

Robust Watermarking for Video Forgery Detection with Improved Imperceptibility and Robustness

Yangming Zhou
School of Computer Science
Fudan University, Shanghai, China
ymzhou21@m.fudan.edu.cn

Qichao Ying
School of Computer Science
Fudan University, Shanghai, China
shinydotcom@163.com

Xiangyu Zhang
School of Computer Science
Fudan University, Shanghai, China
xyzhang20@fudan.edu.cn

Zhenxing Qian*
School of Computer Science
Fudan University, Shanghai, China
zxqian@fudan.edu.cn

Sheng Li
School of Computer Science
Fudan University, Shanghai, China
lisheng@fudan.edu.cn

Xinpeng Zhang
School of Computer Science
Fudan University, Shanghai, China
zhangxinpeng@fudan.edu.cn

Abstract—Videos are prone to tampering attacks that alter the meaning and deceive the audience. Previous video forgery detection schemes find tiny clues to locate the tampered areas. However, attackers can successfully evade supervision by destroying such clues using video compression or blurring. This paper proposes a video watermarking network for tampering localization. We jointly train a 3D-UNet-based watermark embedding network and a decoder that predicts the tampering mask. The perturbation made by watermark embedding is close to imperceptible. Considering that there is no off-the-shelf differentiable video codec simulator, we propose to mimic video compression by ensembling simulation results of other typical attacks, e.g., JPEG compression and blurring, as an approximation. Experimental results demonstrate that our method generates watermarked videos with good imperceptibility and robustly and accurately locates tampered areas within the attacked version.

Index Terms—Video Technology, Forgery Detection, Multimedia Watermarking, Forensics, Robustness

I. INTRODUCTION

With the maturity of various video processing and compression technologies, the Online Social Networks (OSNs) are crowded with daily-shared videos for entertainment and reporting. However, the popularization of video technology also breeds malicious even illegal activities such as the generation of fake news caused by video clipping or tampering. Manual video inspection and anomaly detection are time-consuming, labor-intensive and usually with high latency. Therefore, algorithm-based automatic video tampering detection has gained extensive research interest.

Traditional video forgery detection methods [1]–[3] include stitching detection, copy-paste detection, image restoration detection, etc. For example, Subramanyam et al. [2] finds that HoG features are robust against various signal processing manipulations. Aloraini [3] performs sequential analysis by modeling video sequences as stochastic processes. Changes in the parameters of these processes indicate a video forgery. In

recent years, most of the video forgery detection work focuses on the detection of forged faces [4]–[7]. Li et al. [5] considers both temporal and spatial information, and uses 3DCNN to discriminate forged videos. Zhang et al. [6] mined some traces in the frequency spectrum and detected the images generated by GAN in the frequency domain. There are also several image manipulation detection schemes. Wu et al. [8] proposed an end-to-end deep neural network structure, ManTraNet, which first learns image manipulation traces through self-supervision, then extracts local anomalies through Z-score, and detects and locates multiple tampering by judging anomalies. Dong et al. [9] proposed MVSS-Net to augment the differences between the tampered and untampered regions at the boundary, and noise inconsistency and edge supervision are monitored to unveil image manipulation. However, universal video tampering detection is still a hard issue. One reason is that the above methods either focus on a typical distribution of videos, such as facial clips, or cannot generalize well on compressed videos. Another reason is that video post-processing attacks represented by MPEG compression are complicated, and the ways of video tampering and post-processing are indefinite. Thus, it is extremely difficult to find a universal clue for all kinds of tampered videos.

Active forensics based on watermarking is an important alternative for manipulation detection. The goal is to hide a tailored clue into the targeted videos for protection, and once the embedded signal is destroyed by tampering, the recipient can identify the modified areas. Meanwhile, the signal must survive video post-processing attacks to ensure robustness. In the image domain, Imuge [10] presents a robust watermarking scheme that protects images from being tampered with. After the recipient gets the tampered protected image, he can conduct accurate tamper localization and image recovery. Khachaturov [11] proposes a watermarking-like adversarial method, called Markpainting that prevents images from being inpainted. Besides, many video watermarking methods [12]–[14] have been proposed for covert data transmission. But they only focus on hiding as much information as possible into a

This work is supported by National Natural Science Foundation of China under Grant U20B2051, U1936214. * Corresponding author: Zhenxing Qian (zxqian@fudan.edu.cn).

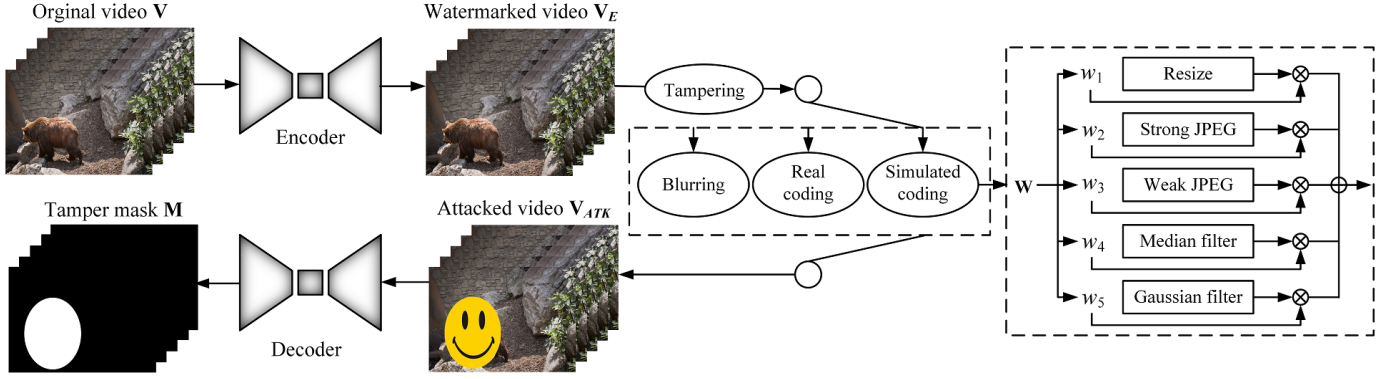


Fig. 1: **Sketch of the pipeline of RWVMD.** A 3D-UNet based encoder is used to embed the watermark information into the video. Video processing attacks in social media are simulated by the attack layer. Finally, the decoder predicts the tampering mask from the watermarked video.

targeted video, and in comparison, there is little work that focuses on video watermarking against tampering attacks.

We propose an end-to-end Robust Watermarking network for Video Forgery Detection (RWVFD). We jointly train a 3D-UNet-based encoder for imperceptible watermark embedding and a decoder for tampering localization. We design an attack simulation module that combines simulated video encoding, real video encoding, and other obfuscation attacks to improve the robustness of watermarking. Considering that there is no off-the-shelf differentiable video codec simulator, we propose to mimic video compression by ensembling simulation results of other typical attacks, e.g., JPEG compression and blurring, as an approximation. The experimental results on the dataset YouTube-VOS demonstrate that our watermarking scheme simultaneously achieves satisfactory imperceptibility and robustness, and the accuracy of tampering localization is much higher compared to existing passive forensics methods.

Our contributions are mainly as follows: 1) We use deep networks for video watermarking against tampering; 2) We propose a tailored attacking layer for enhanced robustness against typical video post-processing attacks; 3) Our method can achieve higher accuracy in locating tampered regions and is robust against multiple kinds of post-processing attacks.

II. METHOD

A. Approach Overview

Fig. 1 illustrates the network design of the proposed method. Our watermarking scheme follows the traditional data hiding pipeline, which mainly contains three phases, namely, watermark embedding, attacking simulation and forgery detection. We use two independent three-dimensional U-shaped architecture [15] to hide auto-generated watermark into an original video, and localize the tampered areas on receiving the attacked version, respectively. In detail, given an original video \mathbf{V} , we transform the original video \mathbf{V} into the watermarked video \mathbf{V}_E using the encoder. The attacking layer performs both tampering and benign video post-processing attacks on \mathbf{V}_E to generate \mathbf{V}_{ATK} . In this stage, the hidden information

might be globally or locally destroyed. On the recipient's side, the decoder produces the predicted tampering mask $\hat{\mathbf{M}}$ to see which parts of the video are tampered with. The architectures of the encoder and decoder are shown in Fig. 2.

The objective functions include the embedding loss \mathcal{L}_{emb} and the localization loss \mathcal{L}_{loc} . We respectively employ the Mean Squared Error (MSE) loss and the Binary Cross-Entropy (BCE) loss as their implementation.

$$\mathcal{L}_{emb} = \|\mathbf{V} - \mathbf{V}_E\|_2, \quad (1)$$

$$\mathcal{L}_{loc} = -(\mathbf{M} \log \hat{\mathbf{M}} + (1 - \mathbf{M}) \log(1 - \hat{\mathbf{M}})), \quad (2)$$

where \mathbf{M} represents the ground-truth tampering mask. The total loss is listed in Eq. (3), where α is a hyper-parameter.

$$\mathcal{L} = \mathcal{L}_{emb} + \alpha \cdot \mathcal{L}_{loc}. \quad (3)$$

B. Attack Simulation

Attack simulation plays a critical role in robustness training. To simulate the video redistribution stage, we first perform tampering on \mathbf{V}_E , and afterwards, common video post-processing attacks are performed to generate the attacked video \mathbf{V}_{ATK} . To begin with, we select some critical areas within the original video \mathbf{X} to form \mathbf{M} . In our scheme, we use binarized segmentation masks as \mathbf{M} . We then tamper the watermarked video \mathbf{V}_E as \mathbf{V}_{imp} by replacing contents within \mathbf{M} with that within \mathbf{R} , a randomly-selected video frame from an irrelevant video. Next, we implement typical video post-processing attacks using $IP(\cdot)$ to simulate that \mathbf{V}_{imp} must be lossily processed during transmission and storage, where the attacker wants to conceal the tampering behavior. In sum, the attacked video \mathbf{V}_{ATK} is generated according to Eq. (4).

$$\mathbf{V}_{ATK} = IP(\mathbf{X} \cdot (1 - \mathbf{M}) + \mathbf{R} \cdot \mathbf{M}). \quad (4)$$

The video post-processing attacks implemented in our scheme, i.e., $IP(\cdot)$, include the following attacks. (1) **Median filtering**, where each output pixel is computed as the median value of the input pixels under a 5×5 window. (2) **Gaussian blurring**, which convolves the image with a Gaussian

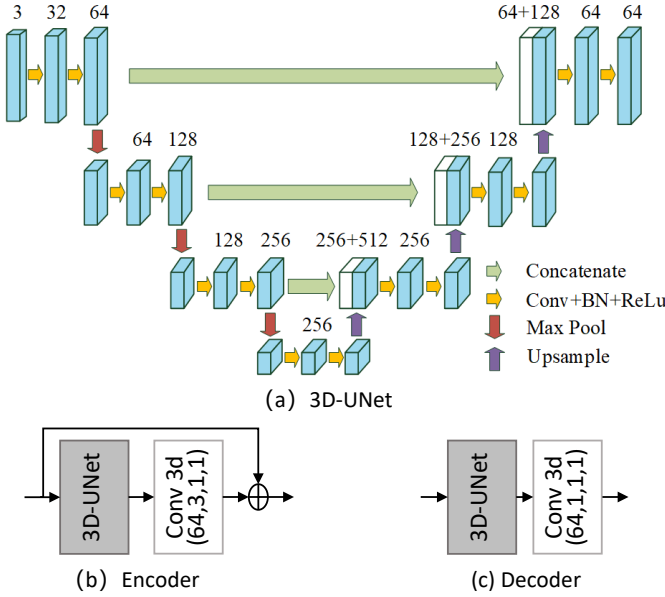


Fig. 2: Architectures of the encoder and decoder, which are based on the 3D-Unet [15].

5×5 -sized kernel. (3) **Rescaling**, which randomly scales the video frames up or down and back to the original size. (4) **Image lossy compression**, which lossily compresses the video frame by frame. (5) **Video compression and decompression (codec)**, which lossily compresses the video using both spatial and temporal characteristics of the video. These attacks might introduce missing or distorted higher-frequency details in the ultimate videos compared to the original ones. For example, JPEG compression is a widely used compression that includes DCT transformation, quantization on the coefficients and lossless encoding, in which the quantization process discards many details for file size shrinkage, and can cause chess-board artifacts. Video codec attack is similar yet more complicated than the other attacks. The decompressed video might have lower quality than the original video because there is insufficient information to accurately reconstruct the original video. Typical video codec standards are MPEG, H.264, HEVC (H.265), etc.

The issue mainly lies in how to effectively simulate video codec attack, since the leading three kinds of attacks can be easily implemented by simple differentiable methods [16]. As for image compression, many effective JPEG simulators have been proposed in the past literature, such as Diff-JPEG [17], MBRS [18] and HiDDeN [16]. In contrast, there is no off-the-shelf differentiable video codec simulator so far. The reason is that many internal steps, such as motion estimation and compensation, are hard to be differentiated. However, we find that JPEG and H.264 compression both share the process of DCT-based coefficient quantization, suggesting that losses introduced by video codec might share some common characteristics with those made by the rest of the attacks. Therefore, we are motivated to design a video

codec simulator by ensembling the attacked videos generated by blurring, scaling, JPEG compression, etc. First, we use a real video codec to compress \mathbf{V}_{ATK} according to different coding standards and Constant Rate Factors (CRFs), i.e., $\mathbf{V}_{ATK}^{codec} = \text{Real_Codec}(\mathbf{V}_{tmp}, \text{CRF})$, where *Real_Codec* is the H.264 codec, and $\text{CRF} = \{17, 23, 29\}$. Note that CRF is a tunable content-specific offset to the frame’s quantization parameter, with lower values indicating less compression and higher quality. We select the H.264 codec for its overwhelming popularity. The attacked video by the video codec attack simulation is generated according to Eq. (5).

$$\begin{aligned} \hat{\mathbf{V}}_{ATK}^{codec} = & \alpha_0 \cdot \text{Resize}(\mathbf{V}_{tmp}) + \alpha_1 \cdot \text{Med}(\mathbf{V}_{tmp}) \\ & + \alpha_2 \cdot \text{Gauss}(\mathbf{V}_{tmp}) + \alpha_3 \cdot \text{JPEG}(\mathbf{V}_{tmp}, \text{QF}_W) \\ & + \alpha_4 \cdot \text{JPEG}(\mathbf{V}_{tmp}, \text{QF}_S), \end{aligned} \quad (5)$$

where $\text{QF}_S \in \{40, 50, 60\}$ and $\text{QF}_W \in \{70, 80, 90\}$, respectively represent strong and weak JPEG compression attack. $\alpha = \{\alpha_0, \dots, \alpha_4\}$ are learnable parameters that weight the generated results of the five attacks to let \mathbf{V}_{ATK} more close to \mathbf{V}_{GT} . For \mathbf{V}_{ATK}^{codec} with totally three different combinations of codec and CRF, we employ three different sets of parameters α . On training the simulator, given a \mathbf{V}_{tmp} , we randomly sample a codec and CRF and generate \mathbf{V}_{ATK}^{codec} , we let the simulator update the corresponding α to make closer $\hat{\mathbf{V}}_{ATK}^{codec}$ and \mathbf{V}_{ATK}^{codec} . We use the MSE loss \mathcal{L}_{simul} as the supervision on the simulator.

$$\mathcal{L}_{simul} = \|\hat{\mathbf{V}}_{ATK}^{codec} - \mathbf{V}_{ATK}^{codec}\|_2. \quad (6)$$

Then, in the training phase, the benign attack $IP(\cdot)$ is evenly and iteratively switched within the range of the above attacks. Empirically, we find that robustness against video codec simulation is much more important than robustness against the rest of the attacks. Therefore, we further propose two strategies in our adversarial training mechanism. First, we observe that the blurring results produced by median filtering and Gaussian blurring are close. We again use the ensembling strategy to linearly combine $\text{Gauss}(\mathbf{V}_{tmp})$ with $\text{Med}(\mathbf{V}_{tmp})$ as the mixed filtering operation. The benefit is that we moderately lower the importance of robustness against blurring, and further introduce randomness within the model. Second, in some cases, we directly let $\hat{\mathbf{V}}_{ATK}^{codec} = \mathbf{V}_{ATK}^{codec}$, and address the non-differentiable problem by using the noise-addition strategy proposed by Zhang et al. [19]. That is, the residual $e = \mathbf{V}_{ATK}^{codec} - \mathbf{V}_{tmp}$ is detached and directly added onto \mathbf{V}_{tmp} . Therefore, \mathbf{V}_{ATK} in different training iteration *iter* is as follows.

$$\mathbf{V}_{ATK} = \begin{cases} \text{Resize}(\mathbf{V}_{tmp}), & \text{iter} \% 4 = 0 \\ \beta \cdot \text{Med}(\mathbf{V}_{tmp}) + \gamma \cdot \text{Gauss}(\mathbf{V}_{tmp}), & \text{iter} \% 4 = 1 \\ \hat{\mathbf{V}}_{ATK}^{codec}, & \text{iter} \% 4 = 2 \\ \text{stop_grad}(e) + \mathbf{V}_{tmp}, & \text{iter} \% 4 = 3 \end{cases}, \quad (7)$$

where $\beta, \gamma \in [0, 1], \beta + \gamma = 1$. The reason for using hybrid real-world and simulated video codec is to reduce temporal complexity, where real video codecs are much slower than the proposed video simulator.

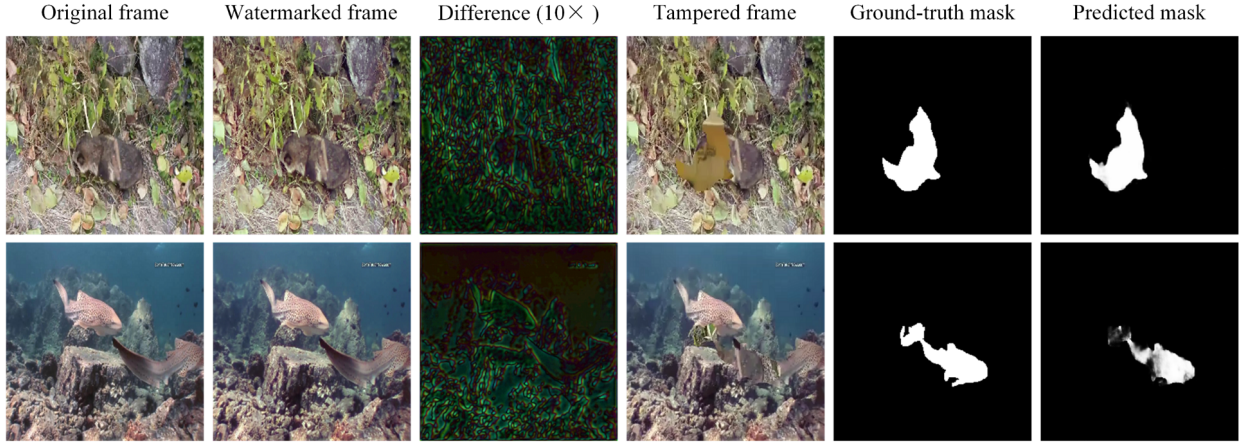


Fig. 3: **Performance against tampering with FFMPEG compression (CRF = 17). The average PSNR and SSIM on YouTube-VOS dataset are 37.78 dB and 0.987, respectively.** The difference between the original video and the watermarked video cannot be perceived by the human eye, and the tampered area can still be accurately detected in the compressed video.

TABLE I: **Results of the learned parameters of the codec simulator under different attacks.** Compression-based attacks are more preferred than the rest kinds of attacks. The highest and lowest value are respectively marked red and blue.

CRF	Resize	Strong JPEG	Weak JPEG	Median Blurring	Gaussian Blurring
17	-0.7575	1.0723	1.4917	-1.3091	-0.8802
23	-0.7568	1.0685	1.3155	-0.8185	-1.1666
29	-0.1529	1.1229	0.5848	-0.2182	-0.7920

III. EXPERIMENTS

A. Experimental Setup

We empirically set the hyper-parameter as $\alpha = 0.9$. The batch size is set as four. We use Adam optimizer [21] with the default parameters. The learning rate is 1×10^{-4} with manual decay. The default frame rate of the video is set to 25 frames per second. We binarize the prediction mask by setting the threshold Th as 0.5.

We use two popular object segmentation datasets, namely, Davis [22] and YouTube-VOS [23] in the experiment. During training, the original videos V are prepared by selecting the data from the whole Davis dataset and YouTube-VOS train set. We use the YouTube-VOS test set to test our model. The tampering masks are the annotation images corresponding to the video frames in the object segmentation datasets.

We compare RWVFD with two passive methods for tamper detection, which detect universal manipulations or deepfake, namely, MVSS-Net [9] and Xception [20]. We employ the peak signal-to-noise ratio (PSNR) and the Structural Similarity [24] to evaluate the image quality. The value of SSIM ranges from zero to one. A higher structure similarity is indicated by a high SSIM closer to one. We employ the

Precision, Recall, and F1 score to measure the accuracy of tamper localization. Higher F1 value indicates more accurate result.

B. Imperceptibility of watermark embedding

In Fig. 3, we showcase the first frames from three randomly selected test videos from YouTube-VOS dataset. We can observe that the differences before and after watermark embedding are almost imperceptible, and the overall quality of the watermarked frames is satisfactory. The watermark information is distributed in the whole set of video frames, mainly hidden in the higher frequencies. The embedded watermark is robust to video processing attacks and can be used for tampering localization. Instead of finding a ubiquitously existing trace to unveil video modification behavior, in our scheme, tampering will result in local pattern inconsistencies, allowing the network to efficiently detect and locate the tampered regions. We have conducted the embedding experiments on the entire test dataset of YouTube-VOS and the average PSNR and SSIM are 37.78 dB and 0.987, respectively.

C. Accuracy and robustness of tampering localization

In Table II, we clarify the robustness of RWVFD with the presence of different video processing attacks. It can be seen from the results that RWVFD has strong robustness to common video processing behaviors. Even if there is a high-intensity H.264 video compression attack, the performance will not be significantly degraded. It proves the effectiveness of our codec simulator for improving watermarking robustness. From Fig. 3, RWVFD can accurately detect forged regions even under compression attacks. In addition, RWVFD is also robust to typical temporal attacks such as frame deletion and frame rate transformation.

Table I shows the learned parameters of the video codec simulator. On simulating a video codec, the simulator prefers the simulated images of JPEG compression, and those of blurring attacks will be suppressed, indicating that the

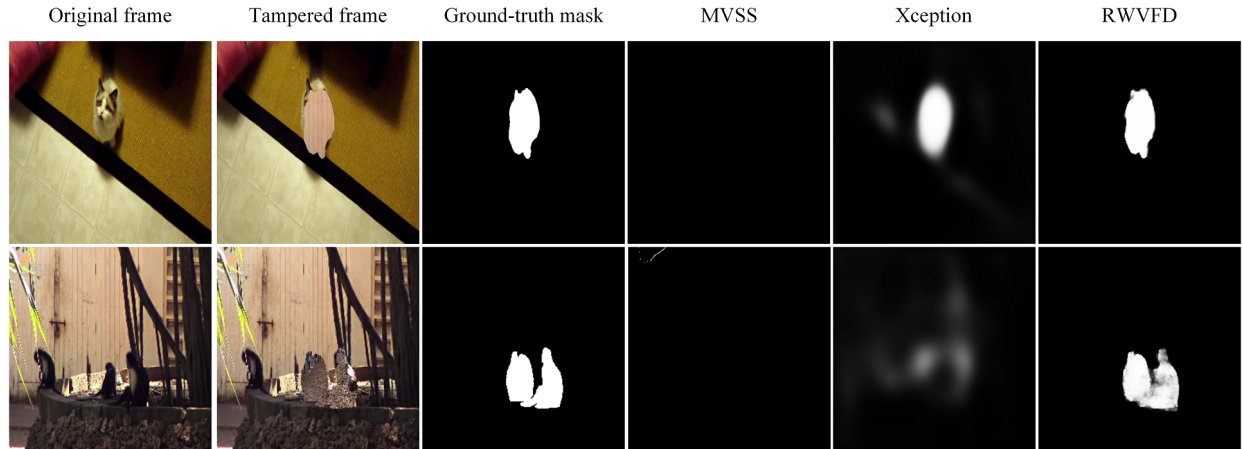


Fig. 4: Comparison of tamper localization under FFMPEG compression (CRF = 17) among RWVFD and several state-of-the-art passive schemes. RWVFD can accurately localize the tampered areas even with the presence of post-processing attack. In contrast, many passive schemes have low detection accuracy and are reported to not have robustness.

TABLE II: Performance comparison for tamper detection among our scheme and the state-of-the-art passive methods. NA: No-attack, GB: Gaussian Blur, MB: Median Blur, VC: Video Compression, FD: Frame Dropping, FRC: Frame Rate conversion, CRF: Constant Rate Factor, DN: Dropping Number, FR: Frame Rate.

Attack		MVSS [9]			Xception [20]			RWVMD			
		Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	
VC	NA	0.05	0.00	0.01	0.64	0.59	0.57	1.00	0.99	0.99	
	GB	0.06	0.00	0.01	0.55	0.21	0.25	0.99	0.95	0.96	
	MB	0.05	0.00	0.01	0.64	0.60	0.58	0.99	0.99	0.99	
	CRF = 17	0.05	0.00	0.01	0.66	0.46	0.49	0.95	0.83	0.87	
	CRF = 23	0.05	0.00	0.01	0.67	0.45	0.49	0.95	0.80	0.84	
	CRF = 29	0.05	0.00	0.01	0.68	0.42	0.46	0.90	0.70	0.75	
	FD	DN = 1	0.05	0.00	0.01	0.64	0.59	0.57	0.93	0.98	0.95
		DN = 2	0.05	0.00	0.01	0.64	0.59	0.57	0.97	0.99	0.97
	FRC	FR = 20	0.05	0.00	0.01	0.67	0.46	0.49	0.96	0.92	0.93
FR = 30		0.05	0.00	0.01	0.66	0.55	0.55	0.87	0.87	0.84	
FR = 35		0.05	0.00	0.01	0.66	0.55	0.55	0.88	0.86	0.84	

video coding distortion shares more characteristics with two-dimensional JPEG compression, such as chess-board artifact.

In comparison, passive forensics methods, e.g., MVSS and Xception, do not perform well on the test sets. We use the pre-trained models provided by the authors of MVSS, and find that the model fails to detect forged regions when there exist video codec attacks. The reason is most likely that video frames after codec attack are out of distribution of natural images that MVSS detects. We train Xception using the same training dataset as RWVFD. We also equip the scheme with the attacking layer proposed by us for adversarial training. However, the accuracy is still not high. Fig. 4 showcases the experimental comparison of tamper localization among RWVFD, MVSS, and Xception.

D. Ablation Study

We explore the influence of 3D-Unet and codec simulator in our scheme. In the first experiment, we changed the 3D-Unet architecture into 2D-Unet as RWVMD without 3D-Unet.

In the second experiment, we train the network without using the codec simulator. For fair comparisons, we separately train the model from scratch until it converges, perform the same video post-processing attacks and the same tampering attack in each test.

We summarize the average results on the test dataset in Table III. The results show that the complete implementation of RWVFD has higher PSNR and F1 scores than the ablated versions. In comparison, RWVMD without 3D-UNET cannot perform forgery detection when we apply video codec attacks. It suggests that considering video frames as independent images and embedding temporarily-inconsistent watermarks is less effective in countering typical video attacks. RWVMD without the proposed codec simulator also performs worse in the overall accuracy under video compression. This shows that applying Zhang et al. [19] alone is not enough, and proves the necessity of applying the codec simulator.

TABLE III: Ablation study of RWVFD using varied partial settings.

Attack		RWVMD w/o 3D-Unet			RWVMD w/o codec simulator			RWVMD		
		(PSNR = 34.95dB, SSIM = 0.958)			(PSNR = 35.82dB, SSIM = 0.987)			(PSNR = 37.78dB, SSIM = 0.987)		
		Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
VC	CRF = 17	0.72	0.94	0.80	0.76	0.86	0.80	0.95	0.83	0.87
	CRF = 23	0.66	0.94	0.75	0.71	0.87	0.77	0.95	0.80	0.84
	CRF = 29	0.49	0.95	0.61	0.44	0.89	0.56	0.90	0.70	0.75
FD	DN = 1	0.98	0.98	0.97	0.44	0.99	0.56	0.93	0.98	0.95
	DN = 2	0.98	0.98	0.97	0.49	0.99	0.61	0.97	0.99	0.97
FRC	FR = 20	0.90	0.96	0.92	0.31	0.97	0.42	0.96	0.92	0.93
	FR = 30	0.90	0.96	0.93	0.21	0.99	0.31	0.87	0.87	0.84
	FR = 35	0.91	0.96	0.92	0.21	0.99	0.31	0.88	0.86	0.84

IV. CONCLUSION

In this paper, we propose a deep learning-based video watermarking method RWVFD for video forgery detection. We encode the original video into a watermarked video, in which the tampering attack area can be accurately located. To improve performance, we propose a video codec simulator along with simulation of other typical attacks to enhance the robustness against common video post-processing attacks. We conduct experiments on a popular video dataset and the results demonstrate the effectiveness of RWVFD in tampering localization.

REFERENCES

- [1] Nitin Arvind Shelke and Singara Singh Kasana, "A comprehensive survey on passive techniques for digital video forgery detection," *Multimedia Tools and Applications*, vol. 80, no. 4, pp. 6247–6310, 2021.
- [2] A Venkata Subramanyam and Sabu Emmanuel, "Video forgery detection using hog features and compression properties," in *2012 IEEE 14th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2012, pp. 89–94.
- [3] Mohammed Aloraini, Mehdi Sharifzadeh, and Dan Schonfeld, "Sequential and patch analyses for object removal video forgery detection and localization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 3, pp. 917–930, 2020.
- [4] Yuezun Li and Siwei Lyu, "Exposing deepfake videos by detecting face warping artifacts," *arXiv preprint arXiv:1811.00656*, 2018.
- [5] Haoliang Li, Peisong He, Shiqi Wang, Anderson Rocha, Xinghao Jiang, and Alex C Kot, "Learning generalized deep feature representation for face anti-spoofing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 10, pp. 2639–2652, 2018.
- [6] Xu Zhang, Svebor Karaman, and Shih-Fu Chang, "Detecting and simulating artifacts in gan fake images," in *2019 IEEE international workshop on information forensics and security (WIFS)*. IEEE, 2019, pp. 1–6.
- [7] Baogen Zhang, Sheng Li, Guorui Feng, Zhenxing Qian, and Xinpeng Zhang, "Patch diffusion: A general module for face manipulation detection," 2022.
- [8] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan, "Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9543–9552.
- [9] Xinru Chen, Chengbo Dong, Jiaqi Ji, Juan Cao, and Xirong Li, "Image manipulation detection by multi-view multi-scale supervision," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14185–14193.
- [10] Qichao Ying, Zhenxing Qian, Hang Zhou, Haisheng Xu, Xinpeng Zhang, and Siyi Li, "From image to image: Immunized image generation," in *Proceedings of the 29th ACM international conference on Multimedia*, 2021, pp. 1–9.
- [11] David Khachaturov, Ilia Shumailov, Yiren Zhao, Nicolas Papernot, and Ross Anderson, "Markpainting: Adversarial machine learning meets inpainting," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5409–5419.
- [12] Qichao Ying, Jingzhi Lin, Zhenxing Qian, Haisheng Xu, and Xinpeng Zhang, "Robust digital watermarking for color images in combined dft and dt-cwt domains," *Mathematical Biosciences and Engineering*, vol. 16, no. 5, pp. 4788–4801, 2019.
- [13] Yifei Wang, Qichao Ying, Zhenxing Qian, Sheng Li, and Xinpeng Zhang, "A dt-cwt-svd based video watermarking resistant to frame rate conversion," *arXiv preprint arXiv:2206.01094*, 2022.
- [14] Md Asikuzzaman, Md Jahangir Alam, and Mark R Pickering, "A blind and robust video watermarking scheme in the dt cwt and svd domain," in *2015 Picture Coding Symposium (PCS)*. IEEE, 2015, pp. 277–281.
- [15] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2016, pp. 424–432.
- [16] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei, "Hidden: Hiding data with deep networks," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 657–672.
- [17] Richard Shin and Dawn Song, "Jpeg-resistant adversarial images," in *NIPS 2017 Workshop on Machine Learning and Computer Security*, 2017, vol. 1.
- [18] Zhaoyang Jia, Han Fang, and Weiming Zhang, "Mbrs: Enhancing robustness of dnn-based watermarking by mini-batch of real and simulated jpeg compression," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 41–49.
- [19] Chaoning Zhang, Adil Karjauv, Philipp Benz, and In So Kweon, "Towards robust deep hiding under non-differentiable distortions for practical blind watermarking," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 5158–5166.
- [20] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner, "Faceforensics++: Learning to detect manipulated facial images," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1–11.
- [21] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [22] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 724–732.
- [23] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang, "Youtube-vos: A large-scale video object segmentation benchmark," *arXiv preprint arXiv:1809.03327*, 2018.
- [24] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.