Robust Video watermarking based on deep neural network and curriculum learning

1st Zehui Ke
School of Software Engineering
South China University of Technology
Guangzhou, China
kezh98@foxmail.com

2nd Hailiang Huang
Guangzhou Easefun Co. Ltd.
Information and Technology
Guangzhou, China
sean@polyv.net

3rd Yingwei Liang
Guangzhou Easefun Co. Ltd.
Information and Technology
Guangzhou, China
alvinliang@polyv.net

4th Yi Ding
Guangzhou Easefun Co. Ltd.
Information and Technology
Guangzhou, China
dingyi@polyv.net

5th Xin Cheng
School of Software Engineering
South China University of Technology
Guangzhou, China
chengxin_19@aliyun.com

6th Qingyao Wu*
School of Software Engineering
South China University of Technology
Guangzhou, China
qyw@scut.edu.cn

***Abstract*—With the rapid development of multimedia and short video, there is a growing concern for video copyright protection. Some work has been proposed to add some copyright or fingerprint information to the video to trace the source of the video when it is stolen and protect video copyright. This paper proposes a video watermarking method based on a deep neural network and curriculum learning for watermarking of sliced videos. The first frame of the segmented video is perturbed by an encoder network, which is invisible and can be distinguished by the decoder network. Our model is trained and tested on an online educational video dataset consisting of 2000 different video clips. Experimental results show that our method can successfully discriminate most watermarked and non-watermarked videos with low visual disturbance, which can be achieved even under a relatively high video compression rate(H.264 video compress with CRF 32).**

***Index Terms*—robust video watermark, deep neural network, copyright protection, curriculum learning**

I. Introduction

With the development of easy access to the Internet and the popularity of social media platforms, digital media such as images, video, and audio account for most of the current Internet traffic. This has been accompanied by a dramatic increase in video copyright infringement cases. Due to the anonymity of the Internet, video copyright infringement is often difficult to trace. Some methods embed information indicating copyright(such as trademarks, words, and other patterns) directly into the video as a visual watermark. However, visible watermarks not only affect the viewing experience but are also easily distorted or even removed by attackers through various disturbances.

For the above reasons, there have been many studies on how to add invisible watermarks to videos. According to the domain of watermark embedding, traditional video watermarking methods can roughly divide into three schemes: spatial domain, transform domain, and compressed domain. In addition, with the development of deep learning, many image watermarking methods based on deep learning have emerged in recent years, mostly based on CNN-based auto-encoder and generative adversarial networks.

This paper aims to introduce a robust video watermarking method based on deep learning and curriculum learning. To ensure the model can perform better in H264 video compression, we incorporate video compression noise losses with different weights at different stages of training, which makes the watermark added by the model more robust in relatively high compression rate scenarios based on H264.

The article's structure is as follows: In Section 2, we will discuss existing video watermarking methods, including traditional and deep learning based methods. In Section 3, we will introduce our method, including model structure, training strategy, and a segmentation-based video watermark embedding and extraction process. In Section 4, we will discuss

the experimental setting and results and analysis of the proposed method. The last section concludes the paper.

II. Related work

A. Traditional methods

According to the domain of watermark embedding, traditional video watermarking methods can be roughly divided into three schemes: spatial domain, transform domain, and compression domain.

The spatial domain embedding watermark [**1**, **2**] is to decode video into frames and embed the watermark into frames sequence, and then encode the embedded video. The Least Significant Bits [**8**, **9**] (LSB) is a representative scheme. Although this solution can take advantage of image watermarking technology and combine the characteristics of video frames to form a video watermarking solution, the embedded video is likely to lose the watermark after being compressed and encoded. Compared with the simple embedding in the spatial domain, The transform domain scheme can design a more robust watermarking algorithm. This type of scheme embeds watermarks in transform domain coefficients in conjunction with the encoding process. Common transforms include discrete Fourier transform [**10**] (DFT), discrete cosine transform [**11**] (DCT), and more complex transforms such as discrete wavelet transform [**12**] (DWT) and dual-tree complex wavelet transform [**13**] (DC-CWT) et al. However, when the video compression rate is high, the problem of losing the watermark still does not solution well.

There are also recent pieces of literature on compressing the domain. Song [**28**] proposed a watermarking method based on the mapping rules between motion vector resolution and watermarking for AVS-encoded video. Qiu et al. [**29**] embed the robust watermark and fragile watermark together in the video during H.264/AVC encoding. The fragile watermark is embedded in the motion vector by changing the components of a set of selected motion vectors. By changing the quantization in the I frame, The AC coefficients of the robust watermark are embedded in the DCT domain. The scheme of compressed domain embedding has lower computational complexity. However, most of these methods [**7**] are format-specific, do not support using alternative encoders for conversion, and have poor resistance to channel interference.

B. Deep learning based methods

Since encoding and decoding tasks are the core of the digital watermarking process, the encoder-decoder deep learning framework is well suited for digital watermarking models. Therefore, the most current deep learning methods in the field of digital watermarking rely on Auto-encoder (AE) architectures combined with convolutional neural networks. ReDMark [**17**] uses two Full Convolutional Neural Networks (FCN) for watermark embedding and extraction. And a differentiable attack layer to simulate different distortions. ReDMark can learn many embedding patterns in different transform domains and can be trained for a specific attack or a range of attacks. Lee et al. [**18**] uses a simple CNN for embedding and extraction without any resolution-dependent layers, which means that images of any resolution can be used as input to the system for watermarking. The model also employs an intensity scaling factor that controls the model's trade-off between robustness and imperceptibility. The discriminator/adversarial network in the Generative adversarial network (GAN) is very suitable for the analogy of the various interferences and attacks encountered in the digital watermark transmission or to identify the fidelity quality of the generated image. Therefore, generative adversarial network architecture is also a lot of work. HiDDeN [**14**] is one of the earliest deep learning methods for image watermarking. The model consists of a trained encoder, decoder, and discriminator network. The encoder network is trained to embed information strings into the cover image while minimizing the perceptual perturbation of the encoded image. The decoder network receives an encoded image and tries to extract information. This discriminator network uses the principle of GAN to force the encoder to generate a watermark image that is as similar as possible to the original image. Zhang et al. [**27**] propose to add an attention module before watermark embedding and extraction so that the model can learn better embedding regions.

III. Proposed Methods

A. Model architecture

In the encoder-decoder model design, we use the encoder and decoder structure of RivaGAN [**27**]. It incorporates an attention module with shared parameters in the encoder and decoder, which helps the encoder and decoder to pay more attention to image regions that are more conducive to embedding. At the same time, there are only multiple convolutional layers in its encoder and decoder model, allowing the model to process input frames of arbitrary resolution.

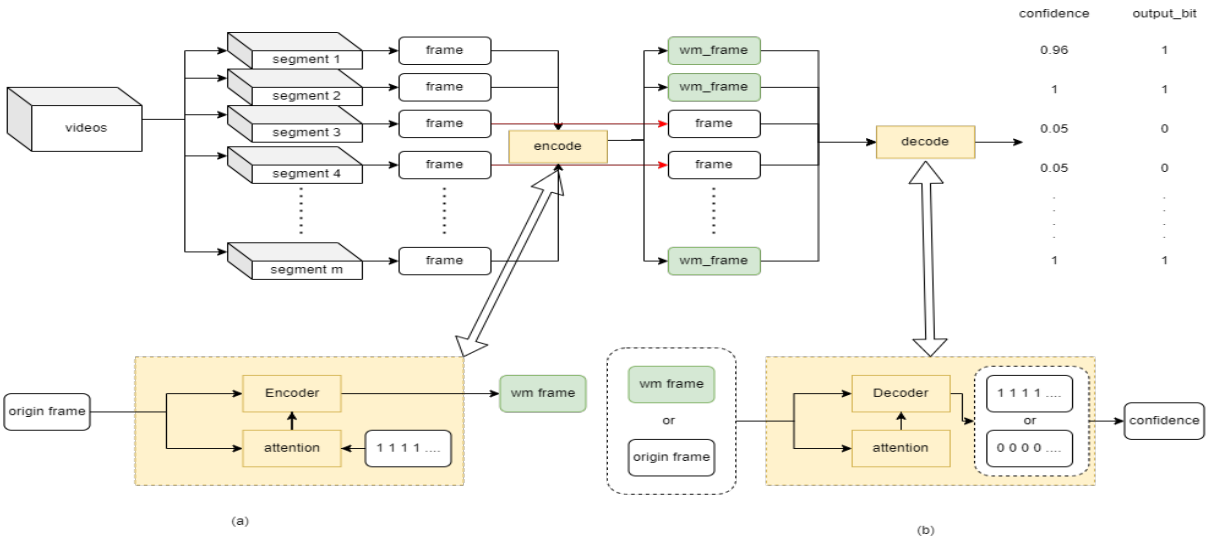

Fig. 1. Overall watermark embedding and extracting process. To embed [1 1 0 0 ..... 1] into a video, we select the first frame of the video clip whose corresponding bit is 1, and input it into the encoder together with the watermark to obtain the watermarked video frame. The specific watermark embedding and extraction process of each frame is shown in (a) and (b)

B. Watermark Embedding and Extraction Process

Different from the previous deep learning image watermarking schemes, which usually embed a specific 01-bit watermark for each image. When the watermark contains a large amount of information, it tends to be distorted at higher video compression rates. Therefore, we do not embed and extract a watermark of a certain length for each frame but determine whether the frame is embedded with a watermark by a certain method. Since the video often contains many frames, this method can still obtain enough information space for embedding and extraction. We divide each video into different segments according to a certain number of frames. For each video segment, we only embed the watermark on the first frame or a specific frame that follows certain rules. Compared to embedding watermarks on all frames, this method is more affected by video compression and thus more challenging.

As shown in Figure 1, a specific watermark embedding extraction scenario is as follows. For a complete video, if its length is N frames, we use k as the number of frames to split the video, then there are m = N/k video segments in total, and we can choose to embed or not embed a watermark in each segment. When it is detected that the fragment contains a watermark frame, output 1, otherwise output 0, finally we can obtain a bit sequence of length m. Similar to the previous digital watermarking literature, we can decode this bit sequence into a specific ID or character.

C. Noise layer

1) Cropping: The cropping layer is used to randomly crop out watermarked video frames as the input to the decoder. It can ensure that the encoder learns to embed data bits with sufficient spatial redundancy and improves the robustness of the model to Cropping noise.

2) Scaling: Scaling is also a common noise for video. To improve the robustness of the model to scaling, we add this noise to the noise layer.

3) H264 compression: The H.264 compression algorithm is currently the most common compression algorithm, which includes intra-frame compression and inter-frame compression. The specific method of this noise is to directly input the watermark frame and its adjacent frames into the compression algorithm and output it to the disk in mp4 format, then read the video file from the disk, and extract the corresponding compressed watermark frame. We use the Constant Rate Factor (CRF) as a tuning parameter for compression ratio.

D. Training process

For the purpose of enabling the decoder to distinguish the watermark-embedded frame from the source video frame,when the frame input to the decoder is embedded with the watermark by the encoder

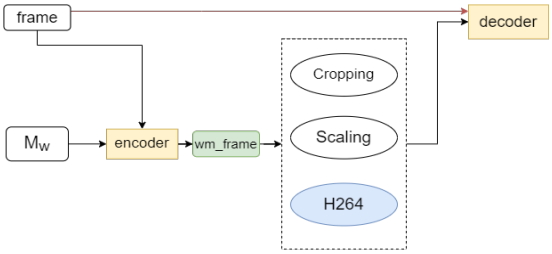

Fig. 2. Training process. During training, when the original frame is input to the decoder, the output of the decoder needs to be as close as possible to M_o (in other words, ground truth is M_o). When the watermark frame is input to the decoder, the output of the decoder needs to be as close as possible to M_w probably close (in other words, ground truth is M_w).

in advance, the decoder needs to output a specific bit sequence M_w, which is used to indicate that the input frame contains a watermark, on the contrary, when the input frame comes from the original video, the decoder needs to output another specific bit sequence M_o that should be as inconsistent as possible with M_w. For simplicity, we set $M_w$to be an all-one bit sequence of length N and M_o to be an all-zero bit sequence of length N, as shown in Figure 1 (b).

During training, the original frame $frame_o$ and M_w are input into the encoder to obtain the watermark frame $frame_m$. After the watermark frame passes through different noise layers, different scrambled watermark frames will be obtained. Among them, the cropped watermark frame is $Crop(frame_m)$, the scaled watermark frame is $Scale(frame_m)$, and the H.264 compressed watermark frame is $H264(frame_m)$. After these different scrambled frames are input to the decoder, a set of output bit sequences will be obtained. We will calculate the cross-entropy loss of these bit sequences with M_w, as shown in Figure 2.

for original frame, decoder should output M_o,the loss function is(CE means Cross Entropy):

$$Loss_o = CE(M_o, Decoder(frame_o)) \quad (1)$$

For watermark frames and various watermark frames after noise layer, we set the decoder output ground truth as M_w, as follows:

$$Loss_m = CE(M_w, Decoder(frame_m)) \quad (2)$$

$$Loss_{cm} = CE(M_w, Decoder(Crop(frame_m)) \quad (3)$$

$$Loss_{sm} = CE(M_w, Decoder(Scale(frame_m))) \quad (4)$$

$$Loss_{hm} = CE(M_w, Decoder(H264(frame_m))) \quad (5)$$

To realize curriculum learning, we assign corresponding weights to different loss functions in different training stages according to their difficulty. We will mainly focus on simple loss function optimization at the beginning of training. When the model has learned a certain watermark embedding and extraction ability, the weight of difficult samples will be increased to make the model more robust to those noises. It is specifically implemented as a two-stage training scheme. Let w_o, w_m, w_{cm}, w_{sm}, and w_{hm} be the weights of $Loss_o$, $Loss_m$, $Loss_{cm}$, $Loss_{sm}$, and $Loss_{hm}$, respectively. Their values in different training epochs are shown in Table 1(E is the total number of epoches for training).

TABLE I
WEIGHTS OF DIFFERENT EPOCHS

weight	0 to E/2	E/2 to E
w_o	1.0	1.0
w_m	0.8	0.3
w_{cm}	0.1	0.2
w_{sm}	0.1	0.2
w_{hm}	0	0.2

At each stage of training, the target loss function is:

$$Loss = w_o * Loss_o + w_m * Loss_m + w_{cm} * Loss_{cm} + w_{sm} * Loss_{sm} + w_{hm} * Loss_{hm} \quad (6)$$

IV. Experiment and Visualization

In this section, we first describe the software and hardware environments of the experiments, followed by the datasets we used and the experimental settings. Finally, we present the experimental results and their related statements.

A. Environment

For this experiment, the training and testing environments of the model are the same. The operating system is ubuntu18.04 (Linux version 4.15.0-180-generic, GCC version 7.5.0), and we use python as the implementation language of the algorithm. The model building and training coding environment is python3.7.13, pytorch1.4.0, and CUDA version 10.1.

B. Dataset

We use an online educational video dataset provided by Easefun, which includes a total of 2000 video clips extracted from online educational videos. We divided 1700 videos from Easefun's online instructional video dataset as a training set and the remaining 300 videos as a testing set. Each video clip is about 3 minutes. The types of videos include screen recording teaching, live-action shooting teaching, etc. The content involves software tutorials, programming teaching, professional skills teaching, etc.

C. Setting

For the training parameters, we set the lengths of M_w and M_o to 16, the watermark visibility to 0.032 (to ensure the quality of the generated video frames), and the total number of training epochs is 200 . We use the Adam optimizer, where the initial learning rate is set to 0.0005, and the batch size is set to 24. H264 CRF is set to 22, and the number of adjacent frames for video compression noise is 16. During training, the video will first be randomly cropped into a region of size 256*256, which is then fed into the model for training.

D. Result

For watermark invisibility, the primary evaluation metric is PSNR. For all test video results, our draw PSNR is 36.9. The visualization results are shown in Figure 3

(a) ori_frame (b) wm_fraame (c) wm_frame after compressed

Fig. 3. (a), (b) and (c) are the original frame, the watermark frame, and the H264 compressed watermark frame, respectively

For testing watermark robustness, we mainly evaluate the model's watermark frame classification accuracy and the decoder's confidence in the watermark frame and the original frame. The confidence of the watermark frame is calculated as follows:

$$sum(output)/Length(output)$$

For example, if the output bit sequence is 11000111, the confidence of this frame will be 5/8 = 0.625. We set a classification threshold of 0.5. When the confidence of a frame is greater than the threshold, it is judged as a watermarked frame. Otherwise, it is a non-watermarked frame (i.e., the original video frame). For each test video, we divide it into many video segments, as shown in Figure 3. Here, we set k =125. There are 10384 125-frame video segments in total. The specific process of the test is as follows. For each video segment, we input it into the encoder model for watermark embedding while retaining the original video so that each video segment will have a watermark version as well as the original version. After that, the watermarked version video is encoded and compressed by H264 with different CRF values to obtain different compressed versions. Finally, the original version and the compressed version are respectively input to the decoder to obtain the corresponding average confidence and classification accuracy, as shown in table 2.

TABLE II
RESULT OF DIFFERENT CRF

metrics	CRF 22	CRF 28	CRF 32
wm_accuracy	0.99	0.98	0.95
wm_confidence	0.99	0.98	0.94
ori_accuracy	0.99	-	-
ori_confidence	0.04	-	-

V. CONCLUSION

In this paper, we propose a robust video watermarking model based on deep neural network and curriculum learning. The watermarking results show that the model can maintain high watermarking accuracy at a higher compression rate. It can already deal with most video piracy scenarios, and it is a method that can be applied to actual scenarios. However, although our method can guarantee sufficient accuracy and robustness, there is still room for further exploration of the quality of watermarked frames and the generation efficiency of watermarked frames. In future work, we will further improve these aspects.

ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China (NSFC) 61876208 and 62272172, Tip-top Scientific and Technical Innovative Youth Talents of Guangdong Special Support Program (2019TQ05X200) and 2022 Tencent Wechat Rhino-Bird Focused Research Program Research (Tencent WeChat RBFR2022008).

REFERENCES

[1] Hui Zhou, Tao Xu, Xiaochuan Wu. Resist the collusion attack of a digital video watermarking algorithm. Computer Applications 2006; 26 (04):812-814.

[2] Hefei Ling, Zhengding Lu, Fuhao Zou. New real-time watermarking algorithm for compressed video in VLC domain. International Conference on Image Processing 2004; 24: 2171–2174.

[3] Hua Cao, Jing-li Zhou, Sheng-sheng Yu, Shuguang Su. Based on H.264 Low bit rate video stream 264 semi-fragile watermarking algorithms. Electronics 2006; 34 (01):40-44.

[4] Desheng Fu, Jianrong Wang. Based on H. 264 of the video watermarking technology. Computer Applications 2009; 29 (04):1174-1176.

[5] Lihe Zhang, Hongtao Wu, Changli Hu. A Gabor transform based video watermarking algorithm . Software 2004; 15 (08):1252-1258.

[6] Yafei Shao, Guowei Wu, Li Zhang, Xinggang Lin. Digital Video Broadcasting in the compressed domain watermarking algorithm. Electronics 2003; 31 (10):1562 -1565.

[7] Noorkami M, Mersereau R M. Compressed-domain video watermarking for H. 264[C]//IEEE International Conference on Image Processing 2005. IEEE, 2005, 2: II-890.

[8] Deepshikha Chopra, Preeti Gupta, Gaur Sanjay, and Anil Gupta. Lsb based digital image watermarking for gray scale image.IOSR Journal of Computer Engineering, 6(1):36–41, 2012.2

[9] Ton Kalker, Geert Depovere, Jaap Haitsma, and Maurice JJJB Maes. Video watermarking system for broadcast monitoring. InSecurity and Watermarking of Multimedia contents, volume 3657, pages 103–112. International Society for Optics and Photonics, 1999.2

[10] Frederic Deguillaume, Gabriela Csurka, Joseph JK O'Ruanaidh, and Thierry Pun. Robust 3d dft video watermarking. InSecurity and Watermarking of Multimedia Contents, volume 3657, pages 113–124. International Society for Optics and Photonics, 1999.2

[11] Nilanjan Dey, Poulami Das, Anamitra Bardhan Roy, Achintya Das, and Sheli Sinha Chaudhuri. Dwt-dct-svd based intravascular ultrasound video watermarking. In2012 World Congress on Information and Communication Technologies, pages 224–229. IEEE, 2012.2

[12] Pik-Wah Chan and Michael R Lyu. A dwt-based digital video watermarking scheme with error correcting code. In International Conference on Information and Communications Security, pages 202–213. Springer, 2003.2

[13] Lino E Coria, Mark R Pickering, Panos Nasiopoulos, and Rabab Kreidieh Ward. A video watermarking scheme based on the dual-tree complex wavelet transform.IEEE Transactions on Information F orensics and Security, 3(3):466–474, 2008.2

[14] Zhu J, Kaplan R, Johnson J, et al. Hidden: Hiding data with deep networks[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 657-672.

[15] Seung-Min Mun, Seung-Hun Nam, Haneol Jang, Dongkyu Kim, and Heung-Kyu Lee. 2019. Finding robust domain from attacks: A learning framework for blind watermarking.Neurocomputing337 (2019), 191–202.

[16] Xin Zhong, Pei-Chi Huang, Spyridon Mastorakis, and Frank Y. Shih. 2020. An Automated and Robust Image Watermarking Scheme Based on Deep Neural Networks.IEEETransactionsonMultimedia(2020).

[17] Mahdi Ahmadi, Alireza Norouzi, S. M. Reza Soroushmehr, Nader Karimi, Kayvan Najarian, Shadrokh Samavi, and Ali Emami. 2020. ReDMark: Framework for Residual Diffusion Watermarking on Deep Networks.ExpertSystemswith Applications 146 (2020), 113157.

[18] Jae-Eun Lee, Young-Ho Seo, and Dong-Wook Kim. 2020. Convolutional Neural Network-Based Digital Image Watermarking Adaptive to the Resolution of Image and Watermark.AppliedSciences10, 19 (2020).

[19] Bingyang Wen and Sergul Aydore. 2019. ROMark: A Robust Watermarking System Using Adversarial Training. arXiv:1910.01221 [cs.CV]

[20] Ippei HAMAMOTO and Masaki Kawamura. 2020.IEICE Transactions on Information and Systems E103.D (01 2020), 33–41.

[21] Honglei Zhang, Hu Wang, Yuanzhouhan Cao, Chunhua Shen, and Yidong Li. 2021. Robust Watermarking Using Inverse Gradient Attention. (2021). arXiv:2011.10850 [cs.CV]

[22] Yang Liu, Mengxi Guo, Jian Zhang, Yuesheng Zhu, and Xiaodong Xie. 2019. A Novel Two-Stage Separable Deep Learning Framework for Practical Blind Watermarking. In MM'19 : Proceedings of the 27th ACM International Conference on Multimedia (Nice, France) (MM '19). Association for Computing Machinery, New York, NY, USA.

[23] Aayush Mishra, Suraj Kumar, Aditya Nigam, and Saiful Islam. Vstegnet: Video steganography network using spatiotemporal features and micro-bottleneck. InThe British Machine Vision Conference, page 274, 2019.3

[24] V . Vukoti´c, V . Chappelier, and T. Furon. Are Deep Neural Networks good for blind image watermarking? InProc. of the IEEE Int. Workshop on Information F orensics and Security (WIFS), pages 1–7, Dec 2018.

[25] Haribabu Kandi, Deepak Mishra, and Subrahmanyam R.K. Sai Gorthi. Exploring the learning capabilities of convolutional neural networks for robust image watermarking.Computers Security, 65:247 – 268, 2017.

[26] Bengio Y, Louradour J, Collobert R, et al. Curriculum learning[C]//Proceedings of the 26th annual international conference on machine learning. 2009: 41-48.

[27] Zhang K A, Xu L, Cuesta-Infante A, et al. Robust invisible video watermarking with attention[J]. arXiv preprint arXiv:1909.01285, 2019.

[28] Song X, Su Y, Liu Y, et al. A video watermarking scheme for AVS based on motion vectors[C]//2008 11th IEEE International Conference on Communication Technology. IEEE, 2008: 738-741.

[29] Qiu G, Marziliano P, Ho A T S, et al. A hybrid watermarking scheme for H. 264/AVC video[C]//Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004. IEEE, 2004, 4: 865-868.