligned} Sobel_x(i,j) = &-2f(i-1,j-1) \\ &+0f(i,j-1)+2f(i+1,j-1) \\ &-2f(i-1,j)+0f(i,j)+2f(i+1,j) \\ &-2f(i-1,j+1)+0f(i,j+1) \\ &+2f(i+1,j+1) \end{aligned} \tag{1}$$

$$\begin{aligned} Sobel_y(i,j) = &-2f(i-1,j-1)-2f(i,j-1) \\ &-2f(i+1,j-1)+0f(i-1,j) \\ &+0f(i,j)+0f(i+1,j) \\ &+2f(i-1,j+1) \\ &+2f(i,j+1)+2f(i+1,j+1) \end{aligned} \tag{2}$$

$$Laplacian(f) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{3}$$

Here, $f(i,j)$ denotes the grayscale value of pixel (i,j) in the image, while $Sobel_x(i,j)$ and $Sobel_y(i,j)$ respectively represent the responses of the Sobel filter in the horizontal and vertical directions. $Laplacian(f)$ corresponds to the Laplacian response of the image f, and $\frac{\partial^2 f}{\partial x^2}$ and $\frac{\partial^2 f}{\partial y^2}$ signify the second-order derivatives of the image in the horizontal and vertical directions. In practical applications, it is common to apply Sobel and Laplacian filters to each channel of the image (e.g., RGB) and then combine their results to comprehensively consider variations in brightness across different color channels. The utilization of these filters not only aids in detecting splicing edges but also facilitates the extraction of brightness features, particularly in addressing inconsistencies in lighting when detecting the presence of forged objects against the background.

B. Texture Feature Extraction Module

We employ CNN convolutional kernels to implement Gabor filters for the extraction of texture features. Gabor filters efficiently capture various texture information within images through multi-scale and multi-directional analysis. They exhibit excellent sensitivity to texture features of different frequencies, orientations, and polarities, making them suitable for a wide range of texture analysis tasks. The mathematical expressions for Gabor filters are as follows:

$$G(x,y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right)\cos\left(2\pi\frac{x'}{\lambda} + \phi\right) \tag{4}$$

$$G(x,y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right)\sin\left(2\pi\frac{x'}{\lambda} + \phi\right) \tag{5}$$

Here, x and y represent the spatial coordinates of the image, x' and y' represent the coordinates after rotation and scale transformation, σ controls the bandwidth of the filter, λ determines the center frequency of the filter, γ is an attenuation factor, and ϕ denotes the phase offset. After applying Gabor filters, we obtain texture responses of the image at different scales and orientations. These texture response images can be used to construct a representation of texture features, which is instrumental in detecting inconsistencies and disguises between forged objects and the background. The extraction of texture features and the application of Gabor filters play a crucial role in enhancing the performance of video object splicing detection.

C. Inconsistency in Quality Features Extraction Module

Frequency domain information extraction plays a crucial role in the analysis of video quality features. Discrete Cosine Transform (DCT) is a commonly used frequency domain transformation method employed to convert video signals from the time domain to the frequency domain. Through DCT, videos can be decomposed into different frequency components, and the magnitude and phase information of these components can be used to assess video quality. Specifically, higher magnitudes often indicate stronger signals, while lower magnitudes may suggest signal loss or noise interference. By analyzing these magnitudes and phase information, we can evaluate quality metrics such as video sharpness and distortion degree, thus enhancing the accuracy and robustness of splice forgery detection and ensuring the credibility and integrity of video content. The mathematical expression of DCT is as follows:

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} S(x,y) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right) \tag{6}$$

In this simplified formula, $F(u,v)$ represents coefficients in the frequency domain, $S(x,y)$ represents pixel values in the block, u and v are frequency domain coordinates, and N is the block size. In our PyTorch-based frequency filter model, we employ four different block sizes to handle image information in different frequency ranges. These filters are used to process: (1) low-frequency image features, typically including the overall structure and larger features of the image. (2) mid-frequency image features, encompassing image features between low and high frequencies. (3) high-frequency image features, capturing image details and texture information. (4) the entire spectral range, covering low, mid, and high frequencies, for frequency domain transformation of the entire image. Finally, the outputs of these four filters are concatenated to form the final output.

D. Multi-View Feature Joint Learning

Multi-view feature joint learning aims to generate pixel-level predictions that closely resemble the ground truth of forged regions. To achieve this goal, multi-view feature joint learning combines various types of features, including edge brightness features, texture features, and video quality information features. During the feature fusion stage, we leverage the strengths of both CNN and ViT models. By using MaxPooling in conjunction with a CNN-based ResNet module and InterlacedFormer, our objective is to extract both local and global features. We employ MSE loss to efficiently and accurately locate forged regions, thereby enhancing the performance and reliability of video forgery detection.

We employ a joint learning approach with CNN and ViT, capitalizing on their respective strengths. CNN models excel at extracting local features, particularly in capturing details such as edges and textures within images. Consequently, we design a CNN-based ResNet specifically for extracting local features, replacing the self-attention mechanism with MaxPooling and CNN. This approach aids in better capturing local details of forged regions, enhancing sensitivity to edges and textures, and improving the ability to pinpoint forged regions. Meanwhile, ViT is a powerful model for extracting global features and excels in semantic understanding of the entire image. We use

the InterlacedFormer module based on ViT to extract global features, facilitating a better understanding of the overall content and background information of the image. By integrating global features, the model gains a better understanding of the relationship between forged objects and their surroundings, thereby increasing the accuracy of forged region localization.

IV. Experiments

A. Experimental Setup

1) Dataset: To evaluate the performance of object-level video splicing detection, we conducted experiments using the Video Splicing (VS) dataset[16]. The VS dataset is specifically designed for video splicing detection and consists of a training set with 795 forged videos and a test set with 30 carefully crafted forged videos. Each forged video is paired with 30 genuine videos. This dataset provides diverse and challenging examples, allowing us to comprehensively assess the robustness and accuracy of our method in detecting object additions.

2) Evaluation Metrics: We employed multiple evaluation metrics to measure the performance of our method, including mean Intersection over Union (mIoU), Area Under the Curve (AUC), F1 score, and pixel-level accuracy. These metrics quantitatively assess the performance of the method in terms of overlap with ground truth data, similarity to ground truth data, accuracy, completeness, and pixel-level precision, among other aspects. Higher values of mIoU, AUC, F1 score, and pixel-level accuracy indicate superior performance. Given the lack of open-source video splicing detection methods, we compared our approach with the existing UVL-Net as well as several leading video semantic segmentation methods, including PoolFormer[9], MetaFormer[9], and HRFormer[11]. We retrained these baseline models on the VS dataset.

3) Implementation Details: We utilized a GPU with 24GB of memory for model training. Each video was treated as an input sequence containing 4 frames, and we employed a batch size of 10 with a learning rate of 1e-4 for training. To enhance the diversity of the training data, we employed various common data augmentation techniques. During the model training process, we selected the Mean Squared Error (MSE) as the loss function and used the Adam optimizer for optimization. The advantage of using MSE loss is that it directly compares the pixel-level results generated by the model with the ground truth, making the training process more targeted. This helps the model gradually approach the true distribution of forged regions, thereby improving the performance and reliability of video splicing detection. This training approach aids the model in better understanding and capturing the features of forged regions, enabling it to detect video forgery more accurately.

B. Experimental Results

C. Comparative Experiments

Table I presents comparative experiments between our method and existing video splicing detection methods, as well as methods based on semantic segmentation. Due to the enhancement of multi-view inconsistency features in our approach, our method significantly outperforms semantic segmentation-based methods like PoolFormer, MetaFormer, and HRFormer. UVL-Net aims to propose a universal framework for video tampering localization but focuses more on detecting forged traces. In contrast, our method pays more attention to the differences between two spliced genuine videos, aiming to extract more generalized inconsistency features, thus performing better in terms of performance.

Table I: Comparison Experiments on the VS Dataset with Baseline Methods. Here, E, T, F represent the application of edge-brightness, texture, and frequency domain features, while N represents no application of any inconsistency trace extraction module.

Methods	None mIoU/F1	Compression mIoU/F1	Detail mIoU/F1	Gaussian mIoU/F1	Blur mIoU/F1	Median mIoU/F1	Filp mIoU/F1
PoolFormer	0.08/0.14	0.09/0.14	0.07/ 0.13	0.08/0.14	0.09/0.14	0.07/0.13	0.08/0.14
MetaFormer	0.10/0.14	0.08/0.14	0.08/0.12	0.10/0.14	0.08/0.14	0.08/0.12	0.10/0.14
HRFormer	0.40/0.48	0.31/0.38	0.36/0.45	0.28/0.37	0.27/0.35	0.38/0.46	0.46/0.55
UVL-Net	0.46/0.57	0.49/0.62	0.46/0.58	0.34/0.44	0.41/0.51	0.68/0.77	0.55/0.67
Ours+N	0.50/0.61	0.39/0.51	0.52/0.63	0.26/0.36	0.38/0.44	0.68/0.78	0.53/0.64
Ours+T	0.57/0.68	0.50/0.63	0.51/0.63	0.30/0.43	0.74/0.80	0.80/0.85	0.59/0.70
Ours+E	0.55/0.67	0.51/0.64	0.53/0.64	0.33/0.45	0.60/0.72	0.74/0.83	0.58/0.70
Ours+F	0.55/0.67	0.46/0.58	0.49/0.63	0.31/0.42	0.71/0.77	0.78/0.86	0.58/0.69
Ours+E+T+F	0.61/0.73	0.65/0.78	0.63/0.76	0.32/0.49	0.78/0.87	0.82/0.90	0.57/ 0.68

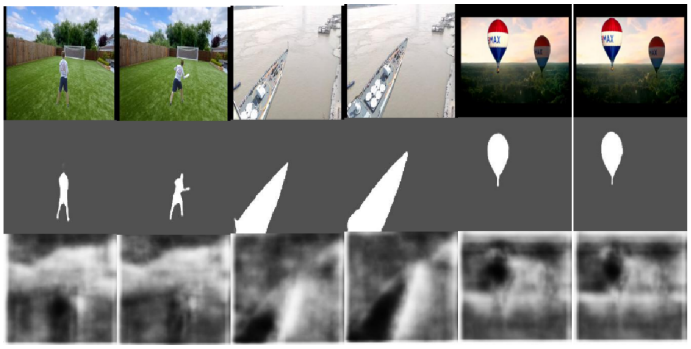

Fig. 2. Visualization results with baseline methods.

D. Robustness Evaluation

Table I also displays the results of robustness evaluations against common video processing techniques. Considering that videos uploaded to the internet often undergo various treatments like compression and enhancement, robustness evaluation holds significant importance. We assessed the performance under various video processing operations. Experimental results demonstrate that our method exhibits robustness across different processing scenarios, outperforming the baseline methods significantly. However, for Gaussian blur and blur operations, there is a more noticeable performance decline, indicating that these noise reduction operations might reduce the inconsistency features between the two spliced videos, especially the impact of physical noise.

E. Visualization Results Analysis and Discussion

Figure 2 highlight the exceptional performance of our approach in detecting video object-level splicing forgery. This success can be attributed to the integration of multi-view learning and the utilization of both local and global features. Multi-view learning allows us to accurately capture subtle differences in forged regions, particularly in fine-grained edge

details. Simultaneously, the combination of local and global feature information enables a more comprehensive understanding of the nature of the forgery, enhancing precise localization across the entire region. These factors collectively contribute to higher accuracy and robustness in video forgery detection, ultimately safeguarding the authenticity and integrity of multimedia content.

V. Conclusion

In this study, we have proposed a novel approach for detecting object-level video splicing forgery. Through the joint learning of multi-view features, we have leveraged a variety of feature types, including edge brightness, texture, and video quality, as well as the strengths of CNN and ViT models. This approach effectively addresses the challenges of video splicing forgery detection, enhancing accuracy and robustness in detection. Our experimental results demonstrate significant performance advantages on datasets related to video object splicing forgery, providing strong support for preserving the authenticity and integrity of multimedia content. In the future, we will continue to refine our method to adapt to evolving video forgery techniques, ensuring the trustworthiness and reliability of video content.

References

[1] C. Dong, X. Chen, R. Hu, J. Cao, and X. Li, "Mvss-net: Multi-view multi-scale supervised networks for image manipulation detection,"IEEE TPAMI, vol. 45, no. 3, pp. 3539–3553, 2023.

[2] C. Yan, S. Li, and H. Li, "Transu2-net: A hybrid transformer architecture for image splicing forgery detection,"IEEE Access, vol. 11, pp. 33313–33323, 2023.

[3] Y. Liu, X. Zhu, X. Zhao, and Y. Cao, "Adversarial learning for constrained image splicing detection and localization based on atrous convolution,"IEEE Transactions on Information Forensics and Security, vol. 14, no. 10, pp. 2551–2566, 2019.

[4] X. B. andZhipeng Zhang andBin Xiao, "Reality transform adversarial generators for image splicing forgery detection and localization," inICCV, (Montreal, QC, Canada), pp. 14274–14283, 2021.

[5] X. Shi, P. Li, H. Wu, Q. Chen, and H. Zhu, "A lightweight image splicing tampering localization method based on mobilenetv2 and SRM,"IET Image Process., vol. 17, no. 6, pp. 1883–1892, 2023.

[6] M. K. Johnson and H. Farid, "Exposing digital forgeries by detecting inconsistencies in lighting," inProceedings of the 7th workshop on Multimedia & Security, MM&Sec 2005, New York, NY, USA, August 1-2, 2005, 2006 (A. M. Eskicioglu, J. J. Fridrich, and J. Dittmann, eds.), pp. 1–10, ACM, 2005.

[7] Y. Niu, B. Tondi, Y. Zhao, R. Ni, and M. Barni, "Image splicing detection, localization and attribution via JPEG primary quantization matrix estimation and clustering,"IEEE Trans. Inf. Forensics Secur., vol. 16, pp. 5397–5412, 2021.

[8] Y. Zhang, G. Zhu, L. Wu, S. Kwong, H. Zhang, and Y. Zhou, "Multi-task se-network for image splicing localization,"IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 7, pp. 4828–4840, 2022.

[9] Q. Diao, Y. Jiang, B. Wen, J. Sun, and Z. Yuan, "Metaformer: A unified meta framework for fine-grained recognition," inCVPR, (New Orleans, Louisiana, USA), IEEE, 2022.

[10] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," inCVPR, (Long Beach, CA, USA), pp. 5693–5703, IEEE, 2019.

[11] Y. Yuan, R. Fu, L. Huang, W. Lin, C. Zhang, X. Chen, and J. Wang, "Hrformer: High-resolution vision transformer for dense predict," inNeurIPS, (Virtual), pp. 7281–7293, 2021.

[12] H. Ding, L. Chen, Q. Tao, Z. Fu, L. Dong, and X. Cui, "Dcu-net: a dual-channel u-shaped network for image splicing forgery detection,"Neural Comput. Appl., vol. 35, no. 7, pp. 5015–5031, 2023.

[13] B. C. Hadwiger and C. Riess, "Deep metric color embeddings for splicing localization in severely degraded images,"IEEE Transactions on Information Forensics and Security, vol. 17, pp. 2614–2627, 2022.

[14] B. C. Hadwiger and C. Riess, "Deep metric color embeddings for splicing localization in severely degraded images,"IEEE Transactions on Information Forensics and Security, vol. 17, pp. 2614–2627, 2022.

[15] Y. Rao and J. Ni, "Self-supervised domain adaptation for forgery localization of JPEG compressed images," inICCV, (Montreal, QC, Canada), pp. 15014–15023, 2021.

[16] P. Pei, X. Zhao, J. Li, Y. Cao, and X. Lai, "Vision transformer based video hashing retrieval for tracing the source of fake videos,"Security and Communication Networks, vol. 2023, 2023.